SLR sentiment enhanced

Martina Colombari
270396@studenti.unimore.it

Omayma Moussadek
253399@studenti.unimore.it

Nicolò Rossi
272179@studenti.unimore.it

Università degli studi di Modena e Reggio Emilia

ABSTRACT

With this paper we would like to explore many different ways to enhance Sign Language Recognition, also known as SLR. We built the foundation of this project on an already working SLR system [2] we chose this particular implementation for its experimental approach with skeleton extraction, we'll talk more about this later. So to enhance the prediction we work on 3 different solution:

- *Emotion enhancement: facial expression is often used in Sign language to convey better words, so words with emotional meaning after comes with the respective facial expression from the user. Hence detecting it can provide a better context to predict a word more accurately.*

- *Skeleton rotation: as introduced earlier [2] use a skeleton extraction to feed a Graph Convolutional Network, for this reason we would like to change the orientation of the skeleton in a manner that will always face the camera, by doing this we would like to make the system resilient to user orientation changes.*

- *Similar Sign retrieval: from the high-level feature representation of the SL-GCN branch, we extract the embedding space, in which we are able to find similar video. This similarity provides more information to work in the prediction, hence could improve accuracy.*

In the following paragraphs we'll explore how we started, how we implemented each component and we'll talk about our results.

1 Introduction

Sign languages are the primary means of communication within deaf communities, characterized by a rich and expressive visual language performed through dynamic hand gestures, body postures, and facial expressions. Understanding and using sign language requires a

considerable time of learining and training which is not practical and feasbible for the public. Furthermore, sign languages are dynamic and diverse, influenced by geographic regions (e.g., *American Sign Language* - ASL, *Italian Sign Language* - LIS, *Chinese Sign Language* - CSL) and cultural contexts, which further complicates its popularization.

To bridge the gap between deaf and hearing communities, there is a growing interest in Sign Language Recognition (SLR) systems, which leverage advancements in machine learning and computer vision to automatically interpret sign language. However, SLR is more challenging than conventional action recognition. First, sign language requires both global body motion and delicate arm/hand gestures to distinctly and accurately express its meaning, furthermore, facial expression can be utilized to express an emotion that is linked to that gloss. Second, different signers may perform sign language differently(e.g. speed, localism, left-hander, right-hander, dialect), making SLR more challenging.

Traditional SLR methods mainly deploy handcrafted features as HOG and SIFT [11, 5] with conventional classifiers like KNN or SVM. However, with the advent of deep learning, more sophisticated models, including RNNs, LSTMs, and 3D CNNs, have emerged.[6]. Recently, skeleton-based methods have gained popularity due to their adaptability to dynamic environments and complex backgrounds, which is crucial for developing inclusive and versatile SLR systems.

Inspired by SAM-SLR[2] that uses a graph convolutional networks (GCNs) to model keypoints and uses an ensemble method to improve overall performance, we propose a novel skeleton-based SLR approach aimed at improving model accuracy. Our contributions include:

- We constructed a new branch for classify the emotion conveyed by the signer, integrating this information into the GCN ensemble.
- We introduced a geometric component that should allow for pose correction through skeleton rotations. With this we aim to improve the strengthness of the skeleton pose used in sign recognition.
- We proposed a Similar Sign Retrieval (SSR) system to find similar signs in a database, enhancing the model's robustness in inference.
- We developed a MLP for the final prediction of the sign considering the signer's emotion, the pose's prediction and the retrieval score.

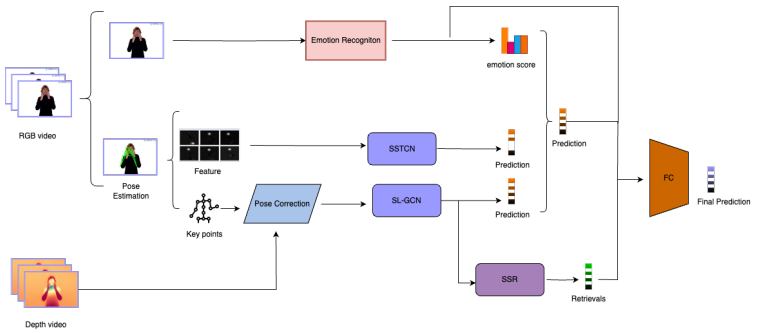


Figure 1: End-To-End schema of our SLR sentiment enhanced model.

2 SAM-SLR

It's the start point of this project, the foundation which holds our implementation. It was developed for the CVPR 2021 Challenge: Looking at People Large Scale Signer Independent Isolated Sign Language Recognition, in which ranked first. It's an ensemble of completely different approach to solve SLR:

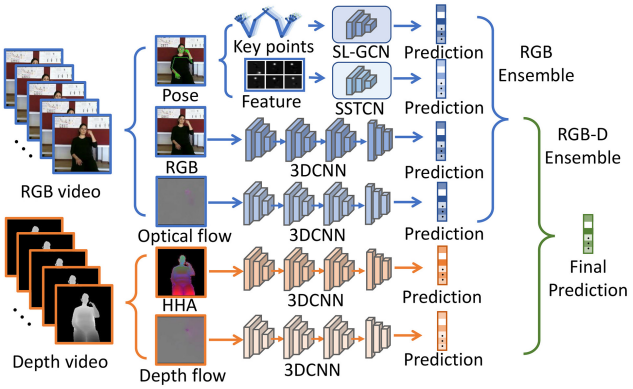


Figure 2: Complete ensemble architecture

For this project we focused on the RGB video, neglecting the Depth Video, due to a lack of availability of depth video for SLR.

SAM-SLR first use the 133 keypoints estimated from a pretrained whole-body pose estimation [7] for generate a spatio-temporal graph. It can be constructed by connecting the adjacent keypoints in the spatial dimension according to the natural connections of human body, and connecting all keypoints to themselves in the temporal dimension. SAM keeps only 27 of the 133 keypoints estimated. This representation is than amplified in 4 different streams: joint, bone, joint motion and bone motion. All the streams is give as input of SL-GCN, a decoupled spatial convolutional network, with a self-attention and graph dropping module. Besides using key point coordinates generated from the whole-body pose network, SAM also propose a SSTCN(Separable Spatial Temporal Convolution Network) model to recognize the sign language from whole-body features. At the end SAM uses a simple ensemble method to ensemble all four modalities above and SSTCN result. For the RGB, SAM also propose a 3DCNN, that we trained but didn't used it in our project.

2.1 Datasets

In this brief paragraph we present the two datasets that we use for training and evaluating our proposal. We use WLASL dataset for training and testing our model and we also create a custom dataset for test the pose correction module.

2.1.1 WLASL

WLASL2000 [3] is an American Sign Language dataset, of videos, with a vocabulary size of 2000 words. It is a massive collection extracted from the web, and performed by 119 signers. It is split into three different subsets, which composition is shown in Table 1.

2.1.2 Custom

Our custom dataset is composed of 111 videos performed by 4 signers. The dataset was born with the idea of testing if the pose adjustment that we performed influences the model accuracy. For that case we take as vobabulary only 12 words (Love, Book, Help, Clock, Happy, Sad, Angry, Surprise, Disgust, Italy, Rock, Look at) and than we perform this sign

Subsets	Samples
Training	8,811
Testing	2,067
Validation	2,723
Total	13,601

Table 1: A statisitcal summary of the WLASL dataset

3 times: one with the signer look at the camera, one were the signer look at the left of the camera and one were the signer look at the right of the camera.

Figure 3: Example of how custom dataset videos are. These three frames refer to 3 different videos where the same word is performed but with different orientations.

3 Emotional Enhancement

From an article of The National Center for Biotechnology Information, which is part of the United State National Library of medicine, a branch of the National Institute of Health which is the primary agency of the United State government responsible for biomedical and public health research:
"Facial and head movements are used in sign languages at all levels of linguistic structure. At the phonological level some signs have an obligatory facial component in their citation form (Liddell, 1980; Woll, 2001)[...]. Facial actions mark relative clauses, content questions and conditionals, amongst others, although there is some controversy whether these markings should be regarded as syntactic or prosodic (cf. Liddell, 1980; Baker-Shenk, 1983; Aarons et al., 1992; Nespor and Sandler, 1999; Sandler, 1999; Wilbur and Patschke, 1999; Neidle et al., 2000; Dachkovsky and Sandler, 2009; Wilbur, 2009)".
[1].
Since facial expressions convey information about the conversation we thought to explore the possibility to use this information to enhance word prediction in the system, in this chapter we'll talk about how we implement this and our results.

3.1 DAN

To extract facial expression from the user we used a Multi-head Cross Attention Network for Facial Expression Recognition [8] which is a pyTorch implementation of the DAN, Distract Your Attention [9]. This network takes in input images and it's able to detect 8 different emotional facial expressions: neutral, happy, sad, surprise, fear, disgust, anger and contempt. Since this network was implemented to have in input images, the emotional state of the sign language user was analyzed in the video frame by frame and then by evaluating the accuracy of the prediction among all the frames, is computed a vector of 8 elements representing the distribution of probability for each emotional state. This solution was introduced instead of just predicting a single label for the whole video to estimate better video with more than one facial expression estimation.

3.2 Enhancement through score augmentation

The First approach to enhance the prediction was to take the ensemble score of the SL-GCN network, which takes in input the prediction of the Graph Convolutional Network for Joint, Joint Motion, Bone and Bone Motion and output a weighted sum, and compute a new score based on the ensemble one and the emotion probability distribution. To understand how an emotion should change the score we needed an association between emotion and resulting word, so based on word meaning and relative video in the WLASL dataset were defined the following relationships:

Happy	Enjoy Happy Fun Relax Peaceful
Sad	Cry Tired Sad Sorry Lonely Depressed Death
Surprise	Surprise Shock Amazing
Fear	Fear Afraid Scared Worry
Disgust	Awful Bitter Disgust
Anger	Quiet Rage Upset
Contempt	Not Cheat

This is the last iteration of multiple tests to understand which word actually portrays facial expression, and which doesn't. We haven't defined any relationship with the emotion 'neutral', since we are not enhancing the score for this one.
As the final stage, each score value produced by the SL-GCN ensemble, that is present in the table shown earlier, is multiplied by a factor

$$\alpha(E) \tag{1}$$

defined as:

$$\alpha(E) = 1 + Emotion\_score(E) \tag{2}$$

with respect to the relative emotion E. "Emotion_score(E)" returns the emotion score obtained earlier by the DAN for the emotion E.

In the end the enhancement of the score was applied only for the highest 10 score labels, due to the amount of False positives, this decision was based on the result shown in the next chapter.

3.3 Result of enhancement through score augmentation

The network, SL-GCN, is trained using the "train" subset of WLASL dataset, and tested with "val".

Baseline result without emotional enhancement:

Measure	Value
Top1 accuracy	0.2824
Top5 accuracy	0.5997
Top10 accuracy	0.7000
Mean loss normal	731.9924
Mean loss Softmax	1.4986

Table 2: Metrics without enhancement

(Results are approximated to the fourth decimal number, for readability).

- With *Top-n* we refer to the accuracy of the model considering a positive prediction if the correct answer is in the first highest n scores.
- With *mean loss normal* we refer to the average of the difference between the normalized score and the target score: 1 for the actual label and everyone else 0.
- With *mean loss Softmax* we refer to the average of the difference between the normalized score, normalized with the softmax function, and the objective score.

Having different measurements of accuracy based on the position of the right label can be useful to understand how we improve the score, so that even if the final prediction is wrong we are able to see if the right prediction score had some improvement. Mean loss normal and Mean loss softmax measure both how much the predicted score was wrong with respect to the target, the main difference is that with the first one by keeping the proportion between value in the normalization actually gives a bigger penalty for score scattered more between the score value's range. And for the second one since push values between 0 and 1, while keeping the sum 1, don't give the same penalty, it has minimum value, or mostly disregardable, as long as the bigger score index is equal to the label.

To make sure the number of False positive and False negative obtained by the emotional enhancement was at a reasonable level, it was studied in the "val" subset of the dataset. Results were:

Measure	Value
True positive	32
False positive	2296
True negative	376
False negative	19

Table 3: Confusion Matrix values

- TP: videos that should be enhanced and are, for the right label.
- FP: videos that shouldn't be enhanced and are, for any label.
- TN: videos that shouldn't be enhanced and are not, for any label.
- FN: videos that should be enhanced and are not, for the right label.

Precision and Recall measurement:

- Precision: 0.0137
- Recall: 0.6274

To reduce the number of False Positives, we tried to enhance only the 10 highest score labels, since they were the one with the highest possibility to be correct, we wouldn't risk enhancing a label with a really low possibility to be right.
Results were:

Measure	**Value**
True positive	26
False positive	98
True negative	2574
False negative	25

Table 4: Confusion Matrix values, only the 10 highest score enhanced

Precision and Recall measurement:

- Precision: 0.2097
- Recall: 0.5098

We kept a reasonable Recall value while increasing Precision, this solution even though reduced the number of true positives, reduced the amount of False positives. This result as mentioned earlier convinced us to use only the first 10 highest score label for the enhancement.
Result with emotional enhancement:

Measure	**Value**
Top1 accuracy	0.3023
Top5 accuracy	0.6277
Top10 accuracy	0.7239
Mean loss normal	726.2018
Mean loss Softmax	1.4614

Table 5: Metrics after emotional enhancement

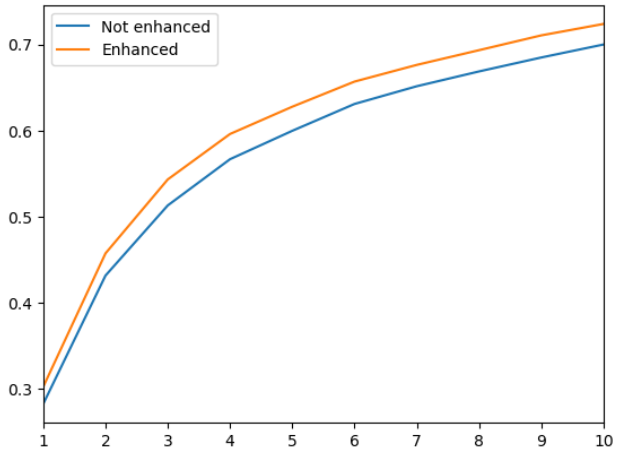


Figure 4: This Graph represent the Top-n accuracy for both enhanced results and not

It is worth noting that "signer9" (sign language user labeled with number 9 in the dataset) was excluded in the testing due to poor results for emotion recognition, possibly caused by the low quality video provided for this signer.

All metrics perform better, we have better accuracy for each possible Top-n measure, and both mean losses are lower than earlier.

It's quite interesting also seeing the result specifically only for videos that should be having an enhancement.

Results only for video that should be enhanced:

Measure	**Value**
Top1 accuracy	0.4359
Top5 accuracy	0.7949
Top10 accuracy	0.7949
Mean loss normal	649.4471
Mean loss Softmax	1.1504

Table 6: Metrics only for the videos that should be enhanced

There's a significant improvement, up to a 36.19% for the Top-n accuracy, also both mean losses are lower, this shows that there's an improvement over the average statistics for video that should be enhanced.

3.4 Enhancement through a Neural Network

Another approach used to enhance prediction with facial expression information was through a Fully Connected Neural Network.

3 different network sizes were tested, each one had an input dimension of 2008, 2000 from the normalized score from the SL-GCN ensemble and 8 from the vector provided earlier from DAN which is the same one used in the "enhancement through score augmentation", and an output of 2000 for each possible label.

First one had 2 hidden layers, the first one with 2006 perceptron and the second one with 2004, after the first hidden layer was implemented a policy of dropoff with a probability of 20%, obviously only applied during training, as a measure to contrast overfitting. After 3100 epoch the results were the following:

Measure	Value
Train accuracy	0.954
Test accuracy	0.194
Val accuracy	0.213

Table 7: Accuracy for different subset of the dataset, first architecture

Second one had only 1 hidden layer in an attempt to reduce overfitting, this hidden layer has 2000 perceptrons, it still had a dropoff policy after the first hidden layer with a probability of 20%.
After 2000 epoch the results were the following:

Measure	Value
Train accuracy	0.931
Test accuracy	0.214
Val accuracy	0.228

Table 8: Accuracy for different subset of the dataset, second architecture

The third one had no hidden layer, and no dropout policy, just an input layer of 2008 and an output layer of 2000. After 24100 epoch the results were the following:

Measure	Value
Train accuracy	0.871
Test accuracy	0.230
Val accuracy	0.226

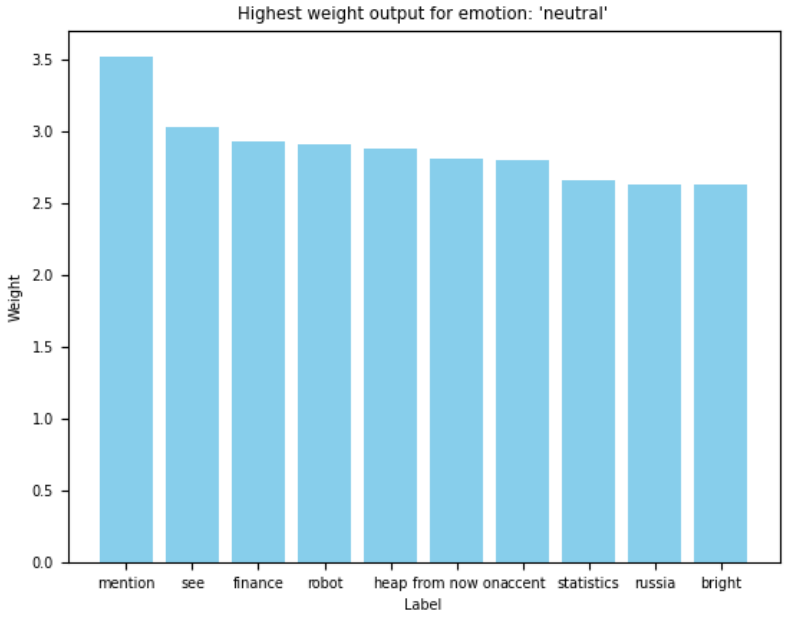
Table 9: Accuracy for different subset of the dataset, third architecture

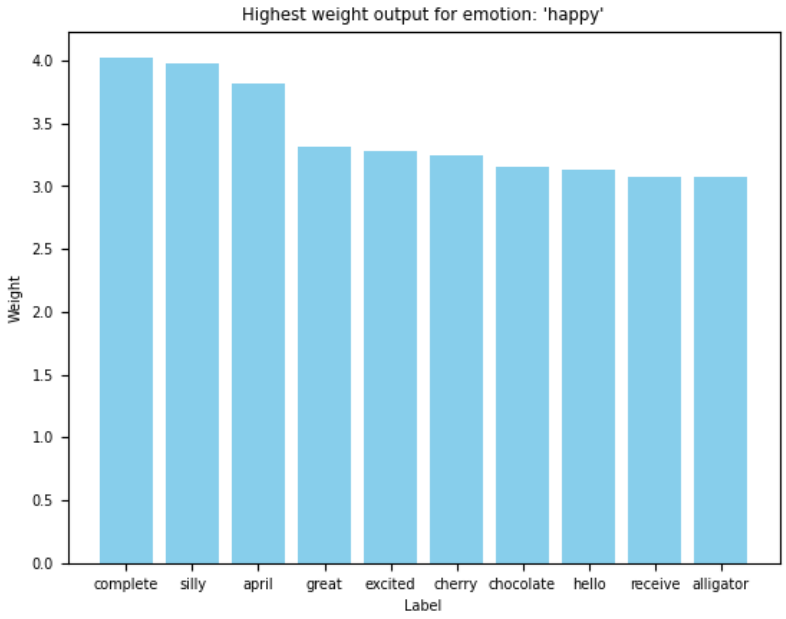
Even Though results improved after each iteration of the network, they never reached the same as the one before the enhancement or with the "enhancement through score augmentation" seen earlier.

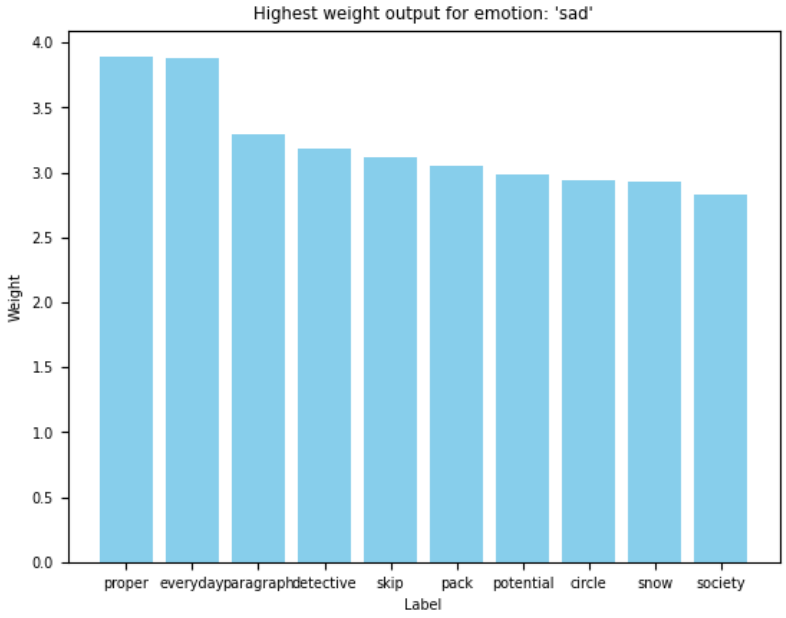
To investigate why this may happen, we estimated for each emotion which label was the one with the biggest enhancement.

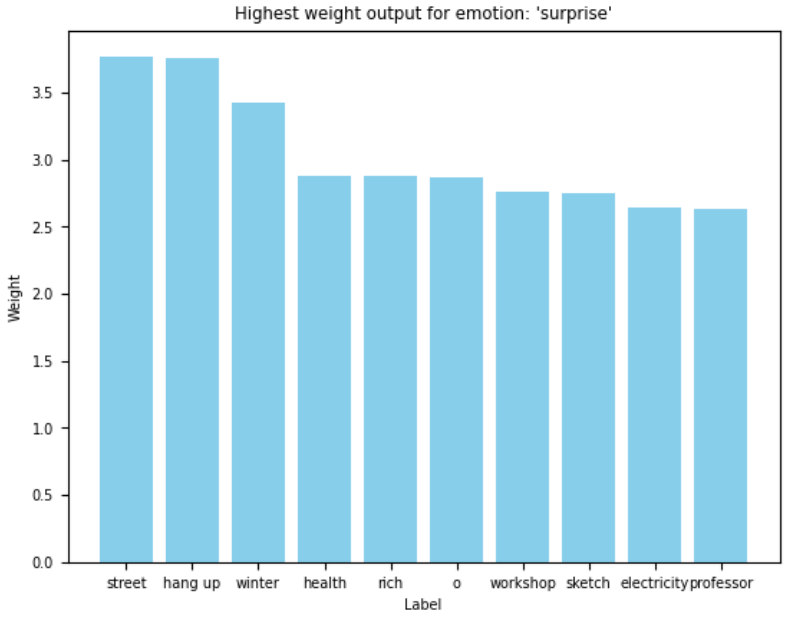
To achieve this we took the latest network architecture's weight and proceeded for each emotion to search which were the output most affected by that emotion. A bigger weight for a particular input emotion (which was normalized between 0 and 1) corresponds to a bigger score as output, and so a higher probability that the label would be the prediction.

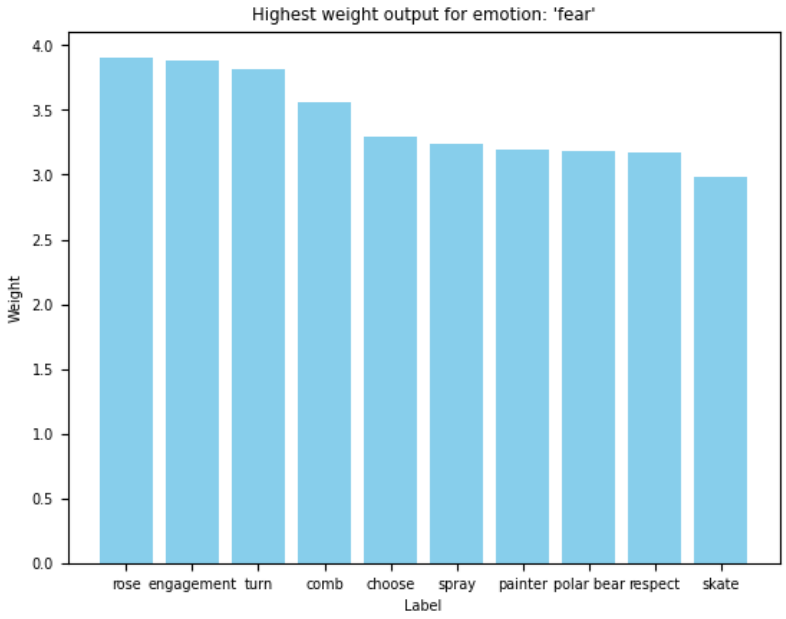
The following graph displays only the highest 10 labels for each emotion to achieve a less congested graph.

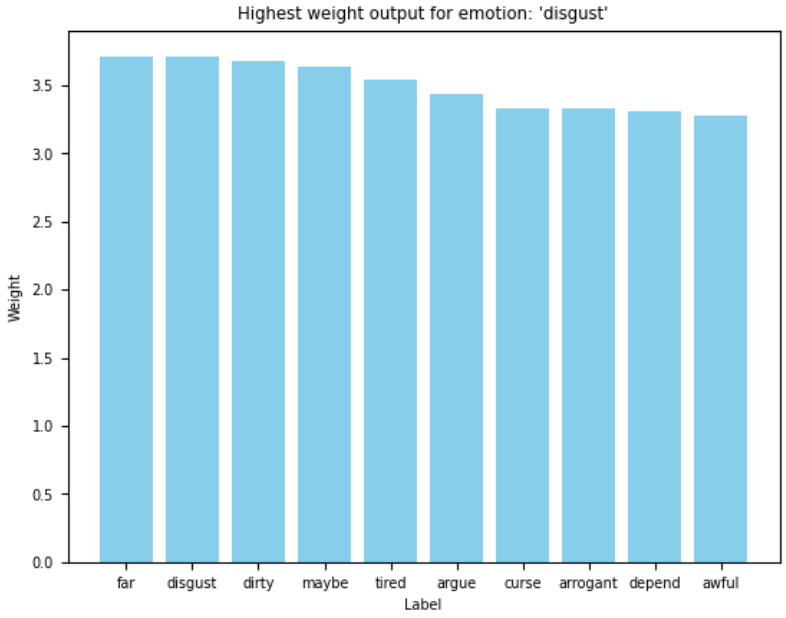

Highest weight output for emotion: 'neutral'
Weight
Label
mention
see
finance
robot
heap
from now on
accent
statistics
russia
bright

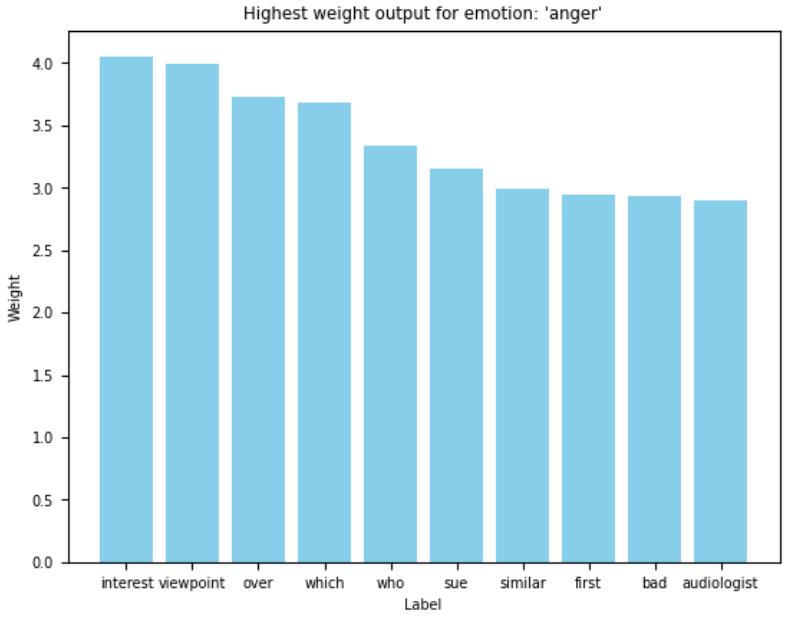

Highest weight output for emotion: 'happy'
Weight
Label
complete
silly
april
great
excited
cherry
chocolate
hello
receive
alligator

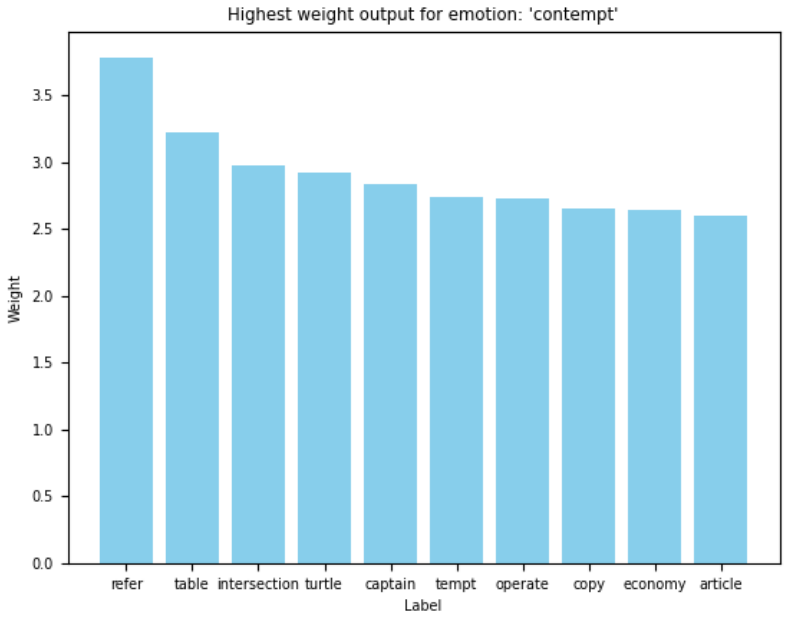

Highest weight output for emotion: 'sad'
Weight
Label
proper
everyday
paragraph
detective
skip
pack
potential
circle
snow
society


Highest weight output for emotion: 'surprise'
Weight
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
street
hang up
winter
health
rich
o
workshop
sketch
electricity
professor
Label


Highest weight output for emotion: 'fear'
Weight
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
rose
engagement
turn
comb
choose
spray
painter
polar bear
respect
skate
Label


Highest weight output for emotion: 'disgust'
Weight
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
far
disgust
dirty
maybe
tired
argue
curse
arrogant
depend
awful
Label





By the graph proposed previously we can see how in many cases the network was able correctly to understand how a certain word is affected by a certain emotion, for example:

- The word ‘excited’ for the emotion ‘happy’.
- The word ’disgust’ for the emotion ‘disgust’.
- The word ’bad’ for the emotion ‘anger’.

But we can see many more examples of words that shouldn’t be enhanced at all by any emotion, for example:

- The word ‘rose’ for the emotion ‘fear’.
- The word ‘circle’ for the emotion ‘sad’.
- The word ‘alligator’ for the emotion ‘happy’.

if we investigate further we can see how the average score of the emotion 'fear' in the subset "Train" for the video which label corresponds to the word 'rose' is 1.0000, we have 2 file which corresponds to the word rose and in both we find a score of 1 for the emotion 'fear'. For the same word and label in the "Test" subset of the dataset we get a mean of 0.3531, and for the "Val" subset we get a mean of 0.3049.

For both the word "circle" with the emotion "sad" and the word "alligator" with the emotion "happy" we find similar results:

Subset	**Mean value**
Train	0.5864
Test	0.0000
Val	0.0000

Table 10: Mean score of emotion 'sad' for the word 'circle'

Subset	**Mean value**
Train	0.5000
Test	0.0000
Val	0.0000

Table 11: Mean score of emotion 'happy' for the word 'alligator'

What we can observe from these results is that the relationships between word and facial expression do not match between subsets of the dataset. This fools the network to learn a relation between a facial expression and a certain output, but since this is only true for the training subset, it leads to overfit the network.

The previous results may be caused by an inaccuracy of the facial expression prediction, which could lead to a series of prediction differences, for the same label, in different subsets. Or could happen due to the dataset itself, from the actual facial expression of the sign language user, that could not match the actual meaning of the word.

3.5 Conclusion

As we seen in the emotional enhancement with the Neural Network, improving the score with facial expression is a non-trivial problem that requires a lot of work both on the architecture and the dataset. Since we got worse results than the ensemble without the enhancement we misled the network towards overfitting, providing information on the emotional expression of the sign language user.

We can deduce, based also on the result of the enhancement through score augmentation, that hand picking the relationship between word and emotion could have some improvement over the results, but require a lot of work to achieve this and may require more human intervention in the future to maintain changes in relationship, that may occur due to a change in society perception of a word. An enhancement with Neural networks can automate this process, but still require really accurate training data to achieve decent results.

4 Geometric transformations

4.1 Motivations

After identifying keypoints of the human body, a graph is created to connect adjacent pairs of points, representing the natural connections present in the human body. Since sign language recognition is aware of the skeleton, it's important to accurately identify the joints to distinguish between similar gestures with different meanings.

To improve the robustness of the skeleton pose used in sign recognition, we suggest introducing a geometric component that should allow for pose correction through transformations of the points in 2D and then in 3D space. In the latter case, the depth informations are extracted from processed videos.

In both cases, the transformations are applied to a graph obtained from a reduction on the whole-body skeleton graph, which trims down the 133 nodes to 27 nodes. This is done in order to reduce the noise that affects the recognition by isolate the joints and edges which perform the actual gesture.

4.2 Pre-processing

Depth Anything [10] is an image-based depth estimation system that we used on the custom dataset videos to generate a monocular depth estimation, which is based on the size of objects, shadows, and other two-dimensional information. The authors provide three models of varying scales for robust relative depth estimation, and we specifically used ViT-L.

The videos are generated using a set of colors obtained with *cv2.applyColorMap(depth, cv2.COLORMAP_INFERNO)*. This is one of the various colormaps provided by OpenCv to enhance visualization. To improve usability, we inverted the obtained colormap by implementing a function that extracts the depth, since OpenCV does not have a built-in function to provide the depth from the color. Then we encoded this value in the blue channel of the video's frames.

4.3 2D rotation

Input: 2D graph in which each point is represented by (x, y, s), where x-y are the 2D coordinates and s is the confidence score.

To correct and straighten the skeleton pose, a rotation must be applied to the skeleton frame by an angle corresponding to the deviation of the skeleton's reference frame from the base frame. The angle is represented in the Figure 8.

The skeleton's reference frame is defined using the shoulders as cardinal points to determine the x-axis, with the origin set at the center of the segment connecting the shoulders. The y-axis is orthogonal to the x-axis and passes through this origin. The coordinates of the points in the graph are assigned according to the reference of each frame (captured from the video where the gesture is performed). This is used to define the rotation angle to be applied. The final orientation in which the pose is straightened is obtained once the frames are aligned.

The point's coordinates in the graph are defined relative to each frame's reference captured from the video, which helps determine the rotation angle needed for alignment. The alignment process involves rotating the skeleton's reference frame to align with the frame's reference.

In 2D space, this is viewed as projecting 3D points onto the XY plane, with the rotation occurring around the z-axis passing through the skeleton's origin. The rotation angle is computed based on the deviation from the x-axis of the reference frame, and the points are rotated accordingly.

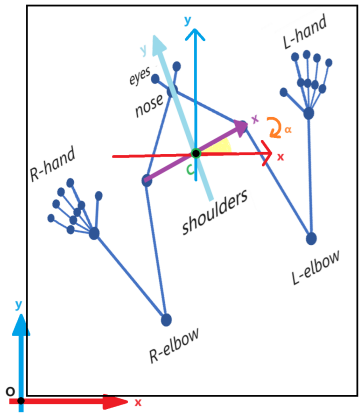


Figure 8: The figure represents the frames used in the 2D rotation. Specifically, the angle α is calculated between the two x-axes.

The new coordinates of the points can be found by multiplying the rotation matrix by the points centered at C. If we have a point $P = (x, y)^T$ [1] and the rotation matrix (both in R^2) around the z-axis by an angle α is given by:

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \tag{3}$$

then the new coordinates P' of the point after rotation can be obtained by:

$$P' = R_z(\alpha) \cdot (P - C) \tag{4}$$

where C is the center of the skeleton reference frame, and the subtraction $P - C$ extract the coordinates of P point with respect to C. The result of this multiplication gives the coordinates of the rotated point in the skeleton's frame. To obtain the coordinates of the rotated points with respect to the original coordinate system, we perform the following operation:

$$P_{\text{final}} = (P' + C) \tag{5}$$

4.4 3D rotation

Input: 3D graph in which each point is represented by (x, y, z, s), where x-y-z are the 3D coordinates and s is the confidence score. Starting from the video in BRG, we sampled the depth value (coordinate z) at the x-y positions for all the points in the skeleton.

The new skeleton points are obtained by applying two consecutive rotations:

- around Z axis: this rotation is used to straighten the skeleton, corresponding to the transformation applied to correct the pose in 2D.
- around Y axis: this rotation aligns the skeleton so that it lies on the desired plane.

To compose these transformations, the two matrices defined with respect to the base frame can be pre-multiplied, as follows:

$$R(\beta, \alpha) = R_y(\beta) * R_z(\alpha) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where α is the angle between the X axes of the two frames, β is the angle between the Y axes of the two frames.

[1] From the 2D graph, we consider only the first two coordinates of the points.

The new coordinates of the points $P = (x, y, z)$[2] can be found by multiplying this new rotation matrix by the points centered at C and the combined rotation matrix (both in R^3):

$$P' = R(\beta, \alpha) \cdot (P - C)$$
$$P_{\text{final}} = P' + C \tag{7}$$

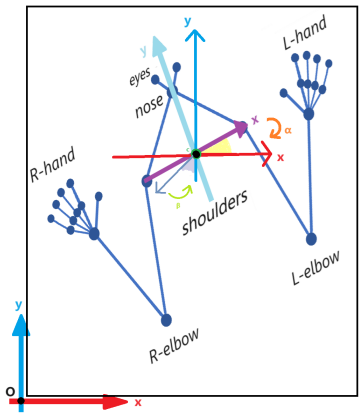


Figure 9: The figure represents the frames used in the 3D rotation. Specifically: the angle α is calculated between the two x-axes; while the angle β is calculated between the z-axes and the XY plane of the base frame.

4.5 Results and limitations

The 2D pose correction was applied to both the WLASL "test" dataset, which includes 2067 skeletons, and our custom dataset of 111 skeletons. In both cases results did not show any change in the network's performance, as significant rotations require a substantial delta y between the shoulders. The histogram in Figure 10 shows an improvement in performance for straight orientations compared to the results obtained by the network without 2D rotation.

Modes	Top-1	Top-5
WLASL(val)	24.53	50.86
WLASL(test)	25.21	52.20
Custom(2D)	18.18	32.73
Custom(3D)	16.36	30.91

Table 12: Geometric transformations results on WLASL and on our custom dataset

3D rotation was applied to a custom dataset of 111 skeletons with three different orientations: Straight, Left, and Right. The results were mainly influenced by the inaccuracy of the joints identified in the samples with lateral orientation, which did not lead to significant improvements. The main issue concerns the calculation of the depth angle used for rotation around the y-axis, which becomes inaccurate when the figure is oriented to the left or right. To address this problem, experiments were conducted to evaluate the effect of the transformation as a function of the depth angle.

- **Discrete angles**: For each video, after discretizing the depth angles of all frames, a single depth angle was used for all frames, corresponding to the value associated with the most frequent bin. The bin granularities used were: 5°, 10°, 20°, 30°. With this modification, an improvement in TOP1 was observed compared to the baseline results and those with only 2D rotation; however, as the granularity increases, the improvement decreases.

[2]From the 3D graph, we consider only the first three coordinates of the points.

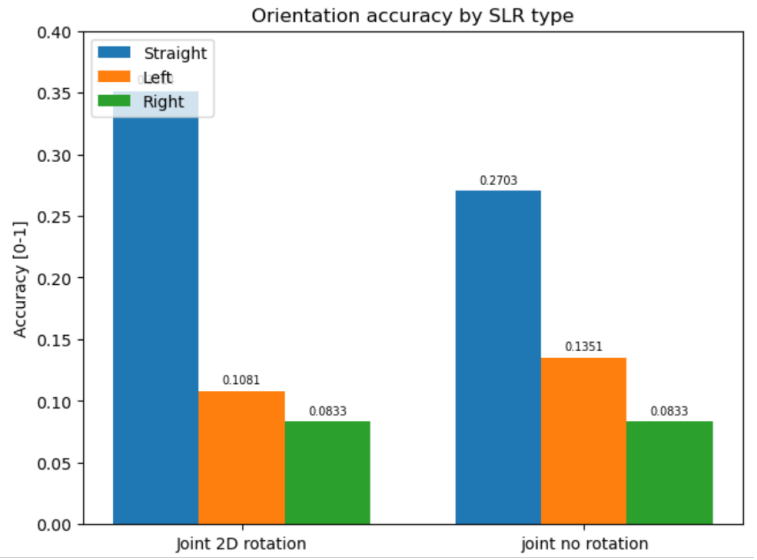


Figure 10: Histograms showing the performance obtained by the network for each orientation when applying 2D rotation and without applying the geometric transformation.

- **Fixed angles**: The angles were chosen based on the orientation of the figure in the video, determined by the sign of delta z (calculated as the difference between the z coordinate of the left shoulder and that of the right shoulder). Two tests were conducted: one with angles of -45°, 0°, 45° for Right, Straight, and Left respectively, and another with smaller angles of -10°, 0°, 10°. In the latter case, using smaller fixed angles, improvements were observed only in TOP1 compared to the basic 3D transformation.

Depth Angles	Top-1	Top-5
normal angles	16.36	30.91
bin5	20.00	30.00
bin10	20.00	29.09
bin20	20.00	27.27
bin30	20.00	29.09
-45°,0°,45°	14.55	28.18
-10°,0°,10°	17.27	30.00

Table 13: Geometric transformations results on our custom dataset using different depth angles

Another experiment involved rotating all the joints and only the shoulder joints. It was observed that there was an improvement in performance in the latter case, although the accuracy based on the angle used varies in the same way regardless of the angle used. These results are represented in the histograms of Figure 11, which show the performance of only three experiments as the orientation changes.

From the results described above, it can be observed that if the figure is oriented in a straight manner, the 2D rotation leads to improvements compared to the results obtained without the geometric component. However, if the orientation changes, an additional component that accounts for depth is necessary. The 3D rotation, however, is limited by the inaccuracy of the measured depth angle. Despite the improvements observed in experiments using fixed angles (specifically -10°, 0°, 10°), the results are not reliable as they are based on an estimate of depth. Therefore, it is necessary to ensure greater precision in the input data, particularly by obtaining x-y coordinates that better identify the joints, and to increase the accuracy of the depth estimation.

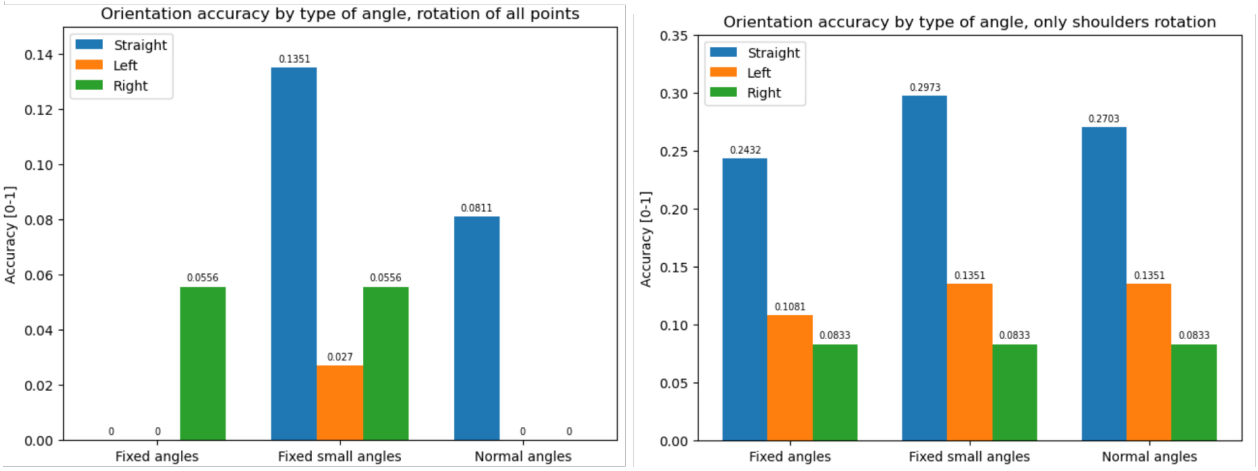


Figure 11: Performance trends obtained for each orientation using three different depth angles: *fixed_angles* (-45°, 0°, 45°), *fixed_small_angles* (-10°, 0°, 10°), and *normal_angles* (angle between the shoulder and the XY plane of the camera). On the left, the results obtained by rotating all the joints are shown, while on the right, the results are shown when only the shoulders are rotated.

5 Similar Sign Retrieval

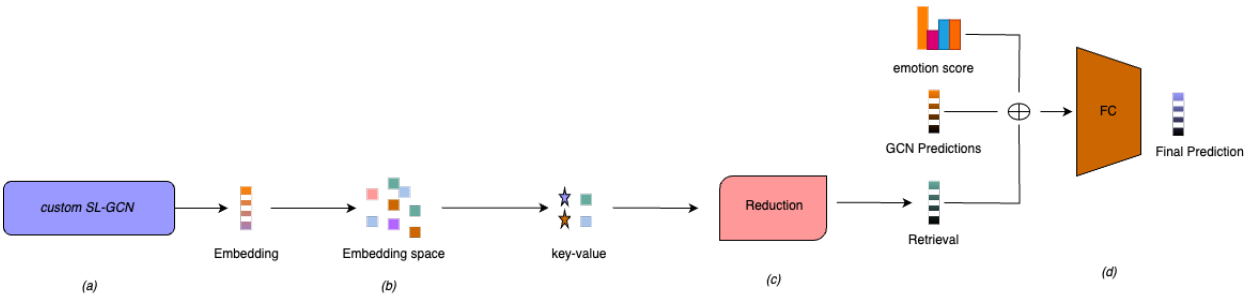


Figure 12: Illustration of the proposed SSR; (a) the embedding feature vector is generated from the SL-GCN modified; (b) compute the cosine similarity in the embedding space for produce the order dictionary of similar signs; (c) reduce the number of elements equals to the number of classes; (d) FC with the other predictions.

Besides using key points corrections and emotion enhancement , we also propose a Similar Sign Retrieval (SSR) module for recognize similar sign languages from the knowledge gained by our model and add it at the final classifier, this makes the prediction of the sign more robust to false positive sample. The Retrieval system rally on the idea that CNNs, given their hierarchical nature, are able to determine the best representation for learning or classification task [4], this allows them to model more abstract representations with respect to other low level feature. In our model, the more discriminant visual feature for detect the sign is skeleton key points. Using that information we took the SL-GCN network and removed the last fully connected layer to return the embedding feature vectors. After this update we had re-pass all the training data inside the network to generate the embeddings space. For matching similar signs we use the Cosine Similarity measure. Mathematically, the modified SL-GCN network outputs a vector e_i of size 1×256 , this vector is than compare with all the vectors of the embeddings space $E = \{e_j | j = \ldots, N\}$ where N is the number of sample we've:

$$\text{similarity}(e_i, e_j) = \frac{e_i \cdot e_j}{max(|e_i|_2 \cdot |e_j|_2, \epsilon)}$$

were ϵ is used for avoiding dividing by zero. We generated 4 embedding spaces from the four types of information coming from the skeleton: joints, bone, joints motion and bone motion. Since different networks are used to train the four parts, retrieval of similar sign from

each space also produce different results. During the experiments we noticed an interesting insights: embeddings with the same label appeared in different parts of the space, this can be explained by the presence of dialects within WLASL, where the same words are perforated differently.

5.0.1 Neural Networks with retrieval part

As illustrated in Figure 12, the SSR block constructs the embedding representation of the input data, and returning dictionary of pairs *sample;similarity* ordered by decreasing similarity. The second stage is a trasformation part, where we reduce the dictionary from N pairs of *sample;similarity* to a vector of C elements, where N is the number of elements in the embedding space and C is the number of classes, We have studied different reduction modalities:

- One Hot reduction: we created a vector of zeros apart in the c_i position, which corresponds to the label of the sign most similar found in the dictionary.
- TopK reduction: we create a vector of zero apart in the positions of the labels to which correspond the top K signs more similar, in those positions we give the same value of the similarity obtained. In our experiments we use $K = \{10, 20, 50\}$.
- Mean reduction: we create a vector where each position is the mean value of the similarities for each label.

5.0.2 Results

In this subsection, the details of training will be presented. As a case of study we decided to take only the SL-GCN of the joints, we train the Neural Networks with one hidden layer of 2004 neurons and a Dropout layer with drop probability set to 20%, learning rate is set to 1×10^{-4} with weight decay set to 1×10^{-4}. We use the Cross Entropy Loss as our loss function with SGD for optimize our parameters. After 10000 epoch the result of our proposed SLR system are reported in Table 14 in terms of Top-1 and Top-5 recognition rate. The Top50 provides the best performance among all five reduction modes, this can be explained by the presence of dialect within WLASL, by increasing the number of k more similar signs you can also increase the probability of including more true positives. All five modalities improve the overall recognition rate, which demonstrates the effectiveness of our proposed SSR module.

Modes	Top-1	Top-5
1Hot	25.30	54.02
Top10	25.52	54.06
Top20	25.30	53.91
Top50	**26.22**	**55.05**
Mean	25.26	54.17

Table 14: Performance of SSR on WLASL

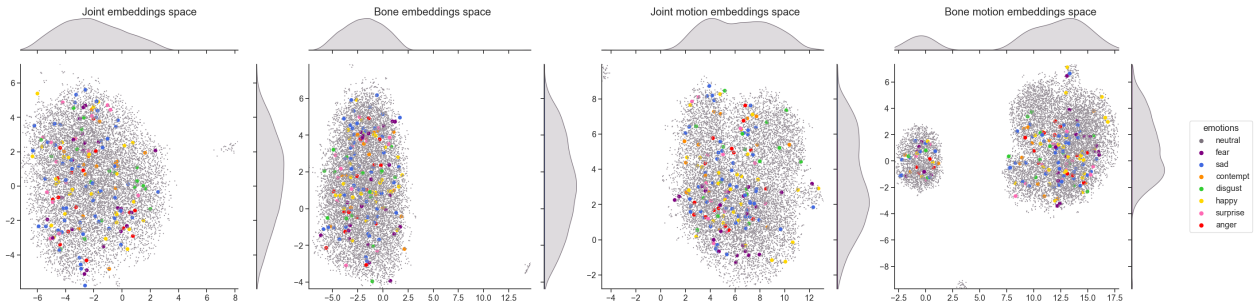


Figure 13: UMAP visualization of embedding spaces for joint, bone, joint motion and bone motion. Points represent embeddings of individual sample in the WLASL dataset. The colored dots represent embeddings whose label is an emotional word.

6 Conclusion and future research based on our results

In this paper, we presented three different components to enhance Sign Language Recognition (SLR) systems by incorporating sentiment analysis and try to improve the robustness of SAM-SLR.

Emotional enhancement of the prediction was revealed to be more complex than expected and even though we were able to slightly improve it, we still faced a significant challenge to train a network to do so. The inaccuracy of emotional facial expression estimation and an unreliable dataset (in terms of facial expression) made it impossible to improve prediction with a NN. To achieve better results it is reasonable to assume that using a reliable dataset, with a facial expression system more accurate could drastically improve results and maybe be able to improve prediction. Even Though we are not sure it would be reasonable to invest in a vast and reliable dataset to achieve an improvement over just a small subset of words that actually could benefit from facial expression emotion extraction.

The 2D geometric transformation, introduced to align the signer's body pose, showed minor improvements under certain conditions (in particular for orientations different from the straight). Instead, the 3D rotation was limited by the inaccuracies in depth estimation, indicating a need for better estimation. A future optimization is to enhance the accuracy of input data, particularly by refining x-y coordinates for an improved joint identification. Having a good depth values will be possible to apply a rotation that considers not only the depth of the shoulders but also the distance between the elbows and wrists and their depth, in order to achieve a pose correction that preserves more orientation properties.

The Similar Sign Retrieval module further strengthened ours model by retrieving visually similar signs from an embedding space constructed from high-level skeleton features. The SSR system has overall improve the robustness of our network, this enriched the power of DNNs as high-level feature extractor.Another interesting fact is the presence of dialects within sign language, this helps our model to better find near true positive example during the reduction phase. A possible research of study could be to try to merge in one only big retrieval system all the skeleton's representations (joint motion, bone and bone motion) and also try different selection methods.

Overall, while our proposed enhancements show significant promise in making SLR systems more accurate and reliable, we weren't able to achieve it with the resources available to us. Future work will focus on refining these methods with better datasets, advanced depth estimation techniques, and more robust neural network architectures to fully capture the nuances of sign language.

Useful links
Model weights
Our repository

References

[1] Eeva A. Elliott and Arthur M. Jacobs. "Facial expressions, emotions, and sign languages". In: *Frontiers in Psychology* 4 (Mar. 2013), p. 115. DOI: 10.3389/fpsyg.2013.00115.

[2] Songyao Jiang et al. "Skeleton Aware Multi-modal S ign Language Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2021.

[3] Dongxu Li et al. "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison". In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 1459–1469.

[4] Yu Han Liu. "Feature Extraction and Image Recognition with Convolutional Neural Networks". In: *Journal of Physics: Conference Series* 1087.6 (Sept. 2018), p. 062032. DOI: 10.1088/1742-6596/1087/6/062032. URL: https://dx.doi.org/10.1088/1742-6596/1087/6/062032.

[5] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[6] Ozge Mercanoglu, Anıl Tur, and Hacer Keles. In: *Isolated Sign Language Recognition with Multi-scale Features using LSTM*. Apr. 2019, pp. 1–4. DOI: 10.1109/SIU.2019.8806467.

[7] Ke Sun et al. "Deep High-Resolution Representation Learning for Human Pose Estimation". In: *CoRR* abs/1902.09212 (2019). arXiv: 1902.09212. URL: http://arxiv.org/abs/1902.09212.

[8] Zhengyao Wen. *DAN*. https://github.com/yaoing/DAN.git. 2022.

[9] Zhengyao Wen et al. "Distract Your Attention: Multi-Head Cross Attention Network for Facial Expression Recognition". In: *Biomimetics* 8.2 (May 2023), p. 199. ISSN: 2313-7673. DOI: 10.3390/biomimetics8020199. URL: http://dx.doi.org/10.3390/biomimetics8020199.

[10] Lihe Yang et al. "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data". In: *CVPR*. 2024.

[11] Qiang Zhu et al. "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 1491–1498. DOI: 10.1109/CVPR.2006.119.