Distractor-Based Evaluation of Sign Spotting

Hollain, N.; Larson, M.; Roelofsen, F.

DOI
10.1109/ICASSPW59220.2023.10193484
Publication date
2023
Document Version
Final published version
Published in
IEEE ICASSPW 2023 Workshop Proceedings (ICASSP 2023)

2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) | 979-8-3503-0261-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSPW59220.2023.10193484

DISTRACTOR-BASED EVALUATION OF SIGN SPOTTING

Natalie Hollain[1] *Martha Larson*[1] *Floris Roelofsen*[2]

[1] Radboud University, Nijmegen, The Netherlands
[2] University of Amsterdam, Amsterdam, The Netherlands
natalie.hollain@ru.nl, m.larson@cs.ru.nl, f.roelofsen@uva.nl

ABSTRACT

Sign spotting is a subtask of sign language processing in which we determine when a given target sign occurs in a given sign sequence. This paper proposes a method for evaluating sign spotting systems, which we argue to be more reflective of the degree to which a system would satisfy the user's requirements in practice than previously proposed evaluation methods. To deal with an incomplete ground truth, we introduce the concept of distractors: signs which are similar to the target sign according to a given distance measure. We assume that the performance of a sign spotting model when distinguishing a given target sign from the associated distractors will reflect the performance of the model on the complete ground truth. We develop a sign spotting model to demonstrate our evaluation method.

Index Terms— Sign language, sign spotting, machine learning

1. INTRODUCTION

Researchers in Sign Language Processing (SLP) [1, 2] have studied recognition, retrieval and spotting of sign language in video footage. In this paper, we focus on sign spotting: given a video of continuous signing, we want to determine when a given target sign occurs. This task is different from recognition and retrieval since it requires determining *when* a sign is performed in continuous footage. Multiple signs can be performed in one segment, while the other tasks use segments with one isolated sign each. Sign spotting is a relatively new field of research, and current approaches could be more successful if some issues were addressed.

We hypothesize that one important challenge is the lack of established evaluation metrics. Metrics are borrowed from other related fields without any adaptations and it is unclear how well these existing metrics reflect users of sign spotting systems. A focus on effective metrics for real-life applications of SLP has been stressed by previous research [1].

We propose an evaluation method and failure analysis for sign spotting. It is based on the *tolerance to irrelevance* idea [3], which is, in turn, based on the assumption that users, when given a certain entry point in a video or audio stream, keep watching/listening until their tolerance to irrelevant elements is reached. Our evaluation incorporates knowledge about sign language and user needs to reflect real-life applications.

In practical situations, it is often necessary to be able to evaluate on video footage that is not fully transcribed. In other words, there is no complete ground truth. To deal with this, we introduce the concept of distractors: signs which are similar to a target sign according to a given distance measure. We use distractors in order to systematically incorporate challenging, difficult-to-distinguish, cases into our evaluation method, pushing it, we assume, to reflect performance on the full ground truth. We develop a sign spotting model that uses linguistic features to demonstrate our evaluation, using a linguistic distance measure. Our code is available on Github[1].

2. RELATED WORK

2.1. Sign language phonology and phonetics

In sign language phonology, the manual articulation of a sign is generally described in terms of four parameters: handshape, location, orientation, and movement [4, 5, 6, 7, 8, 9].

The *handshape* parameter describes how the fingers of the hand(s) are configured. The *location* parameter indicates where in signing space the sign is articulated. Signs can be performed in *neutral space*, the space in front of the signer's torso, or at a specific body part, such as the head or shoulder. In a two-handed sign, the location of the hands relative to each other is relevant. *Orientation* is specified in an absolute or relative manner. Absolute orientation defines the orientation of the hand in space without making reference to how it is positioned relative to the body. By contrast, relative orientation takes the signer's body as the frame of reference. Absolute orientation is the dominant perspective in most phonological models. The *movement* of the hand(s) is specified in terms of path shape, size, direction, tensity and repetition.

Of course, phonological characteristics only partly determine the concrete realisation of signs. Other factors that may play a role include the age and gender of the signer, as well as what signs precede and follow the uttered sign. How such factors affect the concrete realisation of a sign is investigated in sign language phonetics [10, 11]. In addition to a manual component, some signs also involve non-manual components, such as particular mouth shapes, facial expression and/or body postures [12].

Our longer-term goal is to develop more accurate and explainable sign spotting systems by incorporating relevant insights from sign language phonology and phonetics. In this paper, we start by taking the basic phonological parameters into account—handshape, orientation, location, and movement—leaving phonetic factors and the contribution of non-manual articulators for future work.

2.2. Sign spotting

We discuss a selection of work in sign spotting that is relevant to our paper. Previous work in sign spotting has used a variety of tools, including dynamic time warping (e.g. [13, 14]), conditional random fields (e.g. [15, 16]) and hidden Markov models (e.g. [17]). Typically, the datasets these methods were applied to only contained a small set of signs and signers. Recently, the focus has shifted to applying deep learning architectures such as 3D convolution [18, 19].

[1] https://github.com/nataliehh/Evaluating-Sign-Spotting

We highlight in particular the work of [20], which proposed a framework called 'watch, read and lookup' for continuous sign spotting. Sign spotting embeddings are learned from watching sparsely annotated videos, reading subtitles to find candidate signs and looking up examples in a SL video dictionary. The data used consist of interpreted signing, which is distinct from 'natural signing' that is performed faster and less distinctly [1]. Thus, the applicability of this approach to real-life circumstances is unclear.

Notably, the work we surveyed typically operates on raw pixel information. The necessity of using linguistically relevant features for SLP has been stressed by previous work [2]. Our model thus implements approximative phonological features, as opposed to raw pixel input.

2.3. Sign recognition

Some models that use sign language linguistics have emerged for sign recognition. To the best of our knowledge, the papers below extract some form of linguistic information.

In [21], hand landmarks were extracted using Mediapipe. Then, angles and distances between the landmarks were computed and used to recognize which letter of a sign language alphabet is being signed in an image. A similar setup is used in [22]. Alphabet signs were recognized by computing angles between specific landmarks and between the slopes of the fingers. In both papers, only static sign language recognition was tested. The first paper we are aware of that is applied to a small set of isolated, dynamic signs is [23]. The authors extracted both angles as well as distances between the landmarks, which relate to the way the hand is configured and could capture the linguistic handshape. These features are then fed to a model architecture which uses LSTM layers. We build on this work by extending it to sign spotting, where multiple signs can occur in the same video segment. Moreover, we introduce new features to capture the movement and orientation of the hand.

2.4. Evaluation

To the best of our knowledge, few research contributions have proposed evaluation metrics for sign spotting. In [13], a tool to assess the spotting performance on continuous signing was developed. The evaluation is symmetrical, which is not considered ideal for retrieval metrics [24]. In fact, it is not clear how such a metric reflects the real use case of a sign spotting system. Allowing for spottings to begin after the ground truth has started has been shown to be annoying to users [25].

For the task of audio segmentation, the F-measure appears to be one of the more popular metrics [26, 27, 28]. A set of restrictions is imposed on the metric when it is applied to audio. In particular, a tolerance window is used that restricts the boundaries of the predictions to be close to the ground truth boundaries. Typically, two tolerances are used: 0.5 and 3 seconds [26, 27, 28]. The 3 second tolerance window is supported by the fact that users need about 3 seconds to adjust to viewing a result item [3]. For the 0.5 second tolerance, it is most similar to the strict tolerances applied to the task of precise event spotting [29].

Like the sign spotting metric in [13], audio segmentation metrics and tolerances are used in a symmetrical manner. We distinguish our approach from them by proposing an asymmetrical evaluation.

3. DISTRACTOR-BASED EVALUATION OF SIGN SPOTTING

Our sign spotting system presents the user with a list of time points from which the user can start watching the video to see a relevant segment. We adapt *tolerance to irrelevance* (TTI) [3] to create a metric to evaluate sign spotting. This metric only considers the starting time t_{p_i} of a predicted occurrence p_i of the target sign, and the starting time t_{s_i} of an annotated occurrence s_i of the target sign. A prediction is considered correct if its starting time falls within some window of tolerance, tol, before the ground truth: $t_{pi} \in [t_{si} - tol, t_{si}]$. TTI is thus asymmetric, where a prediction is only tolerated when it starts a bit before or at the same time as an annotated occurrence.

The tolerance window tol takes on different values depending on the domain the metric is used in. No specific tolerance window sizes have been determined for SLP. Therefore, we use the tolerance windows of 0.5 and 3 seconds from the related field of audio segmentation. We add a tolerance window of 0.3 seconds, based on an analysis showing that this is the median duration of a sign in our dataset (see Section 4.1). Thus, we use the following tolerance window sizes (in seconds): $W = [0.3, 0.5, 3]$. For footage that runs at 25 fps, like the data we introduce in Section 4.1, these window sizes are equivalent to $[7.5, 12.5, 75]$ frames.

Note that datasets used in SLP are often not fully annotated, as is the case for the dataset we use (see Section 4.1). This means we are dealing with an incomplete ground truth for which a timespan without annotation may simply need to still be annotated, instead of being a timespan where nothing is signed. We address this issue by selecting a subset of annotations over which we have full control, which we take as representative of a full sign-by-sign transcription. This subset is selected based on a distance measure: given a target sign, we find the most similar signs in terms of this measure. We define these similar signs as *distractors*. We assume that, if our model is able to ignore these similar distractors, more dissimilar signs will also be ignored. Thus, the performance on this subset of difficult distractors should reflect the performance on the full ground truth.

In this paper, we use *linguistic distance* as our distance measure: distractors are chosen based on their phonology as described in Section 2.1. We elaborate on the implementation of this distance measure in Section 4.4. Other distance measures can be chosen depending on the application for which this evaluation is used.

To define the true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) of our evaluation, we define the following notation, given a target sign S:

$P(S)$	$= \{p_1, \ldots, p_n\}$	predicted occurrences of S
$T_{P(S)}$	$= \{t_{p_1}, \ldots, t_{p_n}\}$	starting times of $p_1, \ldots, p_n$
$A(S)$	$= \{s_1, \ldots, s_m\}$	annotated occurrences of S
$T_{A(S)}$	$= \{t_{s_1}, \ldots, t_{s_m}\}$	starting times of $s_1, \ldots, s_m$
$D(S)$	$= \{d_1 \ldots d_l\}$	distractors for S
$T_{D(S)}$	$= \{t_{d_1} \ldots t_{d_l}\}$	starting times of $d_1 \ldots d_l$

We then evaluate as follows:

$$
\begin{aligned}
TP: &\quad \exists t_{pi} \in T_{P(S)} : t_{pi} \in [t_{sj} - tol, t_{sj}] \\
FN: &\quad \forall t_{pi} \in T_{P(S)} : t_{pi} \notin [t_{sj} - tol, t_{sj}] \\
FP: &\quad \exists t_{pi} \in T_{P(S)} : t_{pi} \in [t_{di} - tol, t_{di}] \\
TN: &\quad \forall t_{pi} \in T_{P(S)} : t_{pi} \notin [t_{di} - tol, t_{di}]
\end{aligned}
$$

To ensure that the predictions do not overlap with more than one tolerance window, we select $D(S)$ such that their tolerances do not overlap with each other or with the tolerances of $A(S)$. Furthermore, if multiple predictions spot the same annotation, we only count the first TP or FP spotting and assume all other matching predictions can

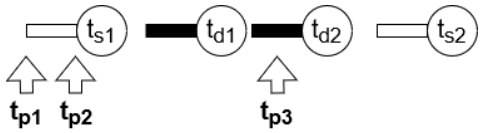

Fig. 1. Example of predicted spottings for a target sign and its distractors

be ignored because they could be aggregated into one prediction. We leave the aggregation of the predictions to future work.

In Figure 1, let us assume we are spotting sign S, such that $T_{A(S)} = \{t_{s1}, t_{s2}\}$. We also define the start times of distractors: $T_{D(S)} = \{t_{d1}, t_{d2}\}$. The white bars show the tolerance windows of the targets, while the black bars indicate the tolerance windows of the distractors.

t_{p1} is a TP spotting because it falls within the tolerance window of t_{s1}. t_{p2} is discarded since t_{s1} is already spotted by t_{p1}. t_{p3} is a FP spotting, as it falls within the tolerance window of t_{d2}. For t_{d1}, we see that there are no predictions at all. Thus, this is counted as a TN. In the case of t_{s2}, which is also not spotted, we find a FN.

From this, we define $accuracy = \frac{TN+TP}{TN+TP+FN+FP}$. The accuracy for a specific tolerance can then be specified as $acc@tol$, where tol is the tolerance in seconds. We can also specify other metrics that are defined for binary classification, such as recall and precision, in the same manner. Failure analysis can be performed by analyzing the false classifications, i.e. the FPs and FNs.

4. APPLYING DISTRACTOR-BASED EVALUATION

4.1. Datasets

We use the Corpus Nederlandse Gebarentaal (CNGT) [30, 31] to assess our sign spotting metric. It consists of 72 hours of video footage, recorded at 25 fps, of 104 signers of Dutch Sign Language (NGT). About 15% of the corpus is annotated, totaling 162k annotations of 3.2k signs.

The signs used in CNGT are labelled based on the NGT lexicon in Global Signbank [32]. This lexicon also includes the phonological features discussed in Section 2.1 for each sign.

We selected CNGT as our dataset because it matches most of the criteria for suitable datasets set out in [1]. CNGT provides footage of 'natural' signing, where the signers are in conversation and are not trying to sign in a more proper manner than usual [30]. Furthermore, the signers have a variety of backgrounds and are native signers. These characteristics make CNGT a good basis for SLP, as the data resemble those encountered in practical use cases [1].

We split the annotations into a training, validation and test set. The training set contains different signers from the validation and test set. We use only signs seen during training and for which the linguistics are annotated in NGT Signbank. This leaves us with 118k annotations of 2.7k signs. Our split is approximately 80/10/10, with 90k training, 10.5k validation and 9.5k test instances. The training data is augmented by mirroring the footage in the training set, effectively doubling the number of training instances.

Annotations in CNGT are of variable length, which we convert to one fixed input length by zero-padding annotations that are too short and undersampling those that are too long. We choose a fixed input length of 10 frames, which is the mean duration of all annotations.

For each frame, we extract landmarks using Mediapipe. The landmarks are already normalised by Mediapipe using the video dimensions, we reverse this to convert them to pixel coordinates. Then, we normalise each landmark by the distance between the shoulders and afterwards center them by subtracting the midpoint of the shoulders as described in [33].

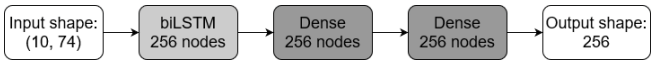

Fig. 2. Model architecture

4.2. Linguistically relevant features

We extract features which aim to represent the basic phonological parameters:

- Handshape: the distances and angles between the fingertips, handpalm and wrist.
- Orientation: the angle of the handpalm relative to the torso and the shoulders.
- Location: the x, y coordinates of the wrist and the fingertips.
- Movement: the x, y coordinates of the wrist and the fingertips as well as the velocity of the wrist.

This results in 174 features, 87 for each hand. The features of the handshape were based on [23] and supplemented with extra features that could help better capture the curvature and spread of the fingers. We then remove any features which have a correlation ≥ 0.90 to help with model convergence, leaving 74 out of 174 features. The discarded features are mostly ones describing distances between landmarks on the hand, as well as coordinates of the fingertips.

We augment the training data again to ensure the mirrored and non-mirrored examples are not too similar. The mirror augmentation only changes which features belong to which hand, with the angles and distances staying identical except for the change of hands. Thus, we add another augmentation step where for each value x, we randomly shift it to a value between $[0.99x, 1.01x]$. We found that this augmentation improves the training stability on our validation set.

4.3. Model architecture

Our sign spotting model is shown in Figure 2. It consists of a biLSTM layer with 256 nodes, followed by two Dense layers that also have 256 nodes. Due to the strength of contrastive loss reported in the literature (e.g. [20]), we use supervised contrastive loss to train our model [34]. We adapt the loss by normalising the vectors that are used in the computation of the dot-product between two embeddings. This effectively introduces cosine similarity into the loss and allows us to make a more direct connection to the testing phase where this similarity measure is used too. The temperature hyperparameter is set to $\tau = 0.07$ as in [20, 35, 36]. We optimize our contrastive loss using the Adam optimizer. Exploratory experimentation on the validation set was used to determine the architecture and hyperparameters of our neural network.

The model is trained for 10 epochs, which is when it typically converges on the validation set. The batch size is set to 128 and the learning rate to 0.001 based on [20].

We now apply our evaluation method to assess the performance of our model. We set up a testing phase for our model to assess the $acc@tol$ metric. First, we take all videos for which we have test set annotations. We then use a sliding window to make embeddings for

the video, taking a window size of 10 frames to match the fixed annotation length of the training data. We can then calculate the cosine distance between each sliding window embedding and a target sign.

Each target sign has been seen multiple times during training, thus using each train embedding individually requires many comparisons. To reduce the number of comparisons, we compute *reference embeddings* for each sign. This is done by first computing embeddings for all occurrences of a given sign in the training set. We then compare these embeddings to each other, finding which of them are on average closest to all other embeddings. These embeddings are then chosen to be the most representative of the sign. We use the top 10% most representative embeddings and average them to make one reference embedding for the sign.

Next, we define how the reference embedding of a target sign is spotted in a test set video. If the cosine distance is less than or equal to our spotting threshold t, we determine that the sign is spotted. The first frame of the spotting is chosen as the prediction's start t_{pi} for the target sign S. We empirically found $t = 0.2$ to be a fitting threshold for our model on our validation set.

4.4. Selecting distractors

Our evaluation uses linguistic distance to determine which non-target annotations count as distractors. To compute the linguistic distance, we use NGT Signbank. We use 14 attributes described in this dataset, which together describe the manual phonology of a sign in terms of handshape, orientation, location and movement. Given two signs, we compare them per attribute. If the signs differ in an attribute, we increase the linguistic distance of the signs by one.

Currently, not all attributes are described for each sign in NGT SignBank. To deal with this, we assume that comparisons of unknown values can be ignored and leave a more fine-grained handling of unknown attributes to future work.

After we compute the linguistic distance between all signs, we can select distractors for our evaluation. For a target sign S, we find its frequency f in a target video. Then, we select f distractors in the same video. Those most similar to the target are chosen, such that we use the top-f most similar distractors. In this paper, if in the given video we cannot find f distractors, we disregard this video for the given target sign. Depending on the application for which the evaluation is used, more or less distractors can be selected.

4.5. Results

In Table 1, the evaluation of our sign spotting model is shown using $acc@tol$, $recall@tol$ ($\frac{TP}{TP+FN}$) and $precision@tol$ ($\frac{TP}{TP+FP}$). Recall increases for higher tolerances, whereas precision decreases. Larger tolerances cause more target annotations to be counted but also more distractors. Thus, a balance has to be struck between recall and precision, which is further demonstrated by the accuracy which decreases at $tol = 3$. Depending on what type of application the evaluation is used for, one of the metrics could be prioritized. The tolerance level and spotting threshold can be adapted to the specific use case. We decided on a fixed threshold since users of our model should not have to adapt such hyperparameters themselves, but it is possible to use metrics that summarize multiple thresholds, such as AUC.

Table 1. Model performance for distractor-based metrics

	$tol = 0.3$	$tol = 0.5$	$tol = 3$
TP	2701	3280	4078
FN	4046	3467	2669
FP	608	827	1971
TN	6139	5920	4776
$acc@tol$	0.655	0.682	0.656
$precision@tol$	0.816	0.799	0.674
$recall@tol$	0.4	0.486	0.604

5. CONCLUSION AND OUTLOOK

We have introduced a distractor-based evaluation method, intended to support research on sign spotting. Moving forward, our approach to extracting features from detected landmarks that correspond to a sign's phonological makeup could be further refined. Moreover, the adopted linguistic distance measure could be improved. Future work may further investigate how to deal with signs with incomplete phonological characterisations. Furthermore, our work did not incorporate mouthing and other non-manual features into the distance computation. Since some signs only differ in these features, it seems promising to study their incorporation. Of course, different distance measures could be implemented to study the effect on the evaluation and which distractors are selected. Lastly, the number of distractors that is optimal for specific applications was not studied and is left to future research.

6. ACKNOWLEDGMENTS

We thank Onno Crasborn for providing us with the data of Corpus NGT. We also thank Jasper de Meijer and Javier Martínez Rodríguez for their valuable support of this project.

7. REFERENCES

[1] Danielle Bragg et al., "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *The 21st Int. ACM SIGACCESS Conf. on Computers and Accessibility*, 2019, pp. 16–31.

[2] Amit Moryossef and Yoav Goldberg, "Sign Language Processing," https://sign-language-processing.github.io/, 2021, Accessed: Jan. 27, 2023.

[3] Arjen P De Vries, Gabriella Kazai, and Mounia Lalmas, "Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit," in *RIAO Conf. Proc.*, 2004, pp. 463–473.

[4] William Stokoe, "Sign language structure, an outline of the visual communications systems of american deaf," *Studies in linguistics occasional paper*, vol. 8, 1960.

[5] Robbin Battison, *Lexical borrowing in American sign language.*, Silver Spring, MD: Linstok Press, 1978.

[6] Els Van der Kooij, *Phonological categories in Sign Language of the Netherlands: The role of phonetic implementation and iconicity*, LOT, 2002.

[7] Diane Brentari, *Sign Language Phonology*, Cambridge University Press, 2019.

[8] Diane Brentari, Jordan Fenlon, and Kearsy Cormier, "Sign language phonology," in *Oxford Research Encyclopedia of Linguistics*. Oxford University Press, 2018.

[9] Wendy Sandler, "The phonological organization of sign languages," *Language and Linguistics Compass*, vol. 6, no. 3, pp. 162–182, 2012.

[10] Onno Crasborn, "Phonetics," in *Sign language: An international handbook*. De Gruyter, 2012.

[11] Martha Tyrone, "Phonetics of sign language," in *Oxford Research Encyclopedia of Linguistics*. Oxford University Press, 2020.

[12] Nina-Kristin Pendzich, *Lexical nonmanuals in German Sign Language: Empirical studies and theoretical implications*, De Gruyter, 2020.

[13] Ville Viitaniemi, Tommi Jantunen, Leena Savolainen, Matti Karppa, and Jorma Laaksonen, "S-pot–a benchmark in spotting signs within continuous signing," in *Proc. of the 9th Int. Conf. on Language Resources and Evaluation (LREC 2014)*. European Language Resources Association (LREC), 2014, pp. 1892–1897.

[14] Srujana Gattupalli, "Sign gesture spotting in american sign language using dynamic space time warping," 2013.

[15] Seong-Sik Cho, Hee-Deok Yang, and Seong-Whan Lee, "Sign language spotting based on semi-markov conditional random field," in *2009 Workshop on Applications of Computer Vision (WACV)*. IEEE, 2009, pp. 1–6.

[16] Hee-Deok Yang and Seong-Whan Lee, "Simultaneous spotting of signs and fingerspellings based on hierarchical conditional random fields and boostmap embeddings," *Pattern Recognition*, vol. 43, no. 8, pp. 2858–2870, 2010.

[17] Mahmoud Elmezain, Ayoub Al-Hamadi, Jorg Appenrodt, and Bernd Michaelis, "A hidden markov model-based continuous gesture recognition system for hand motion trajectory," in *2008 19th international conference on pattern recognition*. IEEE, 2008, pp. 1–4.

[18] Tao Jiang, Necati Cihan Camgöz, and Richard Bowden, "Looking for the signs: Identifying isolated sign instances in continuous video footage," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 2021, pp. 1–8.

[19] Ryan Wong, Necati Cihan Camgöz, and Richard Bowden, "Hierarchical i3d for sign spotting," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer, 2023, pp. 243–255.

[20] Liliane Momeni, Gul Varol, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman, "Watch, read and lookup: learning to spot signs from multiple supervisors," in *Asian Conf. on Computer Vision*, 2020.

[21] Jungpil Shin, Akitaka Matsuoka, Md Al Mehedi Hasan, and Azmain Yakin Srizon, "American sign language alphabet recognition by extracting feature from hand pose estimation," *Sensors*, vol. 21, no. 17, pp. 5856, 2021.

[22] Muhammad Jamil Hussain et al., "Intelligent sign language recognition system for e-learning context," *Computers, Materials & Continua*, vol. 72, no. 3, pp. 5327–5343, 2022.

[23] Youssef Farhan and Abdessalam Ait Madi, "Real-time dynamic sign recognition using mediapipe," in *2022 IEEE 3rd Int. Conf. on Electronics, Control, Optimization and Computer Science (ICECOCS)*. IEEE, 2022, pp. 1–7.

[24] Maria Eskevich, Walid Magdy, and Gareth JF Jones, "New metrics for meaningful evaluation of informally structured speech retrieval," in *Advances in Information Retrieval*. 2012, pp. 170–181, Springer Berlin Heidelberg.

[25] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin, "Auto-summarization of audio-video presentations," in *Proc. of the seventh ACM Int. Conf. on Multimedia (Part 1)*, 1999, pp. 489–498.

[26] Anna Aljanaki, Frans Wiering, and Remco C Veltkamp, "Emotion based segmentation of musical audio," in *Proc. of the 16th Conf. of the Int. Society for Music Information Retrieval (ISMIR 2015)*, 2015, pp. 770–776.

[27] Jordan BL Smith and Elaine Chew, "A meta-analysis of the mirex structure segmentation task," in *Proc. of the 14th Int. Society for Music Information Retrieval Conference, Curitiba, Brazil*, 2013.

[28] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie, "Design and creation of a large-scale database of structural annotations.," in *Proc. of the 12th Int. Society for Music Information Retrieval Conference (ISMIR)*. Miami, FL, 2011, pp. 555–560.

[29] James Hong, Haotian Zhang, Michaël Gharbi, Matthew Fisher, and Kayvon Fatahalian, "Spotting temporally precise, fine-grained events in video," in *Computer Vision–ECCV 2022*. Springer, 2022, pp. 33–51.

[30] Onno Crasborn and Inge Zwitserlood, "The Corpus NGT: an online corpus for professionals and laymen," *Construction and Exploitation of Sign Language Corpora. 3rd Workshop on the Representation and Processing of Sign Languages*, pp. 44–49, 2008.

[31] Onno Crasborn, Inge Zwitserlood, and Johan Ros, "The corpus ngt. an open access digital corpus of movies with annotations of sign language of the netherlands.," *Centre for Language Studies, Radboud University Nijmegen*, 2008.

[32] Onno Crasborn et al., "NGT Signbank," *Centre for Language Studies, Radboud University Nijmegen*, 2014.

[33] Sait Celebi, Ali Selman Aydin, Talha Tarik Temiz, and Tarik Arici, "Gesture recognition using skeleton data with weighted dynamic time warping.," *VISAPP 2013 - Proc. of the Int. Conf. on Computer Vision Theory and Applications*, vol. 1, pp. 620–625, 2013.

[34] Prannay Khosla et al., "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. of the IEEE/CVF conf. on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[36] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. of the IEEE conf. on computer vision and pattern recognition*, 2018, pp. 3733–3742.