Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
17/2 (2024), 195-202. DOI: http://dx.doi.org/10.21609/jiki.v17i2.1284

Temporal Action Segmentation in Sign Language System for Bahasa Indonesia (SIBI) Videos Using Optical Flow-Based Approach

I Dewa Made Bayu Atmaja Darmawan[1], Linawati[2], Gede Sukadarmika[2], Ni Made Ary Esta Dewi Wirastuti[2] and Reza Pulungan[3]

[1]Doctoral Engineering Department, Udayana University, Bali, Indonesia
[2]Electrical Engineering Department, Udayana University, Bali, Indonesia
[3]Computer Science and Electronics Department, Universitas Gadjah Mada, Yogyakarta, Indonesia

E-mail: dewabayu@unud.ac.id

Abstract

Sign language (SL) is vital in fostering communication for the deaf and hard-of-hearing communities. Continuous Sign Language Translation (CSLT) is a work that translates sign language into spoken language. CSLT translation is done by changing continuous forms into isolated signs. Segmenting morpheme signs from phrase signs has several challenges, such as the availability of annotated datasets and the complexity of continuous gesture movements. The Indonesian Sign Language (SIBI) system follows Indonesian grammatical norms, including word formation, in contrast to other sign languages with rules derived from their spoken language. In SIBI, a word can consist of a root word and an affix word. Therefore, temporal action segmentation in SIBI is important to reconstruct the results of translating each sign into spoken Indonesian sentences. This research uses an optical flow approach to segment temporal actions in SIBI videos. Optical flow methods that calculate changes in intensity between adjacent frames can be used to determine the occurrence of sign movement or vice versa to determine the delay between sign movements. The absence of intensity differences between the two frames indicates the boundary between sign gestures. This study tested the use of dense optical flow on videos containing SIBI sentences taken from 3 signers. Evaluation is done on several parameters in the dense optical flow algorithm, such as threshold size, PyrScale, and WinSize, to obtain the best accuracy. This paper shows that the optical flow algorithm successfully performs segmentation, as measured by Perf and F1r. The experimental results showed that the highest Perf and F1r yields were 0.8298 and 0.8524, respectively.

Keywords: *dense optical flow, farneback, sign language, segmentation, sibi*

1. Introduction

Sign language is crucial for fostering accessibility and inclusivity within deaf and hard-of-hearing communities. World Health Organization (WHO) estimates that in 2050, one in every four individuals will have a hearing impairment [1]; thus, sign language proficiency is crucial. Accurate sign language recognition and comprehension are essential for facilitating effective communication between non-sign and sign language users.

Countries that share a common spoken language may also have distinct sign languages. Special education incorporates the Indonesian Sign System for Bahasa Indonesia (SIBI), a formal sign language utilized in Indonesia. SIBI employs Indonesian rules encompassing vocabulary and grammar, which may comprise root and inflectional words. As an illustration, the word *membaca* (reading) is structured by appending the prefix *me* to the root *baca*. For comparison, in Indonesian Sign Language (BISINDO), reading sign words is done with a sign. Likewise, affixes do not have sign movements in American Sign Language (ASL), which is the source of SIBI [2]. This characteristic differentiates SIBI from other sign languages; in fact, SIBI is not a sign language but a sign system that refers to the spoken language Indonesian [3].

Continuous sign language recognition (CSLR) converts some of the words or sentences of sign

language into spoken language sentences. Temporal segmentation for sign language translation is still an open research problem because of the low accuracy obtained. Alternatively, Hidden Markov Model (HMM) and Connectionist Temporal Classification (CTC) have been used to perform segmentation or sequence labeling. However, both methods require a learning stage requiring extensive training data. In addition, annotated video training data requires additional time and costs.

The temporal action segmentation (TAS) method can transform continuous shapes into isolated signs to facilitate translation into oral language at the frame level. TAS involves identifying and isolating individual sign movements in a continuous video sequence. The segmentation process is critical to enable the sign language recognition system to accurately recognize the significance of every gesture to allow a smooth translation. Nevertheless, researchers face the obstacle of achieving accurate segmentation through appropriate sign movement [4]. Recognizing gestures for formed words without segmentation can result in poor accuracy, as was done by Halim and Rakun [5], who applied TensorFlow and Long Short-Term Memory (LSTM) for SIBI.

Temporal action segmentation (TAS) is crucial in sign language video analysis and understanding. It involves dividing a sign language video into smaller segments based on the temporal boundaries of individual signs or actions the signer performs, as depicted in Figure 1. This segmentation process enables researchers and developers to analyze and extract meaningful information from sign language videos, leading to advancements in areas such as sign language recognition. SIBI uses the same rules as a spoken language, making it unique compared to other sign languages. TAS segments untrimmed continuous sign language phrases so that the constituent morphemes can be identified. Although there have been significant advances in temporal segmentation techniques in recent years, applying TAS using an Optical Flow-based method to SIBI sign language videos has never been carried.

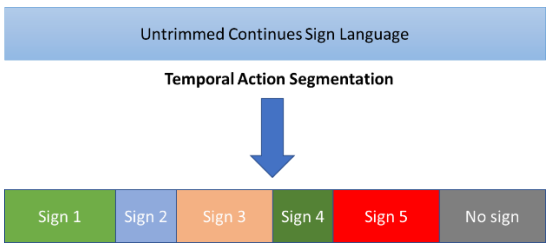


Figure 1. The TAS model segments untrimmed continuous sign language video sequences into several isolated sign language video segments.

Since the emergence of deep learning architectures, many studies have tried to use deep learning approaches with various architectural variations to solve some specific issues. In the video segmentation domain, deep learning architectures have been implemented in at least three types of networks, such as convolution networks [6], [7], [8], [9], [10], [11], Recurrent Neural Networks (RNN) [12], [13], [14] and Transformers (attention-based architectures) [15], [16], [17]. However, it is common knowledge that deep learning requires a lot of data as training data and has high computational costs for building models, so it is not discussed in this paper.

Pre-computed frame-wise features using optical flow have much cheaper computational costs compared to learning models (learning video features) [18], [19]. A method was introduced for the optical flow-based still image to video segmentation models, resulting in increased stability [20]. Optical flow has emerged as an effective tool for motion analysis in video processing. Dense motion vectors between consecutive frames are computed by optical flow, yielding significant temporal insights into pixel motion patterns. This information can be used to identify the beginning and end of a sign gesture.

Furthermore, using optical flow in sign language video segmentation addresses the challenge of co-articulation, where the previous sign influences one sign. The motion feature that uses Optical Flow can be used to detect shot boundaries [21]. In this research, shot boundaries are detected using the projection feature to obtain candidate boundary frames and the motion feature to remove non-boundary frames from the candidate frames provided by the projection feature. The proposed method successfully detects the transition of an abrupt shot from the presence of motion and changes in the illumination of video sequences.

Research on temporal segmentation is also closely related to motion detection or object movement tracking. The Canny Edge and Optical Flow (CE-OF) [22] method can detect and track moving objects with curation and precision above 90%. Apart from object tracking, Optical Flow can also be used to calculate movement speed [23], so it can potentially be used to extract information related to the speed of movement of signs, a non-manual component in sign language.

This research proposes an innovative approach that leverages optical flow-based temporal segmentation for sign language videos. By using dense optical flow, our method aims to achieve precise, robust, and lightweight sign language segmentation, thereby enabling more detailed analysis of sign language sequences at a later stage

while maintaining computational costs. The proposed TAS method is expected to obtain acceptable accuracy to become a solution for the segmentation stage in the SIBI sign language recognition framework [24].

2. Methodology

2.1 Temporal Action Segmentation (TAS)

Temporal action segmentation (TAS) divides a video into segments or intervals representing different actions or activities humans or objects perform. It involves analyzing a video sequence's temporal structure to identify and mark the boundaries between different actions. The aim is to accurately segment and label each action or activity, allowing for a more detailed understanding and analysis of the sequence [18].

Practically, TAS is implemented using pre-computed frame-wise features as input because it avoids the more significant computational load required for learning video features [18]. In cases where the movement of objects is dynamic and there may be dependencies between video frames, a temporal or sequential model that uses a learning model such as a Convolutional Network, RNN, and Transformer is needed. Each approach has a trade-off between complexity and computational speed. Accurate but high computational costs can be less suitable for real-time applications that require fast processing. The model proposed in this paper uses frame-wise features using dense optical flow to determine the boundaries between signs that can be detected because there is a change from stillness to movement by the signer at the beginning or pause between signs.

2.2 Dense Optical Flow

Optical flow algorithms aim to estimate the velocity of an object or pixel between successive frames in a sequence. Optical Flow observes the movement of objects by measuring variations in image intensity over time caused by the object's movement. Optical Flow assumes an object's pixel intensity does not change over successive frames. The intensity of the image *I(x,y,t)*, after time *dt* has moved a distance *dx* and *dy* can be calculated by:

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \quad (1)$$

By using the Taylor's series expansion of (1), the optical flow equation can be derived:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dI}{dy} + \frac{\partial I}{\partial t} = 0. \quad (2)$$

Dense optical flow is a notion in computer vision that refers to the computation of motion vectors for each pixel in an image or frame of a video. It aims to produce a dense representation of the flow field by calculating the motion at each pixel in the picture. Dense optical flow considers every pixel in the image, giving a more thorough and in-depth comprehension of the motion throughout, in contrast to sparse optical flow, which simply estimates motion at particular areas or features.

An approach for dense optical flow that estimates object motion in a video sequence is called the Farneback Optical Flow [25]. The Farneback Optical Flow algorithm uses the Taylor series approximation to estimate the flow between image frames, while polynomial expansion is used to represent the local image structure. It uses polynomial expansion to represent the local image structure after dividing the image into a grid of small regions. Next, by comparing the coefficients of the polynomials, the program calculates the motion between these regions. Farneback Optical Flow estimates each neighboring pixel with a polynomial:

$$f(x) \sim x^T Ax + b^T x + c. \quad (3)$$

The implementation of the Farneback Optical Flow algorithm was made using the functions of OpenCV [26]. The calcOpticalFlowFarneback function compares two adjacent frames as input and returns the computed flow image. In addition, the function accepts 7 input parameters, consisting of the first 4 parameters (pyr_scale, levels, winsize, and iterations) specifying the pyramidal approach, the 2 parameters (poly_n and poly_sigma) correspond to the polynomial expansion in each pixel, and the last parameter is the operation flags which must be done with a priori displacement fields and optical flow estimation smoothing filters.

2.3 Dataset

The data in this study was collected from three special school teachers who mastered SIBI. There are five sentences commonly used by students in the school environment. Each sentence was recorded ten times, so the total data used in this study was 150 videos. This study arranges sentences with affixes to ensure the proposed algorithm correctly segments affixes and root words. Table 1 shows the five sentences used in the study, consisting of some words and their signs. The affixes used in the sentences are as follows: reading, late, exercising, together, and sweeping. Video is captured using a mobile phone camera, with video specifications of 720 × 1280 resolution (portrait layout) and a frame rate of 24 fps.

Table 1. Dataset properties.

No	Sentences	Num. of Words	Num. of Signs
1	*Saya senang membaca buku* (I enjoy reading books).	4	5
2	*Saya terlambat bangun tidur* (I woke up late).	4	5
3	*Saya suka berolahraga* (I like exercising).	3	4
4	*sekarang kita makan bersama* (we eat together now).	4	5
5	*saya tidak suka menyapu di kelas* (I don't like sweeping the class).	6	7

2.4 Performance Metric

This study measures segmentation performance using Perf and F1r performance metrics. Perf [27] combines F1-Score and Accurate Temporal Segmentation Rate (ATSR). Perf provides an analysis of the accuracy of the location of key frames at the frame level and also considers the Precision (P) and Recall (R) used in calculating the F1-Score. Perf can be expressed by:

$$Perf = 10 \times \frac{ATSR \times P \times R}{4 \times ATSR + P + R}, \quad (4)$$

where P is precision and R is recall. ATSR is defined by:

$$ATSR = 1 - \frac{1}{n} \times \sum_{i=1}^{n} ATSE(i), \quad (5)$$

where:

$$ATSE = \frac{\|Start_{GT} - Start_{Alg}\| + \|Stop_{GT} - Stop_{Alg}\|}{Stop_{GT} - Start_{GT}}. \quad (6)$$

F1r [28] is a combination of the F1-score and concordance rate (*r*). F1r solves the problems found in ATSR when it analyzes segments with different segment sizes between ground truth and algorithm results. F1r and Perf return values between 0 and 1. F1r is calculated by replacing ATSR in (4) with concordance rate (*r*), which is given by:

$$r_c(g_i, a_j) = 2 \times \frac{min\left(t^{g_i}{}_{end}, t^{a_j}{}_{end}\right) - max\left(t^{g_i}{}_{start}, t^{a_j}{}_{start}\right)}{t^{g_i}{}_{end} - t^{g_i}{}_{start} + t^{a_j}{}_{end} - t^{a_j}{}_{start}}. \quad (7)$$

3. Experiment

This study uses a quantitative experimental approach to investigate and test several group configurations using statistics. Independent variables from the Farneback parameters include magnitude threshold, PyrScale, and WinSize. Meanwhile, the dependent variable in this study refers to the performance matrix described in the previous section, namely Perf and F1r.

The first independent variable is the magnitude threshold. The magnitude threshold determines how much change in the pixel value is considered movement. The smaller the magnitude threshold value, the more sensitive the algorithm is in assessing whether moving objects are in two frames. The magnitude threshold values used in this research are 5.0, 10.0, and 15.0.

The second independent variable is PyrScale. This variable determines the reduction control factor at each level of the pyramid. A smaller PyrScale value will result in a deeper pyramid with higher resolution at each pyramid level. PyrScale values that can be used range from 0 to 1. The PyrScale values used in this study are 0.25, 0.5, and 0.75.

The third independent variable is WinSize. Its value ranges from 0 to 1. A value closer to 1 gives a smoother value but is less sensitive to small movements. The WinSize values used in this study are 10, 15, 20, and 25.

Other parameters were given the following settings: the pyramid layers were set to 3, number of iterations equal to 3 to get the convergence of the pyramid report transition to the actual image resolution, the size of the neighboring pixels was set to 5 to get a smoother image surface and the standard deviation of the Gaussian filter to 1.2.

4. Result and Discussion

4.1.1 Frames Analysis

Frames analysis is performed on ground-truth videos. A ground-truth video is a video of sign sentences labeled with the boundaries between sign language words. Label determination is carried out by experts based on the results of SIBI recordings. Referring to the label of sign language word boundaries, determining the duration of a sign movement can be calculated from the number of frames divided by the framerate. Table 2 shows the smallest and largest number of frames of each sign taken from the five-sentence dataset. The analysis results show that the shortest sign is 12 frames (0.5 seconds), and the most extended sign is 52 frames (2.16 seconds).

4.1.2 Testing Result

The first test compares the results of Perf and F1r to the minimum use of segment duration. The use of minimum segment duration is expected to avoid the formation of temporal segments that are too small because the applied algorithm is too sensitive to movement. The results of Perf and F1r

for comparisons without (S1) and with (S2) a minimum segment duration are displayed in Tables 3 and 4, respectively. Based on the twelve tests that used a magnitude threshold of 5.0 and a minimum segment duration of 0.5 seconds, the confidence interval results are in the negative area. This means that S1 is worse than S2. In other words, using a minimum segment duration can improve the performance of the Farneback optical flow algorithm.

Table 2. The number of required frames of each word or affix to determine the minimum duration of a sign.

No	Words	Number of Frames	
		Min	Max
1	*saya*	22	36
2	*senang*	20	28
3	*me*	12	26
4	*baca*	16	22
5	*buku*	16	40
6	*ter*	16	16
7	*lambat*	20	24
8	*bangun*	16	18
9	*tidur*	28	40
10	*suka*	18	40
11	*ber*	12	16
12	*olahraga*	32	52
13	*sekarang*	20	22
14	*kita*	20	32
15	*makan*	16	20
16	*sama*	26	34
17	*tidak*	14	20
18	*sapu*	14	18
19	*di*	14	24
20	*kelas*	24	40

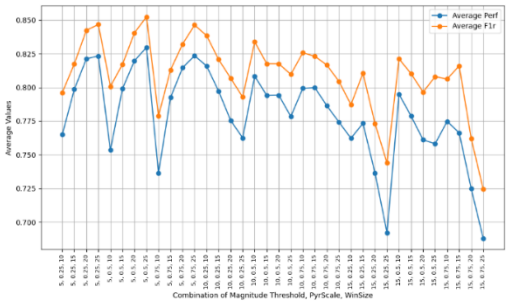
Table 3. Comparison of Perf measurement to minimal segment duration usage.

S1	S2	Cum. Mean Diff.	SD	Confidence Interval	
				Lower interval	Upper interval
0.737	0.765	-0.028	n/a	n/a	n/a
0.783	0.799	-0.022	0.008	-0.023	-0.020
0.801	0.821	-0.021	0.006	-0.022	-0.020
0.805	0.823	-0.021	0.005	-0.021	-0.020
0.727	0.753	-0.022	0.005	-0.022	-0.021
0.761	0.799	-0.024	0.008	-0.025	-0.024
0.804	0.820	-0.023	0.008	-0.024	-0.023
0.813	0.830	-0.022	0.008	-0.023	-0.022
0.715	0.736	-0.022	0.007	-0.023	-0.022
0.768	0.792	-0.023	0.007	-0.023	-0.022
0.798	0.815	-0.022	0.007	-0.022	-0.022
0.803	0.824	-0.022	0.006	-0.022	-0.022

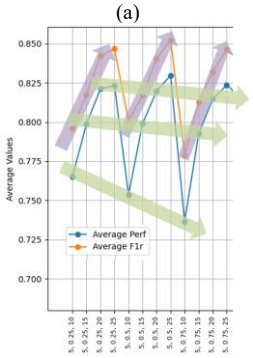
Furthermore, using test scenarios with variations in the value of the independent variable, the test results are obtained, as shown in Figure 2. To make it easier to understand the information in Figure 2, we can distinguish the test results at a threshold magnitude of 5.0 (Figure 2(b)) and above 5.0 (Figure 2(c)). At the magnitude threshold of 5.0, Figure 2(b) shows the PyrScale value, which is inversely proportional to segmentation performance. The green line shows that when PyrScale is increased, the average performance decreases. In contrast, the purple line in Figure 2(b) shows the increasing WinSize values in line with growing performance values.

Table 4. Comparison of F1r measurement to minimal segment duration usage.

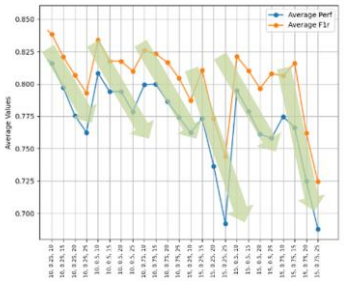
S1	S2	Cum. Mean Diff.	SD	Confidence Interval	
				Lower interval	Upper interval
0.760	0.796	-0.036	n/a	n/a	n/a
0.799	0.817	-0.028	0.012	-0.030	-0.025
0.821	0.842	-0.026	0.009	-0.027	-0.024
0.825	0.847	-0.025	0.008	-0.025	-0.024
0.759	0.801	-0.028	0.010	-0.029	-0.027
0.774	0.817	-0.031	0.011	-0.031	-0.030
0.822	0.841	-0.029	0.011	-0.030	-0.028
0.833	0.852	-0.028	0.011	-0.028	-0.027
0.755	0.779	-0.027	0.010	-0.028	-0.026
0.784	0.813	-0.027	0.010	-0.028	-0.027
0.813	0.832	-0.027	0.009	-0.027	-0.026
0.824	0.846	-0.026	0.009	-0.027	-0.026



(a)



(b)



(c)

Figure 2. Temporal segmentation performance test results with magnitude values (a) 5.0, 10.0, and 15.0, (b) 5.0, and (c) above 5.0.

The test results with a magnitude threshold above 5.0 (Figure 2 (c)) show that the WinSize value is inversely proportional to the method performance. Meanwhile, PyrScale does not significantly affect segmentation performance, although, at a threshold magnitude of 10, it appears that the PyrScale value negatively impacts segmentation performance.

The best temporal segmentation results were produced in a magnitude threshold configuration of 5.0, a PyrScale of 0.5, and a WinSize of 25, with an average Perf of 0.8298 and an F1r of 0.8524. The best performance occurred in subject 1 for sentence 1, with a Perf value of 0.98 and F1r of 0.99. The magnitude threshold of 5.0 means that the optical flow algorithm will recognize the occurrence of a new segment based on the occurrence of object movement after the no-motion condition if there is a movement of the pixel value between two frames with a distance of 5 pixels. Therefore, it is crucial to remember that the magnitude threshold value that produces the best performance on the data only applies to the 360 × 640.

The large WinSize can smooth the estimation results and minimize the effect of noise, thus making the results more stable. However, algorithms can fail to detect subtle movements or detailed frame changes. Combining a small threshold magnitude to capture smoother movements and a considerable WinSize value to minimize noise effects has proven to have the best performance, as seen from the test results.

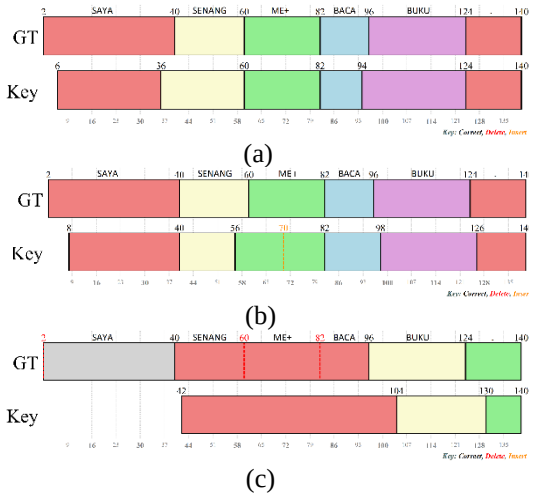


Figure 3. Segment analysis in subject 1, sentence 1, PyrScale 0.75, WinSize 25, (a) threshold 5.0, (b) threshold 10.0, (c) threshold 15.0.

Figure 3 shows some of the temporal segmentation results in the SIBI video sentences. Figure 3(a) has the best performance in temporal segmentation testing because it has different segments based on the ground truth and algorithm results. The segmentation shown in Figure 3(a) can be successful because the algorithm's segmentation results do not remove information from the main movement of the word sign. More details can be seen in Table 5; the 36th and 38th frames have no significant difference. The Farneback algorithm, with a threshold value of 5, determines that the 36th frame is the start of a new segment. This is due to the movement of the body position of the signer, even though the hand gestures do not show any visible changes. Even though these differences do not significantly affect the segmentation results, semantic segmentation can be implemented for signaling hands so that the segmentation results can be carried out more accurately.

Table 5. Analysis of frame sequences from index 34 to 40.

Index	**34**	**36**	**38**	**40**
Frames				

Figure 3(b) shows the algorithm's results, providing an additional segment in the 70th frame. Using a higher threshold magnitude (10) causes the algorithm to detect motion roughly. As seen in Table 6, there is no significant difference in the 66th and 68th frames, so the algorithm assumes that the 68th frame has a pause, closes the segment, and starts a new segment in the 70th frame.

Table 6. Analysis of frame sequences from index 66 to 70.

Index	**66**	**68**	**70**	**72**
Frames				

Figure 3(c) shows unfavorable results for the magnitude threshold and WinSize conditions for the most significant value combination. A large threshold magnitude causes the algorithm to be insensitive to gesture movements, exacerbated when using a considerable window size value. In this test, three segments out of six are compared to ground truth.

Figure 4 shows the performance trend of the segmentation method based on the configuration variations of the experimental scenarios for the three subjects used. Based on the pattern shown in Figure 4, each subject has almost the same pattern, but there are differences in several test variations. This happens because the speed of hand

movements in conveying messages in sign language differs for each signer. However, the most significant average value of the performance of the segmentation algorithm can be obtained by using Farneback Optical Flow of 0.8298 and 0.8524 for Perf and F1r, which are in a magnitude threshold configuration of 5, PyrScale of 10, and WinSize of 25.

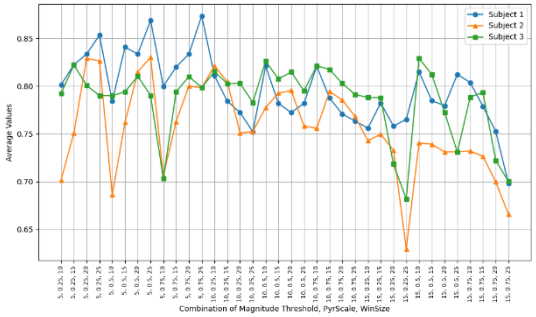


Figure 4. Temporal segmentation performance measurement results for each subject

5. Conclusion

Several conclusions can be drawn based on the experimental results that have been carried out by observing the use of the Farneback optical flow algorithm for temporal action segmentation of SIBI. Each sign produces a variety of movement speeds, resulting in different performance measurement results in different method configurations. Apart from variations in the speed of gestures by different signers, performance is also affected by the complexity of sentence composing signs input to the algorithm. Threshold and WinSize variables negatively correlate to the segmentation algorithm's performance. This can be seen from the results of the measurement visualization, which show that the smallest magnitude threshold has the best performance when the WinSize value is the largest. Performance was measured using Perf and F1r, which yielded the best performance of 0.8298 and 0.8524, respectively. This research has successfully demonstrated using the Farneback optical flow algorithm in performing temporal segmentation on SIBI sign language videos.

Semantic segmentation of the body parts of the hand, which is a component that determines meaning in sign language, can improve the performance of temporal segmentation. Adding machine learning to segmentation can increase performance but at an increased computational cost. The following research is to develop a temporal segmentation method that is adaptive to the possibility of different video attributes and to add a frame selection algorithm for significant movement to the sign segments that have been made. For the record, this research uses a controlled environment in taking videos, such as the subject's background and the camera's distance from the subject. Thus, the movement of objects other than the movement of the subject being observed and changes in light intensity can affect performance. The segmentation results will be used in the SIBI sign language translation.

References

[1] WHO, "Deafness and hearing loss," https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] J. Miller, "Showing TENSE while signing ASL." Accessed: Mar. 13, 2024. [Online]. Available: https://www.signingsavvy.com/article/74/Showing+TENSE+while+signing+ASL

[3] A. S. Nugraheni, A. P. Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat dengan SIBI dan BISINDO pada Mahasiswa Difabel Tunarungu Di Prodi PGMI UIN Sunan Kalijaga," *HOLISTIKA Jurnal Ilmiah PGSD*, vol. 5, no. 1, pp. 28–33, 2021.

[4] E. S. M. El-Alfy and H. Luqman, "A comprehensive survey and taxonomy of sign language research," *Eng Appl Artif Intell*, vol. 114, p. 105198, Sep. 2022, doi: 10.1016/J.ENGAPPAI.2022.105198.

[5] K. Halim and E. Rakun, "Sign Language System for Bahasa Indonesia (Known as SIBI) Recognizer using TensorFlow and Long Short-Term Memory," in *ICACSIS*, 2018, pp. 403–407. doi: 10.1109/ICACSIS.2018.8618134.

[6] Z. Li, Y. A. Farha, and J. Gall, "Temporal Action Segmentation from Timestamp Supervision," Mar. 2021, [Online]. Available: http://arxiv.org/abs/2103.06669

[7] D. Liu, Q. Li, A. Dinh, T. Jiang, M. Shah, and C. Xu, "Diffusion Action Segmentation," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.17959

[8] H. Wang, "Two stage continuous gesture recognition based on deep learning," *Electronics (Switzerland)*, vol. 10, no. 5, pp. 1–14, Mar. 2021, doi: 10.3390/electronics10050534.

[9] S. Asghari-Esfeden, M. Sznaier, and O. Camps, "Dynamic Motion Representation for Human Action Recognition," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2020, pp. 546–555. doi: 10.1109/WACV45572.2020.9093500.

[10] B. Woll, N. Fox, and K. Cormier, "Segmentation of Signs for Research Purposes: Comparing Humans and Machines," 2022. [Online]. Available: https://cvssp.org/projects/extol/

[11] K. Renz, N. C. Stache, S. Albanie, and G. Varol, "Sign language segmentation with temporal convolutional networks," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.12986

[12] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–

1734. doi: 10.3115/v1/D14-1179.

[13] A. Richard, H. Kuehne, and J. Gall, "Weakly Supervised Action Learning with RNN Based Fine-to-Coarse Modeling," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 1273–1282. doi: 10.1109/CVPR.2017.140.

[14] Y. Huang, Y. Sugano, and Y. Sato, "Improving Action Segmentation via Graph-Based Temporal Reasoning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 14021–14031. doi: 10.1109/CVPR42600.2020.01404.

[15] D. and Y. A. Sener Fadime and Singhania, "Temporal Aggregate Representations for Long-Range Video Understanding," in *Computer Vision – ECCV 2020*, H. and B. T. and F. J.-M. Vedaldi Andrea and Bischof, Ed., Cham: Springer International Publishing, 2020, pp. 154–171.

[16] X. and W. J. and C. S. and M. L. and J. Y.-G. Tang Yongyi and Zhang, "Non-local NetVLAD Encoding for Video Classification," in *Computer Vision – ECCV 2018 Workshops*, S. Leal-Taixé Laura and Roth, Ed., Cham: Springer International Publishing, 2019, pp. 219–228.

[17] F. Yi, H. Wen, and T. Jiang, "ASFormer: Transformer for Action Segmentation," in *The 32nd British Machine Vision Conference*, Oct. 2021. [Online]. Available: http://arxiv.org/abs/2110.08568

[18] G. Ding, F. Sener, and A. Yao, "Temporal Action Segmentation: An Analysis of Modern Techniques," Oct. 2022, [Online]. Available: http://arxiv.org/abs/2210.10352

[19] J. Huang, W. Zou, Z. Zhu, and J. Zhu, "An Efficient Optical Flow Based Motion Detection Method for Non-stationary Scenes," *2019 Chinese Control And Decision Conference (CCDC)*, pp. 5272–5277, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:53740608

[20] A. Azulay, T. Halperin, O. Vantzos, N. Bornstein, and O. Bibi, "Temporally stable video segmentation without video annotations," 2021.

[21] E. Hato, "Temporal Video Segmentation Using Optical Flow Estimation," *Iraqi Journal of Science*, vol. 62, no. 11, pp. 4181–4194, Nov. 2021, doi: 10.24996/ijs.2021.62.11.36.

[22] O. Abiodun Adegbola, A. Zinat Alabi, P. O. Idowu, and D. O. Aborisade, "Detection and Tracking of a Moving Object Using Canny Edge and Optical Flow Techniques," *Asian Journal of Research in Computer Science*, pp. 43–56, Jan. 2022, doi: 10.9734/ajrcos/2022/v13i130306.

[23] A. Balasundaram, S. Ashok Kumar, and S. Magesh Kumar, "Optical flow based object movement tracking," *Int J Eng Adv Technol*, vol. 9, no. 1, pp. 3913–3916, Oct. 2019, doi: 10.35940/ijeat.A1317.109119.

[24] I. D. M. B. A. Darmawan *et al.*, "Advancing Total Communication in SIBI: A Proposed Conceptual Framework for Sign Language Translation," in *2023 International Conference on Smart-Green Technology in Electrical and Information Systems (ICSGTEIS)*, 2023, pp. 23–28. doi: 10.1109/ICSGTEIS60500.2023.10424020.

[25] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," in *Image Analysis*, J. Bigun and T. Gustavsson, Eds., Berlin: Springer Berlin Heidelberg, 2003, pp. 363–370. doi: 10.1007/3-540-45103-X_50.

[26] OpenCV, "Object Tracking." Accessed: Dec. 12, 2023. [Online]. Available: https://docs.opencv.org/4.x/dc/d6b/group__video__track.html#ga5d10ebbd59fe09c5f650289ec0ece5af

[27] S. Ruffieux, D. Lalanne, and E. Mugellini, "ChAirGest: A challenge for multimodal mid-air gesture recognition for close HCI," in *ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction*, 2013, pp. 483–488. doi: 10.1145/2522848.2532590.

[28] M. Li and H. Leung, "Graph-based representation learning for automatic human motion segmentation," *Multimed Tools Appl*, vol. 75, no. 15, pp. 9205–9224, Aug. 2016, doi: 10.1007/s11042-016-3480-5.