Hindawi
International Journal of Intelligent Systems
Volume 2024, Article ID 4725216, 16 pages
https://doi.org/10.1155/2024/4725216

WILEY | Hindawi

Research Article

Keyframe Extraction Algorithm for Continuous Sign-Language Videos Using Angular Displacement and Sequence Check Metrics

M. S. Aiswarya and R. Arockia Xavier Annie

Department of Computer Science and Engineering, Anna University, Chennai, India

Correspondence should be addressed to M. S. Aiswarya; aiswarya.smadhavan@gmail.com

Received 5 June 2023; Revised 8 December 2023; Accepted 19 December 2023; Published 10 January 2024

Academic Editor: Riccardo Ortale

Copyright © 2024 M. S. Aiswarya and R. Arockia Xavier Annie. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dynamic signs in the sentence form are conveyed in continuous sign-language videos. A series of frames are used to depict a single sign or a phrase in sign videos. Most of these frames are noninformational and they hardly effect on sign recognition. By removing them from the frameset, the recognition algorithm will only need to input a minimal number of frames for each sign. This reduces the time and spatial complexity of such systems. The algorithm deals with the challenge of identifying tiny motion frames such as tapping, stroking, and caressing as keyframes on continuous sign-language videos with a high reduction ratio and accuracy. The proposed method maintains the continuity of sign motion instead of isolating signs, unlike previous studies. It also supports the scalability and stability of the dataset. The algorithm measures angular displacements between adjacent frames to identify potential keyframes. Then, noninformational frames are discarded using the sequence check technique. Pheonix14, a German continuous sign-language benchmark dataset, has been reduced to 74.9% with an accuracy of 83.1%, and American sign language (ASL) How2Sign is reduced to 76.9% with 84.2% accuracy. A low word error rate (WER) is also achieved on the Phoenix14 dataset.

1. Introduction

Sign language, a visual language, is used by the majority of hard-to-hear people. Both static and dynamic gestures are used to represent words and phrases in sign language. Continuous sign language (CSL) is a collection of sign expressions that can be expressed as a sequence of motions in both space and time. The continuous sign language recognition and translation (CSLRT) task aims to bridge the gap between sign and spoken language by recognizing a series of continuous gestures and translating them into natural language expressions. One sign sentence can contain 100–250 frames (approximately 9 words), depending on the frame rate of the recording device. All these frames are not required for sign interpretation to be performed. Transition frames and noninformative frames can be removed from the frameset, leaving about 1–5 key frames per word. The most informative frames in CSL are keyframes, which contain extensive sign gesture and motion information. This method reduces storage and execution overheads. With a proper keyframe set, neural models can extract spatial and temporal features more precisely. With applications in fields such as action detection [1], video summarization [2], educational video summarization [3], video segmentation [4, 5], and video copyright protection [6], keyframe extraction from videos is one of the thoroughly investigated topics that keep the scientific community interested. Keyframe extraction from sign language videos is considered challenging as we have no indicators to identify the start and end frames of signs in the video and need to identify small motions that may be a part of a sign. Most of the existing keyframe extraction methods are not suitable for sign language video representation, as they do not meet requirements such as multielement identification, minor motion detection, continuity, scalability, and stability.

1.1. Keyframe. Keyframe extraction can be defined as if a video F is represented as a set of frames, F: $\{f_1, f_2, \ldots .. f_n\}$, where n is the total number of frames in F.

The keyframe set is then represented by K such that $K \subset F$ and abstractly represents the original video with a frameset of length m less than n.

The following is a representation of the keyframe extraction algorithm.

Let $\mathscr{H}$ be a keyframe extraction algorithm, then keyframe elements $\mathscr{K}_m$ can be defined as follows:

$$\begin{aligned} \mathscr{H}(F) &= \mathscr{K}_m, \quad 1 \le m < n, \\ \mathscr{K} &= \left\{ K_m \in F \mid K = \bigcup K_m \forall K \in F \right\}, \end{aligned} \tag{1}$$

where $\mathscr{K}$ is the reduced video.

The efficiency of the algorithm $\mathscr{H}$ depends on the reduction rate it attained and the accuracy by which the sign can be recognised. The reduction rate can be expressed as follows:

$$\begin{aligned} \mathscr{R} &= 1 - \frac{\text{Keyframecount}}{\text{Originalframecount}} \\ &= 1 - \frac{m}{n}. \end{aligned} \tag{2}$$

Let $K = \{K_1, K_2, \ldots, K_p\}$ be a keyframe sign sequence and $S = \{S_1, S_2, \ldots, S_q\}$ be the ground truth sign frames on a video F of size r frames, then the accuracy is the proportion at which the frame is correctly recognised against the ground truth.

$$A = \frac{n(K \cap S) + n(F - K)}{n(K \cap S) + n(F - K) + n(S - K)}, \tag{3}$$

where $n()$ gives the total number of frames in the set. The concept of keyframe extraction is depicted in Figures 1 and 2. The sign language representation of the word "LIEB" from the Pheonix4 dataset [7] is illustrated in Figure 1. The keyframes for the same word can be thought of as in Figure 2(a), with two frames handling gesture structure and motion. A graphical depiction of the word "LIEB" based on the signwriting [8] technique taken into consideration for evaluation is shown in Figure 2(b). It is noteworthy that the word with 14 frame length may be reduced to a two-frame representation, which also acts as the keyframe and properly communicates the sign word concept.

The lack of distinct word breaks and continuous gesture transitions make keyframe extraction from CSL video challenging. The gesture position, orientation, and direction of movements must be considered while finding keyframes or in eliminating uninformational frames. Minor and substantial variations in hand forms, motions, positions, nonmanual elements, context, and the signer speed all provide challenges to the keyframe extraction process.

Noninformational frames and informational frames are two instances of frames in CSL videos. The suggested method aims to find a set of keyframes that accurately and efficiently reflect the maximum sign information from a continuous sign-language video with a good reduction rate. The orientation information contained in sign words must also be preserved for a sign to be correctly recognized. By integrating all keyframes, an abstract of a specific sign can be obtained. The motivation for the use of continuous sign-language keyframes is strong since it reduces processing time and storage requirements for representation learning models and other related computer vision tasks.

The various keyframe extraction paradigms utilized in video-related computer vision tasks are cluster methods [9, 10], motion energy-based methods [11], sequence methods [12–16], and machine learning methods [17, 18]. Different sequential approaches and machine learning methods are the most acceptable techniques used in keyframe extraction from continuous sign-language videos.

Keyframe extraction strategies employed in motion analysis, video summarising, or compression cannot directly enhance CSL videos. The spatial, temporal, and directional characteristics of gesture frames in CSL must be evaluated to determine whether they are informative. Certain signs differ solely in the direction of motion of the sign elements. So, the direction of motion is an important information in the sign to fully interpret the link between movements and hence the gesture. This is the first time the concept of gesture orientation has been examined on the keyframe extraction task.

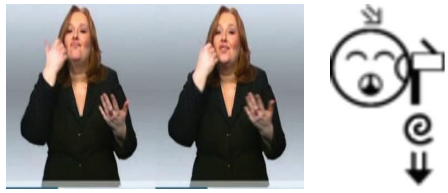
A majority of current sign language key extraction research focuses on dynamic gesture videos (word level), with a few attempts using continuous sign language (sentence level) with the hand as the region of interest [19, 20], leaving the nonmanual elements unresolved. A combination of image entropy and density clustering is used to obtain the keyframes for the hand gesture video in [21]. Minor motions and motion directions cannot be taken into account by this method due to its static threshold value. The method is, therefore, ineffective for CSL videos. The research [22] identifies significant frames and treats each gesture as a separate, isolated gesture using a gradient-based key frame extraction technique. The direction of motion continuity and minute motions are left unresolved. Most sequential approaches use static thresholds such as in [20, 23], which make it difficult to record small, repetitive movements. Specifically, tapping or rubbing does not propagate data over successive frames, preventing static thresholds from distinguishing movements between such frames. Solutions based on threshold values like entropy or sampling do not address scalability or signer independence [14, 24].

This work handled these sign gestures effectively and consistently throughout the huge dataset, which had never been studied before. The proposed work offers an interesting, simple, and efficient approach for extracting successive keyframes from CSL video, which may be fed into a CSLR system for speedy decision, while taking into account hurdles and flaws in earlier works. The following contributions make up this work:

(1) This study proposes a new approach for choosing keyframes from continuous sign video, which significantly reduces computation overhead in time and space dimension

(2) Angular displacement metric is used to evaluate the motion between the frames

Figure 1: Sign representation of the word "LIEB" without frame reduction (initial transition frames also included).



(a) (b)

Figure 2: Keyframes representation of the word "LIEB" and the representation of the sign word "LIEB" using signwriting format. (a) Keyframe representation of the word "LIEB." (b) "LIEB" signwriting format.

(3) The decision of keyframe selection is based on the whole frame; thus, all sign elements are considered

(4) A sequence check metric and frame pixel difference with an adaptive threshold are used to reduce frameset from candidate keyframe set

(5) To analyse and visualise the suggested technique, this work utilise the sign representation method, SignWriting [8].

(6) WER is calculated in conjunction with existing sign language recognition systems to analyse performance of the reduced dataset.

The remaining sections are organized as follows. Section 2 reviews keyframe extraction techniques used in sign language recognition systems. The proposed FSC2 (frame sequence count check) keyframe extraction algorithm is described in Section 3. The experimental results are presented in Section 4. Lastly, a summary of the proposed work and some suggestions for further research are presented.

1.2. Related Work. This section discusses the keyframe extraction techniques that were employed in prior research of sign language recognition tasks.

Keyframe extraction utilising time-varying parameter detection was proposed by [25]. They used statistical analysis of variables such as position, posture, orientation, and motion to detect discontinuities in frames considering only the major motion elements. In [26], fewer gesture motions such as preparation motion and unnecessary movement between sign phrases were deleted using fuzzy partitioning and state automata. For filtering uninformative frames, the authors of [27, 28] employed a gradient-based keyframe extraction method. In [29], the authors randomly sampled 10–50 keyframes from each video and translated directly the sign video representations to spoken language. A method for extracting keyframes in a trajectory density curve using a sliding window is proposed in [19]. In [30], an online low-rank approximation of sign videos to choose keyframes is employed. A method for locating video frames representing single signs in a one-hand finger alphabet is provided in [20], which uses a combination of object tracking and visual attention. In [31], the angular and distance metric of a 3D trajectory skeleton is used for keyframe detection.

The ARSS approach for optimal sampling and alignment of RGB and depth input is proposed in [32], and a relatively complete keyframe set of the video is acquired. In [33], a new sample approach called keyframe-centred clips (KCCs) sampling was given, with the goal of selecting a specific number of frames to describe the entire sign language video. In comparison to other sampling methods, KCC has greater recognition performance. To improve keyframe-centred clips (KCCs) sampling, a new method termed optimised keyframe-centred clip (OptimKCC) sampling was proposed in [14] to optimise the KCC sampling using the DTW distance. In all of the preceding studies, signs are considered as isolated.

The authors proposed two types of distances in [34], interkeyframe distances and model set distances. The sum of the distances to other keyframes and the average distances from the model set are used to pick the set of keyframes K. In [35], Zernike's moments were used to detect the keyframe in a dynamic gesture video clip. A keyframe is one in which Zernike's moments' (ZMs) difference between neighbouring frames is greater than a value (value is set to 50). In [36], a random sampling method is applied. A sequence technique based on the statistical of elements such as colour, picture difference, and weighted frames is proposed in [13] to detect keyframes from dynamic sign-language videos. Edge detection and discrete wavelet transform are used in [37] to extract keyframes. A hybrid clustering approach is provided in [38] and two sets of keyframes are obtained; the spliced original keyframe picture represents the spatial dimension feature, and the optical flow keyframe image represents the time dimension feature. The author of [24] proposed the median of entropy of mean frames (MME) approach for keyframe extraction, which uses the mean of consecutive k frames of video data with a sliding window of size $k/2$ to select the frame that satisfies the median entropy value.

The methodology used in [39] considers multievaluation factors to select critical frames from raw videos. For creating high-quality video clips, essential frames are chosen based on their hand height, hand movements, and frame blurriness levels. In [40], the parameter used for sampling the keyframes was hand coordinates. In [41], the author proposed a clip summary approach to choose the important video clips. In [42], the author used DTW for keyframe extraction.

In comparison to other computer vision tasks that use keyframe extraction such as video summarising and compression, there are few works on keyframe extraction of sign language videos, and it remains a challenging research subject for researchers. The majority of the work is focused on the word level or small phrase extraction which comes under isolated sign. For its complexity, there is very little literature in the realm of continuous sign-language videos. A continuous sign-language sentence stream can have over 250 frames, with a few keyframes functioning as representative frames and the rest as transitional or noninformational frames. Due to the little variation between two consecutive frames and the long length of the input, the demand for modelling temporal sequence of signs at the sentence level is rather stringent.

The majority of early techniques used threshold settings that varied depending on the dataset, which reduced stability and scalability. Repetitive signs and signs with little momentum are disregarded, which results in information loss. Most early research treats the principal hand structure as a single region of interest that is retrieved using a segmentation method in order to condense the gesture space. In addition, each sign phrase's beginning and ending frames were manually chosen and continuous signs were transformed into isolated frames to control the motion. When designing an algorithm for a continuous sign video challenge that heavily relies on continuous data, such restrictions must be minimized.

This work proposes a keyframe extraction algorithm for handling the significant difficulty of keyframe extraction in CSL videos based on the difference in angular displacement of pixels between frames and a sequence check metric.

2. Proposed Method

2.1. Frame Sequence Count Check (FSC2) Keyframe Extraction Algorithm. The FSC2 keyframe extraction algorithm is designed in simple and statistical steps to keep it light weight and effective. The proposed FSC2 keyframe extraction architecture is shown in Figure 3. The FSC2 algorithm has three phases of execution; motion analysis, wrapper, and reduction. Motion analysis uses the Gunnar Farnebäck optical flow algorithm [43] to obtain optical flow data between two nearby neighbouring frames. These data are fed to wrapper where the α value is calculated, which are the mean of the angular displacement obtained from optical flow data. The frames are then arranged in two boxes depending on the α value by the selector and weighed which form the candidate key. The sequencer receives these frames and counts how many of each one can be found in an order and updates the weights depending on the sequence. The frames are sequence checked inside the reducer and then they are reduced using the s-reduction algorithm. S-reduction counts the number of sequences. For a sequence of 3, if the middle element has a positive α value, it is kept and the other frames are discarded; otherwise, the middle element is discarded. In the case of a count of 2 and one is from a box with a negative α value, it will be rejected; otherwise, both will be kept. If the sequence is greater than 3, then the mean pixel difference is used as the threshold and is reduced. The output is a collection of keyframes which form the abstract of signs in CSL videos.

The FSC2 keyframe extraction algorithm evaluates a second-order frame difference by employing a two-frame optical flow calculator (Gunnar Farnebäck) as the first order difference and two successive optical flow differences as the second-order difference. In order to analyse the motion on frames, the algorithm relates to the subsequent three frames. This information allows the algorithm to represent the motion of three subsequent frames, which aids in capturing minute interframe motions.

2.2. Motion Analysis. By obtaining two consecutive frames, the optical flow algorithm calculates the motion of each pixel in a frame. The Gunnar Farnebäck optical flow method was employed for this study to determine the optical flow information between two successive sign frames.

2.2.1. Gunnar Farnebäck Optical Flow. Gunnar Farnebäck [43] is a two-frame motion estimation algorithm developed to produce dense optical flow results. The algorithm is broken down into four steps. Optical flow is determined by quadratic polynomials representing the local neighbourhood of an image in the first step. These quadratic polynomials are used in the second stage to generate a new signal from a global displacement. The following step involves equating quadratic polynomials to calculate global displacements. The coefficient is then calculated by using a weighted least squares estimate of the pixel.

The Gunnar Farnebäck two-frame method was chosen for this study because it can be used to examine each individual pixel displacement between subsequent frames and depends on the notion that sign language frames have a lot of small motion embedded in neighbouring frames.

The mathematical representation of the algorithm is as follows.

Image intensity model with quadratic function for the first frame at pixel location x can be represented as

$$f_1(x) = x^T A x + b_1^T x + c_1, \tag{4}$$

where A is a symmetric matrix, $\mathbf{b}$ is a vector, and c is a scalar. Coefficients are obtained by fitting the weighted least squares to the intensity values in the neighbourhood. For the second frame with global displacement $\mathbf{d}$,

$$\begin{aligned} f_2(x) &= f_1(x - \mathbf{d}) \\ &= (x - \mathbf{d})^T A (x - \mathbf{d}) + {b_1}^T (x - \mathbf{d}) + c_1. \end{aligned} \tag{5}$$

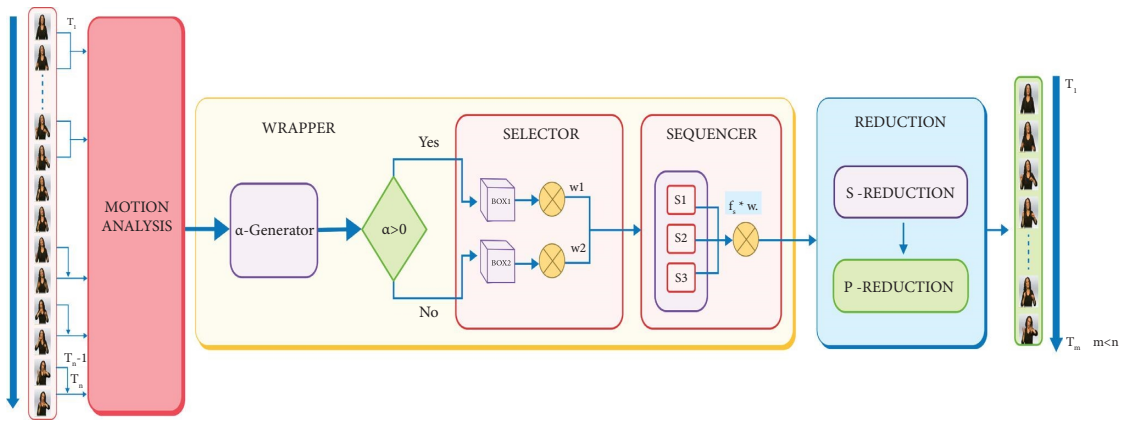


Figure 3: Architecture diagram of frame sequence count check (FSC2) keyframe extraction algorithm.

On expanding and substituting,

$$f_2(x) = x^T Ax + b_2^T x + c_2, \quad b_2 = b_1 - 2A\mathbf{d}. \tag{6}$$

Displacement equation becomes

$$\mathbf{d} = -\frac{1}{2}A^{-1}(b_2 - b_1) = A^{-1}\Delta b. \tag{7}$$

Further reading can be found in the paper [43].

2.3. Wrapper. The Gunnar Farnebäck optical flow algorithm generates an optical flow vector for each pixel that lies between two adjacent frames. By using polarization, angular displacement, $\overrightarrow{\mathbf{A}}$, is calculated from the vector data. The next step is to determine the difference in angular displacement between adjacent pairs of flow data, which corresponds to the angular displacement between three successive frames as shown in equation (8). The parameter utilised for first level candidate keyframe selection, α, is then derived as the mean of the angular displacement difference of the flow data and is represented in equation (9). This process discards a small number of frames.

$$\overrightarrow{\mathbf{A}} = \overrightarrow{\mathbf{A}_i} - \overrightarrow{\mathbf{A}}_{i-1}, \tag{8}$$

$$\alpha = \text{mean}(\overrightarrow{\mathbf{A}}). \tag{9}$$

Thus, the wrapper selects candidate keyframes that may be part of the keyframe set. The selector and sequencer are the two components that determine the rating for the frames. The selector checks the α value, distributes the frames into appropriate boxes, and assigns each frame a weight based on it. Let $f_w = w_g$ be the weight assigned to frames in box with $\alpha > 0$ and $f_w = w_l$ be the weight assigned to the other. This work assumes greater priority to frames in box with $\alpha > 0$, i.e., $w_g > w_l$.

The sequencer uses these weighted frames to determine the sequence check, or the number of frames that follow each other, and divides them into three boxes, designated S1, S2, and S3. Boxes S1, S2, and S3 have score value(s) 2, 3, and 4, respectively. Frames with sequence number two are kept in S1, frames with sequence number three are kept in S2, and frames with sequence number greater than three are kept in S3. Single frames without any adjacent frames are discarded in this step as any abrupt change in motion is considered uninformational.

For example, consider the scenario that a box contains $\{f_1, f_4, f_5, f_6, f_7, f_{10}, f_{11}, f_{12}, f_{20}, f_{21}\}$ set of frames. Then, frame f_1 is discarded, frames $\{f_4, f_5, f_6, f_7\}$ are put in S3 box (for all counts >3, s is set to 4), frames $\{f_{10}, f_{11}, f_{12}\}$ are placed in S2 box, and $\{f_{20}, f_{21}\}$ will get S1 box. Then, each frame's weight in each box is updated in accordance with equation (9).

$$f_w = f_w * s. \tag{10}$$

These weighted frames are what make up the candidate keyframes. In this way, the wrapper initially reduces frames. Then, the frames are combined and sent to the reduction procedure.

2.4. Reduction. Upon receiving the candidate keyframes, the reduction unit starts the reduction process. S-reduction and P-reduction are performed based on the sequence count and pixel difference. The approach is based on the assumption that a significant number of information frames is kept in the box with $\alpha >$ zero.

2.4.1. S-Reduction. There are two types of reductions involved in S-reduction or sequence-check reduction. The first step to determining potential keyframes is to count the continuous frame sequence in the candidate set. For sequence count two, if any one of the frame is from box$_2$, that frame is discarded; otherwise, both frames are kept. From the set $\{f_i, f_{i+1}, f_{i+2}\}$ with sequence count 3, if the frames $\in$box$_1$, then f_{i+1} is discarded; otherwise, $\{f_i, f_{i+2}\}$ are discarded. Frames with a sequence count greater than three is sent for P-reduction.

2.4.2. P-Reduction. A key frame is chosen by comparing pixel differences between succeeding frames to an adaptable threshold. The mean pixel difference of the current sequence set

is used to determine adaptive thresholds. The final output will be the key frameset that represents the sign video abstractly. The algorithmic representation of FSC2 keyframe extraction is given in Algorithm 1. It takes in the frame sequence from the sign video and output the keyframe set K. f_i represents the frame index and f_w represents the frame weight.

The number of frames for a given sign is chosen by the FSC2 algorithm with no reference to any specific parameters. Each sign's motion dictates how many keyframes the FSC2 algorithm selects for it. The choice of keyframes for small signs is one or two. If a sign moves a lot, the algorithm will select more keyframes to identify it.

3. Experimental Results and Analysis

Two datasets were tested using the FSC2 keyframe extraction algorithm, the RWTH-PHOENIX-Weather 2014 dataset [7] and the How2Sign dataset [44]. RWTH-PHOENIX-Weather 2014 includes German sign language weather data captured at 210×260 pixels per frame at 25 frames per second. Extracting the keyframe in an exact way is an important research perspective since the dataset serves as the baseline for all the current sign language research studies. There are more than 80 hours of sign language videos recorded in parallel by 11 signers in How2Sign, a multimodal and multiview continuous American sign language dataset. The backgrounds of both datasets are static. Three sentences of varying length and signer are taken from datasets for analysis and visualization. Table 1 details the sentences used for evaluation and analysis. Two sentences are from the Phoenix4 dataset and one is from the How2Sign dataset.

Figure 4 demonstrates the output achieved for the 176 frames recording "LIEB ZUSCHAUER ABEND WINTER GESTERN loc-NORD SCHOTTLAND loc-REGION UEBERSCHWEMMUNG AMERIKA IX" and the corresponding sign writing notation for each word. The suggested approach reduces the frameset 176 frames to 48 frames, and the figure shows that all informational frames are effectively captured while the directional information is preserved. The sign for the word "LIEB" is well captured as a rubbing gesture as notated in signwriting. The "WINTER" gesture is a modest forward and backward motion of both hands which is also captured well with a low frame count.

3.1. Analysis of the α Value. The α value is the difference between two consecutive angular displacement data obtained from Gunnar Farnebäck optical flow algorithm. In Figure 5, a trace of the α over the ground truth frames from the original video for the sentences in Table 1 is depicted. Sentences with varying word lengths and signers and finger signs are taken at random from the datasets. Estimation of the ground truth is done manually. The αground-truth mapping chart illustrates that most signs appear at $\alpha > 0$. Thus, this study, by prioritizing box_1, is capable of identifying the informational frames of signs. As can be seen from Figure 6, the α value can capture both small and large displacements in a sign and benefit wrapper and reduction algorithms.

3.2. Experimental Setup. The procedure is divided into two sections. The main contribution is the generation of keyframes from continuous sign-language videos. An AMD Ryzen 5000 series CPU system is used for this study. Python 3.10 was used to create the algorithm. Google Colab is utilised for training and testing in the second task, which involves sign language recognition.

3.3. Performance Analysis. Three metrics have been used to evaluate the effectiveness of the proposed algorithm:

(1) Reduction rate (R)

(2) Accuracy (A)

(3) Word error rate (WER)

Table 2 shows the obtained reduction rate and accuracy for two datasets when applied on different keyframe extraction methods. As the value implies, FSC2 performs well on both datasets, capturing the majority of significant frames while eliminating unimportant frames.

In Figure 6, the accuracy chart is presented for different sentences. The keyframes obtained from the FSC2 keyframe extraction algorithm is traced across ground-truth sign frames. A Venn diagram is plotted for the same in Figure 7 to demonstrate the reduction and the accuracy rate. It is demonstrated in Table 3 that the approach is scalable and stable by providing the representation across different sentences, signers, and sentence lengths.

From the figures, it is evident that the FSC2 keyframe extraction algorithm efficiently captures almost all the major and minor gestures in the continuous sign video. WER is evaluated by giving the reduced frameset to two recognition systems. This work chooses SAN [45] and VAC [46] as recognition systems and the obtained results are shown in Table 4. SAN [45] is a transformer-based architecture and with some data augmentation, the network is able to attain better WER when trained with keyframes. VAC [46] uses an iterative training scheme on the CNN framework. Both SAN and VAC are trained and tested with different datasets obtained from methods such as pixel difference, gradient-based approach, Zernik's moment, and the FSC2 algorithm. The outcomes demonstrate that the proposed algorithm efficiently collects information frames while eliminating transitional frames that can be efficient on both global as well as local receptive fields. Figure 8, shows the percentage variance of the WER value obtained after keyframe extraction based on Table 4. The findings indicate that compared to the previous methods, the FSC2 keyframe extraction reduces WER more successfully. As compared to other algorithms, the proposed algorithm reduces the WER relative to the baseline.

3.4. Computational Complexity. The computational complexity of neural network models is commonly assessed using train time complexity, run time complexity, and space complexity. Using the FSC2 keyframe extraction algorithm, the computational complexity can be reduced by a factor of m, where m is the new size of the dataset.

Input: F: set of all frames in a CSL video
Output: K: set of all keyframes
foreach *frame* $f \in F$ **do**
  flow $\longleftarrow$ Gunnar_FarnebäckOpticalFlow (f_1, f_2)
    foreach *element in flow* **do**
      $A \longleftarrow$ polarze (flow$_1$, flow$_2$)
      CK $\longleftarrow$ Call Wrapper (A)
      $K \longleftarrow$ Call Reduction (CK)
    end
end
Function Wrapper (A):
  $\alpha \longleftarrow$ mean ($A_i - A_{i+1}$)//selector
  if$\alpha > 0$, **then**Box$_1 \longleftarrow f_i$, $f_{i_w} \longleftarrow w_g$;
  else Box$_2 \longleftarrow f_i$, $f_{i_w} \longleftarrow w_l$;
  //sequencer
  for each $f \in$ boxes, **do**
    Findsequencecount, s
      if ($s == 2$), **then** $S1 \longleftarrow ff_w \longleftarrow f_w * s_1$;
      else if ($s == 3$), **then** $S2 \longleftarrow ff_w \longleftarrow f_w * s_2$;
      else if ($s > 3$), **then** $S3 \longleftarrow ff_w \longleftarrow f_w * s_3$;
  end
  CK $\longleftarrow S_1 \cup S_2 \cup S_3$
  return CK
  End Function
Function Reduction (CK):
  //Find the sequence count, s
  //S-reduction
  if $s == 2$, **then**
    if f_1 or $f_{i+1} \in$ Box$_2$, **then**
    CK $\longleftarrow$ CK $- f \in$ Box$_2$;
  end
  if $s == 3$, **then**
    if $f_1, f_{i+1}, f_{i+2} \in$ Box$_1$, **then**
      CK $\longleftarrow$ CK $- f_{i+1}$;
    else CK $\longleftarrow$ CK $- f_i - f_{i+2}$;
  end
  if $s > 3$ **then**
    //P-reduction
    for all frames $f \in$ sequence, **do**
        pd $\longleftarrow$ subtract (f_i, f_{i+1})
    end
    $\beta \longleftarrow$ mean (pd)
    if pd$_{i,i+1} < \beta$, **then** CK $\longleftarrow$ CK $- f_i$;
  end
  return CK
End Function

Algorithm 1: FSC2: frame sequence count check.

3.4.1. Space Complexity. Space complexity can be assessed by the amount of space required to store the model input.

Worse case space complexity $= O(n)$, where $n =$ total number of frames without reduction; average or best case space complexity $= O(m)$, where $m =$ total number of keyframes after reduction, $m < n$

Table 2 shows that the algorithm can reach a reduction rate of approximately 75%. As a result, the space complexity is decreased from n to m, which is a 75% reduction and a cost-effective solution.

3.4.2. Time Complexity. Time complexity can be estimated as the train time complexity or run time complexity, when the keyframe set is fed as input to the neural network model.

Worse case time complexity $= O(n)$, where $n =$ total number of frames without reduction. Average or Best case time complexity $= O(m)$, where $m =$ total number of keyframes after reduction, $m < n$

Because the number of input frames is 75% less than in the original dataset, train and run time complexity can be reduced by 75%, allowing the network to extract and learn features faster.

Table 1: The following sentences were taken for the purpose of analysis and visualization.

Sn.	Sentence	FC	Dataset
1	"WOCHENENDE TATSAECHLICH FREIZEIT WOCHENENDE PASSEN"	69	Phoenix4
2	"LIEB ZUSCHAUER ABEND WINTER GESTERN loc-NORD SCHOTTLAND loc-REGION UEBERSCHWEMMUNG AMERIKA IX"	176	Phoenix4
3	"HELLO AGAIN THIS IS OSCAR MORENO WITH MORENO CUSTOM HOME"	104	How2Sign

FC indicates the number of frames a sentence has in its video representation.

Figure 4: The FSC2 keyframe extraction algorithm: generated keyframes. The sentence "LIEB ZUSCHAUER ABEND WINTER GESTERN loc-NORD SCHOTTLAND loc-REGION UEBERSCHWEMMUNG AMERIKA" was condensed from 176 frames to 48 keyframes. The signwriting rendition is shown beneath each word.

3.5. Scalability and Stability. Scalability refers to the capacity of keyframe extraction methods to run on various kinds of datasets captured under a variety of circumstances and yield exact results. The scalability of keyframe extraction techniques can be impacted by variables including data independence, signer independence, and phrase or word length. The FSC2 keyframe extraction algorithm can be used to reduce any sign language dataset, regardless of the type of sign language or the frame rate of the video and thus data independent and scalable. Table 2 shows the reduction rate and accuracy obtained using two datasets. Both the design statistics and the lack of a threshold value lend credence to this benefit. On four distinct signs executed by three distinct signers, the algorithm offers the best and most precise reduction, as shown in Table 3. The signs "MORGEN," "GESTERN," "LIEB," and "KNIFE" are taken into consideration for analysis, and it is discovered that the signs are accurately reduced when performed by various signers. It was, therefore, determined to be accurate and stable, regardless of the signer and language. The word length on 5670 videos in the Pheonix4 dataset is less than 9 words, on average. The How2Sign dataset includes finger signing as

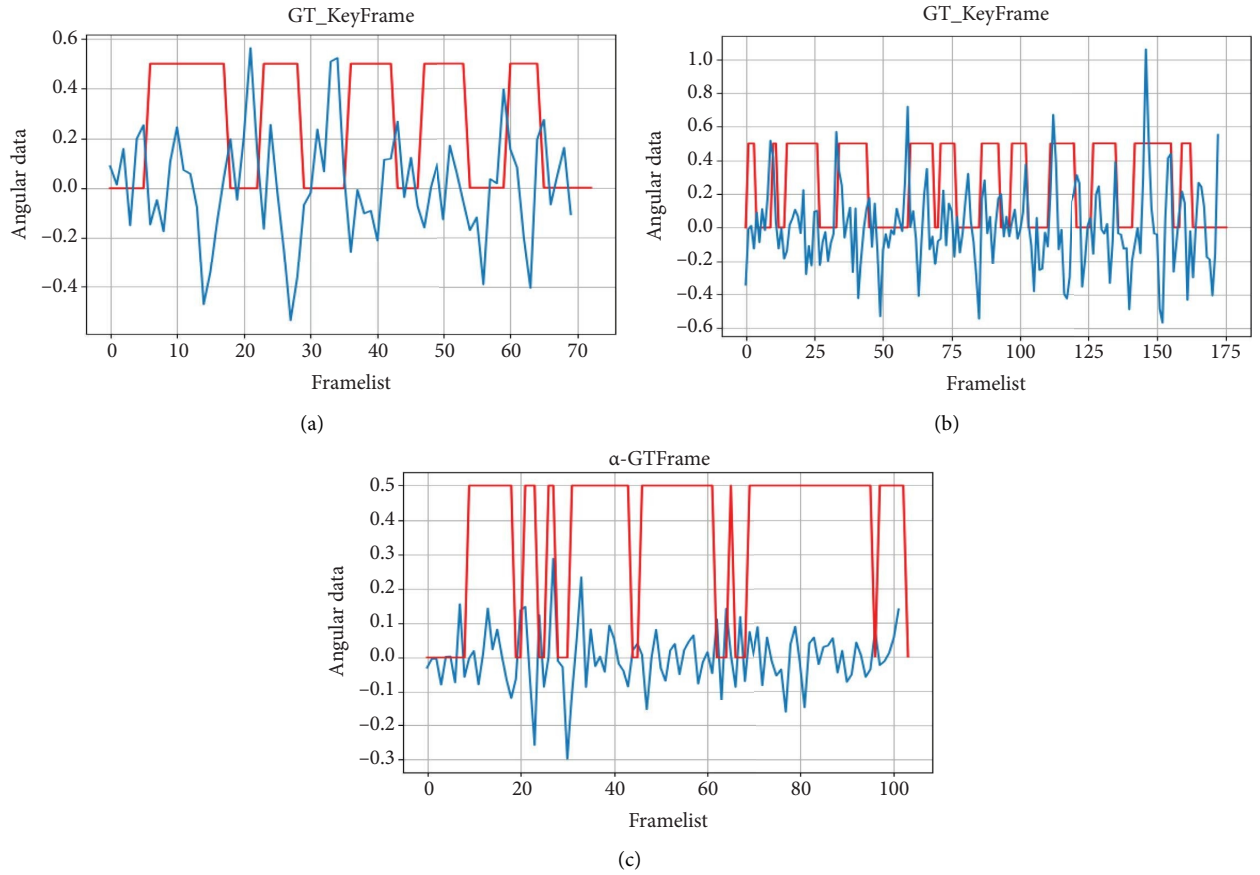
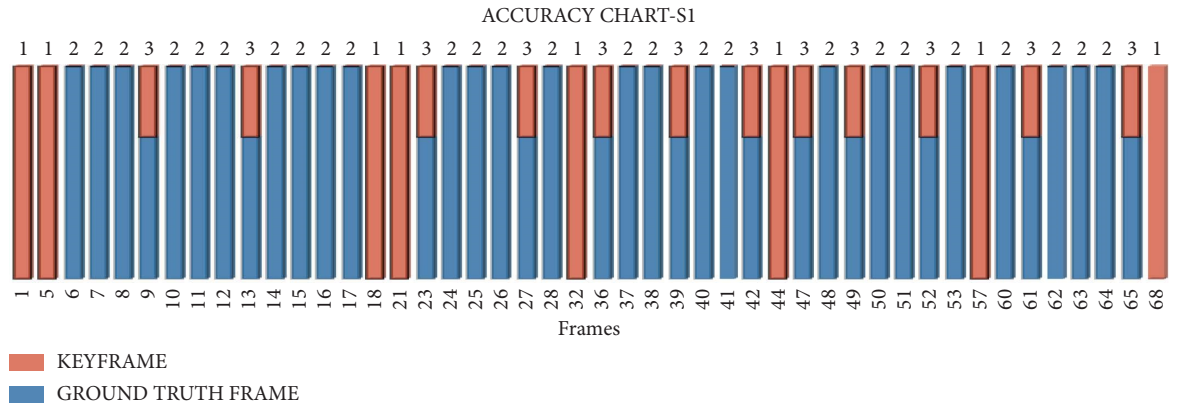
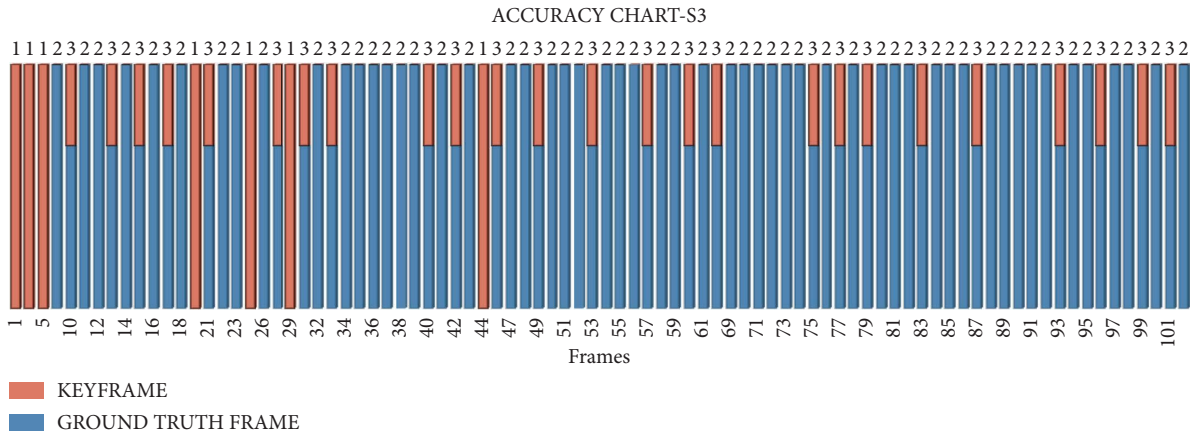


Figure 5: The graphical representation of the relation between the α value and the ground truth sign frames for the sentences in Table 1. (a) Ground truth-α mapping: sentence 1. (b) Ground truth-α mapping: sentence 2. (c) Ground truth-α mapping: sentence 3.

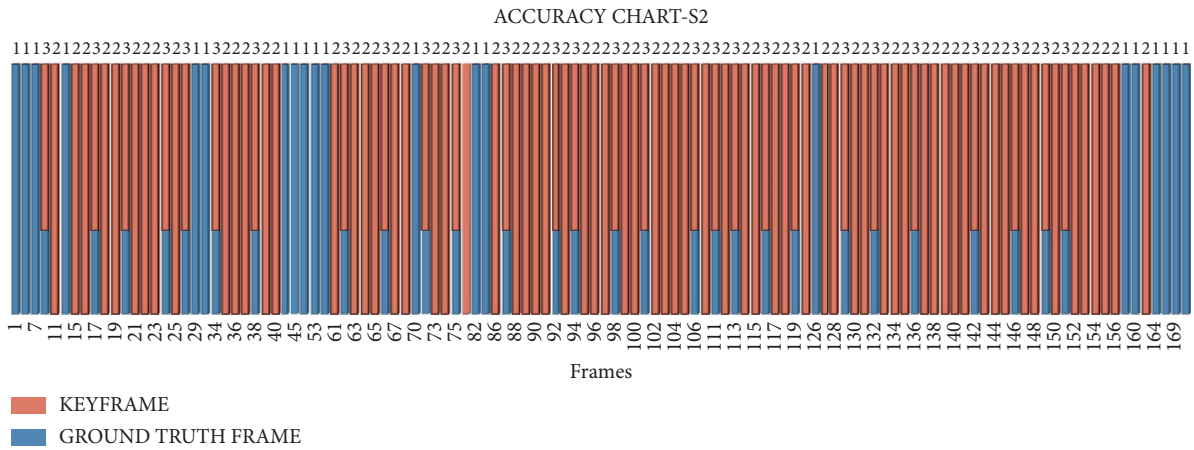


(a)

Figure 6: Continued.



(b)



(c)

Figure 6: A comparison of the ground truth frames and the keyframes produced by the FSC2 algorithm for the sign videos in Table 1. (a) Ground truth-keyframe mapping to show accuracy for sentence 1. (b) Ground truth-keyframe mapping to show accuracy for sentence 2. (c) Ground truth-keyframe mapping to show accuracy for sentence 3.

Table 2: Performance analysis of the FSC2 keyframe extraction algorithm and existing algorithms in terms of the reduction rate and accuracy on Pheonix4 and How2Sign datasets.

Dataset	S	n	Method	m	R (%)	A (%)
Pheonix4 (train)	5670	798629	Pixel difference [13]	203650	74.5	67.3
			Gradient based [27]	214032	73.2	66.4
			Zernik's moment [35]	492754	38.3	70.4
			Sampling [29]	238790	70.1	60.3
			FSC2 (proposed)	206823	**74.1**	**83.1**
How2Sign (test)	32	101793	Pixel difference [13]	30233	70.3	69.7
			Gradient based [27]	35322	65.3	70.1
			Zernike's moment [35]	60974	40.1	74.4
			Sampling [29]	28400	72.1	67.3
			FSC2 (proposed)	24995	**75.4**	**84.2**

S denotes the total input video, n denotes the total frame count before reduction, m denotes the keyframe count after reduction, R is the reduction rate, and A is the average accuracy. The highlighted values shows that the FSC2 algorithm gives a higher reduction rate and accuracy.

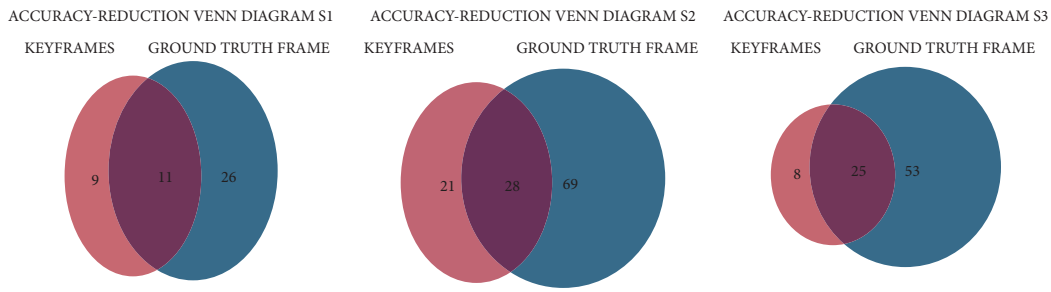


Figure 7: Venn diagram showing the algorithm's accuracy and the reduction rate for three videos.

Table 3: The stability of the algorithm is assessed by comparing the number of keyframes extracted, done by different signers on four sign words from Pheonix14 and How2Sign datasets.

Signs	SignWriting	m	N	Signer ID
MORGEN		8	3	Signer 1
		9	3	Signer 3
		8	3	Signer 8
GESTERN		4	2	Signer 1
		4	2	Signer 4
		4	2	Signer 8
LIEB		3	2	Signer 4
		3	2	Signer 5
		3	2	Signer 7
KNIFE		3	2	Signer 1
		4	2	Signer 2
		3	2	Signer 3

m denotes the number of frames in the original video and n denotes the number of frames in the reduced set.

Table 4: A comparison of the WER metrics obtained by different systems on sign language recognition tasks.

Methods	dev	Test
Deep sign [47]	38.3	38.8
SubUNets [48]	40.8	40.7
SF-Net [49]	35.6	34.9
SAN [45]	29	29.7
SAN [45] + Zernike's moment [35]	38.1	38.2
SAN [45] + pixel difference [13]	40.1	40.2
SAN [45] + gradient based [27]	42.7	43.2
SAN + FSC2 (proposed)	**28.6**	**28.8**
VAC [46]	21.2	22.3
VAC [46] + Zernike's moment [35]	28.1	28.2
VAC [46] + pixel difference [13]	32.1	33.2
VAC [46] + gradient based [27]	31.7	31.8
VAC + FSC2 (proposed)	**20.8**	**21.9**

Tested on the RWTH-PHOENIX-weather 2014 dataset. A lower WER value is better. The highlighted value suggests that FSC2 performs well when integrated with existing systems.

well as long and short sentences. Without using any additional parameters, the FSC2 keyframe extraction algorithm effectively extracts all categories accurately and efficiently.

3.6. Comparison of the FSC2 Keyframe Extraction Algorithm across Various Approaches. Table 5 compares the FSC2 keyframe extraction method with a few benchmark keyframe extraction techniques. Stability (S) examines the ability to extract a keyframe for the same sign done by a different signer in a consistent manner. The static threshold (T) parameter checks the use of any static threshold value for keyframe selection. Scalability measures the data independency (DI) and the application of an algorithm to different datasets without any changes and is tested to ensure that it can work with multisigner, multilanguage, and variable length data. Direction or continuity (C) of sign is also an important element in sign recognition. So, keyframes must also contain directional information. By using transition frames, continuous signs were prevented from

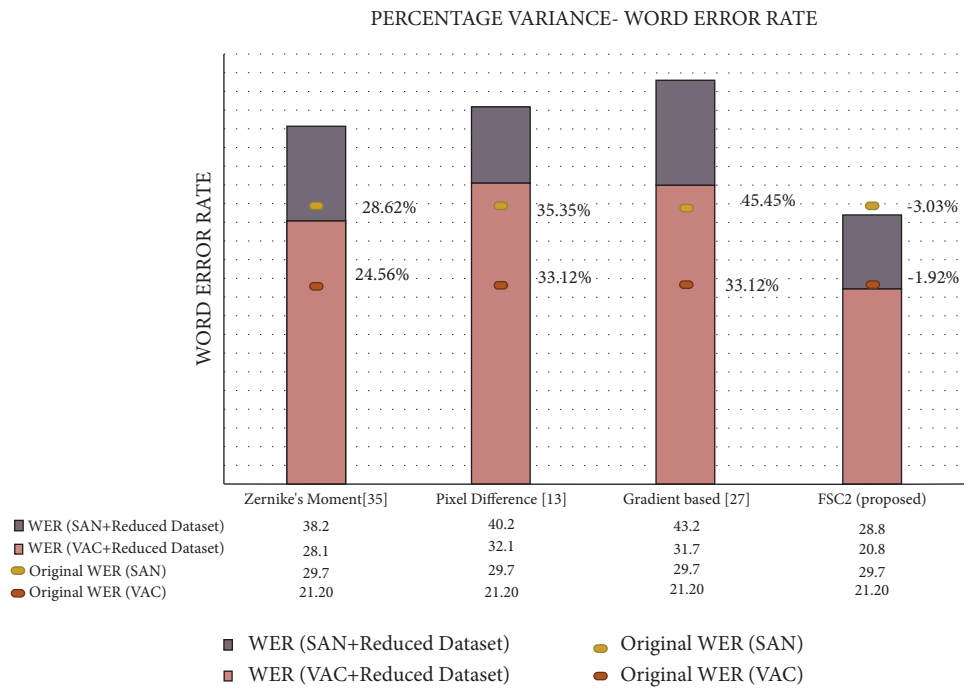


Figure 8: Percentage variation of WER was calculated for four key extraction algorithms based on Table 4. SAN and VAC systems are given new dataset with maximum accuracy obtained and WER is calculated. The percentage variance in WER is shown in each case.

Table 5: Comparison of the FSC2 keyframe extraction algorithm with other algorithms on parameters.

Methods	T	DI	S	C
Pixel difference [13]	✓	×	×	×
DTW [42]	✓	×	×	×
Gradient based [27]	✓	×	×	×
Sampling [29, 33]	✓	×	×	×
Sampling + DTW [14]	✓	×	×	×
Clip summary [41]	×	×	×	×
Zernik's moment [35]	✓	×	×	×
MME [24]	✓	×	×	×
Hand features [39]	✓	×	×	×
Edge detection + DTW [37]	✓	×	×	×
Hybrid cluster [38]	×	×	×	×
FSC2 keyframe extraction (proposed)	×	✓	✓	✓

T: static threshold dependency, DI: data independency, S: stability, and C: continuity.

becoming isolated signs. A qualitative analysis of the keyframe extraction algorithms can be found in Table 5. The analysis shows that the algorithm successfully meets the abovementioned three significant qualities when extracting keyframes from CSL videos.

4. Ablation Study

4.1. Changing the α Criteria. The main notion of FSC2 is that keyframes may be identified at $\alpha > 0$. A study is carried out with $\alpha < 0$. When compared to the previous notion, the obtained result is less precise. There was inadequate similarity between ground truth and keyframes. Figure 9 shows the results for the parameters extracted with keyframe count m, reduction rate R, and accuracy A with two values of α when applied to three sentences. The keyframe count and reduction rate are depicted by bars, while the accuracy is represented by a line. Figure 9 shows that the choice of $\alpha > 0$ gives a better results.

4.2. Motion Analysis Using the Lucas–Kanade Method. The Gunnar Farnebäck algorithm is replaced by the Lucas–Kanade method, and the results demonstrate that the GF algorithm is superior to the LK methodology because the GF algorithm can capture motion between two successive frames and captures all motions in the signs. Figure 10 shows the performance of both optical flow algorithms when applied on three sentences. Accuracy is represented by the line chart and it is clear that Gunnar Farnebäck gives a better value as it can capture the small motions between two frames.

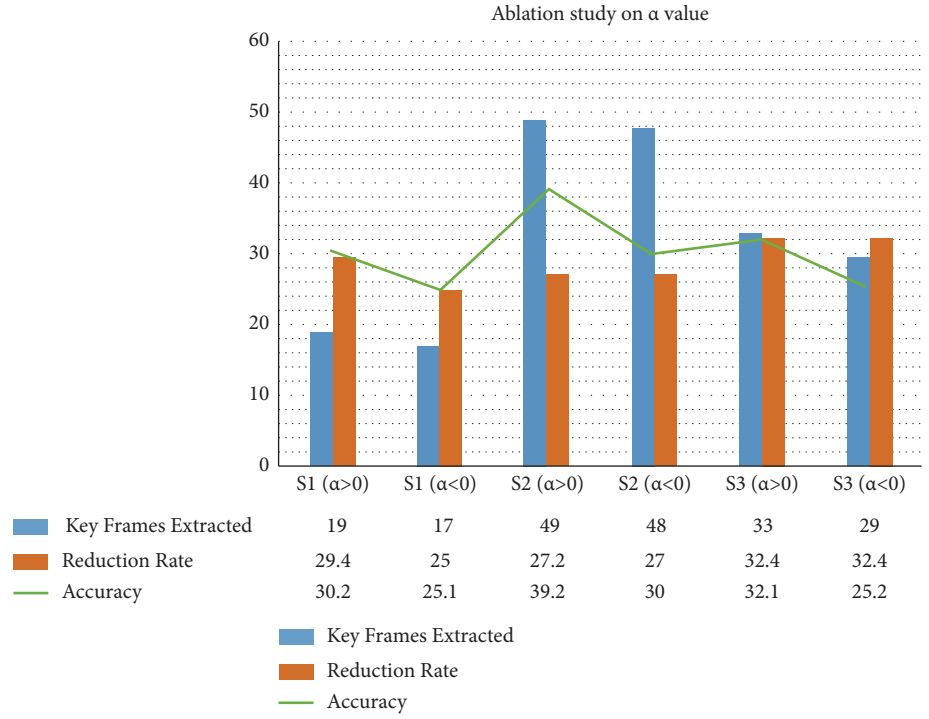


Figure 9: Ablation study on the FSC2 algorithm by altering the α value.

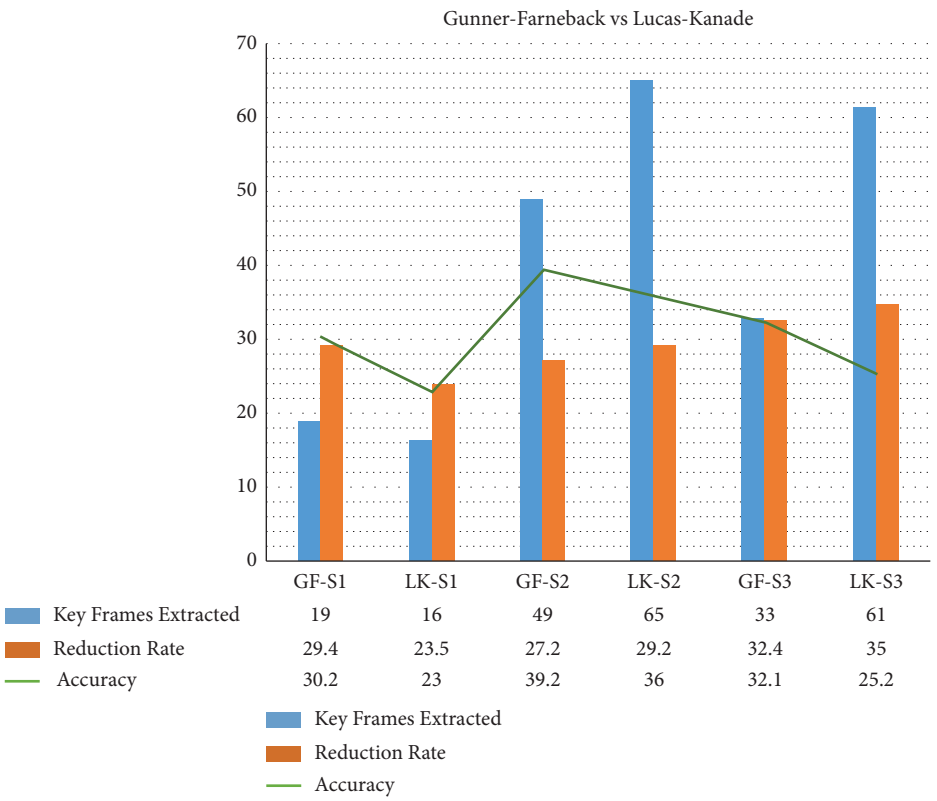


Figure 10: Comparison study of the Gunnar Farnebäck optical flow algorithm and the Lucas–Karnade method when applied to FSC2 on three sentences. Metric considered are the Keyframe count, reduction rate, and accuracy.

4.3. Changing the Sequence Value. For keyframe extraction, the FSC2 algorithm examines three sequence values, i.e., 2, 3, and more than 3 to capture both long and short signs. The sequence values are altered in various orders in order to catch the important frames. The outcome fell short of the standard set by FSC2 in the reduction rate and accuracy.

5. Conclusion

The proposed FSC2 keyframe extraction method is developed to extract keyframes from a video of continuous signs. As a result of the extraction process, every informational frame was successfully extracted and also achieved a high reduction rate. This enables researchers to complete CSL-related tasks in less time, with less sophisticated computational hardware and with less storage. In contrast to previous works, the algorithm extracts gesture information from videos while maintaining factors such as continuity and motion direction. Despite the computationally expensive nature of optical flow techniques, FSC2 keyframe extraction is efficient for both long and short sign sequences in terms of accuracy and stability. With statistical methods on optical flow data that function on all basic hardware, the algorithm design is kept simple. The results showed that the suggested strategy produced highly competitive outcomes when compared to the state-of-the-art approaches. Thus, the algorithm solves six major problems related to keyframe extraction from CSL videos such as stability, scalability, preserving direction information, detecting small and repeated movements in sign, low information loss with great accuracy, and good reduction rate. An evaluation of the algorithm's performance is conducted on existing systems to ensure that it performs the task efficiently. All datasets included in this study have static backgrounds. The angular displacement and optical flow data are impacted by background object movement. As a result, the motion estimate employed in the FSC2 approach cannot precisely determine the sign when the background is changing. Therefore, the proposed algorithm performs poorly compared to how it does with static data. Additional investigation about real-time sign language with different static backgrounds is necessary.

Data Availability

The data used to support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. M. M. E Elahi and Y.-H. Yang, "Online learnable keyframe extraction in videos and its application with semantic word vector in action recognition," *Pattern Recognition*, vol. 122, Article ID 108273, 2022.

[2] N. Ejaz, T. B. Tariq, and S. W. Baik, "Adaptive key frame extraction for video summarization using an aggregation mechanism," *Journal of Visual Communication and Image Representation*, vol. 23, no. 7, pp. 1031–1040, 2012.

[3] C. Choudary and T. Liu, "Summarization of visual content in instructional videos," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1443–1455, 2007.

[4] L. Liu and G. Fan, "Combined key-frame extraction and object-based video segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 869–884, 2005.

[5] M. Awan and J. Shin, "Semantic video segmentation with dynamic keyframe selection and distortion-aware feature rectification," *Image and Vision Computing*, vol. 110, Article ID 104184, 2021.

[6] Y. Shi, H. Yang, M. Gong, X. Liu, and Y. Xia, "A fast and robust key frame extraction method for video copyright protection," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 1231794, 7 pages, 2017.

[7] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015.

[8] V. Sutton, *Lessons in Sign Writing: Textbook*, SignWriting, SignWriting Press, New York, NY, USA, 2014.

[9] G. L. Moir, B. W. Graham, S. E. Davis, J. J. Guers, and C. A. Witmer, "An efficient method of key-frame extraction based on a cluster algorithm," *Journal of Human Kinetics*, vol. 39, no. 1, pp. 15–23, 2013.

[10] A. Ioannidis, V. Chasanis, and A. Likas, "Weighted multi-view key-frame extraction," *Pattern Recognition Letters*, vol. 72, pp. 52–61, 2016.

[11] T. Liu, H.-J. Zhang, and F. Qi, "A novel video key-frame-extraction algorithm based on perceived motion energy model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 10, pp. 1006–1013, 2003.

[12] C. V. Sheena and N. K. Narayanan, "Key-frame extraction by analysis of histograms of video frames using statistical methods," *Procedia Computer Science*, vol. 70, pp. 36–40, 2015.

[13] L. Shurong, H. Yuanyuan, H. Zuojin, and D. Qun, "Key frame detection algorithm based on dynamic sign language video for the non specific population," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 12, pp. 135–148, 2015.

[14] W. Pan, X. Zhang, and Z. Ye, "Attention-based sign language recognition network utilizing keyframe sampling and skeletal features," *IEEE Access*, vol. 8, pp. 215592–215602, 2020.

[15] H. M. Nandini, H. K. Chethan, and B. S. Rashmi, "Keyframe extraction using sobel fuzzified weighted approach," in *Proceedings of the International Conference on Intelligent Systems Design and Applications*, pp. 236–246, Springer, Berlin, Germany, December, 2020.

[16] M. Mentzelopoulos and A. Psarrou, "Key-frame extraction algorithm using entropy difference," in *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 39–45, New York, NY, USA, January, 2004.

[17] C. Huang and H. Wang, "A novel key-frames selection framework for comprehensive video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 577–589, 2020.

[18] N. Ejaz, S. W. Baik, H. Majeed, H. Chang, and I. Mehmood, "Multi-scale contrast and relative motion-based key frame extraction," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, 2018.

[19] Y. Yan, Z. Li, Q. Tao, C. Liu, and R. Zhang, "Research on dynamic sign language algorithm based on sign language trajectory and key frame extraction," in *Proceedings of the 2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pp. 509–514, IEEE, Chengdu, China, May, 2019.

[20] C. Filip, J. Polec, and R. Vargic, "Key frame extraction from video sequences containing asl signs with concealed transmission errors," in *Proceedings of the 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 0208–0213, IEEE, Barcelona, Spain, April, 2017.

[21] H. Tang, H. Liu, W. Xiao, and N. Sebe, "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion," *Neurocomputing*, vol. 331, pp. 424–433, 2019.

[22] R. M. Kagalkar and S. V. Gumaste, "Gradient based key frame extraction for continuous indian sign language gesture recognition and sentence formation in Kannada language: a comparative study of classifiers," *International Journal on Computer Science and Engineering*, vol. 4, no. 9, 2016.

[23] S. Zhang, Z. Zhu, and Z. Hu, "Sign language recognition based on key frame," in *IOP Conference Series: Earth and Environmental Science*, vol. 252, IOP Publishing, 2019.

[24] S. Saqib and S. Kazmi, "Video summarization for sign languages using the median of entropy of mean frames method," *Entropy*, vol. 20, no. 10, p. 748, 2018.

[25] R.-H. Liang and O. Ming, "A real-time continuous gesture recognition system for sign language," in *Proceedings of the third IEEE international conference on automatic face and gesture recognition*, pp. 558–567, IEEE, Nara, Japan, April, 1998.

[26] J.-B. Kim, K.-H. Park, W.-C. Bang, and Z. Z. Bien, "Continuous gesture recognition system for Korean sign language based on fuzzy logic and hidden Markov model," in *Proceedings of the 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, vol. 2, pp. 1574–1579, IEEE, Honolulu, HI, USA, May, 2002.

[27] K. Tripathi and N. B. G. Nandi, "Continuous indian sign language gesture recognition and sentence formation," *Procedia Computer Science*, vol. 54, pp. 523–531, 2015.

[28] V. Joshi, R. Hiray, S. Kale, and S. Mantode, "Hand gesture recognition using gradient based key frame extraction," *Journal of Image Processing and Artificial Intelligence*, vol. 4, 2018.

[29] S.-K. Ko, J. Gi Son, and H. Jung, "Sign language recognition with recurrent neural network using human keypoint detection," in *Proceedings of the 2018 conference on research in adaptive and convergent systems*, pp. 326–328, Honolulu, HI, USA, October, 2018.

[30] H. Wang, X. Chai, Y. Zhou, and X. Chen, "Fast sign language recognition benefited from low rank approximation," in *Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, pp. 1–6, IEEE, Ljubljana, Slovenia, May, 2015.

[31] E. Escobedo-Cardenas and G. Camara-Chavez, "A robust gesture recognition using hand local data and skeleton trajectory," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1240–1244, IEEE, Quebec, Canada, September, 2015.

[32] S. Zhang, W. Meng, H. Li, and X. Cui, "Multimodal spatiotemporal networks for sign language recognition," *IEEE Access*, vol. 7, pp. 180270–180280, 2019.

[33] S. Huang, C. Mao, J. Tao, and Z. Ye, "A novel Chinese sign language recognition method based on keyframe-centered clips," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 442–446, 2018.

[34] R. Yang and S. Sarkar, "Detecting coarticulation in sign language using conditional random fields," in *Proceedings of the 18th International conference on pattern recognition (ICPR'06)*, vol. 2, pp. 108–112, IEEE, Hong Kong, China, August, 2006.

[35] P. K. Athira, C. J. Sruthi, and A. Lijiya, "A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 771–781, 2022.

[36] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied Sciences*, vol. 9, no. 13, p. 2683, 2019.

[37] H. A. Aldelfy, M. H. Al-Mufraji, and T. R. Saeed, "Key frame extraction using hybrid algorithm of dynamic sign language," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 186–190, 2019.

[38] J. Yu, M. Qin, and S. Zhou, "Dynamic gesture recognition based on 2d convolutional neural network and feature fusion," *Scientific Reports*, vol. 12, no. 1, pp. 4345–4360, 2022.

[39] Z. Zhou, V. W. L. Tam, and E. Y. Lam, "Signbert: a bert-based deep learning framework for continuous sign language recognition," *IEEE Access*, vol. 9, pp. 161669–161682, 2021.

[40] Z. Liu, X. Qi, and L. Pang, "Self-boosted gesture interactive system with st-net," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 145–153, Seoul, Republic of Korea, October, 2018.

[41] D. Guo, W. Zhou, A. Li, H. Li, and M. Wang, "Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation," *IEEE Transactions on Image Processing*, vol. 29, pp. 1575–1590, 2020.

[42] F. U. Mangla, A. Bashir, I. Lali, A. C. Bukhari, and B. Shahzad, "A novel key-frame selection-based sign language recognition framework for the video data," *The Imaging Science Journal*, vol. 68, no. 3, pp. 156–169, 2020.

[43] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian Conference on Image Analysis*, pp. 363–370, Springer, Berlin, Germany, 2003.

[44] A. Duarte, S. Palaskar, and L. Ventura, "How2Sign: a large-scale multimodal dataset for continuous American Sign Language," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June, 2021.

[45] F. Ben Slimane and B. Mohamed, "Context matters: self-attention for sign language recognition," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7884–7891, IEEE, Milan, Italy, January, 2021.

[46] Y. Min, H. Aiming, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *Proceedings of the IEEE/CVF International Conference on*

Computer Vision, pp. 11542–11551, Toronto, ON, Canada, December, 2021.

[47] O. Koller, O. Zargaran, H. Ney, and R. Bowden, "Deep sign: hybrid cnn-hmm for continuous sign language recognition," in *Proceedings of the British Machine Vision Conference 2016*, York, UK, September, 2016.

[48] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: end-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE international conference on computer vision*, pp. 3056–3065, Venice, Italy, October, 2017.

[49] Z. Yang, Z. Shi, X. Shen, and Y.-W. Tai, "Sf-net: structured feature network for continuous sign language recognition," 2019, https://arxiv.org/abs/1908.01341.