applied sciences

MDPI

Article

Movement-in-a-Video Detection Scheme for Sign Language Gesture Recognition Using Neural Network

Angela C. Caliwag [1], Han-Jeong Hwang [2,3], Sang-Ho Kim [4,*] and Wansu Lim [1,*]

[1] Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea
[2] Department of Electronics and Information Engineering, Korea University, Sejong 30019, Korea
[3] Interdisciplinary Graduate Program for Artificial Intelligence Smart Convergence Technology, Korea University, Sejong 30019, Korea
[4] Department of Industrial Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea
* Correspondence: kimsh@kumoh.ac.kr (S.-H.K.); wansu.lim@kumoh.ac.kr (W.L.); Tel.: +82-054-478-7656 (S.-H.K.); +82-54-478-7489 (W.L.)

Abstract: Sign language aids in overcoming the communication barrier between hearing-impaired individuals and those with normal hearing. However, not all individuals with normal hearing are skilled at using sign language. Consequently, deaf and hearing-impaired individuals generally encounter the problem of limited communication while interacting with individuals with normal hearing. In this study, a sign language recognition method based on a movement-in-a-video detection scheme is proposed. The proposed scheme is applied to extract unique spatial and temporal features from each gesture. The extracted features are subsequently used to train a neural network to classify the gestures. The proposed movement-in-a-video detection scheme is applied to sign language videos featuring short, medium, and long gestures. The proposed method achieved an accuracy of 90.33% and 40% in classifying short and medium gestures, respectively, compared with 69% and 43.7% achieved using other methods. In addition, the improved accuracies were achieved with less computational complexity and cost. It is anticipated that improvements in the proposed method, for it to achieve high accuracy for long gestures, can enable hearing-impaired individuals to communicate with normal-hearing people who do not have knowledge of sign language.

Keywords: artificial intelligence; dynamic sign language; frame extraction; motion detection; sign language recognition

check for updates

Citation: Caliwag, A.C.; Hwang, H.-J.; Kim, S.-H.; Lim, W. Movement-in-a-Video Detection Scheme for Sign Language Gesture Recognition Using Neural Network. *Appl. Sci.* **2022**, *12*, 10542. https://doi.org/10.3390/app122010542

Academic Editor: Athanasios Nikolaidis

Received: 31 August 2022
Accepted: 13 October 2022
Published: 19 October 2022

1. Introduction

Sign language is used by deaf and hearing-impaired individuals worldwide to communicate with each other and with individuals with normal hearing [1,2]. The World Health Organization (WHO) reported on 20 March 2019, that approximately 466 million individuals exhibit hearing disability [3,4]. This figure is estimated to increase up to 900 million by 2050 [3]. Unlike individuals with normal hearing capability, deaf and hearing-impaired individuals use hand and arm gestures and poses as means of communication [1,5]. In addition, they encounter the problem of limited communication while interacting with individuals with normal hearing. Therefore, mastering sign language is critical for them to express themselves and contribute to society. However, most of the individuals with normal hearing are unskilled in sign languages because it is not a necessity in their daily lives.

To eliminate the limitations imposed by hearing disability on affected individuals, studies have been conducted with the aim of developing tools that can automatically recognize sign languages and convert them into text [6]. Sign language recognition is a technology that bridges the communication gap between the deaf communities and members of society with normal hearing capability [7]. In this technology, sign language gestures are converted into text or speech. Thereby, a new means of human–computer interaction has been produced [7]. The sign language recognition problem is classified into two categories: image-based and sensor-based approaches. Image-based approaches require only a camera to capture a sign language gesture. In contrast, sensor-based approaches require special types of sensors [8]. An example of special sensors is a real-time depth sensor capable of measuring the depth of images. Real-time depth sensors such as Kinect and Leap Motion are generally integrated with a camera to enable recognition of sign languages in three dimensions, with depth as the third dimension.

Sign language gestures are captured in either two or three dimensions. The captured data are used to train a machine learning model for gesture recognition and classification. Machine learning models include a convolutional neural network (CNN), recurrent neural network (RNN), and deep convolutional neural network (DCNN). Kim et al. [9] utilized a CNN to recognize the equivalent hand gestures from an impulse-radio (IR) signal. The IR signals are obtained from a gesture recognition sensor developed by the same authors. The sensor acquires the waveforms reflected by a hand when a gesture is performed. Each hand gesture reflects a distinct waveform, which is used to train a CNN. Six hand gestures in American sign language are recognized with an accuracy of 90%.

Cuszyński et al. [10] proposed a RNN and a gated recurrent unit (GRU) to recognize hand gestures. It training, a dataset of 27 gestures acquired with an optical linear sensor is used. In addition, the effect of data pre-processing on the accuracy of recognition is investigated. The experimental result indicates that the accuracy of the classification of complex gestures by the proposed RNN and GRU can be increased by applying raw data from the optical linear sensor.

Kiran Kumar et al. [11] proposed multiple CNN layers for 3D motion-capture sign language recognition. Herein, the information on each sign is interpreted using joint angular displacement maps.

Rao et al. [12] designed and tested different CNN architectures to recognize Indian sign language by utilizing a front camera. Rather than video inputs, a series of frames are fed to the deep CNN architecture. Frames are still images that are displayed sequentially to present a motion picture. The authors obtained an accuracy of 92.88%.

Bao et al. [13] proposed a DCNN to directly classify gestures in images without the need to localize and segment the specific location of a hand during their realization. A hand occupies approximately 10% of the image area. The proposed methods yield an accuracy of 97.1% using a dataset with simple backgrounds and 85.3% using a dataset with complex backgrounds.

Masood et al. [14] proposed a combination of CNN and RNN to recognize sign language gestures in a video sequence. The spatial and temporal features of the video sequence are utilized to train the CNN and RNN, respectively. Masood et al. [14] claim that their proposed sign language recognition model has an accuracy of 95.2%.

Several other researchers have been proposing complex sign language recognition models and frameworks to overcome the difficulty in recognizing dynamic sign language gestures. That is, the difficulty in recognizing gestures is resolved using models with complex architectures. However, this further results in another issue: model complexity. To resolve the issue of model complexity while still retaining the capability of recognizing the dynamic gestures, an end-to-end sign language gesture recognition framework is proposed in this study: Movement-in-a-Video Detection Scheme. The key idea here is that the model does not find and track the location of the hands. Instead, it detects the movements necessary in performing a gesture, aggregates the movements, and stores it into a single image. This study is aimed at resolving two issues observed in the previous methods: the computation complexity and cost involved in recognizing sign language gestures in videos captured by a camera. First, training a CNN from scratch with a small dataset is complex because it becomes susceptible to overfitting and convergence problems. Transfer learning, which involves the use of pre-trained CNN, is adopted to resolve this issue. [15] In transfer learning, a CNN model is initially trained for a task. The knowledge learned and experiences gained in training a CNN model from that task is then used for another task [16]. Several different model architectures can be used with transfer learning but it must contain a CNN layer. CNN layer is a vital layer in the proposed framework due to its capability in extracting features on images. Second, sensors such as real-time depth sensors, IR sensors, and optical linear sensors are not readily accessible to most of the hearing-impaired individuals. To resolve this, a camera is utilized to capture sign language gestures. Cameras are conveniently accessible to almost all individuals, including the deaf and hearing-impaired. All types of cameras capable of recording a video clip can be used.

This study proposes a movement detection scheme for sign language recognition application. The proposed scheme involves the extraction of spatiotemporal information on videos to an image. The images obtained are used with a pre-trained CNN (to reduce the complexity of training) and any type of camera capable of recording a video (to ensure accessibility). In particular, CNN is trained to recognize sign language gestures. The gestures are captured using a camera and stored as a video file. The information in each video file is extracted and stored as an image. Each sign language gesture is mapped to a distinct image, which is then used to train the CNN.

The objectives of this study are as follows:

1. To extract both the spatial and temporal information in a video and store it to an image.
2. To classify a dynamic sign language gesture using a neural network (NN) trained with images containing the spatiotemporal information in a video.

The main contributions of this paper are as follows:

1. A movement-in-a-video detection method that is proposed for dynamic sign language recognition application is capable of reducing computational complexity and cost by using an image containing both spatial and temporal information rather than a sequence of frames.
2. Compared with the current methods, the proposed method is capable of solving the dynamic sign language gesture recognition task using any CNN that is pre-trained for an image classification task.

2. Sign Language Recognition Method

The proposed sign language recognition method is shown in Figure 1. The input to the system is a dataset of videos containing sign language gestures. The video sequences are processed to extract the information in each frame, and that information is stored in an image. Each sign language gesture renders a distinct image representing the gesture. These images are used to train an NN. Finally, the performance of the NN in solving sign language recognition tasks is evaluated.

A flow chart of the proposed sign language recognition method is shown in Figure 1a. As shown in the figure, the blocks containing the processes are labeled from one to five. The recognition method can be described as follows:

1. In the first block, a series of frames are extracted from each video. The number of frames per video depends on the duration of each gesture. The input to the first block is a video of sign language, as shown in Figure 1b. Meanwhile, the output is a series of frames, as shown in Figure 1c.
2. In the second block, each extracted frame is subtracted from its preceding frame. The input to the second block is a series of extracted frames, as shown in Figure 1c. Meanwhile, the output is a series of frame differences, as shown in Figure 1d. Frame difference is the difference between two consecutive frames.
3. In the third block, the differences between consecutive frames obtained in the second block are added. The input to the third block is an image representing frame differences, as shown in Figure 1d. Meanwhile, the output is an image representing the sum of frame differences, as shown in Figure 1e.

In the fourth and fifth blocks, the images obtained in the third block are used to train and test the NN for recognizing sign language gestures.

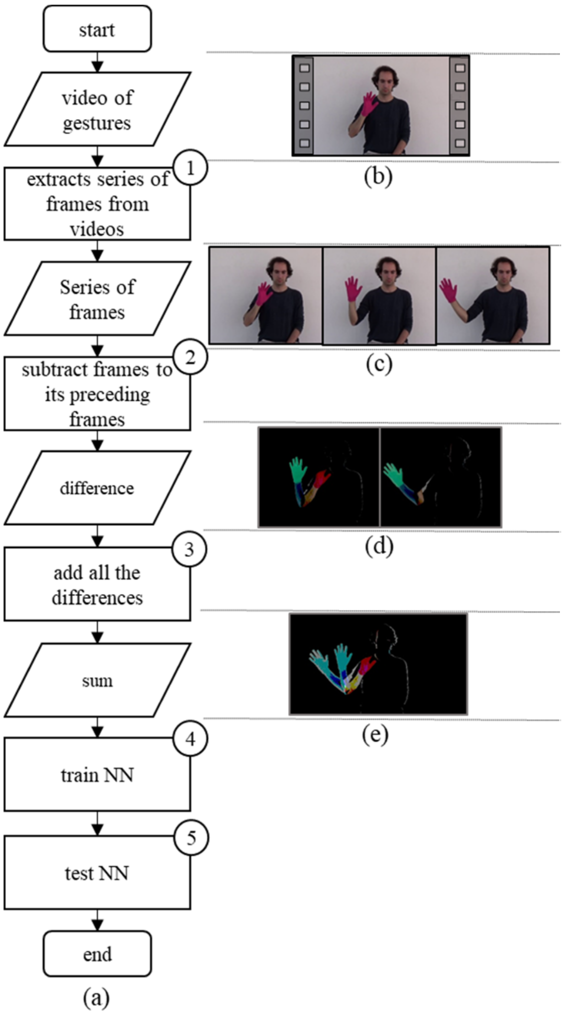


Figure 1. Proposed sign language recognition method. (**a**) Flow chart of proposed sign language recognition method. (**b**) Videos of sign language gestures. (**c**) Frames extracted from videos of sign language gestures. (**d**) An image representing the difference between two consecutive frames. (**e**) Sum of all frame differences.

The concept of obtaining the sum of frame differences is adopted from the sum of absolute difference (SAD) architecture used in block-based motion estimation (ME). [17–19] Block-based ME is widely used in video coding standards owing to its advantage of efficient removal of temporal redundant frames. [18] In block-based ME, each frame is divided into equal-sized M × N blocks. Each block in the current frame is compared with the blocks in the reference frame. The best matching blocks in the current and reference frames are obtained using SAD. The displacement between the best matching current block and reference block is then stored in a motion vector (MV). SAD involves two calculations: first, the calculation of the absolute difference and then, the accumulation of absolute differences. This is expressed as:

$$SAD(x,y,i,j) = \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} \left| A_{(x+l,y+k)} - B_{(x+i+l,y+j+k)} \right| \quad (1)$$

where (x, y) are the coordinates of the block in the reference frame; (i, j) are the x and y distances, respectively, of the block in the current frame with the block in the reference frame; (l, k) are the coordinates of the pixels in the equal-sized $M \times N$ blocks; and $A_{(x+l,y+k)}$ and $B_{(x+i+l,y+j+k)}$ are the pixels in the blocks of the current and reference frame, respectively [17,19].

2.1. Sign Language Dataset

To demonstrate the effectiveness of the proposed movement-detection scheme in detecting the movements necessary in a sign language gesture, we first used a dataset of gestures of short lengths: Argentinian sign language (LSA64) [20]. It consists of only two to three hand poses and positions per gesture. We then considered sign language gestures of medium and long lengths: DEVISIGN-D [21] and sign language recognition (SLR) dataset [22], respectively.

2.1.1. Argentinian Sign Language (LSA64)

LSA64 is an Argentinian sign language dataset containing the most common vocabularies in the LSA lexicon. It contains videos of 10 non-expert gesturers performing sign language gestures for 64 vocabularies (including verbs and nouns). Each gesturer performs each sign language gesture five times. The dataset comprises two sets, which differ in terms of the nature of the recording. This is intended to provide differences in illumination between signs. The first and second sets were recorded outdoors (natural lighting) and indoors (artificial lighting), respectively. The second set is divided further into two: one-handed and two-handed gestures. Figure 2a–c show a few examples of the two sets of the LSA64 dataset. Figure 2a is an image captured from a video recorded outdoors (natural lighting), Figure 2b is an image captured from a video of a one-handed gesture recorded indoors (artificial lighting), and Figure 2c is a snippet of video of a two-handed gesture recorded indoors (artificial lighting). Among the datasets used in this study, only the LSA64 dataset includes information on lighting in the videos.

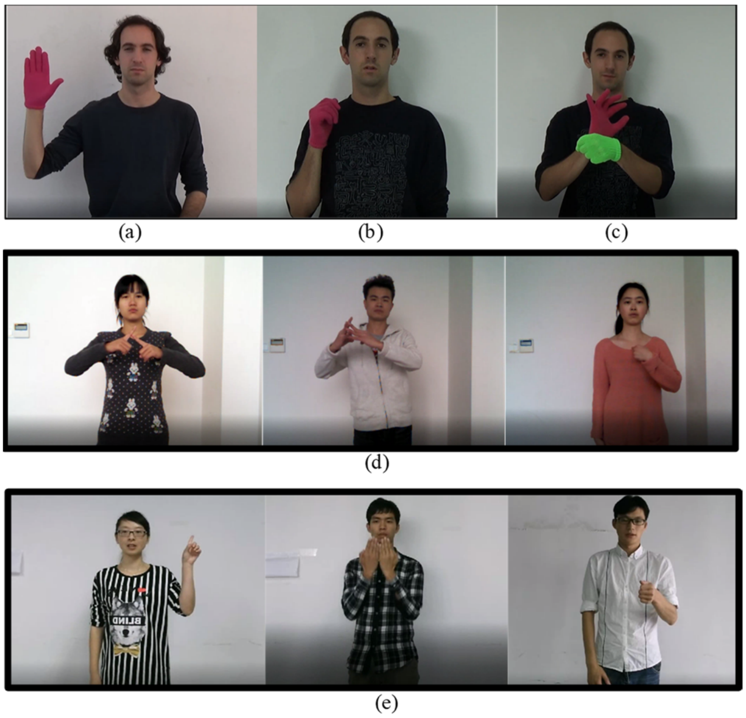


Figure 2. Argentinian sign language: (**a**) one-handed gesture recorded outdoors with natural lighting, (**b**) one-handed gesture recorded indoors with artificial lighting, and (**c**) two-handed gesture recorded indoors with artificial lighting. Chinese sign language: (**d**) DEVISIGN-D and (**e**) SLR_dataset.

2.1.2. DEVISIGN-D Dataset

DEVISIGN-D is a Chinese sign language dataset of 500 commonly used words. It contains videos of eight gesturers: four gesturers (two males and two females) performing each sign language gesture two times, and the remaining four gesturers (two male and two female) performing it one time. Figure 2d shows an example of the DEVISIGN-D dataset.

2.1.3. Sign Language Recognition Dataset (SLR_Dataset)

The SLR_Dataset is a Chinese sign language dataset of 100 Chinese sentences. It contains videos of 50 gesturers performing each sign language gesture five times. Figure 2e shows a few examples of the SLR_Dataset.

2.2. Sum of Frame Difference

The difference between two consecutive frames is obtained using Algorithm 1. In this method, only three images are stored as the information is processed:

1. Current frame, $frame_k$—contains the newly imported frame.
2. Preceding frame, $frame_{k-1}$—contains the preceding frame of $frame_k$.
3. Sum-of-differences frame, $frame_s$—sum of the differences between $frame_k$ and $frame_{k-1}$.

In Algorithm 1, Step 1 sets the $frame_{count}$ to the total number of extracted frames. Step 2 initializes the value of $frame_s$ to zero or to an empty frame. Steps 4–22 are repeated for each of the frames extracted in a video. Steps 4–22 are explained below:

In Step 4, each frame is imported, and in Step 5, it is saved to the current frame. In Steps 6–10, the imported frame is examined to ascertain whether it is the first frame. If that is the case, it indicates that the preceding frame, $frame_{k-1}$, is empty. This step is necessary because the current frame will always be subtracted from the preceding frame, as in Step 7. The difference between the current frame and its preceding frame is set as the difference frame, $frame_d$. Given that the first frame has no preceding frame, it is directly set as the preceding frame for the next frame, as in Step 9.

Algorithm 1 Difference of two consecutive frames

```
1   $frame_{count}$ = total number of frames extracted;
2   $frame_s = 0$;
3   for each frame, $frame_i$, do
4       import $frame_i$;
5       $frame_k = frame_i$;
6       if $i \neq 1$ do
7           $frame_d = frame_{k-1} - frame_k$;
8       else
9           $frame_{k-1} = frame_k$;
10      end if
11      for each element in $frame_d$, $frame_{d,jk}$ do
12          if $frame_{d,jk} > 200$ or $frame_{d,jk} < 55$ do
13              $frame_{d,jk} = 0$;
14          end if
15      end for
16      for each element in $frame_d$, $frame_{d,jk}$ do
17          if $frame_{d,jk} \neq 0$ do
18              $frame_{d,jk}{}^* = \lfloor \frac{255}{frame_{count}-1} \rfloor$;
19          end if
20      end for
21  $frame_s = frame_s + frame_d$;
22  $frame_{k-1} = frame_k$;
23  end for
```

Steps 12–14 are repeated for each pixel in the difference frame, $frame_d$. These steps assign a pixel value of zero to the pixel above and below the upper and lower thresholds, respectively.

An example of a pixel value and its corresponding color in grayscale mode is shown in Figure 3. Pixel values from 0 to 255 render a color gradient from black to white. A pixel value of 0 corresponds to black, whereas a pixel value of 255 corresponds to white. Pixel values between 0 and 255 render colors with different shades of gray.

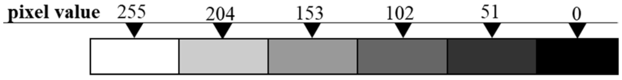
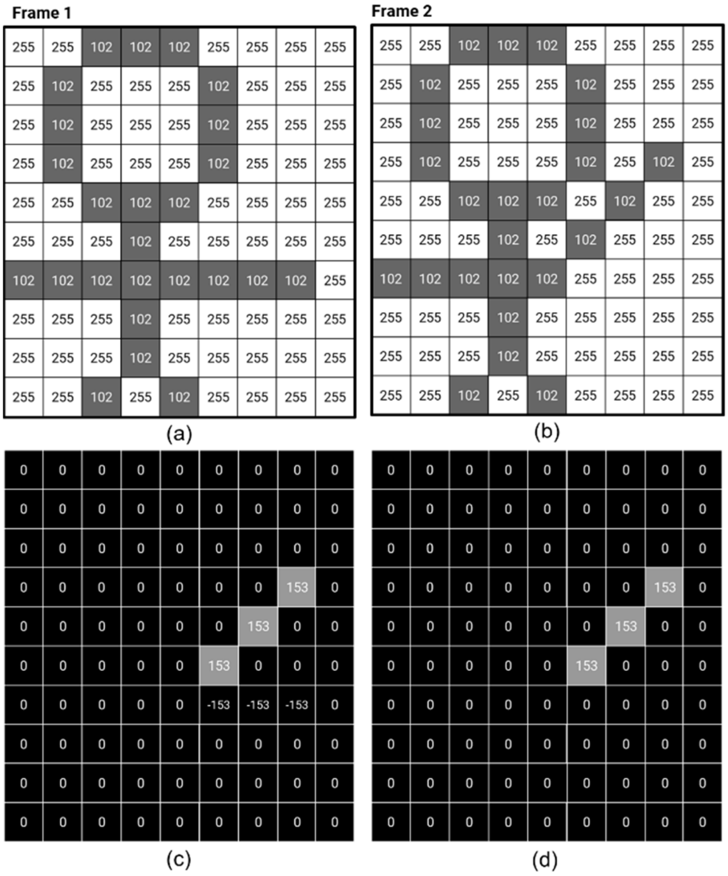


Figure 3. Pixel values and their corresponding colors in grayscale mode.

An example of movement detection between two consecutive frames is shown in Figure 4. The positions of the object in the first and second frames are shown in Figure 4a,b, respectively. In these figures, the background and subject take pixel values of 255 and 102, respectively. The pixel values are selected randomly to provide an example. The transition from Figure 4a to Figure 4b clearly indicates an upward left arm movement of the subject.



Frame 1

255	255	102	102	102	255	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	255	102	102	102	255	255	255	255
255	255	255	102	255	255	255	255	255
102	102	102	102	102	102	102	102	255
255	255	255	102	255	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	102	255	102	255	255	255	255

(a)

Frame 2

255	255	102	102	102	255	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	102	255
255	255	102	102	102	255	102	255	255
255	255	255	102	255	102	255	255	255
102	102	102	102	102	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	102	255	102	255	255	255	255

(b)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	153	0
0	0	0	0	0	0	153	0	0
0	0	0	0	0	153	0	0	0
0	0	0	0	0	-153	-153	-153	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	153	0
0	0	0	0	0	0	153	0	0
0	0	0	0	0	153	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Figure 4. Example of movement detection between two consecutive frames: (**a**) position of subject in the first frame; (**b**) position of subject in the second frame; (**c**) result of subtracting the pixels in the first frame from those in second frame; (**d**) result of setting pixel values higher and lower than the upper and lower thresholds, respectively, to zero.

The result of subtracting the first frame from the second frame is shown in Figure 4c. Only the specific part of the subject that moved was retained.

In Figure 4d, the pixels with values higher and lower than the upper and lower thresholds, respectively, are set to zero. The figure clearly shows the portion of the subject that moved and its final position in the whole frame. The values that yield the least noise in the output image are selected as the thresholds by trial and error.

In Steps 12–14, although pixels with negative values are mapped automatically to black, they are set to zero for computational evidence.

Steps 17–19 are repeated for each pixel in the difference frame, $frame_d$. These steps assign a pixel value of

$$frame_{d,\ jk}{}^{*} = \frac{255}{frame_{count}-1} \tag{2}$$

to a non-zero pixel. Here, * is used to differentiate the current value of $frame_{d,\ jk}$ (used as a reference) to its new assigned value; (j,k) are the coordinates of the non-zero pixel. In the example in Figure 4, the number of frame count is two. This means that

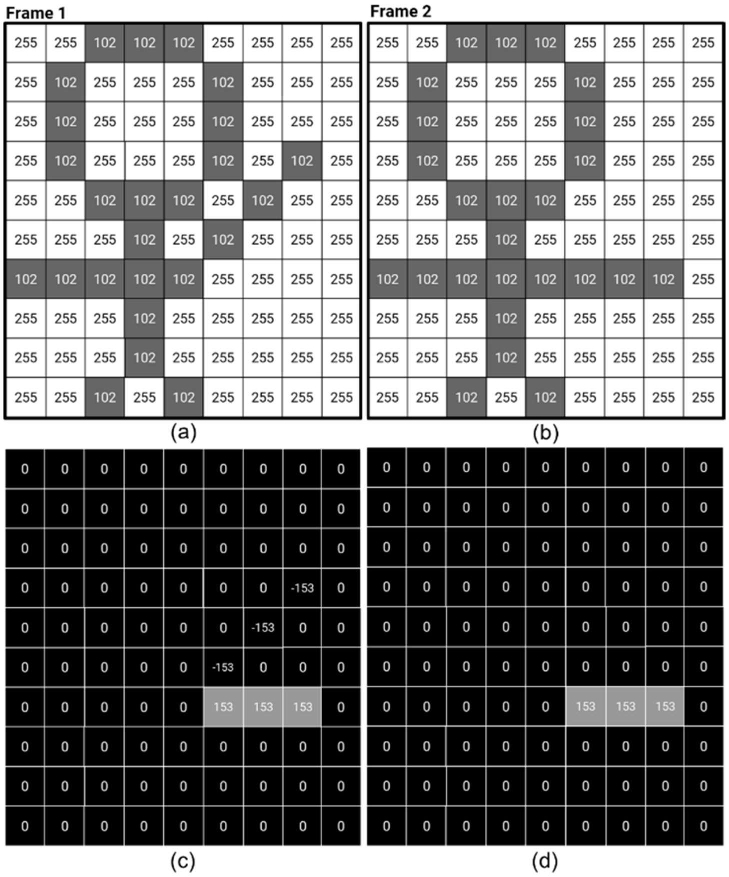
$$frame_{count} = 2 \tag{3}$$

Therefore, the non-zero pixels in Figure 4d are set to $\lfloor 255/(2\text{-}1) \rfloor$ or 255. The floor value is selected to ensure that the sum of the pixels in all the frames, $frame_d$, does not exceed 255.

These steps are necessary to determine which portion of the frame is occupied by the subject while performing the movement. Pixel values closer to 255 indicate the portion of the frame occupied by the subject in more than one frame.

In sign language, the order of gestures is important for conveying a message. An advantage of the use of images obtained from the sum of frame differences over that of a sequence of original frames is that the information regarding the order or sequence of frames is also preserved and stored.

An example is shown in Figure 5 for the purpose of visualization. In the figure, the order of the first and second frames in Figure 4 is reversed. The resulting frame differences in Figure 5c,d show that different orders of frames yield different frame differences. This is because the difference between two frames captures the motion from one frame to another with respect to a reference frame.



Frame 1

255	255	102	102	102	255	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	102	255
255	255	102	102	102	255	102	255	255
255	255	255	102	255	102	255	255	255
102	102	102	102	102	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	102	255	102	255	255	255	255

(a)

Frame 2

255	255	102	102	102	255	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	102	255	255	255	102	255	255	255
255	255	102	102	102	255	255	255	255
255	255	255	102	255	255	255	255	255
102	102	102	102	102	102	102	102	255
255	255	255	102	255	255	255	255	255
255	255	255	102	255	255	255	255	255
255	255	102	255	102	255	255	255	255

(b)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-153	0
0	0	0	0	0	0	-153	0	0
0	0	0	0	0	-153	0	0	0
0	0	0	0	0	153	153	153	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

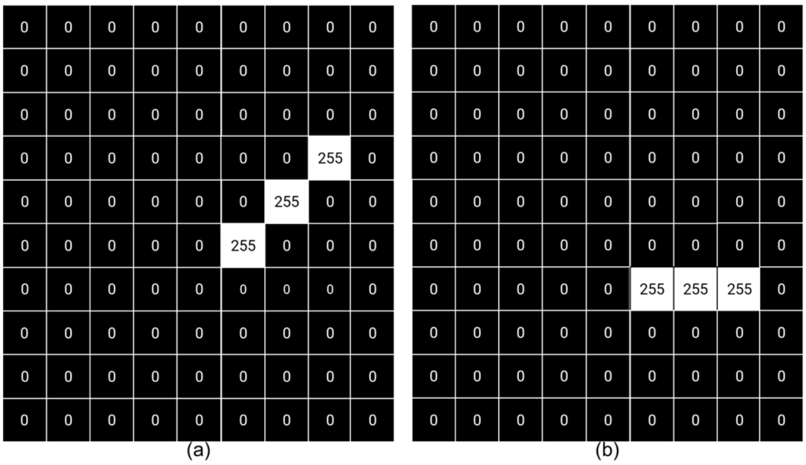
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	153	153	153	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Figure 5. Example of movement detection between two consecutive frames wherein the frames are in a reverse order to that of Figure 4. (**a**) position of subject in the first frame; (**b**) position of subject in the second frame; (**c**) result of subtracting the pixels in the first frame from those in the second frame; and (**d**) result of setting pixel values higher and lower than the upper and lower thresholds, respectively, to zero.

Because both the movements and their order are stored in the images by using the proposed movement-in-a-video detection scheme technique, the sign language recognition problem can be considered as a simple image classification problem. Thereby, the complexity of the sign language recognition problem can be reduced. That is, the proposed method is capable of extracting both the spatial and temporal features of a video sequence and storing them in one image.

In Step 21, the sum-of-differences frame, $frame_s$, is updated by adding the newly obtained difference frame, $frame_d$. The results of performing Steps 17–21 for the examples in Figures 4 and 5 are shown in Figure 6a,b, respectively.



0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	255	0
0	0	0	0	0	0	255	0	0
0	0	0	0	0	255	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	255	255	255	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Figure 6. Sum and difference frame result of examples illustrated in (**a**) Figure 4 and (**b**) Figure 5.

In Step 22, the current frame, $frame_k$, is assigned as the new preceding frame, $frame_{k-1}$. This is a preparation step for importing the next frame.

3. Result and Discussion

In this study, videos of Argentinian and Chinese sign language gestures from LSA64, DEVISIGN-D, and SLR_Dataset were used. First, the frames of each video were extracted. The number of frames depends on the duration of the gestures. The results of frame extraction from the videos in the three datasets are shown in Figure 7.

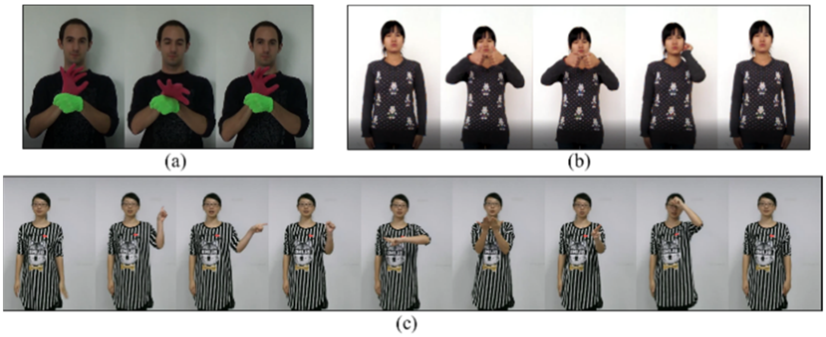


Figure 7. Examples of frame sequences extracted from a video in (**a**) LSA64, (**b**) DEVISIGN-D, and (**c**) SLR_Dataset.

As shown in the figure, LSA64 contains videos of sign language gestures with an average of three hand and arm poses. Therefore, these are classified as short gestures. DEVISIGN-D contains videos of sign language gestures with an average of five hand and arm poses. Therefore, these were classified as medium gestures. SLR_Dataset contains videos of sign language gestures with more than five hand and arm poses. Therefore, these are classified as long gestures.

For the purpose of analysis, further examples of the result of extracting the frame sequence of short gesture videos (LSA64) are shown on Figure 8. In the figure, both the motion and pose of the hands while performing the sign language gesture can be inferred. Thus, the sign language gesture can be visualized regardless of the absence of frames in between those shown.

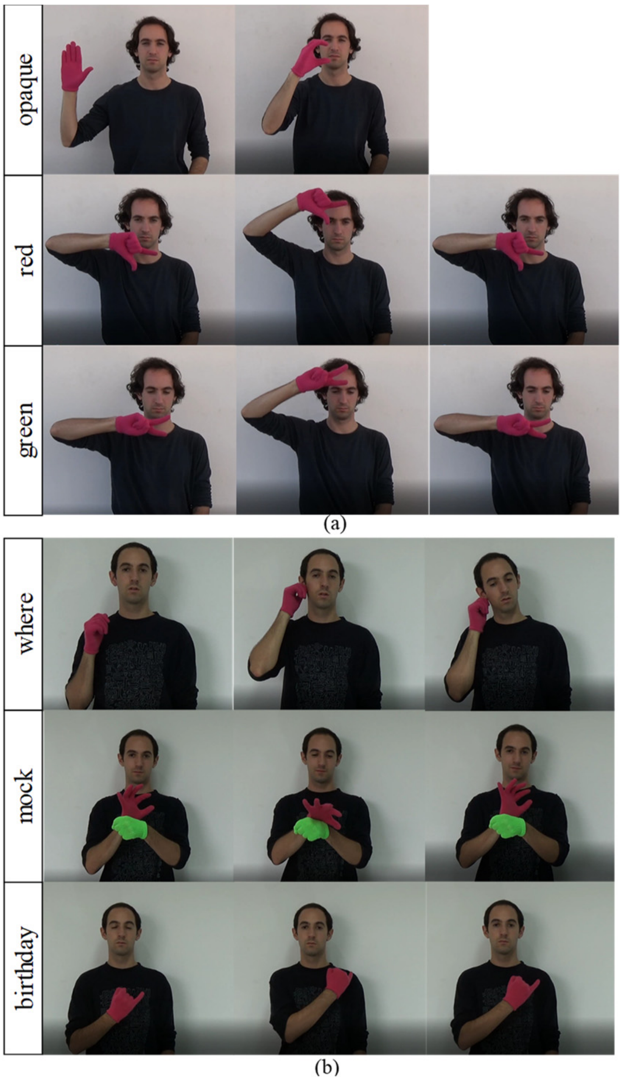


Figure 8. Six examples (out of 64) of frame sequences extracted from videos of sign language gesture recorded (**a**) outdoors and (**b**) indoors.

Figure 8a shows a few of the frames extracted from a video recorded outdoors (natural lighting). The sign language gesture for the word "opaque" involves two hand poses at two positions. This gesture can be performed by first copying the hand's position and pose in the first frame and then copying the hand's position and pose in the second frame. Regardless of the speed of transition between two hand poses and positions, the combination of the two constitutes the gesture for "opaque". Unlike the sign language for "opaque", those for "red" and "green" involve one hand pose each. However, to perform the gesture, the hand is situated in different positions. The gesture begins with the hand positioned below the face. Then, it is raised above the face and finally dropped below the face again, returning it to its initial position. Figure 8b shows a few of the frames extracted from a video recorded indoors (artificial lighting). It is evident that, similar to Figure 8a, the gesture can be performed by copying the pose and position of the hand in each frame consecutively. An exception to this is the sign language gesture for "where". Herein, apart from the hand pose and position, the gesture involves movement of the head as well. An examination of the frames enables visualization of the sign language gesture when the frames are in order. Thereby, the need for viewing the video is circumvented.

To demonstrate the effect of the proposed method further, we classify the gestures in the LSA64 dataset into three based on similarity: (1) identical, (2) similar, and (3) unique gestures. This is apart from the classification according to lighting. An example of each gesture classification is shown in Figure 9a,b. These figures reveal the following: (1) The gestures for "red" and "green" exhibit highly identical paths of hand and differ only in the hand pose. (2) The gestures for "away" and "colors" exhibit a similar path of hand. However, notwithstanding the similarities in the path of hand (rightward away from the gesturer), there is an observable difference between these. (3) The paths of hand for unique gestures are different. These observations are expected to be reflected in the result of the sum and frame difference method. This is explained next.

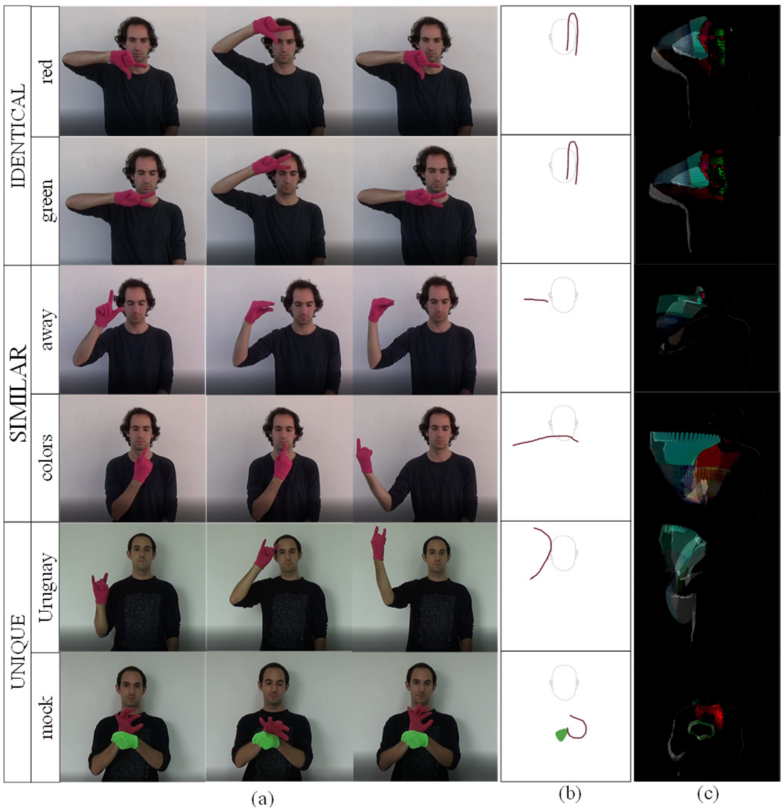


Figure 9. Examples of gestures classified into identical, similar, and unique: (**a**) frame sequence, (**b**) path of the hand, and (**c**) result when the sum of frame differences method is applied.

After extracting the frame sequence, each frame was subtracted from the previous frame. Thereafter, the differences were added. Examples of images obtained by applying the sum of frame differences on the sequence of frames from the three datasets are shown in Figure 10. The observations made based on this figure are as follows:

1. Each video of the three datasets results in a distinct image after the sum-of-the-frame-differences method is applied. Evidently, these images can be used to train an NN to classify the sign language gesture corresponding to the image.
2. As the gesture length increases, the clarity of the paths of hands and arms decreases. That is, whereas the paths of hands and arms can be depicted for short and medium-length gestures, these cannot be easily depicted accurately for long gestures. This is expected because longer gestures have more frames in the sequence.

3. As the gesture length increases, the visibility of the outline of the gesturer's body also increases. This is because the longer the gesture is, the higher is the tendency of the gesturer to make small body movements.

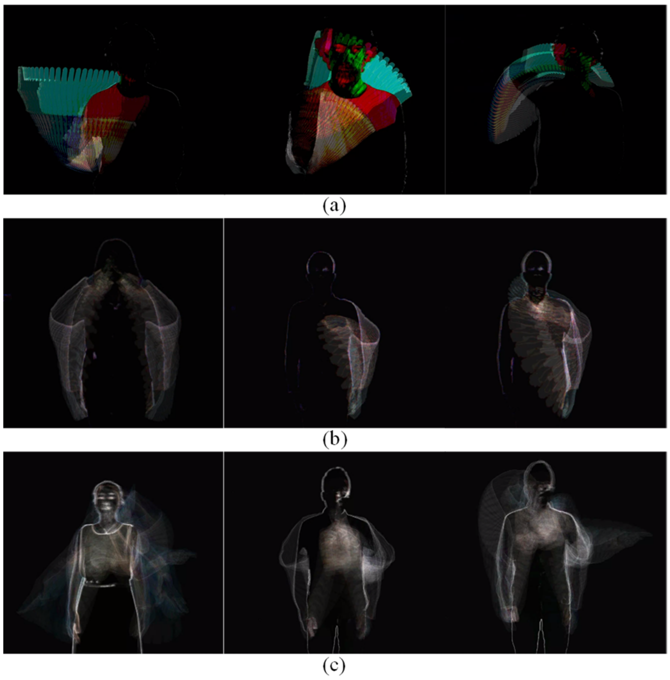


Figure 10. Examples of sum of frame differences using (**a**) LSA64, (**b**) DEVISIGN-D, and (**c**) SLR_Dataset.

To demonstrate the effect of the sum of the frame differences method, we illustrated further examples of the result of applying the method on the LSA64 dataset in Figure 8. Figure 11a is from videos recorded outdoors (natural lighting), and Figure 11b from videos recorded indoors (artificial lighting). The result in Figure 11 clearly shows that the sum-of-the-frame-differences method is successful in detecting the movement in the video regardless of the lighting condition under which the gesture is recorded.

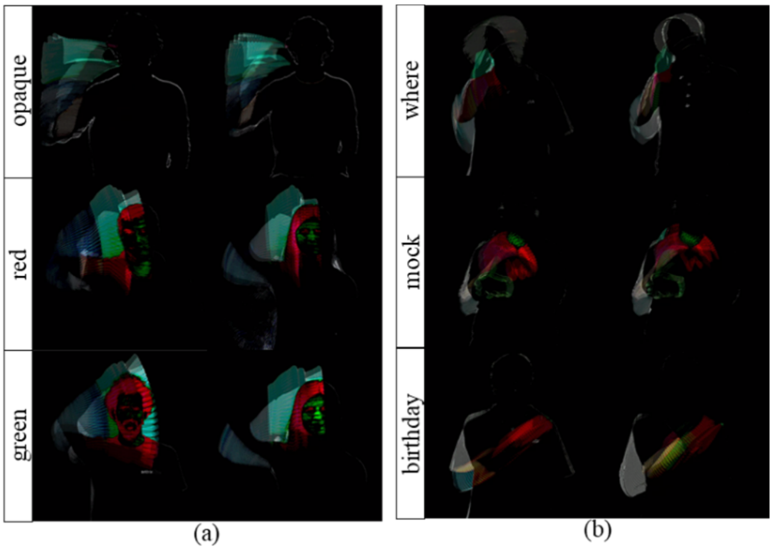


Figure 11. Resulting images of the sum of frame differences method applied on LSA64 dataset videos recorded (**a**) outdoors (natural lighting) and (**b**) indoors (artificial lighting).

In addition to that, except for the images representing "red" and "green", all the resulting images can be easily distinguished. This enables the mapping of the resulting image to its corresponding gesture without the need for the information in the corresponding video. Meanwhile, a closer examination of the images representing "red" and "green" is necessary to identify the corresponding gesture. This is because the two hand gestures differ only in the hand pose. Even when actually performed, it is difficult to identify the gesture unless we observe the hand pose. This is expected because the gestures are classified as identical gestures. To observe the result of applying the proposed method on (1) identical, (2) similar, and (3) unique gestures, examples are given in Figure 9c. The images resulting from the application of the sum of the frame differences method to the sequence of frames in Figure 9a are shown in Figure 9c. The following observations are made:

1. The images corresponding to the sum of frame differences of identical gestures are also identical. This is expected because the gestures differ only in hand pose.
2. The resulting images from the sum of frame differences of similar gestures are distinguishable. This is expected because notwithstanding the similarity in the path of the hand while the gesture is performed, the hand pose and direction are different.
3. Similar to the results for identical gestures, the images resulting from the sum of frame differences for the unique gesture is distinguishable. This is expected because the trajectory and hand pose while performing the gesture is different from those of the remaining gestures.

Previous observations are focused on comparing the images resulting from the frame differences for three gesture categories. The following observations are focused on the images resulting from the sum of frame differences in general:

1. The gesture must be performed with a still background. All movements, variations in position of objects, alterations in scenery, etc., are considered by the proposed method as a part of the gesture. All three images shown in Figure 12 contain information on the gesture expressing the word "bright." However, the gesture involves only the movement shown under the yellow-marked area. The portions of the image under the blue-marked area correspond to the movement of the signer's hair during the implementation of the gesture.
2. Variations in the images are expected given that the presentation of each gesture depends on the signer, occupied area on the image, length of time a gesture is performed, etc. A few of the images trace the figure of the signer. Although the upper body is ideally stationary, small movements are unavoidable. It is important to note that in Figure 13, the areas occupied for this particular gesture are different although the same word "bright" is performed by 10 signers.
3. As shown in Figure 14, the images obtained by taking the sum of frame differences are similar for the words "red" and "green" and the words "bright" and "skimmer". Only the hand pose in performing the gesture is different.

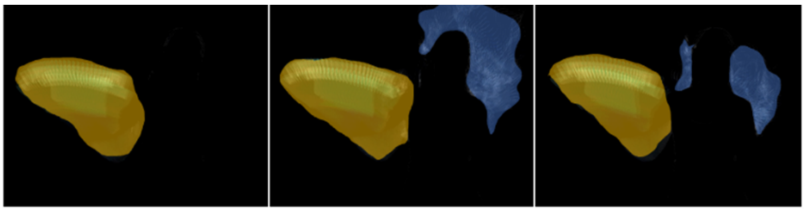


Figure 12. Variation in the image obtained by taking the sum of frame differences for the gesture "bright" performed three times by a signer.

Finally, the images containing the sum of differences of consecutive frames for each gesture were used to train Inception V3. Other types of models with a CNN layer on its architecture can be used. CNN layer is a vital layer as it is mainly responsible for extracting the features on images. In this study, we will only use Inception V3.

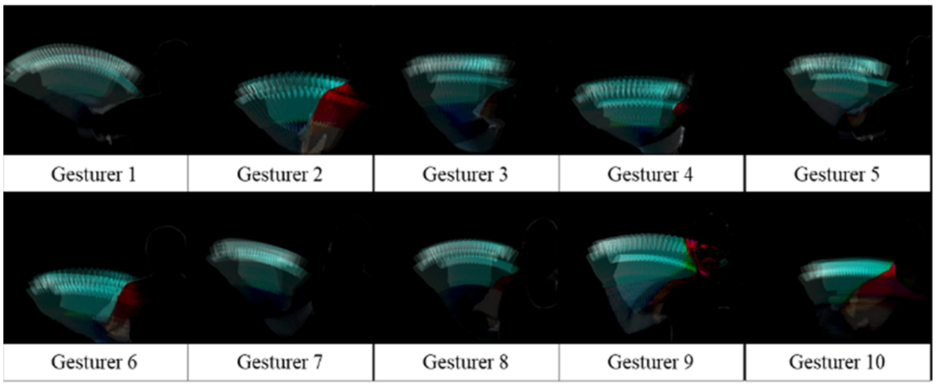


Figure 13. Variations in the image obtained by taking the sum of frame differences for the gesture "bright" performed by 10 different signers.

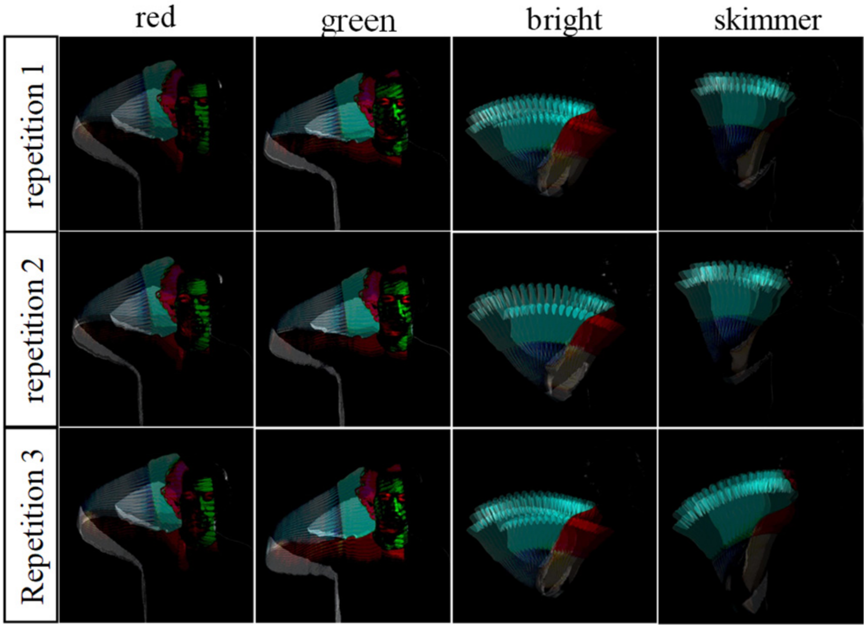


Figure 14. Images obtained by taking the sum of frame differences for the gestures "red", "green", "bright", and "skimmer" performed three times by a signer.

We first classified sign languages with short gestures (LSA64) and compared the result with those of other recent work [14]. LSA64 contains videos of 10 gesturers performing 64 gestures five times each, i.e., 50 videos per gesture and a total of 3200 videos. To prevent overfitting, 70% of the data are used for training (35 videos per gesture), 10% for validation (5 videos per gesture), and the remainder for testing (10 videos per gesture). The processes and time consumption in training each method are illustrated in Table 1. Two types of information are necessary for accurate sign language recognition: (1) spatial and (2) temporal information. Because CNN can capture only the spatial information, training a model with frame sequence requires additional network to preserve the temporal information in a sequence. The commonly used network to extract the temporal information is the RNN. However, because both the spatial and temporal information in the video can be extracted and stored in an image using the proposed method, there is no need for an additional network or RNN.

As shown in Table 1, the proposed method reduced the training time consumption substantially (by 95.62%). The process that consumed more time is the prediction of the spatial feature. In that process, the class at which each frame in the sequence, is predicted. That is, the output of this process is a sequence of n-dimensional vectors containing the probability of each class. Herein, *n* is the number of classes. Because the proposed method uses images that contain both spatial and temporal information of the videos, this process is eliminated. Thereby, the time consumption in training is reduced substantially. The accuracy of prediction of the gestures in the LSA64 dataset is presented in Table 2.

Table 1. Time consumption (in seconds, s) by each training process.

		LSA64	
Process	**Data**	**CNN + RNN [14]**	**Proposed Method**
preprocessing	Train	3225	7221
	Test	783	1817
retrain CNN	Train	804	499
predict spatial	Train	202,502	-
	Test	10,022	-
train RNN	Train	170	-
TOTAL		217,506	9537

Table 2. Summary of accuracies of classifying the gestures in LSA64, DEVISIGN-D, AND SLR_Dataset.

	LSA64	**DEVISIGN-D**	**SLR**
CNN + RNN [14]	0.6900	-	-
BLSTM-3D [7]	-	0.437	0.502
proposed method	0.9033	0.40	0.2236

Apart from the short gesture, we considered the medium and long gestures. We also compared the results of the proposed method with the combination of CNN and RNN architectures. Liao et al. [7] uses long short-term memory (LSTM) rather than simple RNN. LSTM is used because of its capability of learning long-term dependencies. Two experiments are performed: (1) without hand localization and (2) with hand localization. Because hand localization is out of the scope of the study, we compared our results with those of the second experiment. Apart from the accuracy in predicting the gestures in the LSA64 dataset, Table 2 also lists the accuracies of predicting the gestures in DEVISIGN-D and SLR_Dataset.

As illustrated in Table 2, the proposed method exhibits a higher accuracy (90.33%) of predicting short gestures (LSA64) compared with the combination of CNN and RNN (69%). Because the time consumptions listed in Table 1 are less using the proposed method, the results evidently reveal that the proposed method is effective in predicting short gestures.

The method used by Liao et al. [7] exhibits higher accuracy than the proposed method in predicting medium and long gestures (by 7.5% and 27.84%, respectively). However, the main objective of this study is to reduce both the complexity and cost of sign language recognition by extracting both the spatial and temporal information in a video and storing it in an image. This necessitates only a network capable of extracting spatial features and eliminating the need for other networks to extract the temporal features. Whereas Liao et al. [7] used both CNN and RNN (specifically LSTM), this method used only CNN. Therefore, considering the reduction in the computing complexity and cost, a difference of 7.5% is reasonable. However, we consider the fact that accuracy in recognizing sign language is important.

Liao et al. [7] applied hand localization to increase the accuracy of gesture recognition in DEVISIGN-D and SLR_Dataset. Hand localization can be applied to improve the accuracy obtained using the proposed method as well.

The results obtained in predicting long gestures (SLR_Dataset) is expected because more than five hand and arm poses and positions are required to complete a gesture. Because of the long duration needed to perform long gestures, it is inevitable that the gesturers make small movements. These small movements are also detected by the proposed algorithm. To avoid these issues, these small movements must be filtered out, with focus only on the gestures that have a big difference between the initial and final arm poses and positions. In addition, Inception V3 is tuned only for the images containing spatial and temporal information of short gestures. As shown in Figure 9, the path of hands and arms are clear for short gestures and unclear for long gestures. Hence, to improve the model accuracy for long gestures, a feature extractor must be tuned for this type of images.

4. Conclusions and Future Works

In this paper, a sign language recognition method with the movement-in-a-video detection scheme is proposed. In the proposed method, the movement-in-a-video detection scheme was applied to extract unique spatial and temporal features from each gesture. The extracted features were subsequently used with a pre-trained CNN to classify sign language gestures. In the movement-in-a-video detection scheme, a series of frames were extracted from each video. Thereafter, each extracted frame was subtracted from its preceding frame. Finally, the differences between consecutive frames were added together. The proposed method was used to identify sign language with short, medium, and long gestures in the Argentinian and Chinese sign language datasets. The proposed method identifies sign language with higher accuracies of 90.32% and 40% for short and medium gestures, respectively, compared with those (69% and 43.7%, respectively) achieved using methods in other recent works. The results clearly show that the proposed method can be used to recognize sign language gestures. Although the proposed method is tuned for short gestures only, the results show that it can be extended to medium gestures. This will enable real-time processing and recognition of videos and avoid overestimation by using methods with complex and expensive computation for short and medium gestures.

In future works, we intend to extend the study by improving the extraction of information from videos and storage to an image for long gestures. Furthermore, apart from Inception V3, other types of CNNs that are capable of solving image classification tasks will be used.

Author Contributions: Conceptualization, A.C.C., H.-J.H., S.-H.K. and W.L.; methodology, A.C.C. and W.L.; software, A.C.C. and W.L.; validation, H.-J.H., S.-H.K. and W.L.; formal analysis, A.C.C.; investigation, A.C.C. and W.L.; resources, H.-J.H., S.-H.K. and W.L.; data curation, A.C.C.; writing—original draft preparation, A.C.C. and W.L.; writing—review and editing, H.-J.H., S.-H.K. and W.L.; visualization, A.C.C. and W.L.; supervision, H.-J.H., S.-H.K. and W.L.; project administration, H.-J.H., S.-H.K. and W.L.; funding acquisition, H.-J.H., S.-H.K. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) funded by the MSIT (2020R1A4A1017775).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Ethics Committee of Kumoh National Institute of Technology (202007-HR-003-04).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mittal, A.; Kumar, P.; Roy, P.P.; Balasubramanian, R.; Chaudhuri, B.B. A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion. *IEEE Sens. J.* **2019**, *19*, 7056–7063. [CrossRef]
2. Kamal, S.M.; Chen, Y.; Li, S.; Shi, X.; Zheng, J. Technical Approaches to Chinese Sign Language Processing: A Review. *IEEE Access* **2019**, *7*, 96926–96935. [CrossRef]
3. World Health Organization. Deafness and Hearing Loss. Available online: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed on 10 May 2019).
4. Oliveira, T.; Escudeiro, N.; Escudeiro, P.; Rocha, E.; Barbosa, F.M. The Virtual Sign Channel for the Communication between Deaf and Hearing Users. *IEEE Rev. Iberoam. Tecnol. Aprendiz.* **2019**, *14*, 188–195.

5. Tubaiz, N.; Shanableh, T.; Assaleh, K. Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode. *IEEE Trans. Hum.-Mach. Syst.* **2015**, *45*, 526–533. [CrossRef]
6. Xiao, Q.; Qin, M.; Guo, P.; Zhao, Y. Multimodal Fusion Based on LSTM and a Couple Conditional Hidden Markov Model for Chinese Sign Language Recognition. *IEEE Access* **2019**, *7*, 112258–112268. [CrossRef]
7. Liao, Y.; Xiong, P.; Min, W.; Min, W.; Lu, J. Dynamic Sign Language Recognition Based on Video Sequence with BLSTM-3D Residual Networks. *IEEE Access* **2019**, *7*, 38044–38054. [CrossRef]
8. Deriche, M.; Aliyu, S.O.; Mohandes, M. An Intelligent Arabic Sign Language Recognition System Using a Pair of LMCs with GMM Based Classification. *IEEE Sens. J.* **2019**, *19*, 8067–8078. [CrossRef]
9. Kim, S.; Han, H.; Kim, J.; Lee, S.; Kim, T. A Hand Gesture Recognition Using Reflected Impulses. *IEEE Sens. J.* **2017**, *17*, 2975–2976. Available online: https://ieeexplore.ieee.org/document/7874149/ (accessed on 12 October 2018). [CrossRef]
10. Cuszyński, K.; Rumiński, J.; Kwaśniewska, A. Gesture Recognition With the Linear Optical Sensor and Recurrent Neural Networks. *IEEE Sens. J.* **2018**, *18*, 5429–5438. Available online: https://ieeexplore.ieee.org/document/8357549/ (accessed on 15 October 2022). [CrossRef]
11. Kumar, E.K.; Kishore, P.; Sastry, A.; Kumar, M.; Kumar, D. Training CNNs for 3-D Sign Language Recognition with Color Texture Coded Joint Angular Displacement Maps. *IEEE Signal Process. Lett.* **2018**, *25*, 645–649. Available online: https://ieeexplore.ieee.org/document/8319435/ (accessed on 15 October 2022). [CrossRef]
12. Rao, G.; Syamala, K.; Kishore, P.; Sastry, A. Deep Convolutional Neural Networks for Sign Language Recognition. In Proceedings of the Conference on Signal Processing and Communnication Engineering Systems, Vijayawada, India, 4–5 January 2018.
13. Bao, P.; Maqueda, A.; del-Blanco, C.; Garcia, N. Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Trans. Consum. Electron.* **2017**, *63*, 251–257. Available online: https://ieeexplore.ieee.org/document/8103373/ (accessed on 15 October 2022). [CrossRef]
14. Masood, S.; Srivastava, A.; Thuwal, H.C.; Ahmad, M. Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN. *Adv. Intell. Syst. Comput. Intell. Eng. Inform.* **2018**, *695*, 623–632.
15. Swati, Z.N.K.; Zhao, Q.; Kabir, M.; Ali, F.; Ali, Z.; Ahmed, S.; Lu, J. Content-Based Brain Tumor Retrieval for MR Images Using Transfer Learning. *IEEE Access* **2019**, *7*, 17809–17822. [CrossRef]
16. Zhong, X.; Guo, S.; Shan, H.; Gao, L.; Xue, D.; Zhao, N. Feature-Based Transfer Learning Based on Distribution Similarity. *IEEE Access* **2018**, *6*, 35551–35557. [CrossRef]
17. Vanne, J.; Aho, E.; Hamalainen, T.; Kuusilinna, K. A High-Performance Sum of Absolute Difference Implementation for Motion Estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 876–883. [CrossRef]
18. Kim, H.-S.; Lee, J.-H.; Kim, C.-K.; Kim, B.-G. Zoom Motion Estimation Using Block-Based Fast Local Area Scaling. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1280–1291. [CrossRef]
19. Thang, N.V.; Choi, J.; Hong, J.-H.; Kim, J.-S.; Lee, H.-J. Hierarchical Motion Estimation for Small Objects in Frame-Rate Up-Conversion. *IEEE Access* **2018**, *6*, 60353–60360. [CrossRef]
20. LSA64: A Dataset for Argentinian Sign Language. LSA64—Argentinian Sign Language Dataset—UNLP—Index. Available online: http://facundoq.github.io/unlp/lsa64/ (accessed on 26 July 2019).
21. Wang, H.; Chai, X.; Hong, X.; Zhao, G.; Chen, X. Isolated sign language recognition with grassman covariance matrices. *ACM Trans. Access. Comput.* **2016**, *8*, 1–21. [CrossRef]
22. Huang, J.; Zhou, W.; Zhang, Q.; Li, H.; Li, W. Video-based sign language recognition without temporal segmentation. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 1–8. Available online: https://arvix.ord/abs/1801.10111 (accessed on 12 October 2022).