electronics

MDPI

Article

MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition

Gerges H. Samaan [1], Abanoub R. Wadie [1], Abanoub K. Attia [1], Abanoub M. Asaad [1], Andrew E. Kamel [1], Salwa O. Slim [1], Mohamed S. Abdallah [2,3,*] and Young-Im Cho [2,*]

[1] Department of Computer Science, Faculty of Computers and Artificial Intelligence, Helwan University, Helwan 11731, Egypt
[2] Department of Computer Engineering, Gachon University, Seongnam 1342, Korea
[3] Informatics Department, Electronics Research Institute (ERI), Cairo 11843, Egypt
* Correspondence: sameer@gachon.ac.kr (M.S.A.); yicho@gachon.ac.kr (Y.-I.C.)

Abstract: Communication for hearing-impaired communities is an exceedingly challenging task, which is why dynamic sign language was developed. Hand gestures and body movements are used to represent vocabulary in dynamic sign language. However, dynamic sign language faces some challenges, such as recognizing complicated hand gestures and low recognition accuracy, in addition to each vocabulary's reliance on a series of frames. This paper used MediaPipe in conjunction with RNN models to address dynamic sign language recognition issues. MediaPipe was used to determine the location, shape, and orientation by extracting keypoints of the hands, body, and face. RNN models such as GRU, LSTM, and Bi-directional LSTM address the issue of frame dependency in sign movement. Due to the lack of video-based datasets for sign language, the DSL10-Dataset was created. DSL10-Dataset contains ten vocabularies that were repeated 75 times by five signers providing the guiding steps for creating such one. Two experiments are carried out on our dataset (DSL10-Dataset) using RNN models to compare the accuracy of dynamic sign language recognition with and without the use of face keypoints. Experiments revealed that our model had an accuracy of more than 99%.

Keywords: dynamic sign language (DSL); MediaPipe; landmarks; GRU; LSTM; BILSTM

Citation: Samaan, G.H.; Wadie, A.R.; Attia, A.K.; Asaad, A.M.; Kamel, A.E.; Slim, S.O.; Abdallah, M.S.; Cho, Y.-I. MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. *Electronics* **2022**, *11*, 3228. https://doi.org/10.3390/electronics11193228

Academic Editor: George A. Papakostas

Received: 3 August 2022
Accepted: 1 October 2022
Published: 8 October 2022

1. Introduction

The hearing impaired (deaf) community employs a visual gesture language called sign language. In sign language, meaning and extracting information can be expressed by hand gestures, body movements, facial expressions, and emotions. Furthermore, sign languages reduce the communication gap between deaf and regular people, facilitating normal contact. Our aim is to assist these unlucky deaf people in assimilating into society. As a result, a system that makes it easier for regular and deaf people to communicate is necessary. An effective gesture recognition approach that can detect gestures in the sign language video stream is needed to implement this system.

Hand gestures can be classified into two categories: static and dynamic. A static gesture is a single image that represents a specific shape and pose of the hand. A dynamic gesture is a moving gesture, represented by a series of images. Our strategy is built around dynamic hand gesture recognition [1].

Dynamic sign language (DSL) is composed of a sequence of activities that include quick movements with high similarity. As a result, dynamic sign language recognition, which depends on a sequence of actions, faces challenges to deal with diversity, complexity, and a wide range of vocabularies in hand gestures [2]. Furthermore, dynamic sign language recognition faces challenges such as locating the hands, determining the shape plus the direction, and detecting the movement of signs [2,3]. Recognition is complex due to the spatial and temporal variation of gestures carried out by different people.

Dynamic sign language recognition is split into two parameters, the primary parameters, which must be solved and the secondary parameters that are preferred to be solved [3]. The primary parameters point to hand shape, hand and body location, and movement. The distance between the hand and body is a particularly key factor. The shape of the hand may be the same in two vocabularies; however, if the location of the hand in relation to the body changes, the meaning will change. Where DSL relies on a sequence of movements to construct one vocabulary, movement is the most complicated issue to solve.

The secondary parameters indicate palm orientation and facial expressions. The palm hand's orientation refers to whether the palm is pointing up or down, left, right, or in the direction of the body. Many signs have distinguishing characteristics, such as body expressions or facial emotions, which add meaning, feeling, and sentimentality to the statement. The above primary and secondary parameters work in tandem and complement one another. Our approach solves the primary and secondary parameters together, which can obtain an optimal result.

There are several approaches to solving the problems in DSL recognition. Most of the approaches can be categorized into two types [2]. First, an algorithm relies on the hand shape and hand gesture motion trajectory. Second, an approach that relies on each sign language's image sequences.

The method in this work is based on the use of the MediaPipe framework in conjunction with Recurrent neural network (RNN) models: Gated recurrent unit (GRU), Long Short Term Memory (LSTM), and Bi-directional LSTM (BILSTM) [4,5]. MediaPipe recognizes landmarks and extracts keypoints from objects like the hand, body, and face. This framework aids in the resolution of common DSL issues. On the one hand, MediaPipe can determine hand and body shape and location. It also addresses the issues of palm orientation and facial expressions, which are secondary parameters. In addition, the RNN models mentioned above can solve the problem of sign movement.

Because of the lack of video-based DSL datasets, the DSL10-Dataset was created in this paper, along with the guiding steps for creating such one.

The main contributions of this paper are that the problems of recognizing sign language of the dynamic type, i.e., movement dependent, have been solved. Where issues related to the basic principles of dynamic sign language that must be known to learn about sign language have been solved, such as solving the problem of movement, recognizing facial expressions, and the direction, location, and shape of the palm of the hand. We created a new video-based dataset with ten vocabularies and were able to create a sign language recognition model with it. We made comparisons between the diverse types of RNNs and noted how each one is used. Additionally, we made comparisons between including and excluding face points and stated the benefits and drawbacks of using them.

In summary, the following are the paper's main contributions:

- Build an end-to-end model using the MediaPipe framework combined with RNN to solve the issues of DSL recognition.
- Provide three RNN models GRU, LSTM, and BILSTM [4,5] for the recognition of the DSL.
- Created a new video-based dataset (DSL10-Dataset) consisting of ten vocabularies.

The rest of this paper is organized as follows. Firstly, Section 2 shows the related work. Next Section 3 for the methodology. Then, Section 4 provides information on DSL10-Dataset. In Section 5, the experimental results are discussed. Finally, a conclusion for the entire research and the future work is in Section 6.

2. Related Research

This section covers two aspects of sign language recognition, methods based on the dynamic motion of the hands and methods based on static hand shape.

2.1. Dynamic Sign Language Recognition

2.1.1. Motion Trajectory and Hand Shapes Methodologies

Motion trajectory and hand shapes are conventional methodologies for the challenges of DSL recognition. The approach examines the attributes and the characteristics of hand shapes and movement trajectories of hand gestures. Some related works are based mostly on evaluating the characteristics of hand forms. Kim et al. [6] used a deep neural network to solve the challenge of finger spelling recognition based on hand form characteristics. The method based on hand shapes may represent the simple meaning of hand gestures such as the alphanumeric characters. Nevertheless, it was still limited to complicated motion gestures due to the exclusive considering of the hand form with the absence of hand movements.

On the other hand, some related works were focused solely on evaluating the motion trajectory of hand gestures. The system developed by Mohandes et al. [7] used long short-term memory LSTM to distinguish hand gestures exclusively relied on hand motion trajectory.

The works [8–12] classified hand gestures using motion trajectory information obtained from sensors that included gyroscope, Kinect, accelerometer, electronic gloves, and depth camera. The above techniques are restricted to just a few simple hand gestures like waving and moving the hand up and down.

As a result, several relevant studies are built on analyzing hand shape aspects, hand movements, and hand gesture motion trajectory. Ding and Martinez [13], for example, proposed a method for receiving the 3D shape of every significant finger and then expressing the hand movement as a 3D trajectory. Dogra et al. [14] developed a model for multisensory hand gesture recognition. Finger and palm positions are captured from two different perspectives using Leap Motion and Kinect sensors. Such systems that rely on sensor devices have user comfort drawbacks.

In Persian sign language, Zadghorban and Nahvi [15] developed a method for detecting word boundaries. This method converts hand gestures into words by utilizing motion and hand shape features. Tomasi et al. introduced 3D tracking for hand finger motions using a real time mixture of 2D image classification and 3D motion interpolation [1].

DSL recognition based on hand shape features and motion trajectory has several obvious flaws. This strategy works well for some aspects of signs, such as alphanumeric characters, but it becomes more difficult when the recognized model contains many signs and vocabularies. In conclusion, using hand form features and motion trajectory with DSL has drawbacks.

2.1.2. Video Sequence Methodologies

DSL recognition based on the video sequence does not require hand sensors. Some related work that has concentrated on video-based sign language recognition identifies hand motions using deep learning algorithms [16,17].

Zhang et al. [18] created a recurrent convolutional neural network (RCNN) relying on video for dynamic hand sign recognition. Kishore et al. [19] presented recognizing Indian sign language (ISL) gestures using CNN. The selfie mode sign language video methodology was used in this study. Manikanta et al. [20] recognized sign language by fuzzy classifying dynamic gesture videos using a mixture of shapes and tracking features.

To achieve DSL recognition, many methods can be used such as extracting features of hand gestures with CNNs [21,22], approaches to learning video sequence with RNNs [23], and approaches to learning spatiotemporal sequence characteristics by integrating CNNs with RNNs [24,25].

When compared to the methodologies that relied on hand shapes and movement trajectory, such methodologies achieve a significant DSL recognition based on the video sequence.

2.2. Static Sign Language Recognition

This approach relies on image processing techniques to recognize the gestures. A hand posture is a static hand pose. The static image represents a particular shape and posture of the hand. Making a fist and holding it in a certain position, for example, is considered hand posture. The sign is recognized statically with only one hand represented. Feature descriptors techniques such as Histogram of Oriented Gradients (HOG) and Zernike Invariant Moments (ZIM) were used to extract features from images [26]. These features allow them to be represented based on specific attributes, like edges, contours, textures, colors, shapes, and so on.

This approach has some problems, such as image segmentation and the similarity between shapes. The problem of image segmentation is to build an object detection model to segment hands from the background.

Allam and Hemayed [27] demonstrated a system for recognizing the the Arabic sign language alphabet and transform it into speech. The system converts the image into the YCbCr color space in order to extract skin areas under various lighting conditions.

Althagafi et al. [28] proposed a new form that is completely based on the convolutional neural nets, which are fed with a large set of real data, which aims to identify the Arabic sign language, and 28 characters are recognized using the CNN model with RGB as input, and the results obtained were not bad. They trained the CNN model on 1080 images and obtained an accuracy of 92.9% in the training and testing phases.

3. Methodology

Most DSL methods cannot recognize gestures accurately due to the primary and secondary parameter issues. To solve this problem, our approach is split into two sections. The first is the features extraction, which extracts the keypoints by the MediaPipe framework. The second is the DSL recognition module, which can analyze sign movement and output the sign label.

3.1. Input Data

Experiments for dynamic sign language recognition are carried out using our own dataset (DSL10-Dataset), which was generated and analyzed during the current study. The DSL10-Dataset consists of 750 videos divided randomly for training and testing. All videos in DSL10-Dataset were recorded in indoor environments with regular lighting and an average mobile camera.

Each video was recorded at 30 frames per second (FPS) and has the same frame count and duration. However, in different datasets with an unequaled frame count, the total frame count for each clip should be equalized, as the models require a constant number of frames per sign language clip.

3.2. Features Extraction Using MediaPipe

Sign language depends on using hands and poses estimation; however, DSL faces many difficulties as a result of the continuous movement. Those difficulties are locating the hands, determining their shape, and direction. MediaPipe was used as a solution for these problems. It extracts the keypoints for the three dimensions X, Y, Z of both hands and poses estimation for each frame as shown in Figure 1.

The pose estimation technique was used to predict and track the hand location regarding the body. The output of the MediaPipe framework is a list of keypoints for hands and pose estimation.

For each hand, MediaPipe extracts 21 keypoints [29] as shown in Figure 2. The keypoints are calculated in the three-dimension space: X, Y, and Z for both hands. Thus, the number of extracted keypoints of hands is calculated as follows:

$$\text{keypoints in hand} \times \text{Three dimensions} \times \text{ No. of hands} = (21 \times 3 \times 2) = 126\ keypoints. \quad (1)$$

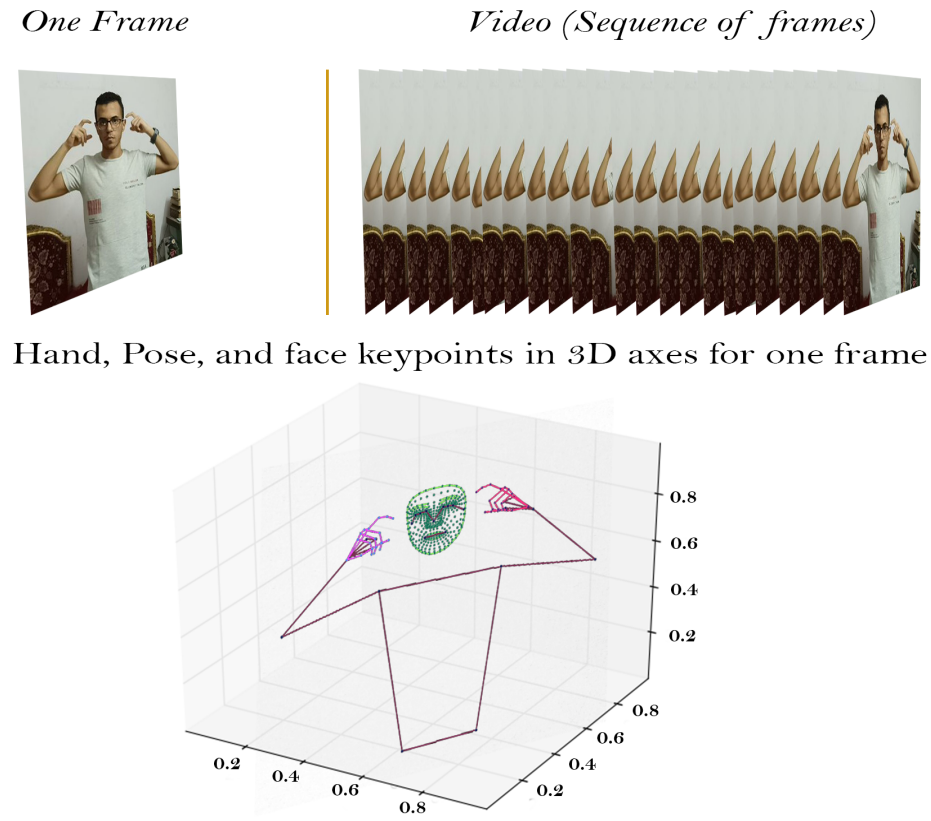

Figure 1. The sequence of frames and keypoints of hands and body in 3D space.

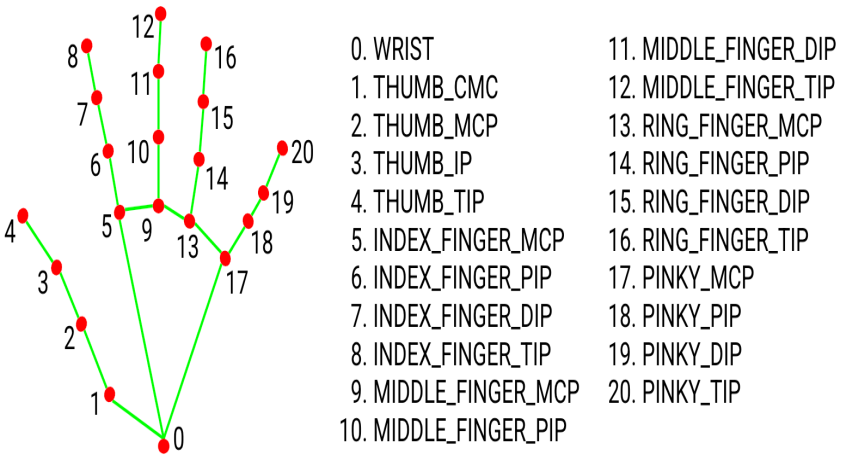

Figure 2. The order and labels for keypoints that exist in the hands of MediaPipe [30].

For Pose Estimation MediaPipe Extracts 33 keypoints [29] as shown in Figure 3. They are calculated in the three-dimension space: X, Y, and Z in addition to the visibility. The visibility is a value indicating if the point is visible or hidden (occluded by another body part) on a frame. Thus, the number of extracted keypoints from the pose estimation is calculated as follows:

$$\text{keypoints in pose} \times (\text{Three dimensions} + \text{Visibility}) = (33 \times (3+1)) = 132\ \textit{keypoints}. \quad (2)$$

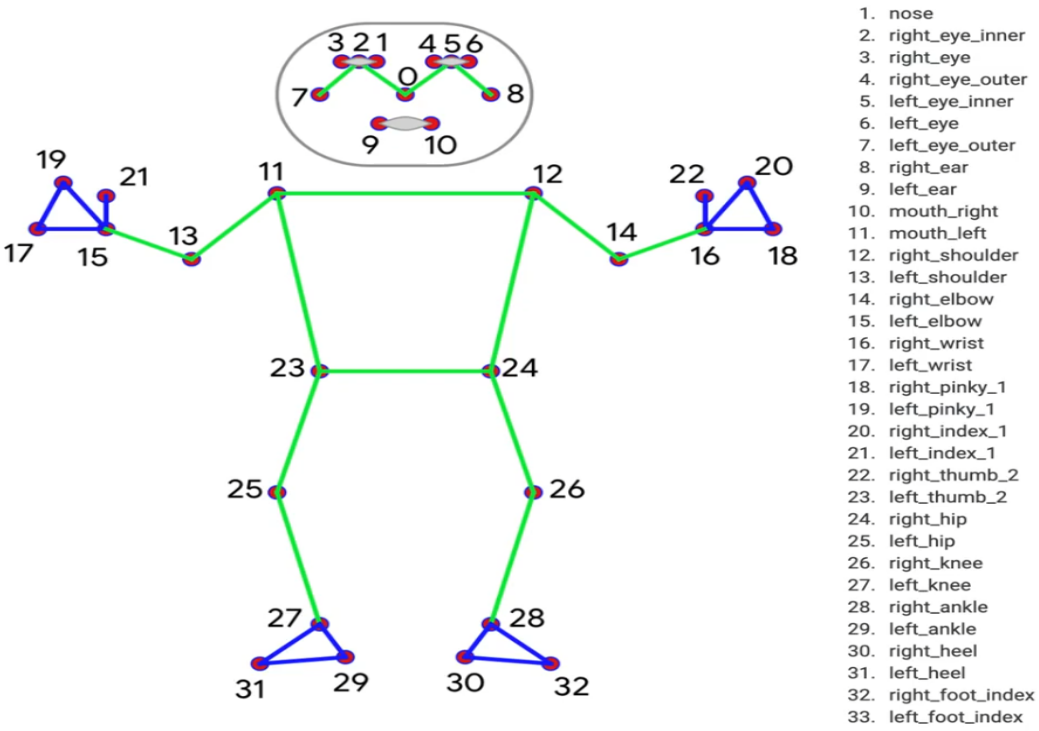

Figure 3. The order and labels for keypoints that exist in the pose [31].

For the face, MediaPipe extracts 468 keypoints [29], as shown in Figure 4. Contours around the face, eyes, lips, and brows are represented by lines connecting landmarks, while the 468 landmarks are represented by dots. They are calculated in the three-dimension space: X, Y, and Z. Thus, the number of extracted keypoints from the face is calculated as follows:

$$\text{keypoints in face} \times \text{Three dimensions} = (468 \times 3) = 1404\ keypoints. \quad (3)$$

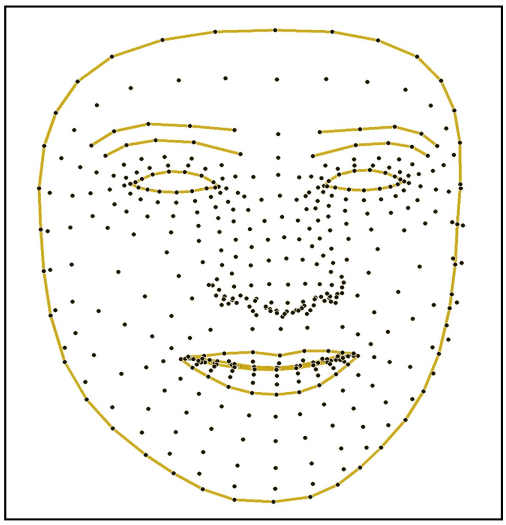

Figure 4. Face landmarks [32].

Without including face keypoints, the total number of keypoints for each frame is calculated as follows:

$$\text{keypoints in hands} + \text{keypoints in pose} = (126 + 132) = 258\ keypoints. \quad (4)$$

With including face keypoints, the total number of keypoints for each frame is calculated as follows:

$$\text{keypoints in hands + in pose + in face} = (126 + 132 + 1404) = 1662\ keypoints. \quad (5)$$

This operation is repeated in the whole video to extract the keypoints for each frame. The location of hand, body, and face are detected with a determination for their shape and direction in all videos of DSL10-Dataset.

Face Detection, Face Mesh, Hands, and Pose are some of the solutions provided by the MediaPipe framework for processing time-series data. However, there are some more drawbacks, which are the low recognition accuracy in most DSL recognition and potential problems of sign movement. Our work proposes solutions to these drawbacks using three models: Gated recurrent unit (GRU), Long Short Term Memory (LSTM), and Bi-directional LSTM (BILSTM).

3.3. The Models

Recurrent neural networks (RNNs) are a kind of artificial neural network (ANN) that utilizes time series and sequential data. RNNs are referred to as recurrent since they carry out the same function for each element in the sequence, with the dependency of previous states computations. The key feature of the RNN is that the network has feedback connections [33]. Another way to think about RNNs is that they contain a memory that stores information about previous states computations. Our work proposes a solution that includes three RNN-related models: GRU, LSTM, and Bi-LSTM.

The GRU is similar to an LSTM with a forget gate; however, it has lower complexity and less parameters. LSTMs were created to address the gradient vanishing issue that can occur while trying to train conventional RNNs. Bi-LSTM is an amalgamation of LSTM and Bi-directional RNNs [34].

Their output is based on the sequence of input, which improves the movement detection of the DSL. The next figures illustrate the summary and structure for each model in this work.

The structure of the models is shown in Figure 5–7. The first three layers belong to the RNN model while the last three layers are dense layers. The layers are then compiled by selecting the best value of the optimizer parameter [4] as shown in Table 1. On using each model, the value of the layer's parameters could be adjusted by selecting any value from Table 1 in order to prepare for the training phase.

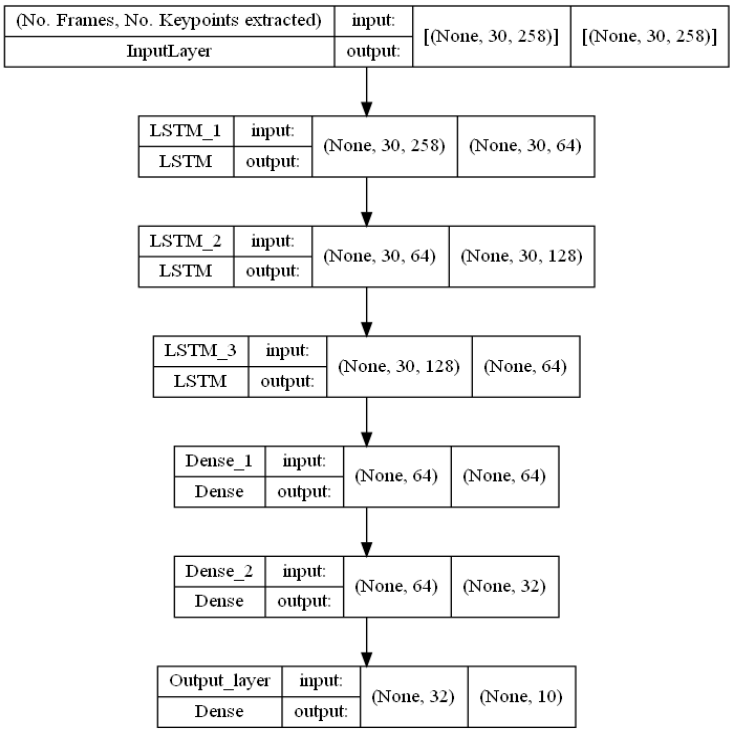

Figure 5. LSTM model structure.

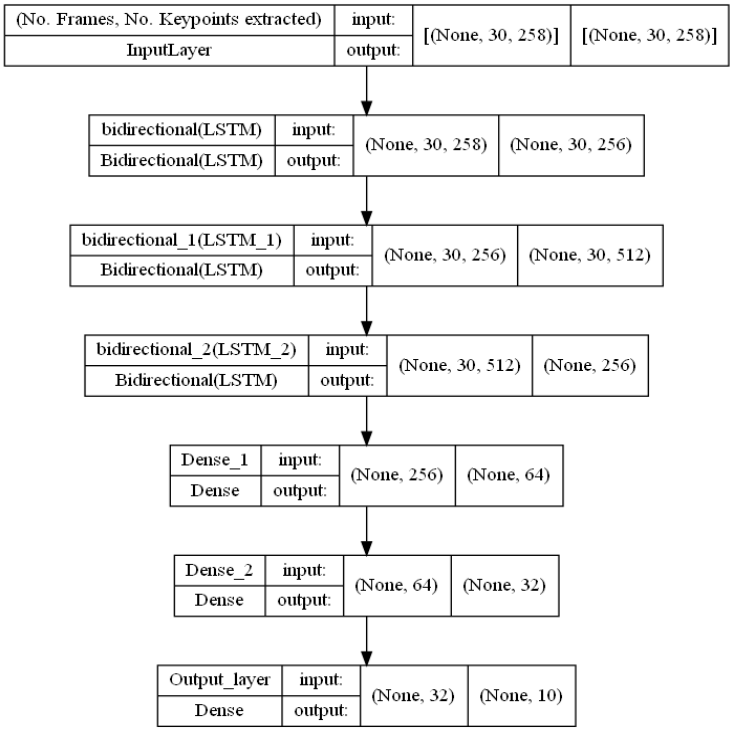

Figure 6. Bi-LSTM model structure.

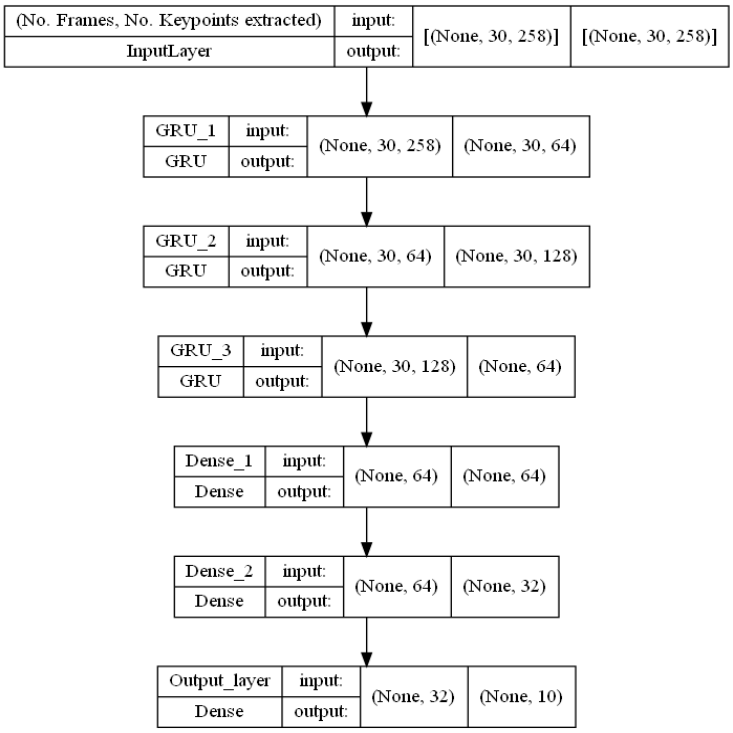

Figure 7. GRU model structure.

The inputs for the models are the sequence length and the total number of keypoints. Sequence length is the frame count contained in each clip. The total number of keypoints is 258 without including the face or 1662 with including the face.

Table 1. Model layer's parameters.

Parameters	Value
RNN Model	GRU, LSTM or BiLSTM
Number of Nodes	Between (64,256)
Activation	'Relu' or 'Softmax'
Optimizer	'Adagrad', 'Adamax', 'Adam' or 'RMSprop'

The model is now ready to receive the dataset and start the training phase based on the sequence of keypoints that have been extracted from videos. Thus, the sign movement is analyzed and the hand gesture label could be predicted. Therefore, the DSL could be recognized effectively.

4. Datasets

DSL10-Dataset is a dynamic sign language dataset created by the authors of this paper to adapt the methodology which does not require high-quality recorded data. The DSL10-Dataset used in this paper is available and can be accessed [35] (See data availability Section). DSL10-Dataset contains ten daily vocabularies for DSL, which are: Hello, Please, Sorry, Thanks, How are you?, Love, Mask, Wear, No, and Yes. The videos were captured by five signers, each of whom recorded 15 videos for each vocabulary, which led to a total number of videos = $10 \times 5 \times 15 = 750$. The data were recorded using OPPO Reno3 Pro mobile through a USB cable with a VideoCapture function in the OpenCV library. The duration for each video is one second with 640×480 resolution and 30 FPS. A simple model was created to record the videos with the chosen requirements.

The following instructions should be applied for creating the dataset.

- Signer body: the full signer's body must appear in all the frames of the video as shown in Figure 8A.
- Signer movement: the whole movement details must be clear and bounded between the camera frame as shown in Figure 8B.
- Background: it is better to record the dataset in a stable background that does not contain any other hands or faces except those of the signer.
- Lighting: it is preferred to record in good lighting conditions to make sure all the keypoints will be clear as shown in Figure 8C.
- Camera: set up your camera on a fixed stand to ensure that the videos are as unshakable and focused as possible as shown in Figure 8D.
- Video duration and frame count: the clip duration and number of frames should be determined before the recording process.
- Quality: any camera with a 640 × 480 resolution sensor can be used for the recording process since the most common sensors on the market are available in this size or higher.

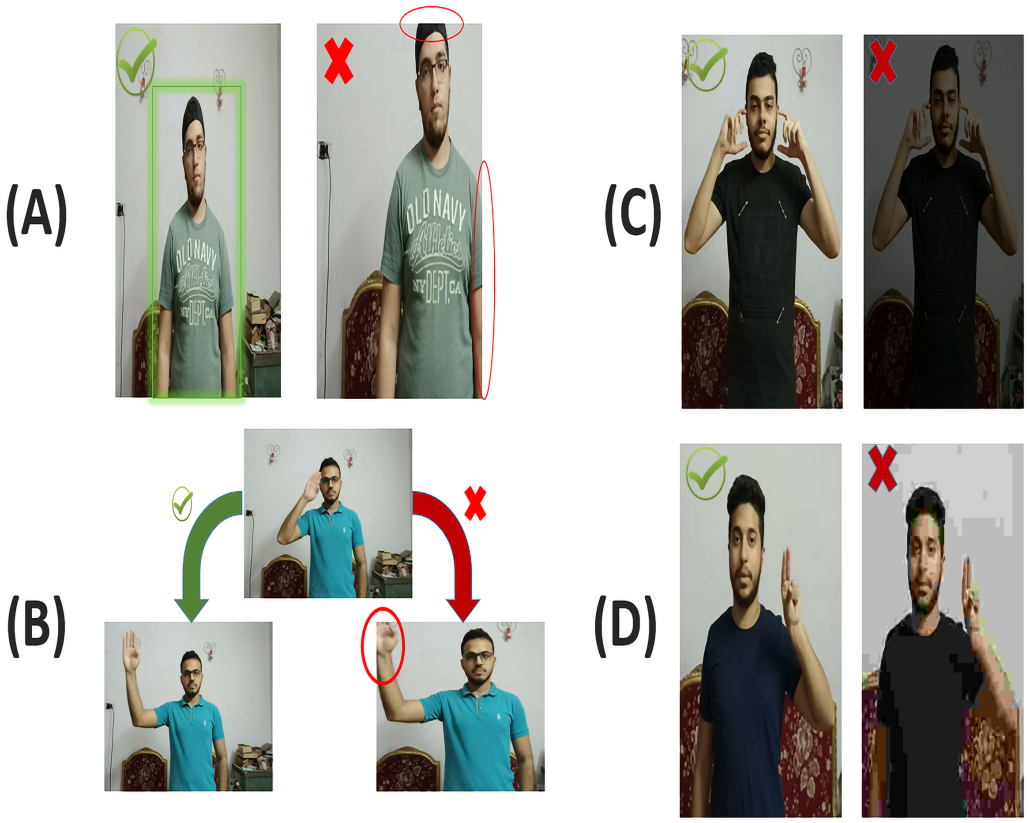

Figure 8. Instructions for recording a clean and clear dataset regarding (**A**) Signer body, (**B**) Signer movement, (**C**) Lighting and (**D**) Camera quality.

5. Experimental Results

Using the DSL10-Dataset, an experiment is carried out to assess the functionality of the proposed system. To conduct the experiments, the DSL10-Dataset was randomly divided into 60% for training and 40% for testing, yielding 450 clips for training and 300 clips for testing. So, every set is generated at random in order to exclude the random factors found in the experiments. The experiments were conducted on a PC with Intel Core i3-10100 of 3.6 GHz clock speed CPU, 16 GB RAM, and Crucial P5 Plus 500GB SSD.

5.1. MediaPipe without Including Face Keypoints

In this experiment, MediaPipe extracted 258 keypoints for the hands and the pose as shown in Figure 9, then inserted them into RNN models. Notice that the keypoints existing on the face are not face keypoints, but they belong to pose estimation as explained Figure 3.

Table 2 shows the train and test accuracy for each model with the taken number of epochs. With 258 keypoints, each model takes an estimated time between 20 to 45 min to train. This experiment achieves high accuracy in both the training and test.

As it appears in Table 2, the results of the GRU, the LSTM, and the BI-LSTM are very close, and the difference may not be noticeable, although if one is preferred, the GRU model will be because it is lighter and faster [34,35], although it took more epochs; however, it is lighter and faster in predicting and also it was observed during the training that the GRU in the one epoch takes much less than the one in the LSTM and the BI-LSTM.

Another experiment will be conducted to explain the difference between training with and without including face keypoints.

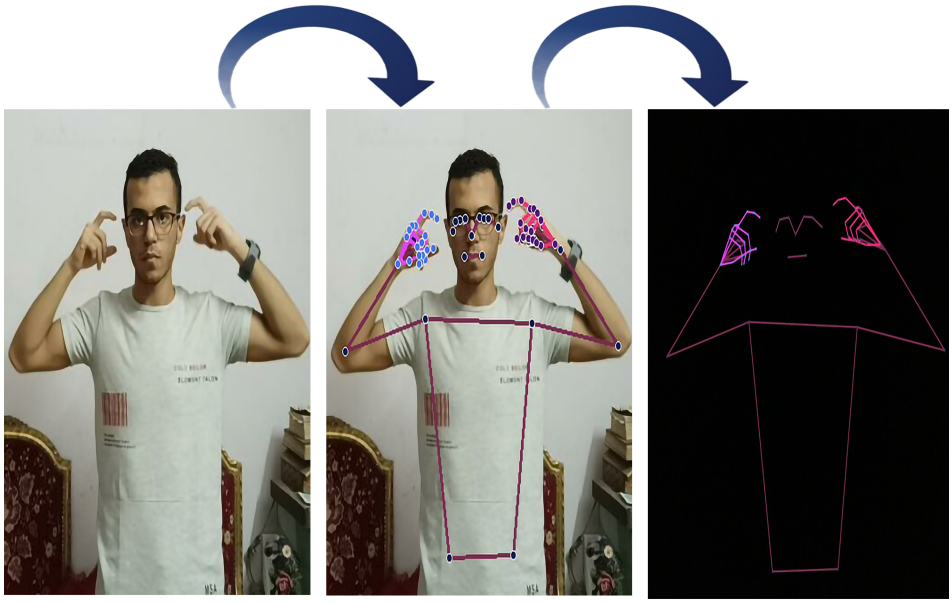

Figure 9. The stages of extracting the keypoints from the frame (Without include face keypoints).

Table 2. Models accuracy without including face keypoints.

	GRU	LSTM	BILSTM
Train accuracy	100%	99.9%	99.9%
Test accuracy	100%	99.6%	99.3%
Number of epochs	241	65	75

5.2. MediaPipe with Face Keypoints

In this experiment, we will include the face keypoints beside the hands and pose keypoints to see if including these face keypoints affects the model positively or negatively. The keypoints extracted from the hand and pose together without the face keypoints equal 258 and the face points alone equal 1404. Therefore, MediaPipe extracted 1662 keypoints for the hands, the pose, and the face as shown in Figure 10.

These 1662 keypoints are the ones that will be inserted into RNN models, which made us increase the size of the features six times the normal situation without including them, which will definitely affect the prediction time because the MediaPipe will take time to extract the 1404 points of the face, and the model will also take time to process these points of the face as well.

Table 3 shows the train and test accuracy for each model with the taken number of epochs. With 1662 keypoints, each model takes an estimated time between 30 to 75 min to train, and again excellent accuracy was reached in both train and test.

Table 3 also shows that the results are close and that there is almost no difference between the types of models. As previously stated, the preference will be for GRU because it is lighter and faster, especially since this experiment contains six times the normal number of keypoints (features), necessitating the lightest model.

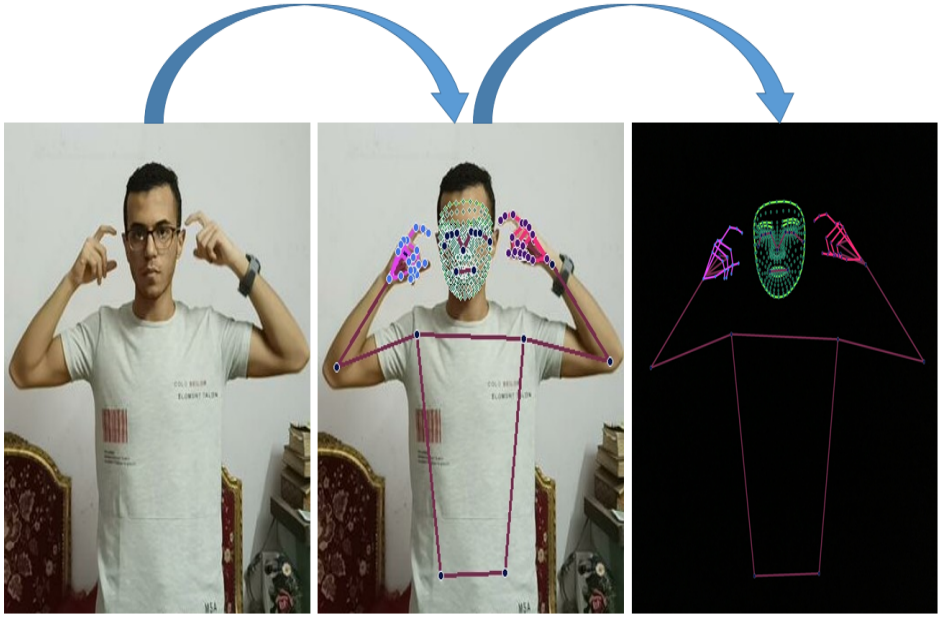

Figure 10. The stages of extracting the keypoints from the frame (including face keypoints).

Table 3. Models accuracy with face keypoints.

	GRU	LSTM	BILSTM
Train accuracy	100%	99%	99.9%
Test accuracy	100%	99.6%	99%
Number of epochs	250	36	49

After conducting the two experiments, it was found that the accuracy is very close at both of them. Hence, the preferred technique depends on the problem situation. For example, if the model will be used in a country that depends on facial expression in sign language, it is preferred to use face keypoints and vice versa. However, it is recommended not to use face keypoints while using the model in real-time because the total keypoints for the face technique is six times bigger than the other technique and it will take a long time to predict. In addition, using face keypoints will force hearing-impaired communities to make a facial expression for every vocabulary, which makes it harder to predict the correct vocabularies. However, it is useful in other situations as mentioned before.

Notice that GRU achieves the highest accuracy in all types of experiments. Although, GRU is not always the best choice. It depends on some factors, the most important one is that GRU easily surpasses LSTM and BILSTM models on lower parameters and less complexity sequences. On the other hand, LSTMs and BILSTMs outperform high-complexity sequences [34]. That is due to DSL10-Dataset being less complex whether in the number of frames per clip or the number of videos in each vocabulary. LSTM and BILSTM work best for larger and complex datasets [34,35]. Thus, the LSTM and BILSTM require a high number of parameters and nodes, while GRU requires the smallest number

of parameters and nodes [4,35]. The GRU model is the perfect choice for mobile phones due to its faster prediction speed [35].

6. Conclusions and Future Research

This paper presented Dynamic Sign Language Recognition with three RNN models, GRU, LSTM, BILSTM on DSL10-Dataset. MediaPipe framework was used for the features extraction phase. Two experiments were conducted to show the DSL recognition from two aspects, and the results show an outperformance in both of them.

The proposed method could be put into action in three steps. Prepare a dataset of videos with equal frame counts first. Then, pass the input data to the MediaPipe framework, which will extract the hand, face, and pose keypoints from each frame of the videos. Finally, for the training phase, insert the extracted keypoints into one of the prepared RNN models, GRU, LSTM, or BIlSTM.

On low complexity sequences, GRUs outperform LSTM networks, while on high complexity sequences, LSTMs outperform GRUs. LSTM and BILSTM require more parameters and nodes than GRU. The train and test accuracy for including face keypoints and without including them are very close. The keypoints with including the face are six times bigger than the keypoints without including the face, and this negatively affects the time of the prediction and learning.

In the future, dynamic sign sentences with a complex environment in real time video will be considered. Additionally, creating a larger dataset and developing an algorithm for preprocessing the datasets with different videos duration.

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

Funding: This research was supported by the MSIT, Korea, under the ITRC support program(IITP-2022-2017-0-01630) supervised by the IITP and by Korea Agency for Technology and Standards in 2022, project number is K_G012002236201.

Data Availability Statement: DSL10-Dataset used in this paper is available and can be accessed in the Google Drive repository https://bit.ly/3gpJhUj.

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

References

1. Abdalla, M.S.; Hemayed, E.E. Dynamic hand gesture recognition of arabic sign language using hand motion trajectory features. *Glob. J. Comput. Sci. Technol.* **2013**, *13*, 27–33.
2. Liao, Y.; Xiong, P.; Min, W. Weiqiong Min, and Jiahao Lu. Dynamic sign language recognition based on video sequence with blstm-3d residual networks. *IEEE Access* **2019**, *7*, 38044–38054. [CrossRef]
3. Escobedo, E.; Ramirez, L.; Camara, G. Dynamic sign language recognition based on convolutional neural networks and texture maps. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 28–30 October 2019; pp. 265–272.
4. Chaikaew, A.; Somkuan, K.; Yuyen, T. Thai sign language recognition: An application of deep neural network. In Proceedings of the 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, Cha-am, Thailand, 3–6 March 2021; pp. 128–131.
5. Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet Things J.* **2019**, *6*, 10639–10651.[CrossRef]
6. Kim, T.; Keane, J.; Wang, W.; Tang, H.; Riggle, J.; Shakhnarovich, G.; Brentari, D.; Livescu, K. Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation. *Comput. Speech Lang.* **2017**, *46*, 209–232. [CrossRef]

7. Mohandes, M.; Deriche, M.; Liu, J. Image-based and sensor-based approaches to Arabic sign language recognition. *IEEE Trans. Hum.-Mach. Syst.* **2014**, *44*, 551–557. [CrossRef]
8. Sonawane, T.; Lavhate, R.; Pandav, P.; Rathod, D. Sign language recognition using leap motion controller. *Int. J. Adv. Res. Innov. Ideas Edu.* **2017**, *3*, 1878–1883.
9. Li, K.; Zhou, Z.; Lee, C.H. Sign transition modeling and a scalable solution to continuous sign language recognition for real-world applications. *ACM Trans. Access. Comput. (TACCESS)* **2016**, *8*, 1–23. [CrossRef]
10. Yang, X.; Chen, X.; Cao, X.; Wei, S.; Zhang, X. Chinese sign language recognition based on an optimized tree-structure framework. *IEEE J. Biomed. Health Inform.* **2016**, *21*, 994–1004. [CrossRef] [PubMed]
11. Liu, T.; Zhou, W.; Li, H. Sign language recognition with long short-term memory. In Proceedings of the 2016 IEEE international conference on image processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2871–2875.
12. Ma, Z.; Lai, Y.; Kleijn, W.B.; Song, Y.Z.; Wang, L.; Guo, J. Variational Bayesian learning for Dirichlet process mixture of inverted Dirichlet distributions in non-Gaussian image feature modeling. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 449–463. [CrossRef] [PubMed]
13. Ding, L.; Martinez, A. Three-Dimensional Shape and Motion Reconstruction for the Analysis of American Sign Language. In Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), New York, NY, USA, 17–22 June 2006; pp. 146–146. [CrossRef]
14. Kumar, P.; Gauba, H.; Roy, P.P.; Dogra, D.P. A multimodal framework for sensor based sign language recognition. *Neurocomputing* **2017**, *259*, 21–38. [CrossRef]
15. Zadghorban, M.; Nahvi, M. An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands. *Pattern Anal. Appl.* **2018**, *21*, 323–335. [CrossRef]
16. Moussa, M.M.; Shoitan, R.; Abdallah, M.S. Efficient common objects localization based on deep hybrid Siamese network. *J. Intell. Fuzzy Syst.* **2021**, *41*, 3499–3508. [CrossRef]
17. Abdallah, M.S.; Kim, H.; Ragab, M.E.; Hemayed, E.E. Zero-shot deep learning for media mining: Person spotting and face clustering in video big data. *Electronics* **2019**, *8*, 1394. [CrossRef]
18. Cui, R.; Liu, H.; Zhang, C. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7361–7369.
19. Rao, G.A.; Syamala, K.; Kishore, P.; Sastry, A. Deep convolutional neural networks for sign language recognition. In Proceedings of the 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), Vijayawada, India, 4–5 January 2018; pp. 194–197.
20. Kishore, P.; Kumar, D.A.; Goutham, E.; Manikanta, M. Continuous sign language recognition from tracking and shape features using fuzzy inference engine. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 23–25 March 2016; pp. 2165–2170.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60* , 84–90. [CrossRef]
22. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
23. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using lstms. In Proceedings of the International Conference on Machine Learning. PMLR, Lille, France, 7–9 July 2015; pp. 843–852.
24. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. Sequential deep learning for human action recognition. In *Proceedings of the International Workshop on Human Behavior Understanding*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 29–39.
25. Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
26. Bastos, I.L.; Angelo, M.F.; Loula, A.C. Recognition of static gestures applied to brazilian sign language (libras). In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Bahia, Brazil, 26–29 August 2015; pp. 305–312.
27. Hemayed, E.E.; Hassanien, A.S. Edge-based recognizer for Arabic sign language alphabet (ArS2V-Arabic sign to voice). In Proceedings of the 2010 International Computer Engineering Conference (ICENCO), Cairo, Egypt, 27–28 December 2010; pp. 121–127. [CrossRef]
28. Althagafi, A.; Alsubait, G.T.; Alqurash, T. ASLR: Arabic sign language recognition using convolutional neural networks. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2020**, *20*, 124–129.
29. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.G.; Lee, J.; et al. Mediapipe: A framework for building perception pipelines. *arXiv* **2019**, arXiv:1906.08172.
30. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.L.; Grundmann, M. Mediapipe hands: On-device real-time hand tracking. *arXiv* **2020**, arXiv:2006.10214.
31. Bazarevsky, V.; Grishchenko, I.; Raveendran, K.; Zhu, T.; Zhang, F.; Grundmann, M. Blazepose: On-device real-time body pose tracking. *arXiv* **2020**, arXiv:2006.10204.

32. Kartynnik, Y.; Ablavatski, A.; Grishchenko, I.; Grundmann, M. Real-time facial surface geometry from monocular video on mobile GPUs. *arXiv* **2019**, arXiv:1907.06724.
33. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292.
34. Cahuantzi, R.; Chen, X.; Güttel, S. A comparison of LSTM and GRU networks for learning symbolic sequences. *arXiv* **2021**, arXiv:2107.02248.
35. Khandelwal, S.; Lecouteux, B.; Besacier, L. Comparing GRU and LSTM for Automatic Speech Recognition. Ph.D. Thesis, LIG, Saint-Martin-d'Hères, France, 2016.