computers

Article

Multistage Spatial Attention-Based Neural Network for Hand Gesture Recognition

Abu Saleh Musa Miah , Md. Al Mehedi Hasan, Jungpil Shin *, Yuichi Okuyama and Yoichi Tomioka

School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Fukushima, Japan
* Correspondence: jpshin@u-aizu.ac.jp

Abstract: The definition of human-computer interaction (HCI) has changed in the current year because people are interested in their various ergonomic devices ways. Many researchers have been working to develop a hand gesture recognition system with a kinetic sensor-based dataset, but their performance accuracy is not satisfactory. In our work, we proposed a multistage spatial attention-based neural network for hand gesture recognition to overcome the challenges. We included three stages in the proposed model where each stage is inherited the CNN; where we first apply a feature extractor and a spatial attention module by using self-attention from the original dataset and then multiply the feature vector with the attention map to highlight effective features of the dataset. Then, we explored features concatenated with the original dataset for obtaining modality feature embedding. In the same way, we generated a feature vector and attention map in the second stage with the feature extraction architecture and self-attention technique. After multiplying the attention map and features, we produced the final feature, which feeds into the third stage, a classification module to predict the label of the correspondent hand gesture. Our model achieved 99.67%, 99.75%, and 99.46% accuracy for the senz3D, Kinematic, and NTU datasets.

Keywords: kinematic sensor; multistage attention neural network; CNN; attention model; sign language recognition; gesture recognition

check for updates

Citation: Miah, A.S.M.; Hasan, M.A.M.; Shin, J.; Okuyama, Y.; Tomioka, Y. Multistage Spatial Attention-Based Neural Network for Hand Gesture Recognition. *Computers* **2023**, *12*, 13. https://doi.org/10.3390/computers12010013

Academic Editor: Robertas Damaševičius

Received: 28 November 2022
Revised: 26 December 2022
Accepted: 29 December 2022
Published: 5 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hand gesture recognition has become a crucial part of the HCI and computer vision research domain because of its extensive application. Every day, we consciously or subconsciously use numerous hand gestures to execute different tasks, but still, there are many difficulties in collecting accurate hand gesture information and recognition. With the wearable or vision-based device, problems can be solved by extracting a variety of hand gesture information. Usually, different movement of the hand and fingers to express specific information is known as hand gestures. The information depends on the hand's orientation, which may be specific symbols, digits, or objects. However, some have some specific meaning for some hand gestures, and some have a universal meaning related to the culture or context of the corresponding countries. A hand gesture can be static or dynamic: generally, static gestures are defined as symbolic gestures that only exist in the spatial domain. Different kinds of digits are examples of static gestures such as one, which can be expressed by the index figure or two index and middle fingers. Static hand gestures (SHGs) usually express the singular quality of something that does not have a different meaning but only one meaning and includes no temporal information. On the other hand, dynamic hand gestures (DHGs) include spatial and temporal information, which can carry broader meaning. If we move the left hand from left to right to express the refuge of something, our gesture contains spatial and temporal information. Therefore, DHGs can be defined as the collection of static hand gesture sequences that can express a single meaning. Sign language recognition, virtual reality, human–computer interaction, robotics, computer gaming, physical science, natural science, computer engineering, and industrial areas are the most usable hand gesture recognition applications [1–13]. Although the information embedding capacity of the hand gesture is lower than the speech-controlled, natural language, it is good for considering language barriers, those suffering from vocal fatigue, and the deaf community. Researchers have been working on the capacity of hand gestures. If it is possible to solve the intriguing problem of hand gesture recognition, it could open the way for more real-life applications in different fields [14]. Another advantage of hand gestures is that they exclude physical contact with a hardware device, where the normal input device must establish a permanent connection mechanism. Thus, using hand gestures helps reduce the use of physical media, which can reduce extra burden, extra cost, and complexity, and this is one of the main advantages of hand gestures [15,16].

There has been a broad range of approaches for collecting hand gesture datasets. Among them, sensor- and vision-based systems are the most ubiquitous and prevalent. The sensor-based method includes different gloves [17] and sensors embedded with a bend, proximity, accelerometer, gyroscopes, and inertial sensors. The vision-based method includes RGB images, RGB video, depth images, skeleton information, and infrared-based depth images. Because of the many limitations of the sensor-based approach, such as cost, user comfort, data preprocessing, portability, and naturalness, the vision-based approach is the most appropriate for the researchers [18–21]. However, using a kinetic sensor is one of the effective ways to collect hand gesture datasets that can efficiently detect and segment hand gesture bursts and a series of events instead of gloves [22]. Although kinetic sensors have many problems for large areas, such as capturing the entire human body because they typically only have 640×480 pixels, they can easily cover hand gestures, as they involve a small area. Many research studies have used kinetic sensor information for recognizing hand gestures. Furthermore, they generate RGB and depth map images; some researchers use only RGB of kinetic sensors, while others use RGB and depth maps as multimodal information [1,14,23–26]. Many researchers employed handcrafted feature extraction techniques with machine learning algorithms to classify hand gestures, but recently, most of the research employed deep learning methods. Initially, convolutional neural networks (CNNs) are used to recognize hand gestures, but it is difficult to recognize dynamic hand gestures containing spatial-temporal information. Some researchers employed recurrent neural networks (RNNs), which are mostly similar to CNNs, although CNNs have been more successful [27]. Recently, researchers have used long short-term memory (LSTM) to extract long-term dependency. A combination of CNN and LSTM was used to achieve high-performance accuracy for hand gesture recognition [28]. Long-term dependency needs high computation complexity, which is the main problem of LSTM; by contrast, attention-based neural networks produce short-term dependency, which needs less computational complexity [29–31].

Recently, some researchers have used attention-based approaches to improve performance accuracy and reduce the computational complexity of hand gesture recognition. Another main advantage of the attention-based method is that it usually highlights the potential feature during discards the irrelevant feature with the self-supervised learning approach. However, very few attention-based methods are used to recognize kinetic RGB-based hand gesture recognition, and their performance is unsatisfactory. To overcome these challenges, we proposed a multistage spatial attention-based neural network for hand gesture recognition. Our model includes three stages: The first stage includes spatial features extracted with a self-attention mechanism, which are then multiplied and concatenated with the extracted features and the original dataset. The same process is repeated in the second stage to highlight the effective features, and finally, a classification module was applied (Figure 1) to show the overview of the work. The main contribution of the study is given below:

- We proposed a multistage, attention-based feature-fusion-based deep learning for recognizing hand gestures. We implemented the model in three phases: The first two phases are used for feature extraction, and the third phase is used for classification. The first two phases consist of a combination of feature extractors and spatial attention modules.

- The first phase is used to extract spatial features, and the second phase is used to highlight effective features.
- Finally, we applied a new classification module adapted from CNNs, which demonstrated state-of-the-art classification performance on three hand gesture datasets used as benchmarks, and it outperformed the existing modules.

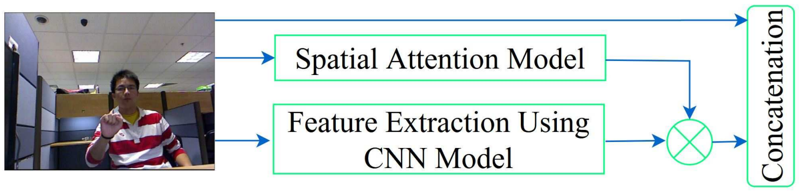

Figure 1. Working procedure of the proposed method.

The presented work is organized as follows: Section 2 summarizes the existing research work and problems related to the presented work, Section 3 describes the three benchmark hand gesture datasets, and Section 4 describes the architecture of the proposed system. Section 5 shows the results obtained from the experiment with a different dataset, followed by a discussion. In Section 6, conclusions are drawn, including our plans for future work.

2. Related Work

There are several studies on hand gesture recognition. To implement the hand gesture recognition system, researchers have used different schemes, the most common of which are handcrafted signals, and machine learning and deep learning algorithms have also been used. Machine-learning-based algorithms can be classified into statistical-based and rule-based approaches. Among the statistical-based methods, Iwai et al. employed a glove-based machine learning system to classify hand gestures [32]. Firstly, they extracted features with the nearest-neighbour approach and then employed a decision tree algorithm for classification. Wilson et al. applied a statistical-based hidden Markov model to recognize hand gestures, where they include a global parametric variation of the output probabilities [33]. In the same way, Kyu et al. extracted input patterns with the likelihood threshold method and generated a confirmation approach to match the gesture pattern with the statistical HMM model [34]. In addition, various particle filtering and condensation algorithms have been used to recognize hand gestures [35,36]. In rule-based methods, fuzzy rules are used to extract features, which are then compared with the encoding rules, and the best match scores are produced to recognize hand gestures [29]. Marin et al. applied a feature extraction technique based on the position and orientation of the fingertips, employed the SVM algorithm and achieved 89.70% accuracy with SVM [14]. In another study, Marin et al. employed distance handcrafted feature extraction based on the centroid, the curvature of hand contour, and hand shape convex [24]. After that, they applied different feature selections to the potential spatial features and the SVM and a random forest algorithm as a classifier, where the SVM produced 96.30% accuracy for the kinetic sensor RGB images. Recently, researchers have applied deep learning algorithms to recognize hand gestures; among them, Nagi et al. applied deep learning algorithms to recognize hand gestures [37]. They mainly focused on the CNN's specific layer, namely the max-pooling layer, and achieved good performance with the RGB hand gesture dataset [37]. The main problem of these studies is the confusion in the detection of hand gestures that involve variations in orientation and partial occlusions. To overcome this challenge, Tao et al. applied a CNN model with multiview augmentation to recognize kinetic-sensor-based hand gestures for the American sign language (ASL) [38]. Furthermore, these methods achieved good performance, but one of their limitations is that they usually consider only spatial information. Because of the temporal information in DHH, recently, researchers have used two-stream networks, 3DCNN and RNN, to overcome this challenge [27]. Based on the recent work, researchers decided that long short-term memory (LSTM) is one RNN network that can learn long-term dependency [27,28]. The combined system of CNN and LSTM network had a high level of performance for leap motion and kinetic devices compared to the other method [39]. The main drawback of LSTM is the long-term dependency, which can increase the computational complexity of the system. To solve the long-term dependency, attention-based neural networks use a different layer of CNN in parallel and produce short-term dependency. As a consequence, attention-based architecture reduces the computational complexity of the system [29–31]. In this study, we proposed a multistage, attention-based neural network architecture to recognize hand gestures to increase performance and reduce computational complexity.

3. Dataset

To evaluate our model, we used three datasets. Each dataset contains multimodal information; however, we only used the RGB modality in this study. These are the Senz3D, NTU, and Kinetice datasets, which are described in Sections 3.1–3.3.

3.1. Creative Senz3d Dataset

The creative senz3D dataset is one of the challenging datasets for hand gesture recognition, comprising RGB images and depth maps, which LibHand uses, and we collected this from the following link https://lttm.dei.unipd.it//downloads/gesture/index.html (accessed on 8 June 2022). A creative Senz3D camera is used to generate this dataset, also known as SoftKinetic DepthSense 325 [26,40,41]. There are 11 different gestures in the dataset, which 4 different people perform. Each person repeats individual gestures 30 times, and a total of 1320 samples are thus acquired. In total, 3200 samples are collected for each of the individual gestures combined with RGB and depth maps. These samples contain different orientations and variations in the position based on the fingers. A sample RGB image for each gesture is visualized in Figure 2, which clearly shows the various gestures included in this dataset, such as gestures with the closest fingers touching each other and the same number of raised fingers. The RGB images have a 640 × 480 resolution, and we used only RGB images in this study.

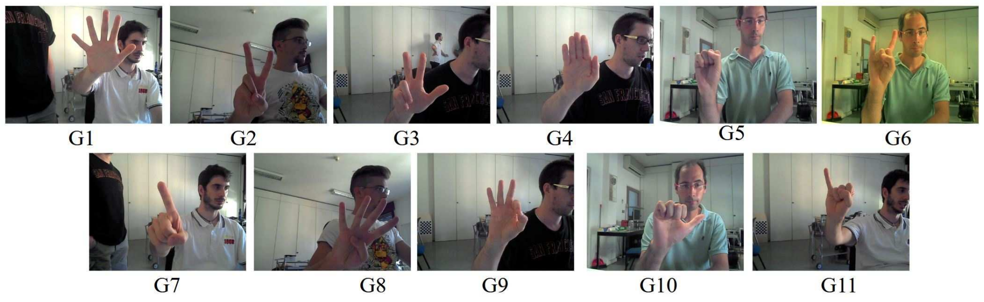

Figure 2. RGB sample frame for each gesture for the Senz3D dataset.

3.2. NTU Dataset

Another challenging gesture dataset is the NTU hand digit dataset, which is also collected with a kinetic sensor consisting of RGB images and depth maps [1,25]. There are 10 gestures (decimal digits 0 to 9) included in the dataset, which are collected from 10 individuals. There are 1000 images in the dataset, and each image's size has a 640 × 480 resolution. This dataset is considered the most challenging for achieving high performance because it includes images with complex backgrounds such as cluttered backgrounds, multiple variations in pose, and multiple articulations and orientations. Figure 3 demonstrates the sample of each gesture.

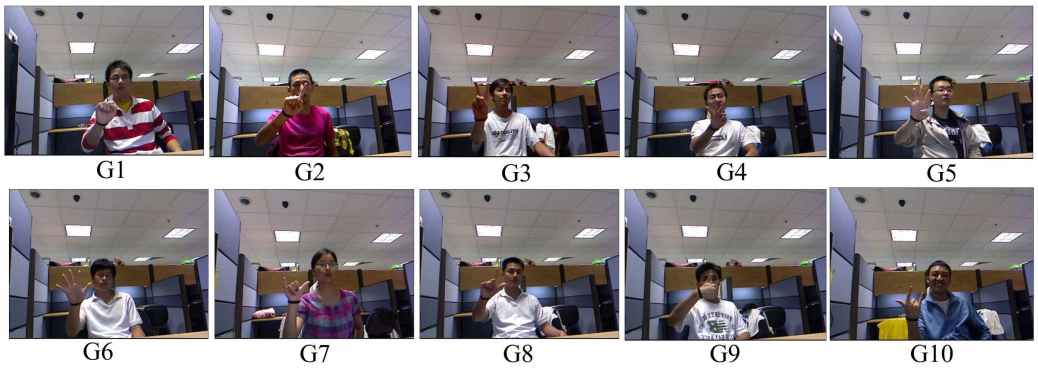

Figure 3. Sample image of the 10 gestures in the NTU dataset.

3.3. Kinetic and Leap Motion Gestures

This is another challenging hand gesture dataset for American sign language (ASL) collected using a kinetic sensor and leap motion [14,24]. There are 10 gestures contained in the dataset collected from 14 people. Each gesture is collected 10 times from each individual, and a total of 1400 different data samples are contained in the dataset. Figure 4 shows the sample of each gesture. This dataset collects RGB and depth maps for each gesture during the recording time, but we used only RGB images in this study. Without any assumption, a kinetic sensor is used to collect the dataset. We downloaded the dataset from the following link: It is available at https://lttm.dei.unipd.it/downloads/gesture (accessed on 8 June 2022).

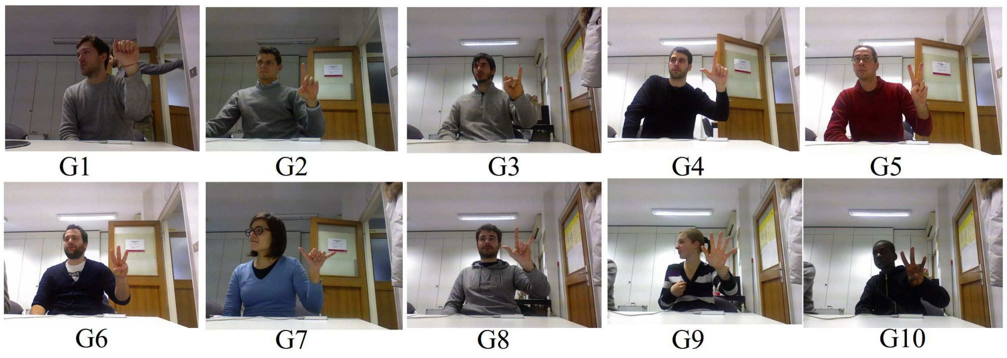

Figure 4. Gestures from American Sign Language (ASL) are contained in the database acquired for experimental results.

4. Proposed Method

Figure 5 shows the proposed method workflow architecture, mainly a modified version of FusAtNet [19,23,42]. The main objective of this study is pixel-based classification by extracting multistage spatial information. The RGB image is written as $X = \{X_R^i\}_{i=1}^n$, where $x_R^i \in R^{M \times N \times C}$ and m = 160, n = 160 expresses the image width and height, and C = 3 is the number of channels of the RGB image. Furthermore, attention modules solved the problems associated with long dependency. In addition, attention modules were used to select and highlight the hotspots of the extracted features to increase the inter-class variance, and as a consequence, the performance and accuracy of the system were automatically improved. Our proposed method was developed in three stages: In the first stage, RGB data passed through the attention module and the feature extractor module, and then we multiplied the output to emphasize the effective information by masking operation. In the second stage, the output of the first stage passed through the second feature extractor and second spatial attention module. We combined the output of these two modules to highlight the important section judiciously and produced the final feature for the classification. We passed the resultant highlighted feature through the classification module. There was a total of six CNN modules employed in our system in the three phases.

In the first phase, an RGB image was used as input of spatial attention A_{ft} and feature extractor F_f, and then we multiplied the output, which produced M_t. After that, we concatenated M_t with the original image X_R^i, which produced M_c. The second phase took M_c as input, which was then passed through the second feature extractor F_s and the second attention module A_{st}, and again these two were multiplied to highlight or select the effective feature with the spatial attention module, which is known as the final feature F_{ss}. In all modules, CNN layers were used, where the size of the kernel was fixed as 3×3, ReLU activation function and softmax activation function were only used in the last layer of the classification module.

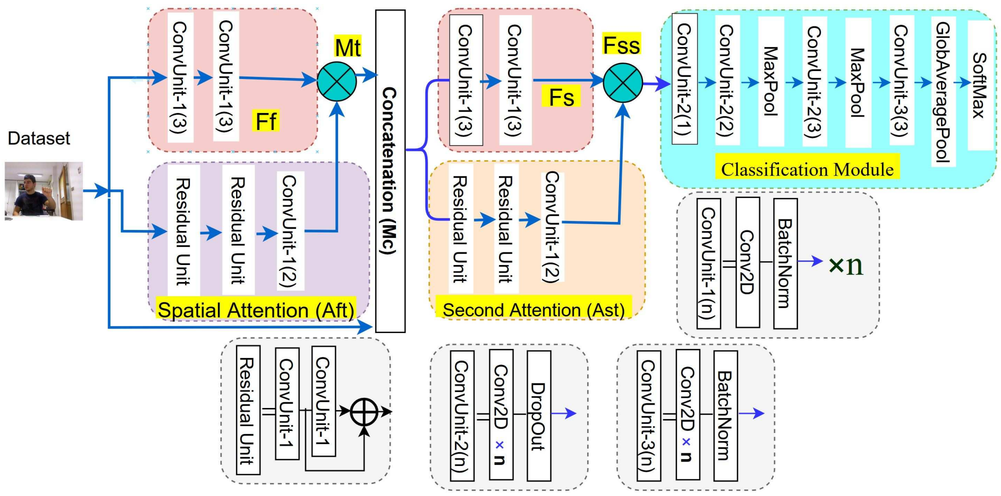

Figure 5. Working flow diagram of the proposed method.

4.1. First-Stage Feature Extractor

The first feature extractor module consisted of CNN's six convolution layers, where spatial features were generated from the RGB images. Each of the convolution operations was employed here with the zero-padding operation. Batch normalization layers were applied after each convolution operation to normalize the value of every mini-batch value. We can represent the feature extractor module as $F_F\left(\theta_{F_f}, x_R^i\right)$ where the weight of the module is represented by θ_{F_f} [23]. The feature extractor output was $160 \times 160 \times 1024$.

4.2. First-Stage Spatial Attention Module

In this study, we employed an attention model to produce the spatial attention model, which can be defined by $A_{ft}\left(\theta_{A_{ft}}, X_R^i\right)$, where the weights of the model are denoted by $\theta_{A_{ft}}$ [23]. This module included six CNN convolution layers used to produce the spatial attention mask from the original dataset. In addition, 128 filters were included in the first three layers and 256 in the remaining three layers. After the second and fourth convolution layers, we used two residual layers to pass the original information. To normalize the output of the convolution layer in each batch, we used batch normalization after each convolution layer. The shape of the output of this module was $160 \times 160 \times 1024$, and it was multiplied by F_f to extract spatially highlighted features denoted by M_T, which can be expressed as the following Equation (1):

$$M_t\left(X_R^i\right) = F_F\left(\theta_{F_f}, x_R^i\right) \bigotimes A_{ft}\left(\theta_{A_{ft}}, X_R^i\right) \tag{1}$$

4.3. Second-Stage Feature Extractor

This block is almost the same as the feature extractor module F_s, which can be written as $F_s\left(X_R^i\right) = F_s\theta_{F_s}, x_R^i)$ and the weight of the architecture is represented by θ_{F_s} [23]. This feature vector was fed with the spatially highlighted M_t features associated with the original dataset X, and its output size was 160 × 160 × 1024, which can be written using the following Equation (2):

$$F_s\left(X_R^i\right) = F_s\left(\theta_{F_s}, x_R^i\right) \bigotimes M_t\left(X_R^i\right) \tag{2}$$

Here, concatenation along the channel axis is represented by the $\bigotimes$ notation.

4.4. Second-Stage Spatial Attention Module

A_{st} is another architecture, which can be written as $A_{st}\left(\theta_{A_{st}}, X_R^i\right)$, where the weight of the architecture is denoted by $\theta_{A_{st}}$ [23]. The main purpose of this block was to create an attention mask for the RGB modality and input to keep it the same as F_s. The working function of the block can be expressed by the following Equation (3):

$$A_{st}\left(X_R^i\right) = A_{st}\left(\theta_{A_{st}}, X_R^i \bigotimes M_t\left(X_R^i\right)\right) \tag{3}$$

The output of the block was 160 × 160 × 1024, which was multiplied with F_s using the following Equation (4):

$$F_{ss}\left(X_R^i\right) = F_s\left(X_R^i\right) \bigotimes A_{st}\left(X_R^i\right) \tag{4}$$

Here, F_{ss} was considered the final spatial feature of the proposed method, which was sent to the classification block.

4.5. Classification Architecture

In the final architecture, we used a classification block that took the final feature F_{ss} as input [19]. The architecture had nine convolutional neural network layers, where the first three layers consisted of 96 filters each, and the next five layers consisted of 192 layers. The ninth layer contained C filters, where C is the total number of classes, and the size of the layer was 1 × 1, and excluding the last layer, all the layers were operated through the ReLu activation. We used the same padding for the first eight layers and valid padding for the ninth layer. There were four drop-out layers, two max-pooling layers, and one batch normalization layer. Finally, instead of the flattened layer, we applied global average pooling associated with the softmax activation function used as a final layer. The classification architecture can be written as $F\left(\theta_f, F_{ss}\left(x_R^i\right)\right)$, where θ_f is the classification weight. The shape of the output of this block was None × C.

4.6. Inference and Training

The final output from the F was associated with the categorical cross-entropy loss used as backpropagation for training the proposed network in an end-to-end fashion shown in Equation (5).

$$L_F = -E_{\left(X_R^i, y^i\right)}\left[y^i log\left[F\left(\theta_f, F_{ss}\left(X_R^i\right)\right)\right]\right] \tag{5}$$

Here, classification loss is denoted by L_F. In the testing phase, the test set of the dataset was x_R^j, which was fed into the model and executed the same instruction as the training samples [23]. The output of the final feature module layer F_{ss} was sent to the classification block F to predict the class label.

5. Experimental Results

This section analyses our experimental results after evaluating the proposed model with the three hand gesture datasets. In Section 5.1, we describe the experimental setting, and in Section 5.2, the performance of the Senz3D and NTU datasets is evaluated. Section 5.3 involves the performance evaluation for the Kinematic dataset. We discuss performance accuracy and state-of-the-art comparison in each section.

5.1. Experimental Setting

We used three datasets to evaluate our model. Each dataset was split into two sets: 70% for training and 30% for testing. Based on the ratio, there were 924 samples considered for training, and 396 samples were considered for testing in the Senz3D dataset. In the same way, for the NTU hand gesture digit dataset, 700 and 300 samples were considered for training and testing, respectively. The kinematic dataset also had the same number of samples for training and testing as the NTU dataset. We used the TensorFlow framework of python programming [43] to implement the experiment in the Google Colab Pro edition environment. There are 25GB GPU in the processing RAM in the environment, known as Tesla P100 [44]. Tensorflow is considered a boon for deep learning models due to being an open-source software program, providing computational graphics, and having adaptability and compatibility with minimum resources. We used the OpenCV python package for the initial image processing task [45]. In addition, the pickle package was used to convert the image dataset into a byte stream for storage. Pandas and Numpy were used for the mathematical and statistical processing of the dataset, and both provide flexibility during matrix manipulation. Matplotlib was used to plot the various types of graph figures [45]. The training was performed for each dataset for 1000 epochs, while we used 0.000005 as the initial learning rate on the account of higher fluctuation during the use of the Adam optimizer with Nesterov momentum [46,47]. However, we used various parameter tuning operations for the learning rate and optimizers for the 10 and 11 classes of this study.

5.2. Performance Result of the Senz3D and NTU Dataset

Table 1 demonstrates the comparative performance of the proposed model with the state-of-the-art system. We compared our results with five of the most related models, namely GestureGAN [48], PoseGAN [49], Ma et al. [50], Yan et al. [51], and PG2 [31]; all of these models were implemented for NTU and Senz3D dataset by the authors in [48]. Ren et al. employed a template-matching approach for recognizing hand gestures, where they achieved 93.20% accuracy [22]. The authors of [41] applied a pose-guided person generation network (PG2), which included a U-Net-like network and produced 93.66% accuracy for the NTU and 98.73% for the Senz3D dataset. The authors in [51] first converted the RGB image into a skeleton and then achieved 95.33% accuracy for the NTU and 99.49% for the Senz3D dataset. In the same way, authors in [49,50] found 95.86% and 96.12% for the NTU dataset and 99.05% and 99.54% accuracy for the Senz3D dataset. The authors in [48] applied the hand gesture generative adversarial network (GestureGAN). They included a single generator and a discriminator, which took the RGB image as the input and the skeleton of the RGB as the target image. They achieved 96.66% accuracy for the NTU dataset and 99.74% accuracy for the Senz3D dataset.

Table 1. Performance results of the Senz3D and NTU dataset with state-of-the-art comparison.

Method	**NTU (%)**	**Senz3D (%)**
Zhou Ren [22]	93.20	Na
PG2 [41]	93.66	98.73
Yan et al. [51]	95.33	99.49
Ma et al. [50]	95.86	99.05
PoseGAN [49]	96.12	99.54
GestureGan [48]	96.66	99.74
Proposed Model	99.67	99.75

5.3. Performance Results of the Kinematic Dataset

Marin et al. employed different feature extraction techniques based on the position and orientation of the fingertips and employed the SVM algorithm, achieving 89.70% accuracy with SVM [14]. Marin et al. applied a feature extraction technique to extract various distance-based features, namely centroid, the curvature of hand contour, and hand shape convex [24]. After that, to select potential features, they applied different feature selections and applied the SVM and a random forest algorithm as a classifier, which produced 96.30% accuracy for the kinetic sensor RGB images. In comparison, our proposed model achieved 99.46% accuracy, which is higher than the existing models shown in Table 2.

Table 2. Performance results of the kinematic dataset and state-of-the-art comparison.

Method	KLP (%)
SVM [14]	89.70
SVM [24]	96.30
Proposed Model	99.46

6. Conclusions

We introduced a novel, multistage, spatial attention-based neural network for hand gesture recognition aiming to produce high performance and generalized properties. In our proposed architecture, multistage attention-learning models were used to learn the joint representation of the RGB modality. We employed a residual connection in the first and second stages to enhance the features by recovering missing values. The performance accuracy achieved for multiple datasets proved the efficiency of the proposed multistage network. In future work, we plan to extend this multistage neural network model to develop hand gesture recognition from a video dataset. We also plan to use multimodalities to improve our proposed system's accuracy and efficiency.

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

Funding: This work was supported by the Competitive Research Fund of The University of Aizu, Japan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest for this research.

References

1. Ren, Z.; Yuan, J.; Meng, J.; Zhang, Z. Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans. Multimed.* **2013**, *15*, 1110–1120. [CrossRef]
2. Wachs, J.P.; Kölsch, M.; Stern, H.; Edan, Y. Vision-based hand-gesture applications. *Commun. ACM* **2011**, *54*, 60–71. [CrossRef]
3. Jalal, A.; Rasheed, Y.A. Collaboration achievement along with performance maintenance in video streaming. In Proceedings of the IEEE Conference on Interactive Computer Aided Learning, Villach, Austria, 26 September 2007; Volume 2628, p. 18.
4. Jalal, A.; Shahzad, A. Multiple facial feature detection using vertex-modeling structure. In Proceedings of the ICL, Villach, Austria, 26–28 September 2007; pp. 1–7.
5. Jalal, A.; Kim, S.; Yun, B. Assembled algorithm in the real-time H. 263 codec for advanced performance. In Proceedings of the IEEE 7th International Workshop on Enterprise Networking and Computing in Healthcare Industry (HEALTHCOM 2005), Busan, Republic of Korea, 23–25 June 2005; pp. 295–298.
6. Jalal, A.; Kim, S. Advanced performance achievement using multi-algorithmic approach of video transcoder for low bit rate wireless communication. *ICGST Int. J. Graph. Vis. Image Process.* **2005**, *5*, 27–32.

7. Jalal, A.; Uddin, I. Security architecture for third generation (3G) using GMHS cellular network. In Proceedings of the 2007 IEEE International Conference on Emerging Technologies, Rawalpindi, Pakistan, 12–13 November 2007; pp. 74–79.
8. Jalal, A.; Zeb, M.A. Security enhancement for e-learning portal. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2008**, *8*.
9. Jalal, A.; Kim, S. The mechanism of edge detection using the block matching criteria for the motion estimation. *한국 HCI 학회 학술대회*. 2005; pp. 484–489. Available online: https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE01886372 (accessed on 8 June 2022).
10. Jalal, A.; Kim, S. Algorithmic implementation and efficiency maintenance of real-time environment using low-bitrate wireless communication. In Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06), Gyeongju, Republic of Korea, 27–28 April 2006; p. 6.
11. Shin, J.; Kim, C.M. Non-touch character input system based on hand tapping gestures using Kinect sensor. *IEEE Access* **2017**, *5*, 10496–10505. [CrossRef]
12. Murata, T.; Shin, J. Hand gesture and character recognition based on kinect sensor. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 278460. [CrossRef]
13. Shin, J.; Matsuoka, A.; Hasan, M.A.M.; Srizon, A.Y. American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors* **2021**, *21*, 5856. [CrossRef]
14. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with leap motion and kinect devices. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1565–1569.
15. Moeslund, T.B.; Störring, M.; Granum, E. A natural interface to a virtual environment through computer vision-estimated pointing gestures. In *International Gesture Workshop*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 59–63.
16. Streitz, N.A.; Tandler, P.; Müller-Tomfelde, C.; Konomi, S. Roomware: Towards the next generation of human–computer interaction based on an integrated design of real and virtual worlds. *Hum.-Comput. Interact. New Millenn.* **2001**, *553*, 578.
17. Dewaele, G.; Devernay, F.; Horaud, R. Hand motion from 3d point trajectories and a smooth surface model. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 495–507.
18. Miah, A.S.M.; Shin, J.; Hasan, M.A.M.; Rahim, M.A.; Okuyama, Y. Rotation, Translation Furthermore, Scale Invariant Sign Word Recognition Using Deep Learning. In *Computer Systems Science and Engineering*. Volume 44. Available online:. (accessed on 1 December 2022). [CrossRef]
19. Miah, A.S.M.; Shin, J.; Hasan, M.A.M.; Rahim, M.A. BenSignNet: Bengali Sign Language Alphabet Recognition Using Concatenated Segmentation and Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 3933. [CrossRef]
20. Erol, A.; Bebis, G.; Nicolescu, M.; Boyle, R.D.; Twombly, X. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.* **2007**, *108*, 52–73. [CrossRef]
21. Murthy, G.; Jadon, R. A review of vision-based hand gestures recognition. *Int. J. Inf. Technol. Knowl. Manag.* **2009**, *2*, 405–410.
22. Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A. Real-time human pose recognition in parts from single depth images. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1297–1304.
23. Mohla, S.; Pande, S.; Banerjee, B.; Chaudhuri, S. Fusatnet: Dual attention based spectrospatial multimodal fusion network for hyperspectral and lidar classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 92–93.
24. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimed. Tools Appl.* **2016**, *75*, 14991–15015. [CrossRef]
25. Zhou, R. Shape Based Hand Gesture Recognition. Ph.D. Thesis, Nanyang Technological University, Singapore, 2020.
26. Biasotti, S.; Tarini, M.; Giachetti, A. Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition. Available online: https://diglib.eg.org/bitstream/handle/10.2312/stag20151288/015-023.pdf (accessed on 1 December 2022).
27. Yuanyuan, S.; Yunan, L.; Xiaolong, F.; Kaibin, M.; Qiguang, M. Review of dynamic gesture recognition. *Virtual Real. Intell. Hardw.* **2021**, *3*, 183–206.
28. Nunez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Velez, J.F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognit.* **2018**, *76*, 80–94. [CrossRef]
29. Su, M.C. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.)* **2000**, *30*, 276–281.
30. Jetley, S.; Lord, N.A.; Lee, N.; Torr, P.H. Learn to pay attention. *arXiv* **2018**, arXiv:1804.02391.
31. Mou, L.; Zhu, X.X. Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 110–122. [CrossRef]
32. Iwai, Y.; Watanabe, K.; Yagi, Y.; Yachida, M. Gesture recognition by using colored gloves. In Proceedings of the 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929), Beijing, China, 14–17 October 1996; Volume 1, pp. 76–81.
33. Wilson, A.D.; Bobick, A.F. Parametric hidden markov models for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 884–900. [CrossRef]
34. Lee, H.K.; Kim, J.H. An HMM-based threshold model approach for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 961–973.

35. Kwok, C.; Fox, D.; Meila, M. Real-time particle filters. In Proceedings of the Advances in Neural Information Processing Systems 15 (NIPS 2002), Vancouver, BC, Canada, 9–14 December 2002; Volume 15.
36. Doucet, A.; De Freitas, N.; Gordon, N.J. *Sequential Monte Carlo Methods in Practice*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 1.
37. Nagi, J.; Ducatelle, F.; Di Caro, G.A.; Cireşan, D.; Meier, U.; Giusti, A.; Nagi, F.; Schmidhuber, J.; Gambardella, L.M. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In Proceedings of the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 16–18 November 2011; pp. 342–347.
38. Tao, W.; Leu, M.C.; Yin, Z. American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Eng. Appl. Artif. Intell.* **2018**, *76*, 202–213. [CrossRef]
39. Naguri, C.R.; Bunescu, R.C. Recognition of dynamic hand gestures from 3D motion data using LSTM and CNN architectures. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 1130–1133.
40. Memo, A.; Zanuttigh, P. Head-mounted gesture-controlled interface for human-computer interaction. *Multimed. Tools Appl.* **2018**, *77*, 27–53. [CrossRef]
41. Ma, L.; Jia, X.; Sun, Q.; Schiele, B.; Tuytelaars, T.; Gool, L.V. Pose Guided Person Image Generation. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
42. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation network. In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
43. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
44. Tock, K. Google CoLaboratory as a platform for Python coding with students. *RTSRE Proc.* **2019**, *2*.
45. Gollapudi, S. OpenCV with Python. In *Learn Computer Vision Using OpenCV*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 31–50.
46. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international artificial intelligence and statistics conference. JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
47. Dozat, T. Incorporating Nesterov Momentum into Adam. 2016. Available online: https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ (accessed on 1 December 2022).
48. Tang, H.; Wang, W.; Xu, D.; Yan, Y.; Sebe, N. GestureGAN for Hand Gesture-to-Gesture Translation in the Wild. In Proceedings of the CVPR 2018 (IEEE), Salt Lake City, UT, USA, 18–23 June 2018; pp. 774–782.
49. Siarohin, A.; Sangineto, E.; Lathuilière, S.; Sebe, N. Deformable GANs for Pose-based Human Image Generation. In Proceedings of the CVPR 2018 (IEEE), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3408–3416.
50. Ma, L.; Sun, Q.; Georgoulis, S.; Van Gool, L.; Schiele, B.; Fritz, M. Disentangled Person Image Generation. In Proceedings of the CVPR 2018 (IEEE), Salt Lake City, UT, USA, 18–23 June 2018; pp. 99–108.
51. Yan, Y.; Xu, J.; Ni, B.; Zhang, W.; Yang, X. Skeleton-aided articulated motion generation. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 199–207.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.