




Article

Next-Gen Dynamic Hand Gesture Recognition: MediaPipe, Inception-v3 and LSTM-Based Enhanced Deep Learning Model

Yaseen ¹, Oh-Jin Kwon ^{1,*}, Jaeho Kim ², Sonain Jamil ³, Jinhee Lee ¹ and Faiz Ullah ¹

¹ Department of Electronics Engineering, Sejong University, Seoul 05006, Republic of Korea; yaseen@sju.ac.kr (Y.); jinee5025@sju.ac.kr (J.L.); faiz@sju.ac.kr (F.U.)

² Department of Electrical Engineering, Sejong University, Seoul 05006, Republic of Korea; kimjh@sejong.ac.kr

³ Department of Computer Science, Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway; sonainj@stud.ntnu.no

* Correspondence: ojkwon@sejong.ac.kr

Abstract: Gesture recognition is crucial in computer vision-based applications, such as drone control, gaming, virtual and augmented reality (VR/AR), and security, especially in human–computer interaction (HCI)-based systems. There are two types of gesture recognition systems, i.e., static and dynamic. However, our focus in this paper is on dynamic gesture recognition. In dynamic hand gesture recognition systems, the sequences of frames, i.e., temporal data, pose significant processing challenges and reduce efficiency compared to static gestures. These data become multi-dimensional compared to static images because spatial and temporal data are being processed, which demands complex deep learning (DL) models with increased computational costs. This article presents a novel triple-layer algorithm that efficiently reduces the 3D feature map into 1D row vectors and enhances the overall performance. First, we process the individual images in a given sequence using the MediaPipe framework and extract the regions of interest (ROI). The processed cropped image is then passed to the Inception-v3 for the 2D feature extractor. Finally, a long short-term memory (LSTM) network is used as a temporal feature extractor and classifier. Our proposed method achieves an average accuracy of more than 89.7%. The experimental results also show that the proposed framework outperforms existing state-of-the-art methods.

Keywords: dynamic hand gesture recognition; hybrid deep learning; dimensionality reduction; temporal data classification; transfer learning; vision-based drone control



Citation: Yaseen; Kwon, O.-J.; Kim, J.; Jamil, S.; Lee, J.; Ullah, F. Next-Gen Dynamic Hand Gesture Recognition: MediaPipe, Inception-v3 and LSTM-Based Enhanced Deep Learning Model. *Electronics* **2024**, *13*, 3233. <https://doi.org/10.3390/electronics13163233>

Academic Editors: Fath U Min Ullah, Estefanía Talavera and Nuno Gonçalves

Received: 12 July 2024

Revised: 6 August 2024

Accepted: 12 August 2024

Published: 15 August 2024



Copyright: © 2024 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 (HGR) is a popular research area due to its scope in applications like human–computer interaction (HCI), drone control, virtual/augmented reality, sign language interpretation, and other interaction-based systems [1,2]. Gestures are natural and intuitive to communicate, sometimes with barely apparent body movements [3]. Computer vision exploits these gestures, making it possible to interact with technology without aided sensors. These gestures can be of two types: static gestures and dynamic gestures. Static gestures represent a still hand gesture or hand posture in a single image frame, i.e., relying fundamentally on spatial information [4]. Dynamic hand gestures are represented by continuous hand motion and are therefore represented by sequences of images, i.e., by temporal information [5]. Compared to static gestures, dynamic gestures can represent and enable a wider range of activities [6]. Vision-based hand gestures provide a touchless interface, and the gestures are recorded by an RGB camera device [7].

Deep learning algorithms, like convolutional neural networks (CNN), are used for hand gesture recognition. This method recognizes hand gestures with high accuracy without using any aided sensors [8,9]. Thus, the user can interact with a computer system more naturally.

Researchers face several challenges, and they have been addressed quite well. However, one common challenge related to dynamic hand gestures is the variability in the length of sequences of image frames in a gesture clip. Variable-length gesture clips pose challenges in data processing, model training, and inference. This also increases the computational time and power. While work has been conducted on achieving high accuracy, less attention has been given to handling the complex shape of gestures. Several authors have reported their work relating to dynamic hand gestures. Recently, Karsh et al. [4] reported their work on the same topic. However, they have tried to focus more on improving the deep learning architecture first rather than on data processing. The authors in [10] have reported work on skeletal-based hand data. For more realistic environments and practical scenarios, datasets need to be more diverse, incorporating images taken in a natural environment. Authors in [3] have also contributed to improving the accuracy by proposing a hybrid CNN model. Their hybrid model consists of a feature extractor 3D CNN module and a classifier module making use of the computational cost and power.

Overall, the work conducted so far on dynamic hand gestures is significant, but little focus has been given to the complexity of their temporal aspects. Considering the research gap, we list our contributions as follows:

- Extending the work conducted by authors [3], we reduce the dimensionality of the 3D feature map to 1D while retaining the temporal aspects of the data.
- Temporal data that is present in the sequence of frames are utilized efficiently by the proposed hybrid model at a lower computational cost than existing methods.
- A lightweight model is proposed, reducing the computation complexity.
- The performance is improved compared to the baseline algorithm. [3]

In summary, we address the temporal complexity of dynamic hand gestures, paving the path for a vision-based interaction system for dynamic hand gesture recognition. The rest of the paper is described as follows: In Section 2, we discuss related work and methods; Section 3 presents our proposed method; Section 4 evaluates the model and presents the results. Section 5 includes our discussion, and finally, in the Section 6 conclusions, we point out the limitations, other applications for our method, and future works.

2. Related Work

In deep learning (DL), CNNs are used simultaneously for feature extraction and classification. Using convolution and pooling optimizes feature extraction and eases the classification process [11]. CNNs were first used in 1990 to recognize handwritten digits. The foundation of modern CNN architecture was laid down by authors [12]. Based on their architecture, more robust classification algorithms were proposed, including VGGNet [13], Resnet50 [14], and GoogleNet [15]. With the increased depth of these CNN networks, their accuracy improved, and so did their demand for high computational power. Currently, researchers are trying to build lightweight CNN models for use in devices with low computing power and in real-world scenarios [16].

2.1. Dynamic Hand Gesture Classification

There are various deep learning architectures for temporal data classification. Some of them are worth mentioning [17], such as 2D CNNs, which are computationally less expensive and are widely used for spatial data feature extraction and classification. The more complex model architectures are 3D CNNs; the third dimension comes from temporal data in these models. Then, two-stream CNNs use 2D CNNs for sparse data classification and recurrent neural networks (RNNs) for temporal data classification. While there are several others, the hybrid approaches are the most popular as they increase accuracy while reducing the computational cost. In hybrid models, CNNs are combined with RNN, gated recurrent neural networks, or long short-term memory (LSTM) [18]. In the domain of hand gesture recognition, there exist numerous state-of-the-art methods [19–21]. Dynamic hand gestures are more complex to process as the temporal data have to be considered, as well as

the optical flow of the data [22]. While dynamic hand gesture recognition authors propose several novel methods, we will discuss the most recent hybrid methods.

Authors in [23] investigated HGR with leap motion sensor-based data, and although they achieved an accuracy of 97%, the work does not provide a fully touchless interface as in the case of RGB data. This work was further extended by Huu [24] while using two-stream approaches, achieving a similar accuracy [23]. In the case of 3D CNNs, authors in the work [25] claimed an accuracy of 90%, but their system is limited to working only under the condition that a person has to sit close to the camera while performing gestures. Another limitation is that they use more complex CNN models for their work. Similarly, in the case of hybrid approaches, another hybrid model was recently proposed by authors [3]. The authors used Google's Inception-v3 architecture as a feature extractor and LSTM for temporal data feature extraction and classification. They have achieved an overall accuracy of 84%. However, their work could not process variable-length sequences of dynamic hand gestures, and there is more room for improvement in the accuracy [3]. Even if the accuracy was much higher, the practical use of this model in real-world scenarios is limited, as during live feed from the camera, the length of a gesture clip is always variable [26], which the authors couldn't address in their work [3].

2.2. Classification with Hybrid Deep Learning

As we know, for a dynamic hand gesture recognition system, spatial data processing is not sufficient; temporal relations in sequences of frames have to be exploited [27]. Various state-of-the-art methods have been reported to work on video classification tasks, especially on dynamic hand gestures using hybrid approaches, i.e., combining CNN or RNN, CNN, and LSTM, and similar other approaches [1]. Various studies have tried exploiting hybrid deep learning models based on CNNs and RNNs for HGR [24,28] and other video classification tasks [29].

In traditional neural networks, neurons are only connected in a feed-forward direction, while in an RNN, the neurons are not only connected in a mixed fashion within a layer but are also connected in opposite directions (both forward and backward) [30]. Furthermore, LSTMs are modified RNNs for the classification of temporal data so that they can store the data in memory cells. These cells contain the output of the previous state relative to the current input. Through memory cells, the temporal relation between the data can be retained [31].

Figure 1 shows this hybrid approach; CNN is used as a feature extractor as they are suitable for recognizing image patterns. At this stage, the spatial information from individual images is only processed using 2D CNN, which does not classify but obtains the hidden patterns. The feature maps for all the frames within the sequence are then passed into the LSTM network. The LSTM extracts features and hidden patterns, exploring temporal relations from the feature maps and performing classification. This yields improved classification and better results [32].

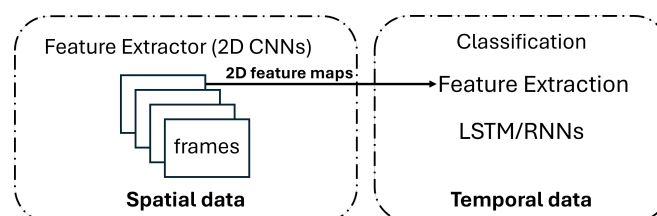


Figure 1. A basic CNN-RNN-based hybrid architecture.

3. Methodology

Compared to static hand gestures, dynamic hand gestures have sequences of frames where the hand moment and position within each frame are recorded concerning the time. For this reason, the hybrid deep neural network architecture that comprises CNN and RNN is well suited for temporal data classification tasks [33]. Our proposed architecture

is inspired by Hax et al. [3]. They have used the Google Inception v3 [34] model for feature extraction and an LSTM for classification. In our case, we have replaced the Google Inception v3 [34] with MediaPipe 0.10.14 in the input layer [35]—to remove unwanted features from the frame and locate the hand gesture efficiently. MediaPipe has a built-in hand landmark model, which returns the knuckle points of the hands in an image with low computational power. Using this advantage, we locate the region of the hand in an image and crop the area called the region of interest (ROI). This cropped image is used in the input for the Inception-v3 layer of feature extraction. The highly efficient approach leads to improved accuracy and lightweight models, reducing the computation complexity and cost. The following points are presented for comparison:

- The use of MediaPipe as an ROI extractor effectively locates the ROI, reducing the image dimension and computational cost.
- It also offers improved efficiency for hand landmark detection compared to Inception v3 alone.
- Time complexity is low, and real-time performance is highly durable compared to Inception v3 without MediaPipe.
- MediaPipe, when employed as an ROI extractor on the input layer in a hybrid architecture alongside an LSTM layer, as shown in Figure 1, exhibits a significant enhancement in terms of classification accuracy.

Comparing our architecture to the baseline, we have proposed the following changes:

- We have moved Google Inception v3 from the input layer to the middle layer [34]. The input layer uses Google MediaPipe as an ROI extractor [35].
- The frame dimensions are lowered from $(1600 \times 900 \times 3)$ to $(50 \times 60 \times 3)$.
- In the case of baseline architecture, ten frames are processed per sequence, and the proposed architecture can process variable-length sequences.

The processed data from MediaPipe is fed into the LSTM block as shown in Figure 2. The LSTM exploits the temporal relationship of the received data for classification. We replicate the same layer architecture as used by the authors [3], including dropout layers and a dense layer (using a softmax function) at the final stage to prevent the model from over-fitting and classification, respectively. The proposed hybrid model consists of three main layers. MediaPipe is used at the top layer to extract the ROI; in the middle layer, Google Inception v3 is used as a feature extractor for input to the third layer, i.e., the LSTM layer, which is used as a classifier [18].

3.1. Dynamic Hand Gesture Dataset

For a fair comparison of our proposed work with that of the baseline architecture, we select the same dataset, “Depth_Camera_Dataset” [36], for training the model, as used by the authors [3]. This dataset has been collected concerning different subjects under various lighting conditions. It has six different gestures, each with 662 sequences, comprising 40 frames per sequence. The list of gestures is as follows: scroll down, scroll up, scroll left, scroll right, zoom in, and zoom out. The background has additional people sometimes engaging in other activities; this is included intentionally to add complexity to the dataset for real-world simulation. A depth camera recorded the dataset so that it also contains depth data, but our focus in this paper is a vision-based hybrid model.

3.2. Data Processing Pipeline

Our proposed deep learning-based hybrid architecture pipeline was implemented in Python using TensorFlow. The proposed model consists of three main blocks: MediaPipe, Inception v3, and the final LSTM classifier, as shown in Figure 2. In the pre-processing stage, frames from the directory are loaded sequence by sequence, and every fourth frame is selected from the sequence, as more frames did not increase the accuracy [3]. Finally, a total of 10 frames are used per sequence. MediaPipe is a hand locator and a region of interest (ROI) extractor. The ROI is cropped for each frame. This technique removes the

unwanted features from a given frame, thus reducing the dimensionality of the data (from $1600 \times 900 \times 3$ to $50 \times 60 \times 3$) per frame. Therefore, the pre-processed data frame from the MediaPipe block is $(10, 50, 60, 3)$ in size. Inception v3 is used as a 2D CNN-based feature extractor, which then outputs a 1D row vector; combinedly, the size is $(10, 2048, 1)$ per sequence. This data frame is then used as an input to the LSTM block for final feature extraction and classification. Also, 70, 20, and 10% of the data is split into training, validation, and testing portions.

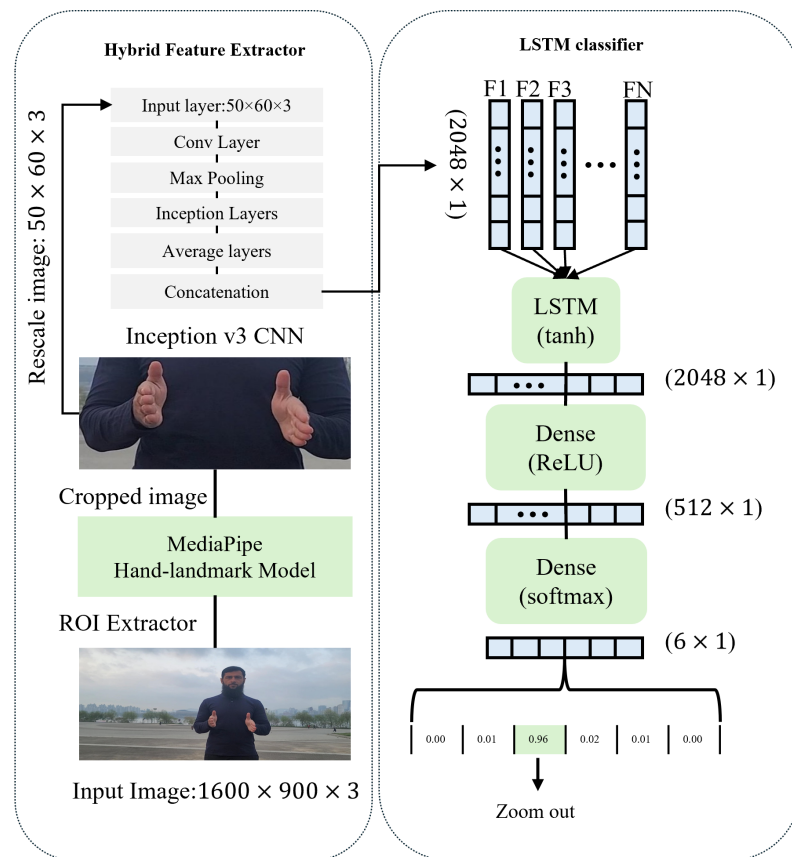


Figure 2. A triple-layer deep learning-based lightweight hybrid architecture.

3.3. Evaluating Criteria

For the model's assessment and evaluation, we used the following metrics:

- **Accuracy:** Reflects the model's performance regarding the overall correctness for the given number of test instances (1).
- **Recall:** Also called true positive rate as it measures the model's ability regarding true positives (2).
- **F1-score:** Reflects the model's ability concerning false positives and negatives (3).
- **Specificity:** Reflects the model's ability to correctly identify true negative instances (4).

Formulas for each of these metrics are given below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (4)$$

Results based on the above metrics are shown in Table 1. For more visually appealing results and deeper insights, we have used a confusion matrix as shown in Figures 3 and 4.

Also, to further evaluate our results for the proposed architecture, we replace the LSTM classifier with a gated recurrent unit (GRU) as these are also the most promising models [18]. We consider the same data split for this architecture as well. We have also used the Resnet50, replacing Inception v3 in our architecture, for further comparison.

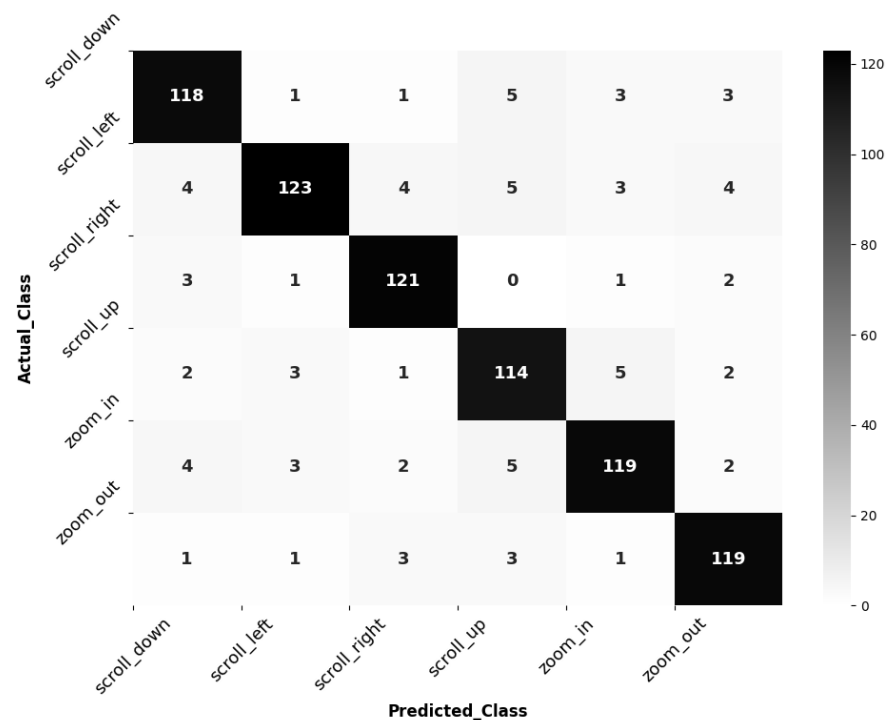


Figure 3. Confusion matrix based on validation data.

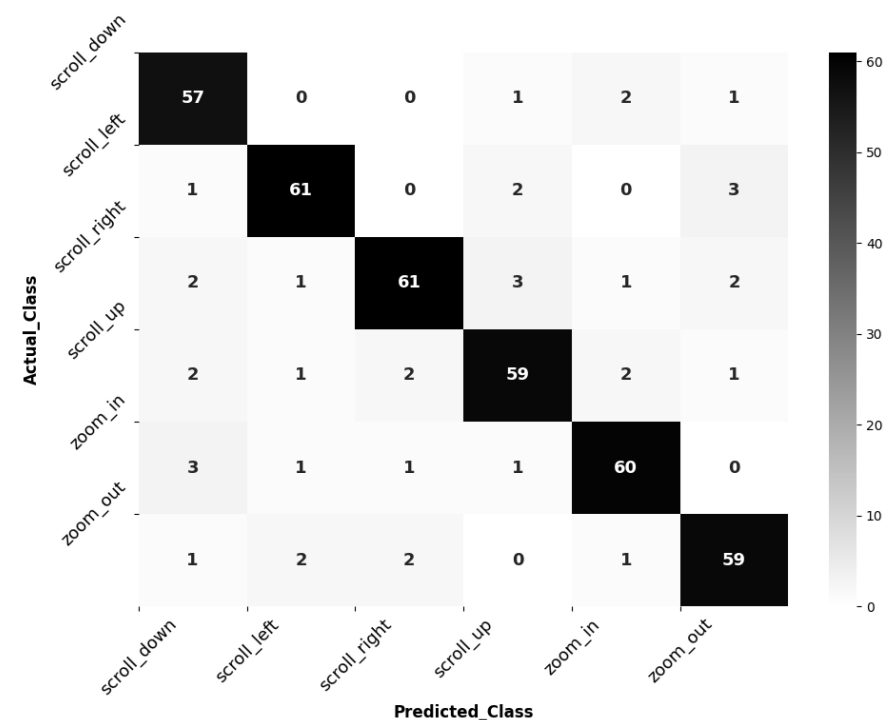


Figure 4. Confusion matrix based on test data.

Table 1. Average results based on the performance metrics for test and validation data.

Data	Recall	F1-Score	Specificity	Accuracy
Validation	89.9%	89.7%	98.0%	90.1%
Test	91.5%	90.0%	97.9%	89.7%

4. Results

This Section presents the experimental setup, performance metrics, and visualization (the reason and rationale behind our model architecture and dataset pre-processing).

4.1. Experimental Setup

The models were trained for 300 epochs, with early stopping enabled to avoid over-fitting, and the initial learning rate was set to 0.001. The categorical cross-entropy cost function was used for training and validation losses for maximum learning. The adam-optimizer was used to iteratively update and adjust the network weights in order to minimize these losses. In the case of activation functions, the tangent hyperbolic function was used for the LSTM layer and the softmax function for the final output layer. For the intermediate layer (Dense layer), the rectified linear unit (ReLU) was employed (see Figure 2). The machine we used for our pipeline has the following specs: We used an HP desktop computer, manufactured by HP Inc., Palo Alto, CA, USA, model Pavilion Gaming Desktop TG01-1xxx, with a RAM of 32 GB, GPU of NVIDIA GeForce RTX 2060 SUPER, and eight cores with clock speed of 2.9 GHz.

4.2. Performance Metrics and Visualizations

The proposed hybrid deep learning-based model for hand gesture recognition demonstrated good performance when evaluated via various metrics, as shown in Table 1. The empirical tests, both on validation and test results, showcase satisfactory results; an average accuracy of 89.7% on test data and 90.1% on validation data was reported. Other metrics were also used to assess the model's ability; we achieved a recall of 91%, an F1-score of 90.0%, and a specificity of 98.0%.

The results were also compared to that of other models [3,37–39], as shown in Table 2. The average accuracy on the test, as well as the validation data, reveal a clear win for our proposed architecture. The Hax et al. [3] model has an average accuracy of 84.7%, while ours achieved 90.1%. Similarly, for test data, the Hax et al. [3] model achieved an accuracy of 83.6%, while ours achieved an average accuracy of 89.7%.

Table 2. Average results based on the performance metrics for test and validation data.

Architecture	Model	Data	Accuracy
Hax et al. [3]	Inception-LSTM	Validation	84.7%
		Test	83.6%
Res3-ATN [37]	Residual-Attention Network	Validation	64.13%
		Test	62.70%
Spat. st. CNN [38]	Two Streams CNN	Validation	55.79%
		Test	54.60%
C3D [39]	3D CNNs	Validation	64.90%
		Test	69.30%
Ours	ROI-Inception-LSTM	Validation	90.1%
		Test	89.7%
	ROI-Inception-GRU	Validation	85.6%
		Test	84.1%

Figures 3 and 4 also present the classification confusion data for each category dataset. The true positive rate for each category is given on the diagonal squares in the confusion matrix. The most precise detected moment in validation is the “scroll_right” category.

The confusion matrix also gives a detailed insight into the true negatives, false positives, and false negatives for each of the categories. By comparing these results with the baseline, our proposed method demonstrates excellent performance, which can be attributed to our modification to the proposed architecture.

We also compared the performance of our model with the state-of-the-art models. The classification results, as shown in Table 2, were quite satisfactory compared to other models. To compare the computational complexity and memory usage of the proposed model with the baseline methods, we have provided a comparison in Table 3. The results clearly show the superiority of the proposed model in terms of weight and low memory usage during inference.

Table 3. Comparison of model parameters and memory size.

Model	Parameters	VRAM (MB)
Hax et. al. [3]	29 M	450
Res3-ATN [37]	26 M	400
Spat. st CNN [38]	38 M	600
C3D [39]	25 M	350
Ours	16 M	150

5. Discussion

Section 4.2 is quite revealing in many ways, as it presents the detailed average results in Tables 1 and 2 and the confusion matrix; the results show uniform error properties across all categories. Gestures like “zoom_in” and “zoom_out” are primarily confused. Still, the true positive rate is above 83%. The Inception-v3 combined with MediaPipe is a booster for enhanced dynamic hand gesture classification. Even though the data were scarce, the model performance demonstrated that this algorithm can perform much better on sufficient data. The proposed model can also be used in other areas of HCI where temporal classification is demanded with minimal data availability for training. As discussed in the related work, in the classification area using CNNs, the trend is to use a lightweight architecture that could achieve better results for practical application scenarios. Our proposed model paved the path for more lightweight architecture. It uses MediaPipe to extract the regions most wanted from the image (ROI) and feed them to the middle layer for feature extraction. This approach reduces the dimensionality of data and increases the availability of rich features; the LSTM layer further utilizes this for temporal data extraction and classification.

Our proposed model can also be used in lightweight real-world applications like drone control where real-time classification is needed. This is because drone control is usually an outdoor activity, and the model can extract rich features with the power of the MediaPipe and Inception-v3 layer together, as well as being able to perform better in the case of dynamic gestures when LSTM is used in the top layer as a classifier and temporal feature extractor. The deep learning models are always data-hungry, which is why the current trend is to use transfer learning to overcome the issue of data scarcity. Using Resnet50, Inception-v3 is one of the many modules used in deep learning models to mitigate the data scarcity issue; the proposed architecture can be used as a transfer learning model in dynamic hand gesture recognition in various HCI applications.

6. Conclusions

In this work, we proposed a hybrid deep learning architecture for classifying dynamic hand gestures (video clips) for use in real-world scenarios. The architecture comprises three layers: The MediaPipe-based ROI extractor is the bottom layer, the middle layer uses 2D CNN (Inception v3) as a feature extractor, and the top layer is a temporal feature extractor and classifier. The proposed model was trained by the Depth_Camera_dataset, a dataset designed explicitly for dynamic hand gestures [36]. With our best-split data, the model achieved an average test accuracy of more than 89.7%. This can be attributed to our novel changes in the proposed hybrid architecture, which includes the addition of

the bottom layer, the MediaPipe-based hand region extractor (ROI), which reduces the dimensionality and computational cost. The second reason is that the feature extractors were doubled; Inception v3 was used as a feature extractor in a focused area, enhancing the features' quality while reducing the computational cost. The third change we made was the input to the LSTM layer, which concatenates features obtained for a sequence of frames per gesture. In this study, we focused on limited gestures that were available in the dataset; further investigations can be carried out on datasets with a large number of gestures. Also, for real world applications, diversity in the datasets is mandatory, as well as the subjective evaluation of the model. Future work includes the subjective testing of the proposed architecture on other benchmark datasets and training the model on variable-length hand gesture clips.

Author Contributions: Conceptualization, Y. and O.-J.K.; data curation, Y.; formal analysis, Y.; funding acquisition, O.-J.K., J.K. and J.L.; project administration, J.L.; resources, J.L.; software, Y., J.L. and F.U.; supervision, O.-J.K.; visualization, Y.; writing—original draft, Y. and S.J.; writing—review and editing, Y. and S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, the Republic of Korea (2023M3C1C1A01098414). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01816) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

HGR	Hand Gesture Recognition
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
ROI	Region of Interest

References

1. Rastgoo, R.; Kiani, K.; Escalera, S.; Sabokrou, M. Multi-modal zero-shot dynamic hand gesture recognition. *Expert Syst. Appl.* **2024**, *247*, 123349. [\[CrossRef\]](#)
2. Balaji, P.; Prusty, M.R. Multimodal fusion hierarchical self-attention network for dynamic hand gesture recognition. *J. Vis. Commun. Image Represent.* **2024**, *98*, 104019. [\[CrossRef\]](#)
3. Hax, D.R.T.; Penava, P.; Krodell, S.; Razova, L.; Buettner, R. A Novel Hybrid Deep Learning Architecture for Dynamic Hand Gesture Recognition. *IEEE Access* **2024**, *12*, 28761–28774. [\[CrossRef\]](#)
4. Karsh, B.; Laskar, R.H.; Karsh, R.K. mXception and dynamic image for hand gesture recognition. *Neural Comput. Appl.* **2024**, *36*, 8281. [\[CrossRef\]](#)
5. Sunanda; Balmik, A.; Nandy, A. A novel feature fusion technique for robust hand gesture recognition. *Multimed. Tools Appl.* **2024**, *83*, 65815–65831. [\[CrossRef\]](#)
6. Shi, Y.; Li, Y.; Fu, X.; Miao, K.; Miao, Q. Review of dynamic gesture recognition. *Virtual Real. Intell. Hardw.* **2021**, *3*, 183–206. [\[CrossRef\]](#)
7. Jain, R.; Karsh, R.K.; Barbhuiya, A.A. Literature review of vision-based dynamic gesture recognition using deep learning techniques. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7159. [\[CrossRef\]](#)
8. Kapuscinski, T.; Inglot, K. Vision-based gesture modeling for signed expressions recognition. *Procedia Comput. Sci.* **2022**, *207*, 1007–1016. [\[CrossRef\]](#)
9. Yaseen; Kwon, O.J.; Lee, J.; Ullah, F.; Jamil, S.; Kim, J.S. Automatic Sequential Stitching of High-Resolution Panorama for Android Devices Using Precapture Feature Detection and the Orientation Sensor. *Sensors* **2023**, *23*, 879. [\[CrossRef\]](#)

10. Abdullahi, S.B.; Chamnongthai, K. American sign language words recognition of skeletal videos using processed video driven multi-stacked deep LSTM. *Sensors* **2022**, *22*, 1406. [\[CrossRef\]](#) [\[PubMed\]](#)
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Saxe, A.; Nelli, S.; Summerfield, C. If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* **2021**, *22*, 55–67. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
15. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
16. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *33*, 6999–7019. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Wang, S.; Cao, J.; Philip, S.Y. Deep learning for spatio-temporal data mining: A survey. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3681–3700. [\[CrossRef\]](#)
18. Ur Rehman, A.; Belhaouari, S.B.; Kabir, M.A.; Khan, A. On the use of deep learning for video classification. *Appl. Sci.* **2023**, *13*, 2007. [\[CrossRef\]](#)
19. Adithya, V.; Rajesh, R. A deep convolutional neural network approach for static hand gesture recognition. *Procedia Comput. Sci.* **2020**, *171*, 2353–2361.
20. Xia, C.; Saito, A.; Sugiura, Y. Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor. *Sens. Actuators A Phys.* **2022**, *338*, 113463. [\[CrossRef\]](#)
21. Dang, T.L.; Tran, S.D.; Nguyen, T.H.; Kim, S.; Monet, N. An improved hand gesture recognition system using keypoints and hand bounding boxes. *Array* **2022**, *16*, 100251. [\[CrossRef\]](#)
22. Rautaray, S.S.; Agrawal, A. Real time gesture recognition system for interaction in dynamic environment. *Procedia Technol.* **2012**, *4*, 595–599. [\[CrossRef\]](#)
23. Naguri, C.R.; Bunesu, 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.
24. Huu, P.N.; Ngoc, T.L. Two-stream convolutional network for dynamic hand gesture recognition using convolutional long short-term memory networks. *Vietnam J. Sci. Technol.* **2020**, *58*, 514–523.
25. Zhang, W.; Wang, J. Dynamic hand gesture recognition based on 3D convolutional neural network models. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 9–11 May 2019; pp. 224–229.
26. Wang, X.; Lafreniere, B.; Zhao, J. Exploring Visualizations for Precisely Guiding Bare Hand Gestures in Virtual Reality. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–14 May 2024; pp. 1–19.
27. Ye, W.; Cheng, J.; Yang, F.; Xu, Y. Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks. *IEEE Access* **2019**, *7*, 67772–67780. [\[CrossRef\]](#)
28. Ma, C.Y.; Chen, M.H.; Kira, Z.; AlRegib, G. TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Process. Image Commun.* **2019**, *71*, 76–87. [\[CrossRef\]](#)
29. Abdullahi, S.B.; Bature, Z.A.; Gabralla, L.A.; Chiroma, H. Lie recognition with multi-modal spatial-temporal state transition patterns based on hybrid convolutional neural network–bidirectional long short-term memory. *Brain Sci.* **2023**, *13*, 555. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Durstewitz, D.; Koppe, G.; Thurm, M.I. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nat. Rev. Neurosci.* **2023**, *24*, 693–710. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Hendrikx, N.; Barhmi, K.; Visser, L.; de Bruin, T.; Pó, M.; Salah, A.; van Sark, W. All sky imaging-based short-term solar irradiance forecasting with Long Short-Term Memory networks. *Sol. Energy* **2024**, *272*, 112463. [\[CrossRef\]](#)
32. Rafiq, I.; Mahmood, A.; Ahmed, U.; Khan, A.R.; Arshad, K.; Assaleh, K.; Ratyal, N.I.; Zoha, A. A Hybrid Approach for Forecasting Occupancy of Building’s Multiple Space Types. *IEEE Access* **2024**, *12*, 50202–50216. [\[CrossRef\]](#)
33. Pan, Y.; Shang, Y.; Liu, T.; Shao, Z.; Guo, G.; Ding, H.; Hu, Q. Spatial-temporal attention network for depression recognition from facial videos. *Expert Syst. Appl.* **2024**, *237*, 121410. [\[CrossRef\]](#)
34. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
35. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.; Lee, J.; et al. Mediapipe: A framework for perceiving and processing reality. In Proceedings of the Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019, Long Beach, CA, USA, 17 June 2019; Volume 2019.
36. Jeeru, S.; Sivapuram, A.K.; León, D.G.; Gröli, J.; Yeduri, S.R.; Cenkeramaddi, L.R. Depth camera based dataset of hand gestures. *Data Brief* **2022**, *45*, 108659. [\[CrossRef\]](#)
37. Dhingra, N.; Kunz, A. Res3atn-Deep 3D residual attention network for hand gesture recognition in videos. In Proceedings of the 2019 International Conference on 3D Vision (3DV), Quebec City, QC, Canada, 16–19 September 2019; pp. 491–501.

38. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
39. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.

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.