# Dynamic Hand Gesture Recognition Based on Deep Learning for Muslim Elderly Care

Hadya Ayeisha Marzuki(✉) and Noramiza Hashim

Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia
1181101256@student.mmu.edu.my

**Abstract.** Gesture recognition for elderly care is an approach to classify gestures performed by the elderly to convey specific messages. Nursing homes or caretakers are often hired to take care of senior citizens and are responsible to keep them safe. Hence, this study will be useful to assist caretakers in providing needs requested by the elderly when they are absent. A collection of dynamic hand gesture recognition (HGR) datasets consisting of ten gesture classes with 630 videos is used to build the models. The ten gestures will represent requests for help to perform daily activities namely eating, toileting and dressing. Specifically for this research, we have infused Islamic hand gestures such as gestures to perform prayers and read the Quran. In this paper, we adopted the action recognition pre-trained model into the HGR by using CNN RNN and Transformer with CNN models. The result of this study shows that Transformer with CNN model has higher accuracy in recognizing hand gestures compared to CNN-RNN model.

**Keywords:** Dynamic Hand Gesture Recognition · Muslim Elderly Care · CNN-RNN · Transformer

## 1 Introduction

Hand gestures enable an intuitive and natural means of connection between humans, and can also serve well for interaction between humans and machines [1]. A wide range of applications incorporating hand gesture recognition (HGR) shows effectiveness in sign language recognition [1, 2], home monitoring systems [3], and the robotic industry [4]. In the application of elderly care, there is a scarcity of research to utilize human-computer interfaces (HCI) based on dynamic (HGR). The studies are limited given the acceptance and practicality of adopting technological solutions for senior citizens. Existing studies by [5–7], adopt HGR as part of a monitoring system to help care providers manage the elderly either from their personal home or care center. The work by [8] involves elderly people performing gestures in front of a camera to get assistances, where the meaning of the gestures is then sent to the caretaker by a text message.


S.-C. Haw and K. Sonai Muthu (Eds.): CITIC 2022, 10, pp. 544–557, 2022.
https://doi.org/10.2991/978-94-6463-094-7_44

The elderly have physical limitations that prevent them from performing daily activities and living independently as they age. The increase in dependency on their own self-care requires a full-time caretaker to help care for their well-being. The main problem is the need for constant communication between the elderly and their caretaker. With HGR, the elderly able to request help in the absence of a caretaker. The predefined gestures need to include basic help requests such as eating, toilet, dressing, and emergency.

It is important for the gesture to be in the most natural way and in a way that it visualizes the action. HGR may differ based on culture, ethnicity, language, and religion. Specifically, for this research, we infused Islamic culture elements to cater to Muslim elderly care. In the collection of datasets, there are two Islamic gestures, namely, the request to help in prayer and reading Quran. In addition, we used an Islamic hand gesture to make Shahadah represent a gesture for reading or listening to Quran. In total, 10 classes of gestures will be adopted in Muslim elderly care.

To create datasets, HGR is performed using wearable sensors until advancements in computer vision and deep learning models widen research using camera sensors. A vison-based method is preferable in elderly care as it offers a remote, contactless, and safer environment. Camera sensors are used to identify both static and dynamic gestures. From spatial features in static gestures, temporal features are added to create dynamic gestures. According to [9], dynamic gesture recognition is widely used because it conveys more information.

References [10–14] use deep learning approaches with spatial-temporal features in their work to achieve dynamic gesture recognition. Three deep learning models that have been built for HGR are 3DCNN, CNN-RNN, and CNN-Transformer. In the process of building dynamic HGR models, CNN is designed with other networks to achieve recognition. The traditional CNN model is built from a sequence of convolutional and pooling layers followed by the hidden dense layer. The main advantage of CNN is that it can automatically extract significant features without any human intervention. As CNN acts as a spatial feature extractor it requires a complementary network to learn the temporal information of the sequence frames. References [11, 12] adopted RNNs in their work to identify hand gestures in video frames.

Whereas, [13–15] developed HGR using transformer-based neural networks to extract temporal features of the video sequence. The architecture of the model achieves sequence modeling by using an encoder-decoder architecture based on attention layers. The current work using this method can replace the traditional recurrent model namely RNNs.

Since there is a limitation in research on Islamic gestures, this paper infused Islamic gestures into the dynamic HGR model. The chosen Islamic gestures will represent the request for help to perform prayers and read the Quran. We propose to compare the performance of two neural networks model namely CNN-RNN and Transformer with CNN to identify ten dynamic hand gestures. Figure 1 shows the overview of the dynamic HGR framework.

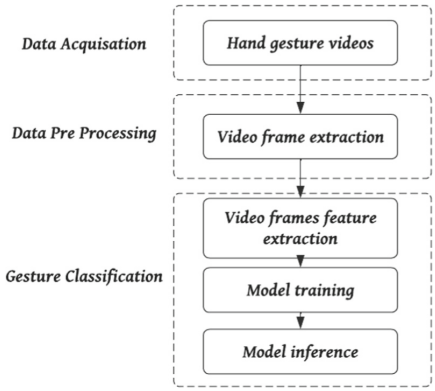



**Fig. 1.** Overview of Dynamic HGR framework

## 2 Methodology

In this section, we proposed two different methods to achieve HGR. Given that both methods have shown remarkable results in action recognition, and commonly can be used in time series data, we adopted the methods for HGR. Each of the methods will be trained individually using the same experimental setup and dataset.

### 2.1 CNN-RNN

For the first proposed gesture recognition method, a combination of convolutional (CNN) and recurrent networks (RNN) is used. This work is inspired by [12] where the author trained a video classifier with transfer learning and a recurrent model for the human activity dataset. The CNN architecture is built for spatial processing and RNN for temporal processing which consists of GRU layers.

To perform spatial processing on the videos, all sequences of 120 RGB frames extracted from the video are given to the pre-trained feature extractor. The feature extractor is taken from the Keras Application named InceptionV3 Model [16] which has been pre-trained using the ImageNet-1k dataset. The model is composed of 42 layers, as shown in Table 1 and Fig. 2, illustrating the architecture of the InceptionV3 model.

As for temporal processing, recurrent network with bidirectional Gated Recurrent Unit (GRU) layers are used. The model is given the features extracted from the InceptionV3 model to make classification. The bidirectional GRU layers are used to overcome vanishing gradient in RNN which resulted in short-term memory problem.

### 2.2 Transformer with CNN

A pre-trained CNN is employed to learn the spatial features and the temporal information is added later to the network by using the attention mechanism in the transformer. The advantage of using Transformer encoder on top of spatial features is twofold: (a) it allows processing a complete video in a single pass, and (b) considerably improves training and inference efficiency by avoiding the expensive 3D convolutions [18].

**Table 1.** InceptionV3 Model Architecture

| Type | Patch Size/Stride | Input Size |
|---|---|---|
| Conv | 3 × 3/2 | 299 × 299 3 |
| Conv | 3 × 3/1 | 149 × 149 × 32 |
| Conv padded | 3 × 3/1 | 147 × 147 × 32 |
| Pool | 3 × 3/2 | 147 × 147 × 64 |
| Conv | 3 × 3/1 | 73 × 73 64 |
| Conv | 3 × 3/2 | 71 × 71 × 80 |
| Conv | 3 × 3/1 | 35 × 35 × 192 |
| 3 × Inception | 3 × 3 | 35 × 35 × 288 |
| 5 × Inception | 1 × 3 and 3 × 1 | 17 17 × 768 |
| 2 × Inception | 8 × 8 | 8 × 8 × 1280 |
| Pool | 8 × 8 | 8 × 8 × 2048 |
| Linear | Logits | 1 × 1 × 2048 |
| Softmax | classifier | 1 × 1 × 1000 |

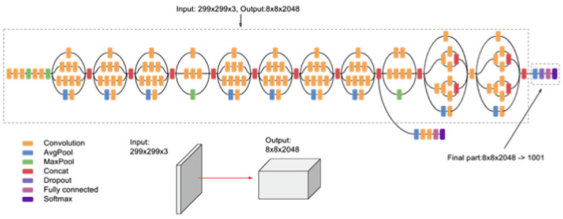



**Fig. 2.** Overview of Inception V3 Model by [17]

The feature maps generated for this model is computed using DenseNets convolution networks. Compared to Inception networks, which also concatenate features from different layers, DenseNets are simpler and more efficient [19]. The architecture requires fewer computations to achieve a promising classification as it is designed with 1 × 1 convolution as a bottleneck layer, reducing a number of feature maps input. Besides, the hyperparameter setting is optimized to reduce the number of hyperparameters.

The Transformer is a model architecture relying entirely on an attention mechanism to draw global dependencies between input and output [20]. The model is built for language modeling, which is similar to video recognition modeling in which the input is words or frames that are represented as a sequence. However, additional processing to maintain the frame sequence is needed because the self-attention layers of the transformer are order-agnostic. The authors in [20] overcome the limitation by introducing positional encoding since the model has no recurrence or convolution to keep the input sequence. Positional encoding vectors are generated using sine and cosine functions of multiple frequencies and summed with the input embeddings in order to inject the positional

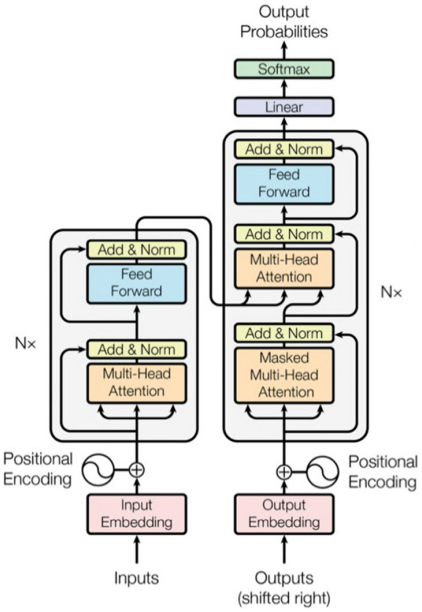



**Fig. 3.** Transformer Model Architecture by [20]

information. Both the main components of transformer, namely, encoder and decoder require positional encoding as the input.

The encoder block consists of a bi-directional self-attention layer, followed by two feed-forward layers with Rectified Linear Unit (ReLU) activation in between to process a long sequence of input vectors. Being an auto-regressive model, the decoder of the Transformer uses previous predictions to output the next word in the sequence [18]. Thus, the decoder has two inputs which are input from encoder and the previous predictions to output the next frame in the sequence. The final classification is made after the output passes through the fully connected layer and software layer. Figure 3 depicts the Transformer model architecture.

## 3 Experimental Setup

The proposed framework is implemented using Keras open source deep learning with TensorFlow backend. The model utilises Nvidia Quadro RTX GPU 16 GB RAM, under Cuda with cuDNN 11.6 on TensorFlow GPU 2.8. Two models are trained for the experiment namely CNN-RNN and Transformer with CNN feature maps. Both model trainings are conducted separately using the same dataset. The model's algorithm is written in Python and run using a Jupyter notebook.

### 3.1 Dataset

The model is trained using a portion of the dataset by [21]. The dataset includes 27 dynamic hand gestures performed thrice by 21 subjects, giving a total of 1701 videos.

The videos are acquired using a 4K HDR video camera with a resolution of 1920 × 1080 at 30 fps. Hence, each video has a total of 120 frames. In particular, the collection is created to motivate the implementation of automatic HGR systems in a wide range of applications namely gaming, multimedia, automotive, and home automation. The dataset is created for general purpose, however, there are gestures identified in the dataset that has Islamic elements. Thus, those gestures will be utilized to represent a request for help to perform Muslim-related actions.

The subjects were carefully instructed before performing the gestures and monitored when performing the gesture. They were prompted to perform each hand gesture with a 3 s sample video of the hand gesture followed by a 3 s countdown. The hand gestures are performed using their right hand and in front of the video camera. In the case of the hand, the gesture was not correct, the subjects had to repeat the movement.

We went through all 27 dynamic gestures in the collections and selected ten gestures that represented basic needs requests from the elderly to their caregivers. Much attention is dedicated to the selection process to accommodate specific elderly needs. For the elderly, the gestures need to be natural, require less learning time, and are easy to remember. Table 2 illustrates the 10 dynamic gestures representing the elderly's common requests.

### 3.2 Model Training and Testing

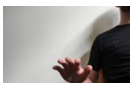
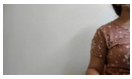
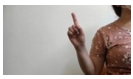
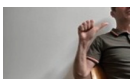
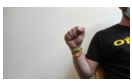
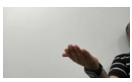
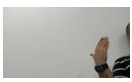
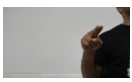
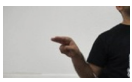
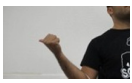
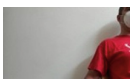
As proposed there are two models built to conduct dynamic hand gesture recognition namely CNN-RNN and Transformer with CNN. The dataset of 630 videos is split as displayed in Table 3.

The first step to data preparation is storing the video path and its true label in a CSV file. However, the video labeling is in string format which is unreadable by the neural networks. Therefore, a numerical representation of the class label is required. In order to feed the model with class labels, the string is converted to numerical form. This is performed by using StringLookup which it encodes the class labels as integers.

The video frames are extracted by using the OpenCV function which is VideoCapture. While reading and storing the frames in an array, the image is resized to an acceptable dimension of 512 × 512. All 120 frames in every video are stored in an array.

The models are trained with the same parameter except for the number of features as it relies on the model input shape. Table 4 shows the parameters set for the model. One key difference is the type of feature extractor of the models. Inception V3 and DenseNets are used to train the CNN-RNN and Transformer-based models, respectively. Based on Figs. 4 and 5, the total parameters for CNN-RNN is significantly smaller than the Transformer model. The total parameter of a model allows us to predict the computational cost to load and train the model. The larger the parameter, the higher random-access memory needs to be allocated. From the model summary, we can extract the input shape required by the model. The original image size of the model is 1980 × 1080 however due to GPU limitations we shrink the frame size to 512 × 512 and trained the model for 100 epochs (Table 5).

**Table 2.** Ten Dynamic Hand Gestures

| Class | | Hand Gestures | |
|---|---|---|---|
| | | Beginning | Ending |
| 01 | Solat |  |  |
| 02 | Quran |  |  |
| 03 | Eat |  |  |
| 04 | Drink |  |  |
| 05 | Medicine |  |  |
| 06 | Sick |  |  |
| 07 | Help |  |  |
| 08 | Toilet |  |  |
| 09 | Go out |  |  |
| 10 | Call |  |  |

## 4 Result and Discussion

The models are able to recognize 10 gestures correctly with an accuracy 89.68% and 70.63% respectively. Transformer-based model produced a higher accuracy value compared to CNN-RNN models. Out of 126 testing videos, 60 videos are chosen randomly to make predictions. Tables 6 and 7 are the recognition result of three videos from the testing set.

**Table 3.** Train-Test Split

| Dataset | Number of video |
|---|---|
| Training | 504 |
| Testing | 126 |

**Table 4.** Model parameters

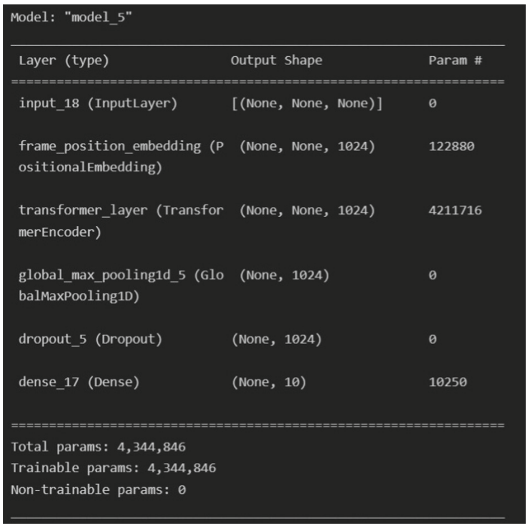
| Model | CNN-RNN | Transformer with CNN |
|---|---|---|
| Max Sequence Length | 120 | 120 |
| Number of Features | 2048 | 1024 |
| Image Size | 512 | 512 |
| Epochs | 100 | 100 |

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=================================================================
 input_3 (InputLayer)        [(None, 120, 2048)]       0         []

 input_4 (InputLayer)        [(None, 120)]             0         []

 gru (GRU)                   (None, 120, 16)           99168     ['input_3[0][0]',
                                                                  'input_4[0][0]']

 gru_1 (GRU)                 (None, 8)                 624       ['gru[0][0]']

 dropout (Dropout)           (None, 8)                 0         ['gru_1[0][0]']

 dense (Dense)               (None, 8)                 72        ['dropout[0][0]']

 dense_1 (Dense)             (None, 10)                90        ['dense[0][0]']

=================================================================
Total params: 99,954
Trainable params: 99,954
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 4.** CNN-RNN model summary

The tables show a large margin in the accuracy of recognition and some of the results have significantly low accuracy. We inspect each of the testing video and analyse its accuracy result. We found out that videos with following situations, has low accuracy:

a. The arm is not fully visible in the video
b. The movement of gesture is not obvious
c. The movement of the gesture is too fast



```
Model: "model_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_18 (InputLayer)       [(None, None, None)]      0

 frame_position_embedding (P  (None, None, 1024)       122880
 ositionalEmbedding)

 transformer_layer (Transfor  (None, None, 1024)       4211716
 merEncoder)

 global_max_pooling1d_5 (Glo  (None, 1024)             0
 balMaxPooling1D)

 dropout_5 (Dropout)         (None, 1024)              0

 dense_17 (Dense)            (None, 10)                10250

=================================================================
Total params: 4,344,846
Trainable params: 4,344,846
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 5.** Transformer with CNN model summary

**Table 5.** Model Training and Testing Result

| Model | CNN-RNN | Transformer with CNN |
|---|---|---|
| Accuracy | 70.63% | 89.68% |
| Feature Extraction Time | 103 min<br>27.8 s | 203 min<br>43.7 s |
| Training Time | 1 min<br>5.1 s | 2 min<br>4.4 s |

The learning curves for Transformer in Fig. 7 decrease to the point of stability which resulted in a good fit model. On the other hand, Fig. 6 depicts an overfit CNN-RNN model. The model has learned the dataset too well in the training set producing a linear graph for both accuracy and loss graph. To improve the model performance, data augmentation such as flipping, rotation, and scaling can be applied to increase the size of the dataset. In addition, regulation techniques namely dropout able to drop a set of neurons during training in each iteration.

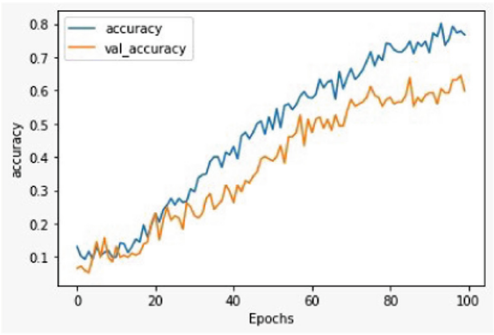
**Table 6.** Recognition Result of Transformer

| Class | | Video 1 (%) | Video 2 (%) | Video 3 (%) |
|---|---|---|---|---|
| 01 | Solat | 99.99 | 99.99 | 100 |
| 02 | Quran | 99.99 | 99.99 | 99.99 |
| 03 | Eat | 100 | 99.99 | 99.99 |
| 04 | Drink | 31.01 | 99.99 | 99.99 |
| 05 | Medicine | 100 | 100 | 100 |
| 06 | Sick | 47.56 | 92.78 | 96.31 |
| 07 | Help | 98.56 | 99.13 | 99.99 |
| 08 | Toilet | 99.99 | 99.99 | 100 |
| 09 | Go out | 99.99 | 100 | 100 |
| 10 | Call | 99.99 | 100 | 100 |

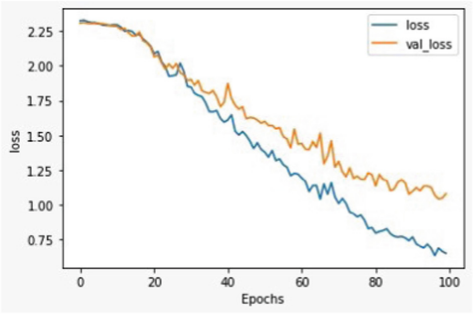
**Table 7.** Recognition Result of CNN-RNN

| Class | | Video 1 (%) | Video 2 (%) | Video 3 (%) |
|---|---|---|---|---|
| 01 | Solat | 19.30 | 61.92 | 80.00 |
| 02 | Quran | 1.45 | 92.28 | 92.77 |
| 03 | Eat | 2.67 | 14.34 | 31.57 |
| 04 | Drink | 4.37 | 27.41 | 76.05 |
| 05 | Medicine | 79.03 | 81.16 | 82.45 |
| 06 | Sick | 52.92 | 55.19 | 66.29 |
| 07 | Help | 48.83 | 60.36 | 71.81 |
| 08 | Toilet | 99.99 | 99.99 | 100 |
| 09 | Go out | 5.97 | 63.52 | 72.49 |
| 10 | Call | 92.22 | 96.25 | 96.83 |

The similarity of the models is the spatial learning process where both take advantage of a pre-computed convolution network to extract the frame's features. However, the types of pre-trained CNN models are different, in which CNN-RNN used Inception V3 while the Transformer-based network adopted DenseNets architecture.




(a)

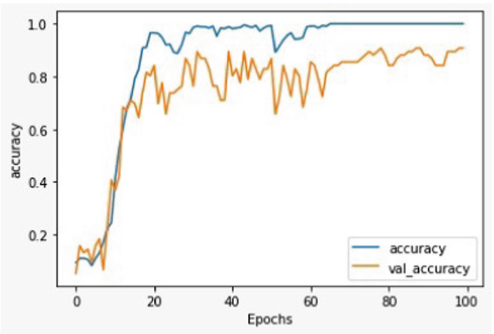



(b)

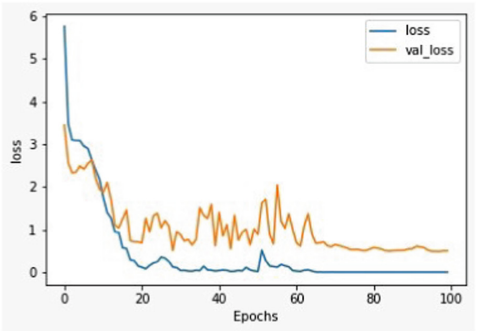
**Fig. 6.** CNN-RNN Learning Curves

## 5 Conclusion

In this paper, we successfully infused Islamic gestures into the dynamic HGR models which are CNN-RNN and Transformer with CNN models. Based on the author's knowledge the dataset has not yet been used in HGR in the application the elderly care. This paper also has contributed to the study on HGR for the Muslim community by adopting praying and reading the Quran gestures into the models. In future work, we aim to improve the CNN-RNN model, study the impact of different types of feature extractors in hand gesture recognition and adopt additional Muslim hand gestures into the models.




(a)




(b)

**Fig. 7.** Transformer with CNN Learning Curves

**Acknowledgments.** The authors would like to acknowledge University of Multimedia for the financial support of the project.

**Authors' Contributions.** Study conception and design, analysis and interpretation of results: Hadya Ayeisha Marzuki; Noramiza Hashim

Data collection, draft manuscript preparation: Hadya Ayeisha Marzuki; Noramiza Hashim

All authors reviewed the results and approved the final version of the manuscript.

## References

1. M. M. Islam, S. Siddiqua, and J. Afnan, "Real time Hand Gesture Recognition using different algorithms based on American Sign Language," Mar. 2017. doi: https://doi.org/10.1109/ICIVPR.2017.7890854.
2. S. Sharma and S. Singh, "Vision-Based Hand Gesture Recognition Using Deep Learning for the Interpretation of Sign Language," *Expert Systems with Applications*, vol. 182, p. 115657, Nov. 2021, doi: https://doi.org/10.1016/J.ESWA.2021.115657.

3. K. Guan, M. Shao, and S. Wu, "A Remote Health Monitoring System for the Elderly Based on Smart Home Gateway," *Journal of Healthcare Engineering*, vol. 2017, 2017, doi: https://doi.org/10.1155/2017/5843504.
4. N. Chen, J. Song, and B. Li, "Providing Aging Adults Social Robots' Companionship in Home-Based Elder Care," *Journal of Healthcare Engineering*, vol. 2019, 2019, doi: https://doi.org/10.1155/2019/2726837.
5. L. Zhang, G. Zhu, L. Mei, P. Shen, S. A. A. Shah, and M. Bennamoun, "Attention in Convolutional LSTM for Gesture Recognition," in *Advances in Neural Information Processing Systems*, 2018, vol. 31. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/287e03db1d99e0ec2edb90d079e142f3-Paper.pdf
6. C. Xi, J. Chen, C. Zhao, Q. Pei, and L. Liu, "Real-Time Hand Tracking Using Kinect," in *Proceedings of the 2nd International Conference on Digital Signal Processing*, 2018, pp. 37–42. doi: https://doi.org/10.1145/3193025.3193056.
7. M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gestures for Elderly Care Using A Microsoft Kinect," *Nano Biomedicine and Engineering*, vol. 12, no. 3, pp. 197–204, 2020, doi: https://doi.org/10.5101/nbe.v12i3.p197-204.
8. M. Oudah, A. Al-Naji, and J. Chahl, "Elderly Care Based on Hand Gestures Using Kinect Sensor," *Computers*, vol. 10, no. 1, pp. 1–25, 2021, doi: https://doi.org/10.3390/computers10010005.
9. Chengfeng Jian and Jianing Li, "Real-time multi- trajectory matching for dynamic hand gesture recognition _ Enhanced Reader," *IET Image Process*, vol. 14, no. 2, pp. 236–244, 2020.
10. C. Jian and J. Li, "IET Image Processing Real-time Multi-Trajectory Matching for Dynamic Hand Gesture Recognition," 2019, doi: https://doi.org/10.1049/iet-ipr.2019.1068.
11. K. B. Prakash, "Accurate Hand Gesture Recognition using CNN and RNN Approaches," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3216–3222, Jun. 2020, doi: https://doi.org/10.30534/IJATCSE/2020/114932020.
12. Sayak Paul, "Video Classification with a CNN-RNN Architecture," 2021. https://keras.io/examples/vision/video_classification/ (accessed Apr. 14, 2022).
13. D. Neimark, O. Bar, M. Z. Dotan, and A. Theator, "Video Transformer Network," 2021. [Online]. Available: https://github.com/bomri/SlowFast/blob/
14. R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video Action Transformer Network." [Online]. Available: http://rohitgirdhar.github.io/ActionTransformer
15. A. D'Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "A Transformer-Based Network for Dynamic Hand Gesture Recognition," *Proceedings - 2020 International Conference on 3D Vision, 3DV 2020*, pp. 623–632, Nov. 2020, doi: https://doi.org/10.1109/3DV50981.2020.00072.
16. C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "InceptionV2/BN-Inception," 2015.
17. C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the Inception Architecture for Computer Vision," 2015.
18. S. Khan, A. Dhabi, M. Naseer, S. Waqas Zamir, and F. Shahbaz Khan, "Transformers in Vision: A Survey," 2022, doi: https://doi.org/10.1145/3505244.

19. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Nov. 2017, doi: https://doi.org/10.1109/CVPR.2017.243.
20. A. Vaswani *et al.*, "Attention Is All You Need," 2017.
21. G. Fronteddu, S. Porcu, A. Floris, and L. Atzori, "A dynamic hand gesture recognition dataset for human-computer interfaces," *Computer Networks*, vol. 205, p. 108781, Mar. 2022, doi: https://doi.org/10.1016/J.COMNET.2022.108781.