Two-Branch 3DCNN-LSTM to Extract Spatial and Temporal Feature Separately for Dynamic Hand Swipe Gesture Recognition

Muhammad Ilham Perdanaa), Mochamad Rahina Bintang Pambayunb), and Irtafa Masruric), Diah Risqiwatid)

Department of Informatics, Universitas Muhammadiyah Malang, Malang, Indonesia

d) Corresponding author: [risqiwati@umm.ac.id](mailto:yos.sumantri@upnyk.ac.id)

a) ilhamperdana@umm.ac.id

b) rahinabintang@webmail.umm.ac.id

c) irtafamasruri@umm.ac.id

**Abstract.**  Recognizing hand gestures is important in human-computer interaction. The primary challenge in recognizing hand gestures lies in their dynamic nature, which constantly changes in both shape and direction over time. A system should be able to extract spatial and temporal information from each sequence of gestures. We proposed a two-branch model that separately extracts spatial and temporal features. The spatial feature was extracted using the 3DCNN model, while the temporal feature was extracted using the LSTM model. The output from each model is subsequently concatenated. This concatenated data is then used for hand gesture recognition. The hand gestures targeted for recognition in this study include three categories, No Gesture, Swipe Left, and Swipe Right. The results obtained from this experiment show promising results, with 0.99 train accuracy, 0.98 validation accuracy, and 0.98 test accuracy. Additionally, we compared our proposed model with another model, specifically those using only 3DCNN and using only the LSTM model. Our proposed method achieved the best results, indicating that the strategy of separately extracting spatial and temporal features using 3DCNN and LSTM is effective.

**Keywords:** Dynamic Hand Gesture Recognition, 3DCNN, LSTM, Two-Branch

# INTRODUCTION

Recent trends in the development of hand gesture recognition have rapidly increased [1], especially in the field of human-machine interaction [2]. Many practical implementations can be built with hand gesture recognition, such as sign language [3], Virtual Reality [4], hand gesture to controll things [5, 6], assistance in healthcare [7] , and industry [8]. Interpreting hand gestures poses significant challenges due to their dynamic nature, as they continually change over time. The system should be able to extract spatial and temporal features. In our context, spatial means the shape of a hand, while temporal implies the movement of a hand before and after the events [9]. Numerous previous works have been done to develop hand gesture recognition systems. These approaches fall into two categories based on how to obtain the spatial and temporal data, utilizing vision-based data and utilizing sensor-based data.

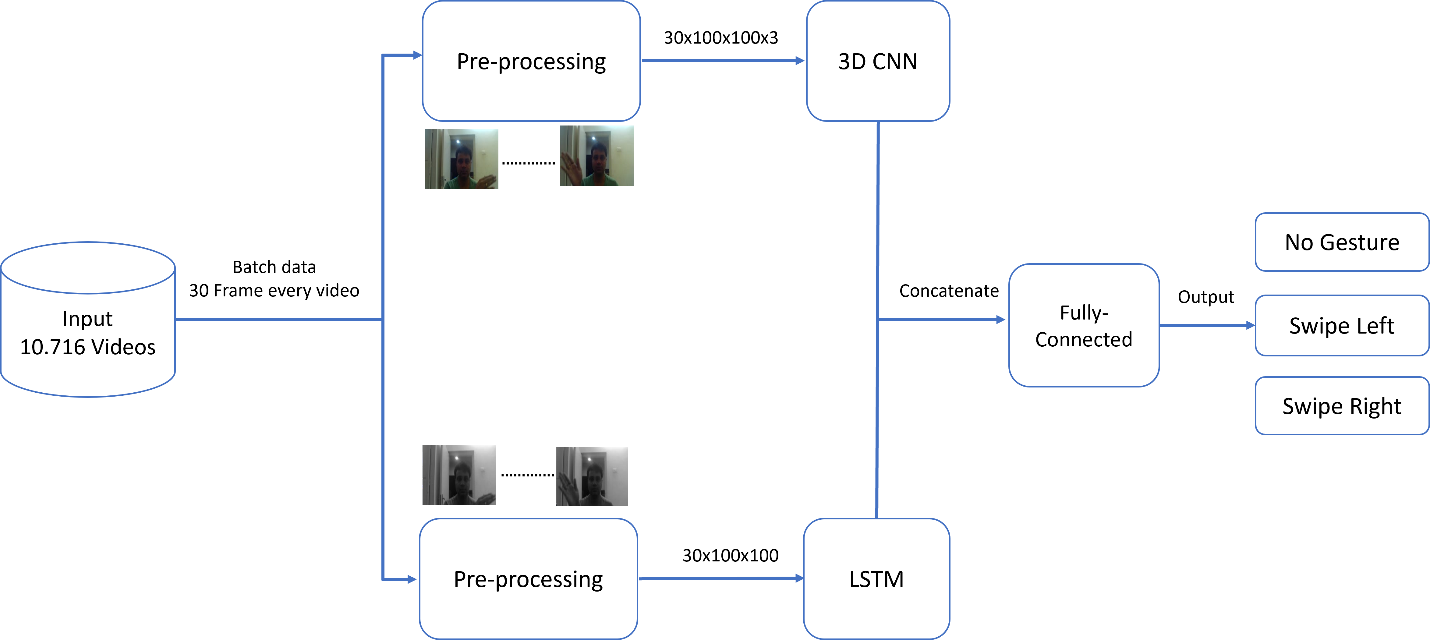
Sensor-based systems can provide precise hand motion tracking data that is less affected by environmental factors such as lighting conditions, offering robustness in various settings. One example of a sensor that can be employed to capture hand movement features to recognize hand gestures is the wearable electromyography (EMG) sensor [10]. Eight points EMG sensors were placed on the wrist and forearm, both locations have 4 sensors. The raw signal then was filtered using a third-order Butterworth high pass filter to remove any low-frequency motion artifacts and applying an Infinite Impulse Response (IIR) notch filter with a Q-factor of 50 to remove noise. Bostros uses Several classification algorithms to classify 5 gestures, Thumb Extension, Index Extension, Pinch Grip, Wrist Extension, and Wrist Flexion [10]. Another sensor type using Frequency-Modulated Continuous Wave (FMCW) radar was also proposed to recognize hand gestures [11]. The signal from FMCW radar was then fed into LSTM to recognize 10 different gestures with a high accuracy of 99.10%. The drawback of using sensors for hand gesture detection is their requirement for additional hardware, which can be invasive. That means users need to wear the sensor devices, potentially causing discomfort and limiting the user's movement.

To achieve greater flexibility and avoid invasiveness, vision-based approaches can be employed. Vision-based systems benefit from rich spatial information. There are two main approaches from vision-based, using RGB camera and RGB-D Camera. By utilizing the depth feature from the RGB-D camera, hand gestures can be accurately and easily interpreted. Numerous previous studies have been conducted on hand gesture recognition using RGB-D cameras with different approaches and strategies [12-15]. The primary challenge in building a hand gesture recognition system occurs when using only RGB cameras. Without the depth feature, the system should be able to recognize hand gestures. Extracting spatial features from videos from RGB cameras can be challenging due to the limited information available compared to RGB-D cameras. Several studies have been done to build a model that is able to extract spatial features. 3DCNN has proven to be effective in performing spatial feature extraction from video data [16-18] . For temporal features, several approaches have also proven to be effective, such as LSTM [19], GRU [20], attention mechanism [21].

In this experiment, our objective is to develop a system capable of recognizing hand-swipe gestures using only RGB camera data. We propose a method to separately extract spatial and temporal features using different models of each branch. For spatial features, we employ the 3DCNN model, while for temporal features we employ LSTM. Output from each branch model is concatenated into one feature to recognize hand gestures in a specific frame. The gestures we aim to recognize include No Gesture, Swipe Left, and Swipe Right. We also compared our approach with models that utilize only 3DCNN and utilize only LSTM.

# Methods

This section outlined our proposed system for building an automatic dynamic hand swipe gesture recognition, which combines Three-Dimensional Convolutional Neural Network (3DCNN) and Long-Short Term Memory (LSTM) to extract spatial and temporal features separately. We propose a system that employs separate branches of 3DCNN and LSTM networks to independently extract spatial and temporal features of dynamic hand-swipe gestures. The illustration of the model can be seen in **FIGURE 1.**



**FIGURE 1**. Block diagram of proposed method

The proposed method is organized into four components, which first describe the dataset used for training and validation, followed by an in-depth explanation of the individual 3DCNN and LSTM branches. Finally, we discuss how the outputs of these branches are combined to form a unified representation for hand swipe gesture recognition.

## DATASET

Recognizing hand-swipe gestures requires a sequential frame. It is hard to recognize what gesture they make if we only look at one single frame image. Several public datasets can be used to build a hand gesture recognition model. In this experiment, we use the 20BN-jester Dataset [22] which consists of 20 label hand gestures. The 20BN-Jester dataset contains 148,092 annotated video clips, each depicting different individuals performing a variety of dynamic hand gestures. Each gesture in the 20BN-Jester dataset has approximately 5000 videos and each video contains an average total of 36 frames, which is separated into training, validation, and test. Our experiment only focuses on building hand swipe recognition only, thus we only use three labels. The three labels that we use are No Gesture, Swipe Left, and Swipe Right. The sample of three gestures can be seen in **FIGURE 2**. Due to the inconsistent total number of frames in each video, we only use videos that contain a total of at least 30 frames. The total dataset that we use is 10.716 for training, 1.315 for validation, and 1.303 for testing.



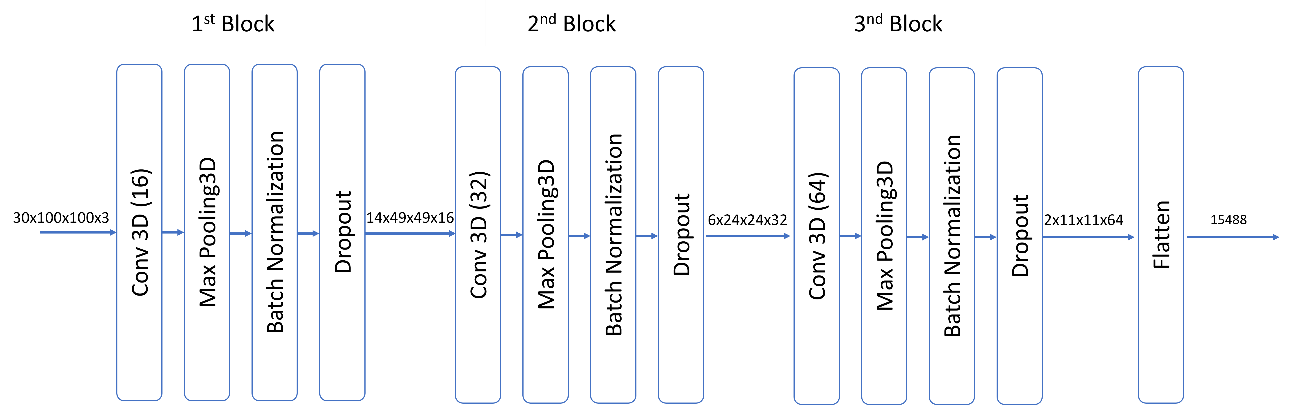
**FIGURE 2.** Sample of 20bn-Jester Dataset. (a) No Gesture (b) Swipe Left (c) Swipe Right

## PREPROCESSING DATA

In this subsection, we outline the preprocessing steps applied to the raw video data. Based on **FIGURE 1**, each branch has a different preprocessing technique. First, on both branches, we resize the frame into 100x100 of each video. The purpose of resizing the frame is to reduce the complexity of the data. Next, in the first branch, we retain the RGB format of the video frames. This preservation of the color information is crucial for the 3DCNN to accurately capture the spatial features of the hand gestures. The RGB frames provide rich information about the texture and color variations that are essential for distinguishing different gestures. However, in the second branch, we convert the frames to grayscale. This transformation aims to further reduce the data complexity, thereby facilitating the LSTM network in learning the temporal patterns of the hand gesture sequences more efficiently. By eliminating the color information, the LSTM can focus on the essential shape and motion patterns within each sequence, enhancing its ability to model the temporal dependencies and variations inherent in dynamic hand gestures. The final preprocessing step involves normalizing the images. This step is crucial for both branches as it simplifies the learning process for each model by scaling the pixel values to a range between 0 and 1. Normalization ensures that the data fed into the models has a consistent scale, which aids in faster convergence during training and helps in mitigating issues related to varying pixel intensities. This step further improves the models’ ability to learn relevant features effectively and contributes to the overall robustness and accuracy of the gesture recognition system. These preprocessing steps ensure that each branch of our system is optimized for its specific task, with the 3DCNN effectively handling spatial feature extraction and the LSTM adeptly capturing temporal dynamics. The distinct preprocessing techniques for each branch are integral to the overall performance of our proposed gesture recognition system.

## 3DCNN FOR EXTRACTING SPATIAL FEATURE

3D Convolutional Neural Networks (3DCNN) have been demonstrated to effectively extract spatial features from data [23] particularly in the specific context of dynamic gesture data [24]. In this section, we provide an in-depth description of our 3D Convolutional Neural Network architecture to extract spatial features from video frames. The illustration of our proposed 3DCNN network can be seen in **FIGURE 3**, which is described as follows:



**FIGURE 3**. Structure Network of Our 3DCNN Branch

The 3D Convolutional Neural Network (3DCNN) is the core component of our proposed architecture. As illustrated in **FIGURE 3**, our 3DCNN comprises three 3D convolutional (3DConv) layers. In the initial 3DConv layer, we implement 16 feature maps. The second 3DConv layer expands to 32 feature maps and the final 3DConv layer further increases to 64 feature maps. Each 3DConv layer employs a kernel size of 3x3x3. The output of each 3DConv layer is passed through the Rectified Linear Unit (ReLU) activation function to introduce non-linearity and enhance the model’s ability to learn complex patterns.

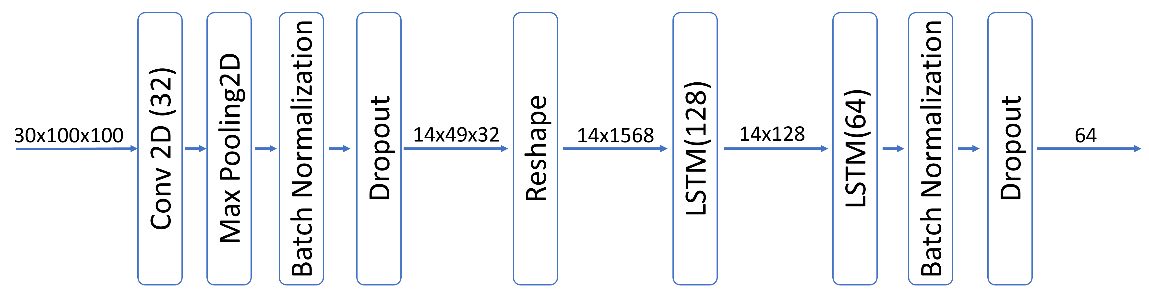
Outputs from each 3DCNN layer are subsequently processed by a 3D MaxPooling layer. The the primary function of 3D MaxPooling is to reduce the spatial dimensions of the feature maps, thereby decreasing the computational requirements. In our architecture, each 3D MaxPooling layer employs a pooling size of 2x2x2. This pooling operation consolidates similar neighboring pixels, effectively combining pixels that have similar values while reducing the overall size of the feature maps.

The Batch Normalization and Dropout layers are employed to reduce overfitting in our model. Batch Normalization is used to stabilize and accelerate the training process by normalizing the outputs of each layer. Then, we apply dropout after the Batch Normalization layer. Dropout is a regularization technique to prevent overfitting. In our model, a dropout value of 0.3 is utilized for the outputs of each 3D MaxPooling layer. This dropout value means that 30% of the features are randomly dropped during training, which helps to enhance model generalization and reduce the risk of overfitting.

Flatten is the last layer. This layer works to flatten the output into 1-dimensional data. We need the Flatten layer to change the size of our model before concatenating with the LSTM branch output.

## LSTM FOR EXTRACTING TEMPORAL FEATURE

The second branch of our model is an LSTM network. This network has proven to be very effective in extracting temporal features [25], especially in the context of dynamic gesture data [13]. This section explains our LSTM network to extract temporal features from sequences of video frames. The illustration of our proposed LSTM network can be seen in **FIGURE 4**, which is described as follows:



**FIGURE 4**. Structure Network of Our LSTM Branch

Given that the input to the second branch consists of grayscale images, which have only two dimensions for each image, a 2D Convolutional Neural Network (2DCNN) is employed. The purpose of utilizing this layer is to extract two-dimensional features from the grayscale images and to reduce the number of parameters before feeding the data into the LSTM network. The 2D convolutional layer is configured with 32 feature maps and employs a kernel size of 3x3.

Similar to the 3DCNN branch, the 2DMaxPooling layer is used to reduce the number of features by concluding the values of neighboring pixels with similar characteristics. This process also aims to decrease computational costs. The pooling size for this layer is 2x2.

LSTM is the main layer of this branch. The purpose is to learn temporal features from each sequence frame of dynamic hand gesture. Before the data is fed to the LSTM network, output from 2DMaxPooling is reshaped into two dimensions. The first index of the data is the sequence and the second index of the data is the feature generated from 2DCNN and 2DMaxPooling. As can be seen in 4, there are two LSTM layers. The first LSTM layer has 128 feature maps, producing an output with dimensions of 14x128. The second LSTM layer comprises 64 units, which generate features with dimensions of 64.

Similar to the 3DCNN branch, Our LSTM Branch also employs Batch Normalization and Dropout layers. The Batch Normalization and Dropout Layers are applied after the Conv2D layer and second LSTM layer. But, the value of each dropout rate is different. The first dropout rate is 0.3 and the second dropout rate is 0.5

## CONCATENATE 3DCNN AND LSTM OUPUT TO RECOGNIZE HAND GESTURE

The output from each branch is concatenated to form a single feature. As can be seen in **FIGURE 1**, the concatenated feature is fed into the connected layer. There are two Fully-Connected layers, The First layer has 128 feature maps and the second layer has 3 features that map into 3 outputs. The output from the last Fully-Connected Layer is the value to recognize No Gesture, Swipe Left, and Swipe Right.

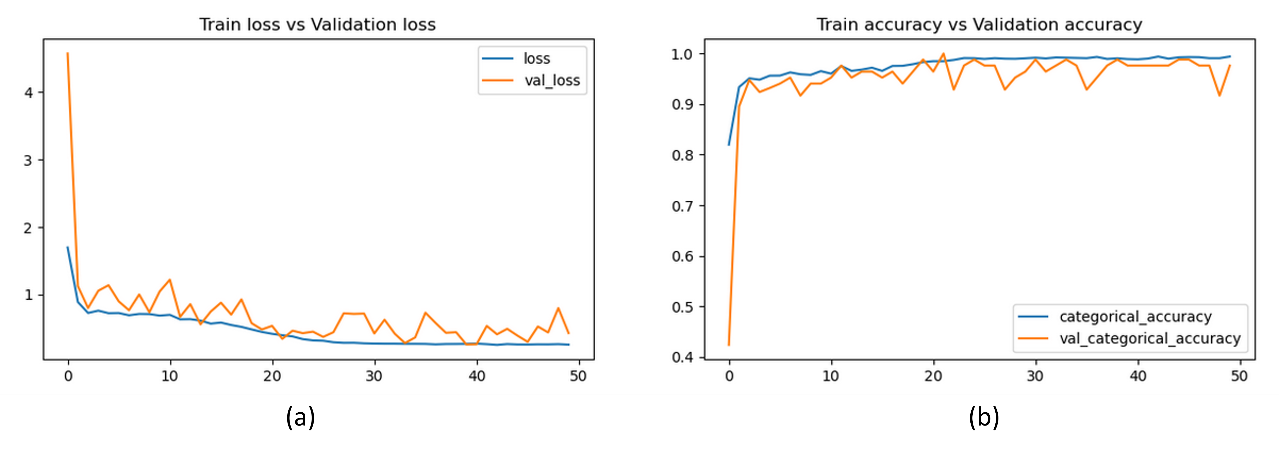
# Result and discussion

This section explains the scenario of our experiment. For building our model, a pc with AMD Ryzen 5 with 16GB RAM was used for computation. We build our two-branch 3DCNN-LSTM model using tensorflow-keras framework to recognize dynamic hand swipe gesture.

## TRAINING RESULT AND DISCUSSION

The training process was conducted over 50 epochs. The training results for each epoch are illustrated in **FIGURE 5**. This figure displays the progression of accuracy and loss for both the training and validation datasets throughout the epochs. After 50 epochs, the model achieved satisfactory performance on both the training and validation sets. Specifically, the training accuracy reached 0.99, while the validation accuracy was 0.98. The training loss was 0.25, and the validation loss was 0.38. Overall, the training process did not show signs of overfitting, as indicated by the minimal difference between the training and validation results.

To ensure the quality of our model, we also run our model on a data test set. We use precision, recall, f1-score, and average accuracy to measure the performance of the model. The result can be seen in **TABLE 1**. As we can see, the average accuracy in the data test is 0.98. The precision value for No Gesture is 0.97, while Swipe Left and Swipe Right have the same value 0.98. The recall value of each label is 0.97 for No Gesture and 0.97 for both Swipe Left and Swipe Right. The F1 score of each label is 0.99 for No Gesture and 0.97 for both Swipe Left and Swipe Right. From these results, it can be concluded that the model demonstrates good performance. Which also does not indicate signs of overfitting. All these results are supported by a confusion matrix that can be seen in **TABLE 2**.



**FIGURE 5**. Proposed Model Accuracy and Loss Performance (a) Loss Value in Each Epoch (b) Accuracy Value in Each Epoch

**TABLE 1**. Performance of Our Model on Data Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1 Score | Accuracy |
| No Gesture | 0.97 | 1.0 | 0.99 | 0.98 |
| Swipe Left | 0.98 | 0.97 | 0.97 |
| Swipe Right | 0.98 | 0.97 | 0.97 |

**TABLE 2**. Confussion Matrix of Hand Swipe Gesture Recognition

|  |  |  |  |
| --- | --- | --- | --- |
|  | No Gesture | Swipe Left | Swipe Right |
| No Gesture | 439 | 1 | 0 |
| Swipe Left | 4 | 416 | 8 |
| Swipe Right | 13 | 12 | 421 |

## COMPARING WITH OTHER MODEL

In this section, we conduct experiments to compare the performance of different model configurations: a model  
using only 3DCNN network and a model using only LSTM network. The comparison result can be seen in **TABLE 3**.

**TABLE 3**. Comparison Performance with 3DCNN Only and LSTM Only

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train Acc | Train Loss | Val Acc | Val Loss |
| 3DCNN | 0.98 | 0.35 | 0.92 | 0.79 |
| LSTM | 0.88 | 0.46 | 0.82 | 0.477 |
| Proposed Method | 0.99 | 0.25 | 0.98 | 0.38 |

We saw 3DCNN Only model has a high train accuracy of 0.98, but validation acc only 0.92. That result indicates signs of overfitting. LSTM Only model has a low train accuracy of 0.88 and low validation accuracy of 0.82. Our proposed system achieves the best result, indicating that the strategy to separately extract spatial and temporal features using the 3DCNN and LSTM model is an optimal approach.

# conclusion

This study aims to develop a model capable of recognizing dynamic hand-swipe gestures. Our strategy involves separately extracting spatial and temporal features, using 3DCNN for spatial features and LSTM for temporal features. This approach demonstrated excellent results, achieving an average accuracy of 0.98 on the test data, utilizing the 20BN-Jester dataset with three labels: No Gesture, Swipe Left, and Swipe Right. We compared our approach with alternative strategies that employ only the 3DCNN model and only the LSTM model. The comparison results show that our approach is optimal.

In the future, we aim to expand our research to include more hand gesture recognition, not only swipe gestures. Additionally, as our research direction is toward implementation in interactive multimedia, we plan to focus on developing a lightweight model that does not require high computational resources.

# References

1. Mohamed, N., M.B. Mustafa, and N. Jomhari, *A Review of the Hand Gesture Recognition System: Current Progress and Future Directions.* IEEE Access, 2021. **9**: p. 157422-157436.

2. Vuletic, T., et al., *Systematic literature review of hand gestures used in human computer interaction interfaces.* International Journal of Human-Computer Studies, 2019. **129**: p. 74-94.

3. Gaikwad, S.A., D. Upasani, and V. Shete. *Review and Trends on Hand Gesture Recognition of Sign Language based on Deep Learning Approaches*. in *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. 2023.

4. Wang, Z., et al., *A study on hand gesture recognition algorithm realized with the aid of efficient feature extraction method and convolution neural networks: design and its application to VR environment.* Soft Computing, 2023: p. 1-24.

5. Mahmood, M., et al. *Design of a low-cost Hand Gesture Controlled Automated Wheelchair*. in *2020 IEEE Region 10 Symposium (TENSYMP)*. 2020.

6. Xia, Z. and F. Xu, *Time-Space Dimension Reduction of Millimeter-Wave Radar Point-Clouds for Smart-Home Hand-Gesture Recognition.* IEEE Sensors Journal, 2022. **22**(5): p. 4425-4437.

7. Alonazi, M., et al., *Smart Healthcare Hand Gesture Recognition Using CNN-Based Detector and Deep Belief Network.* IEEE Access, 2023. **11**: p. 84922-84933.

8. Dumoulin, W., N. Thiry, and R. Slama, *Real Time Hand Gesture Recognition in Industry*, in *Proceedings of the 2021 3rd International Conference on Video, Signal and Image Processing*. 2022, Association for Computing Machinery: Wuhan, China. p. 1–7.

9. Amsaprabhaa, M., Y. Nancy Jane, and H. Khanna Nehemiah, *A survey on spatio-temporal framework for kinematic gait analysis in RGB videos.* Journal of Visual Communication and Image Representation, 2021. **79**: p. 103218.

10. Botros, F.S., A. Phinyomark, and E.J. Scheme, *Day-to-Day Stability of Wrist EMG for Wearable-Based Hand Gesture Recognition.* IEEE Access, 2022. **10**: p. 125942-125954.

11. Choi, J.W., S.J. Ryu, and J.H. Kim, *Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder.* IEEE Access, 2019. **7**: p. 33610-33618.

12. Xu, J., et al., *Robust Hand Gesture Recognition Based on RGB-D Data for Natural Human–Computer Interaction.* IEEE Access, 2022. **10**: p. 54549-54562.

13. D. K. N, C., et al. *SIT-HANDS: A Novel RGB-D Dataset for Static Hand Gesture Recognition Using Kinect Sensor*. in *2023 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*. 2023.

14. Gao, Q., et al., *An Efficient RGB-D Hand Gesture Detection Framework for Dexterous Robot Hand-Arm Teleoperation System.* IEEE Transactions on Human-Machine Systems, 2023. **53**(1): p. 13-23.

15. Tran, D.-S., et al. *Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network*. Applied Sciences, 2020. **10**, 722 DOI: 10.3390/app10020722.

16. Sharma, S. and K. Kumar, *ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks.* Multimedia Tools and Applications, 2021. **80**(17): p. 26319-26331.

17. Vrskova, R., et al. *Human Activity Classification Using the 3DCNN Architecture*. Applied Sciences, 2022. **12**, DOI: 10.3390/app12020931.

18. Zhang, G., et al. *Gesture Recognition Algorithm Based on Lightweight 3DCNN Network*. in *2021 17th International Conference on Computational Intelligence and Security (CIS)*. 2021.

19. Liu, X. and J. Zhou, *Short-term wind power forecasting based on multivariate/multi-step LSTM with temporal feature attention mechanism.* Applied Soft Computing, 2024. **150**: p. 111050.

20. Zhou, Z., X. Yu, and X. Chen *Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer*. Drones, 2023. **7**, DOI: 10.3390/drones7070466.

21. Chang, Q. and S. Zhu, *Human Vision Attention Mechanism-Inspired Temporal-Spatial Feature Pyramid for Video Saliency Detection.* Cognitive Computation, 2023. **15**(3): p. 856-868.

22. Materzynska, J., et al. *The jester dataset: A large-scale video dataset of human gestures*.

23. Tran, D., et al. *Learning spatiotemporal features with 3d convolutional networks*.

24. Singh, D.K., *3D-CNN based Dynamic Gesture Recognition for Indian Sign Language Modeling.* Procedia Computer Science, 2021. **189**: p. 76-83.

25. Tang, Y., et al., *Semi-supervised LSTM with historical feature fusion attention for temporal sequence dynamic modeling in industrial processes.* Engineering Applications of Artificial Intelligence, 2023. **117**: p. 105547.