Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SIGN LANGUAGE RECOGNITION ON VIDEO DATA BASED ON GRAPH CONVOLUTIONAL NETWORK

[1]AYAS FAIKAR NAFIS, [2]NANIK SUCIATI

[1,2]Institut Teknologi Sepuluh Nopember, Department of Informatics, Surabaya, Indonesia

E-mail: [1]ayas.206025@mhs.its.ac.id, [2]nanik@if.its.ac.id

ABSTRACT

Sign language is a very important means of communication for the deaf and the mute. Therefore, it is necessary to automatically recognize sign language by a computer so that non-disabled people can understand the sign language that is used. Many studies on sign language recognition have been carried out, one of which is the sign language alphabet recognition using the Convolutional Neural Network (CNN). However, CNN cannot represent a skeletal data structure that has the graph form. The Graph Convolutional Network (GCN) is a generalization of CNN that can perform feature extraction from graphs in non-Euclidean space. GCN is widely used in action recognition research such as the Shift-GCN method. This study used hand joints position estimated by MediaPipe Hands that shaped like a graph. The graph is processed using the modified Shift-GCN that introduces a shift weighting approach based on the vertices adjacency. The dataset used in this study is hand keypoints extracted from video data of 26 American Sign Language (ASL) alphabets. Based on the experimental results, the proposed method achieved the best accuracy of 99.962%.

Keywords: *Sign Language, Alphabet Recognition, Graph Convolutional Network, Shift-GCN, Skeletal Data*

1. INTRODUCTION

Sign language is a non-verbal language that can be expressed through hand movements, body movements, or facial expressions so that sign language is used as a very important means of communication for the deaf and the mute [1]. An automatic sign language recognition system is useful for facilitating communication between persons with disabilities and non-disabled persons.

Many studies on sign language recognition have been carried out. The type of sign language and the type of data that are the subject of the study distinguish the proposed approach. Based on the type of sign language used, some studies limit recognition problems to only static sign language, while others recognize both static and dynamic sign languages. The types of data used in previous studies are image, video, or skeletal data. Many approaches based on deep learning with Convolutional Neural Network (CNN) have been proposed for sign language recognition on images or video data. Adithya et al proposed CNN-based static hand gesture recognition with the National University of Singapore (NUS) hand posture and American fingerspelling A dataset that can recognize 24 static American Sign Language (ASL) alphabets by ignoring dynamic ASL letters such as the letters 'J' and 'Z' [2]. Aljabar et al have also researched BISINDO sign language recognition with a dataset consisting of two static alphabets and eight words that have movement using CNN for feature extraction and LSTM to overcome words that have movement [3]. However, CNN has a weakness, namely that it cannot represent a skeletal data structure that has a graph form, not a vector sequence or a two-dimensional grid [4]. The skeleton and joints of the human body are features that are not affected by changes in lighting and background variations and are easily obtained by depth sensors or pose estimation algorithms [5].

Graph Convolutional Network (GCN) is a method that can generalize CNN so that feature extraction from graphs in non-Euclidean space can be carried out [5]. The skeleton-based action recognition is one area of research that uses a lot of GCN methods such as Spatial-Temporal Graph Convolutional Networks (ST-GCN) [5] proposed by Yan et al. ST-GCN is widely used as the basis for research on activity recognition based on skeletal data such as the MS-AAGCN (Multi-Stream Attention-enhanced Adaptive GCN) proposed by Shi et al [4], hand gesture recognition such as the proposed HG-GCN (Hand Gesture GCN) by Li et al [6], and skeleton-based sign language recognition such as ST-GCN SL (Sign Language) proposed by Amorim et al [7].

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The ST-GCN-based studies aim to overcome the shortcomings of the ST-GCN or apply the ST-GCN to other domains such as hand gesture recognition and sign language recognition. ST-GCN has the disadvantage that it has high complexity, and the skeleton graph used in ST-GCN is pre-determined heuristically based on the natural connectivity of the human body skeleton which is less than optimal for the task of recognizing human activities [8]. For example, the relationship between two hands is important for recognizing classes such as "clapping" and "reading". However, it is difficult for the ST-GCN to capture the dependence between the two hands because they are located far from each other [4][9][10].

Cheng et al proposed Shift-GCN [8] to overcome the problem of ST-GCN. Shift-GCN is inspired by the shift convolution operation on CNN which is more efficient than the usual convolution process that uses a kernel of a certain size. Shift-GCN consists of a spatial shift graph convolution that extracts spatial features on the frame and a temporal shift graph convolution to remember the movement of the frame on each frame. Non-local shift graph operation is one of the spatial shift graph operations proposed in the Shift-GCN method which allows every feature of a vertex to be shifted to another vertex so that each vertex has the same relationship regardless of whether the neighboring vertices are directly adjacent or not. This causes no difference between vertices that are directly adjacent to other vertices because the relationship of each vertex has the same weight which makes local structural features between directly adjacent vertices neglected. The physical dependence between the joints of the body is very important for understanding human activities [11].

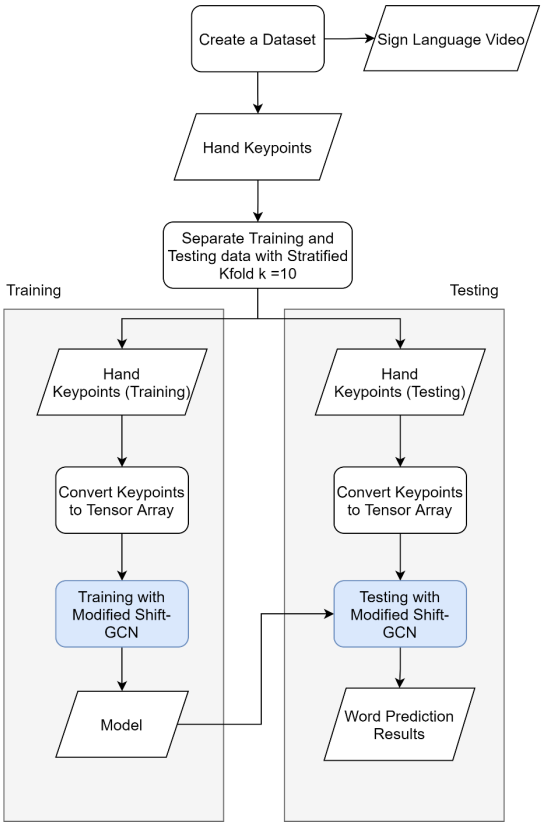


Figure 1: System Overview

Therefore, this study proposes a modification of the non-local shift graph operation in the Shift-GCN method by weighting the number of shifts based on its adjacency. Directly neighboring vertices have a higher number of shifting features than other vertices, which causes the relationship between vertices that are directly neighboring to be stronger than other vertices. This study used the position of bones and joints in the hand estimated by MediaPipe Hands [12] and produced the skeletal data of hand joints position that shaped like a graph. The graph is processed using the modified Shift-GCN. The dataset used in this study is hand keypoints extracted from video data of 26 American Sign Language (ASL) alphabets taken with a camera.

2. RESEARCH METHOD

The main process of this study can be seen in Figure 1 includes create a dataset, separate the dataset into training data and testing data, preprocess the dataset by converting the hand keypoint data into an array tensor, and perform training and testing with modified Shift-GCN.

2.1 Dataset

The dataset used in this study is a video dataset of each ASL sign language letter from the letter A to the letter Z which was collected by the author by recording the hand demonstrating the movement of the hand letter. Each letter has 100 videos in mp4 format, and each video is recorded with a resolution of 340×256 pixels in 30 fps with 2 seconds duration. Hence, each video has 60 frames. Hand keypoint was extracted from each video by MediaPipe Hands that produced 21 hand keypoints. Each hand keypoint has x position, y position, and z relative depth position value. The hand keypoints for all frames in each video are stored in a JSON file. The total of video files is 2600 files and each video file has one hand keypoint JSON file, so the total files in the dataset are 5200 files. Dataset creation flow can be seen in Figure 2 and an example of the hand keypoints stored in JSON can be seen in Figure 3. The example of video frames of static letters can be seen in Figure 4 and the example of video frames of

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

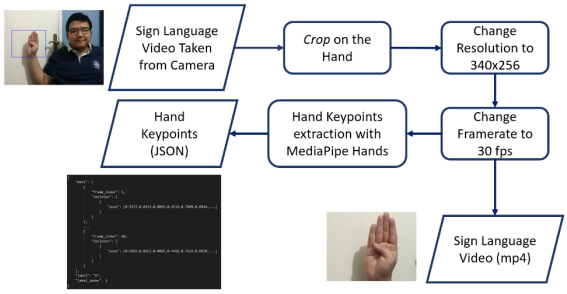


Figure 2: Dataset Creation Flow

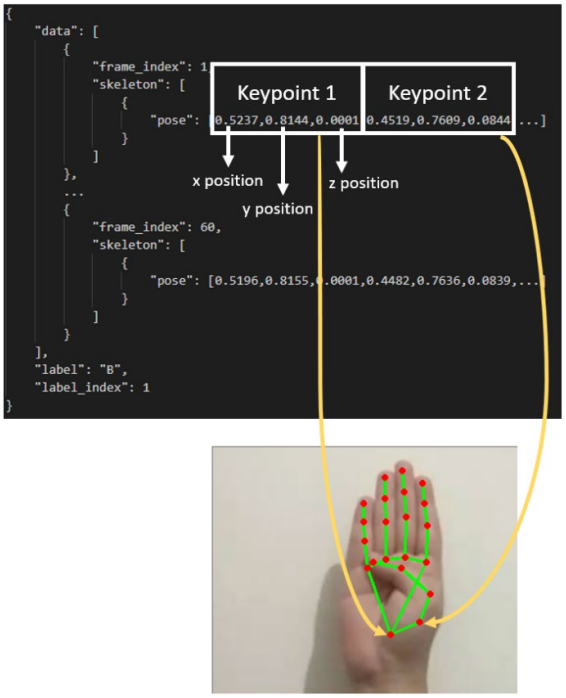
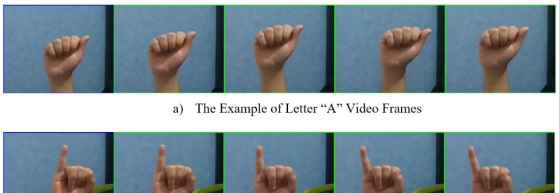


Figure 3: The Example of Hand Keypoint Estimation Results



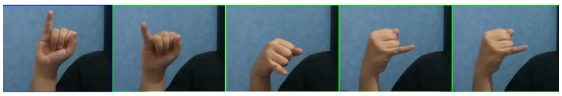
a) The Example of Letter "A" Video Frames

b) The Example of Letter "I" Video Frames

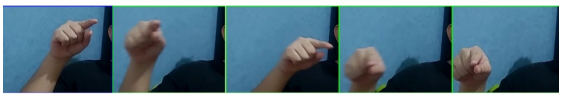
c) The Example of Letter "Y" Video Frames

Figure 4: The Example of Static Letter Frames of a) Letter "A", b) Letter "I", and c) Letter "Y"

dynamic letters like letter "J" and letter "Z" can be seen in Figure 5. The dataset specifications for training and testing can be seen in Table 1.



a) The Example of Letter "J" Video Frames



b) The Example of Letter "Z" Video Frames

Figure 5: The Example of Dynamic Letter Frames of a) Letter "J" and b) Letter "Z"

Table 1: Dataset Specifications for Training and Testing

Description	Specification
Video Resolution	340×256
Frame per Second	30 fps
Duration	2 seconds
Frame Total	60 frames
Video File Extension	.mp4
Hand Keypoints File Extension	.json
Total of Frame in Hand Keypoint File	60 frame (max)
Number of Classes	26 classes
Training Data Total Files	2600
Number of training data file for each fold	2340
Number of testing data file for each fold	260
Video file size	350 - 450 KB
JSON file size	30 - 32 KB
Color Channel	3 (RGB)

B R O W N

Figure 6: The Example of "BROWN" Video Frames for Word Prediction Test

Table 2: Word List for The Word Prediction Test

No.	Word	No	Word
1	THE	6	OVER
2	QUICK	7	LAZY
3	BROWN	8	DOG
4	FOX	9	JIJIK
5	JUMPS	10	ZEBRA

Table 3: Word Prediction Video Specifications

Description	Specification
Video Resolution	340×256
Frame per second	30 fps
Duration	3-7 seconds
Video File Extension	.mp4
Number of Files	10
Video file size	770 KB - 2 MB
Color Channel	3 (RGB)

The dataset used to train and test the model in this study is the hand keypoint data. The dataset is separated into training data and testing data using Stratified KFold with $k = 10$ which helps to reduce

sampling bias [13]. The Stratified KFold that has been generated is saved to a JSON file and reused for other models with different configurations for evaluation purposes like compare the method or change the parameters.

For the word prediction test, the author recorded 10 words with varying letter lengths in sign language in different videos, so that one video contained one word. One word has several letters, for example, the word "BROWN" contains the sign language gestures of the letters "B", "R", "O, "W", and "N" that can be seen in Figure 6. The list of words for the word prediction test can be seen in Table 2. The specification of word videos can be seen in Table 3.

2.2 Preprocessing

The hand keypoint data is converted to $N \times C \times T \times V \times M$ shape tensor array with N number of sample data, C number of channels, T maximum number of frames, V number of vertices or joints, and M number of hands. The N value is the number of sample data that is processed, for example, if the training data is 2340 data, then $N = 2340$. The C value is the number of channels. Initially, the number of channels is 3, which contains the values of the x position, y position, and z position, but the number of channels will increase when the convolution process in the training process is carried out. The T value is the maximum number of frames of the data. Although the maximum number of frames in the dataset used is 60 frames, in this study we used $T = 300$ which is the default value from the original Shift-GCN. The V value is the number of vertexes which is $V = 21$ because there are 21 hand keypoints. The M value is the number of hands which is $M = 1$ because the ASL alphabet only uses one hand.

2.3 Modified Shift-GCN

The hand keypoint data that has been converted to tensor array is trained using the Shift-GCN method which has been modified in the non-local shift graph operation. The architecture used in this study uses the same architecture as the original Shift-GCN that consists of batch normalization layer, TCN_GCN layers, and linear fully connected layer which can be seen in Figure 7. The TCN_GCN layer consists of spatial shift graph convolution and temporal shift graph convolution which can be seen in Figure 8.

The modified non-local shift graph operation is inside of the spatial shift graph convolution. First, the graph generated from hand keypoints is extracted with spatial features with spatial shift graph convolution with modified non-local shift graph operation. The non-local shift graph operation in Shift-GCN performs feature shifts from each vertex to another vertex regardless of whether the neighboring vertices are directly adjacent or not, which causes no difference between the vertices that are directly adjacent to the other vertices so that the local structural features between the vertices are related immediately neglected.

The proposed modification of the non-local shift graph operation is by weighting the number of shifts of vertex features so that directly adjacent vertices have more feature shifts than vertices that are not directly adjacent so that local structural features between directly adjacent vertices are not ignored. For example, suppose there is a simple hand graph with six vertices and edges connecting the vertices (1,2), (1,3), (1,4), (1,5), and (1,6) with 12 channels on each vertex which can be seen in Figure 9.

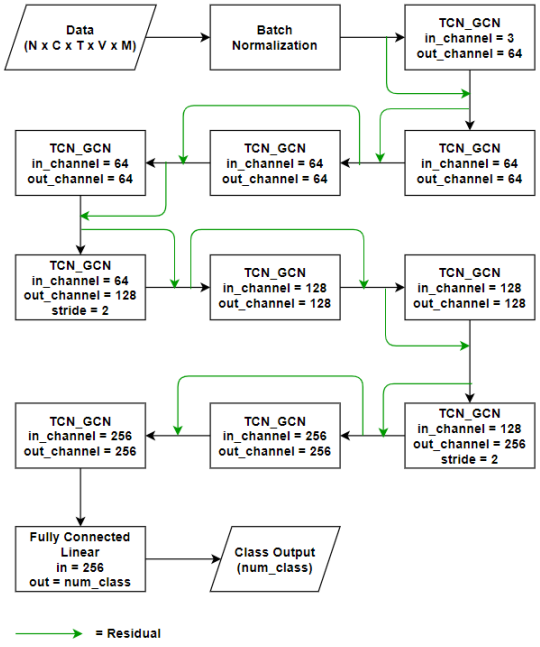


Figure 7: Shift-GCN Architecture

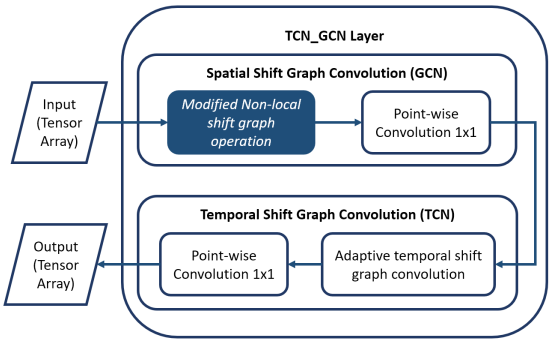


Figure 8: TCN_GCN Layer

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

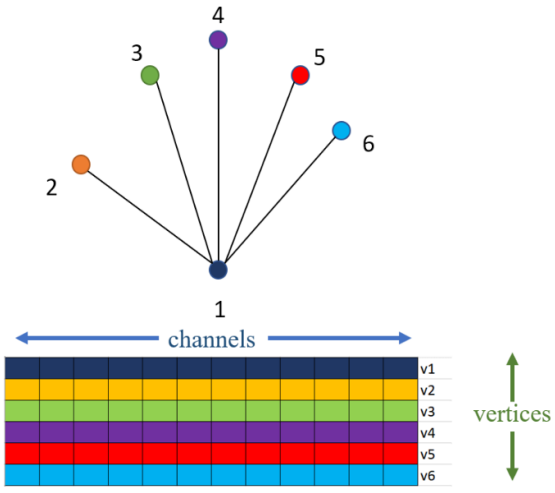


Figure 9: The Example of Simple Hand Graph

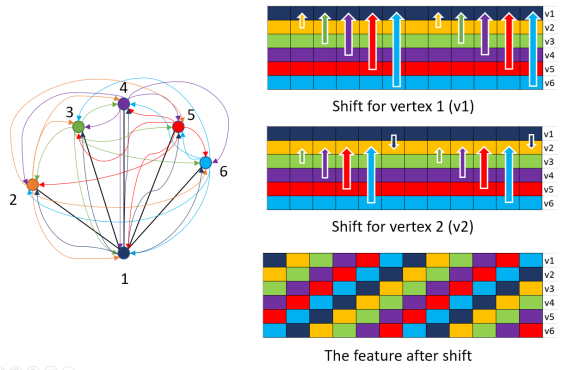


Figure 10: The Example of Original Non-Local Shift Graph Operation Shift-GCN

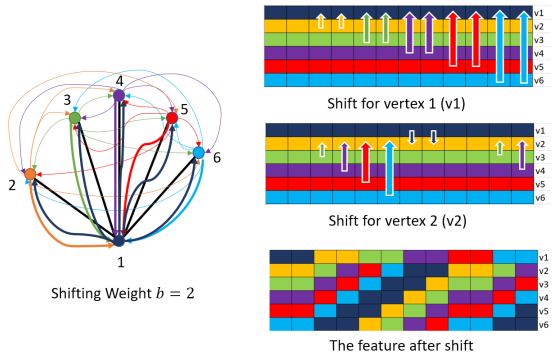


Figure 11: The Example of Modified Non-Local Shift Graph Operation with Shifting Weight $b = 2$

This study introduces shift weighting on non-local shift graph operations which makes shift features of vertices that are directly adjacent to get a larger portion than vertices that are not directly adjacent. For example, if the shifting weight is $b = 2$, then the features that are shifted from the directly adjacent vertices are twice as many as the vertices that are not directly adjacent. This causes the relationship of the vertices that are directly adjacent to be stronger and maintains the natural skeletal structure of the hand and maintains the relationship between the vertices that are not directly adjacent. This proposed method is an extension of the original non-local shift graph operation because if the shifting weight is $b = 1$ in the modified non-local shift graph operation, the result will be the same as the original Shift-GCN non-local shift graph operation. An example of the original non-local shift graph operation can be seen in Figure 10 and an example of the modified non-local shift graph operation can be seen in Figure 11. After performing the modified non-local shift graph operation, then a point-wise convolution process is carried out which performs the convolution process using a 1×1 kernel that iterates through each point. This kernel has as much depth as the channel that the input vertex has. The modified non-local shift graph operation and point-wise convolution processes are part of the spatial shift graph convolution.

Furthermore, a temporal shift graph convolution is used to remember the sequence of frame movements in each frame. The temporal shift graph convolution used in this study is the adaptive temporal shift graph convolution which is an improvement from the naïve temporal shift graph convolution. Temporal shift graph convolution is done by shifting the feature of the next frames on the temporal dimension which causes each frame to have information from its neighboring frames. After the shift operation is done, it is continued with the point-wise convolution process.

3. EXPERIMENTAL RESULTS

The testing process is useful for finding the most effective method. The methods tested in this study include local shift graph operation, non-local shift graph operation, and modified non-local shift graph with three different shifting weights b for testing non-local shift graph operation which has been modified with $b = 2$, $b = 3$, and $b = 4$. The best results from a test scenario is used for the next test scenario. In this section, there are test scenarios and test results, including evaluation on test data, evaluation on one dataset, evaluation on different subjects, and word prediction test.

3.1 Evaluation on Test Data

The experiment is carried out by recognizing each letter of the alphabet in ASL from the letter A to the letter Z. Performance of the system is

Table 4: Hyperparameter Specification

Description	Specification
Number of classes	26
Number of vertices	21
Number of Maximum Frame	300
Batch	16
Number of Epoch	30
Number of Files	2600
Learning Rate	0.1
Step	20, 25

Table 5: Evaluation on Test Data Results with Local Shift-GCN

Fold	Last Accuracy (%)	Best Accuracy (%)	Fold	Last Accuracy (%)	Best Accuracy (%)
1	99.62	100	6	100	100
2	100	100	7	99.23	99.62
3	100	100	8	98.08	98.85
4	99.23	99.62	9	99.23	99.62
5	100	100	10	98.85	98.85

Table 6: Evaluation on Test Data Results with Non-Local Shift-GCN

Fold	Last Accuracy (%)	Best Accuracy (%)	Fold	Last Accuracy (%)	Best Accuracy (%)
1	100	100	6	100	100
2	100	100	7	99.62	100
3	100	100	8	99.23	99.62
4	99.62	100	9	100	100
5	100	100	10	99.23	99.23

Table 7: Evaluation on Test Data Results with Modified Non-Local Shift-GCN $b = 2$

Fold	Last Accuracy (%)	Best Accuracy (%)	Fold	Last Accuracy (%)	Best Accuracy (%)
1	99.62	99.62	6	100	100
2	100	100	7	99.62	100
3	100	100	8	98.85	99.23
4	98.85	99.23	9	99.23	99.62
5	99.23	99.62	10	99.23	99.62

Table 8: Evaluation on Test Data Results with Modified Non-Local Shift-GCN $b = 3$

Fold	Last Accuracy (%)	Best Accuracy (%)	Fold	Last Accuracy (%)	Best Accuracy (%)
1	100	100	6	100	100
2	100	100	7	99.62	100
3	100	100	8	99.23	100
4	99.62	100	9	100	100
5	99.23	99.62	10	99.62	100

Table 9: Evaluation on Test Data Results with Modified Non-Local Shift-GCN $b = 4$

Fold	Last Accuracy (%)	Best Accuracy (%)	Fold	Last Accuracy (%)	Best Accuracy (%)
1	100	100	6	100	100
2	100	100	7	100	100
3	100	100	8	99.62	100
4	99.62	99.62	9	100	100
5	99.62	99.62	10	99.62	99.62

Table 10: Evaluation on Test Data Average Results

Method	Last Accuracy Average (%)	Best Accuracy Average (%)	Best Fold
Local Shift-GCN	99.424	99.656	2,3,5,6
Non-Local Shift-GCN	99.77	99.885	1,2,3,5,6,9
Modified Non-Local Shift-GCN ($b = 2$)	99.463	99.694	2,3,6
Modified Non-Local Shift-GCN ($b = 3$)	99.732	**99.962**	1,2,3,6,9
Modified Non-Local Shift-GCN ($b = 4$)	**99.848**	99.886	1,2,3,6,7,9

evaluated with the accuracy [14] metric. The train data and test data are generated with Stratified Kfold with $k = 10$ to reduce the sampling bias. Each fold is trained using the train data and evaluated with test data with 30 epochs. The initial learning rate is 0.1, on epoch 21 the learning rate is decayed to 0.01, and on epoch 26 the learning rate is decayed to 0.001. The hyperparameter specification can be seen in Table 4. The purpose of training and testing with different methods is to compare the effectiveness of the methods. The results of evaluation accuracy of Local Shift-GCN can be seen in Table 5, Non-Local Shift-GCN can be seen in Table 6, Modified Non-Local Shift-GCN with $b = 2$ can be seen in Table 7, Modified Non-Local Shift-GCN with $b = 3$ can be seen in Table 8, and Modified Non-Local Shift-GCN with $b = 4$ can be seen in Table 9. The last accuracy result is the accuracy result of the last epoch of each fold, while the best accuracy result is the best accuracy result ever achieved in an epoch. The average accuracy of each method can be seen in Table 10.

Based on the results, the Modified Non-Local Shift-GCN with $b = 3$ produces the best accuracy from the average best accuracy of each fold of 99.962% which is better than Local Shift-GCN with 99.656% accuracy and Non-Local Shift-GCN with an accuracy of 99.656% and Non-Local Shift-GCN. Local Shift-GCN with 99.885% accuracy. This proves that the shifting weighting based on neighbors affects the performance of the model. In addition, shifting weights with different weights produce different accuracy.

3.2 Evaluation on One Dataset

In the previous test, the model from each fold was tested with test data from the fold itself, not yet tested with all test data from other folds. The scenario on one dataset is intended to test the model of each method from the best fold with all the data in

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 11: Evaluation on One Dataset Results

Method	Best Model Accuracy (Fold 6) (%)	Worst Model Accuracy (Fold 8) (%)
Local Shift-GCN	99.92	99.81
Non-Local Shift-GCN	100	99.92
Modified Non-Local Shift-GCN ($b = 2$)	100	99.88
Modified Non-Local Shift-GCN ($b = 3$)	100	99.88
Modified Non-Local Shift-GCN ($b = 4$)	**100**	**99.96**

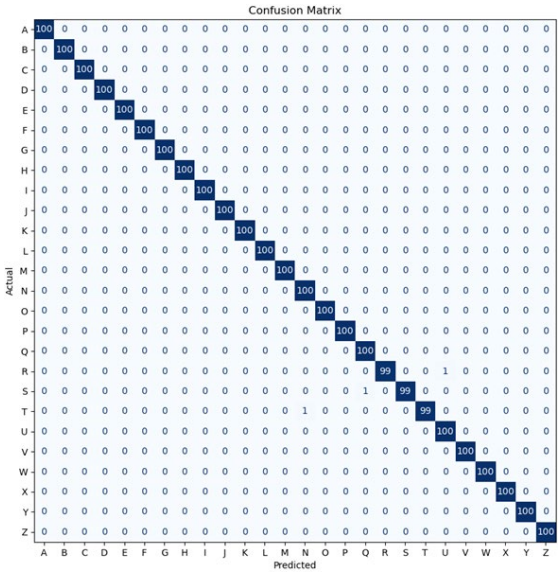


Figure 12: Evaluation on One Dataset Modified Non-Local Shift-GCN $b = 2$ Confusion Matrix

the dataset. This scenario is also carried out to determine whether there is a difference between static and dynamic letters.

The model used in this scenario is the model of fold 6 and fold 8, respectively. The model from fold 6 was chosen because each method produces the highest accuracy on the testing data, while the model from fold 8 was chosen because each method produces the lowest accuracy in the testing data.

The dataset used is a dataset used for training and testing before being separated by Stratified KFold with k=10, so the number of testing files in this scenario is 2600 JSON files of 26 letters of the alphabet from the letter "A" to the letter "Z" with each letter has 100 files. The results of the evaluation on one dataset of each method can be seen in Table 11.

Furthermore, the evaluation on one dataset showed that the accuracy of the best model reached 100%, while the worst model reached 99.92%. To

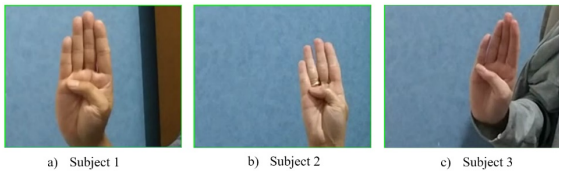


Figure 13: The Example of Hands from Different Subjects

Table 12: Evaluation on Different Subject Results

Method	Accuracy (%)
Local Shift-GCN	74.36
Non-Local Shift-GCN	74.36
Modified Non-Local Shift-GCN ($b = 2$)	**78.21**
Modified Non-Local Shift-GCN ($b = 3$)	**78.21**
Modified Non-Local Shift-GCN ($b = 4$)	**78.21**

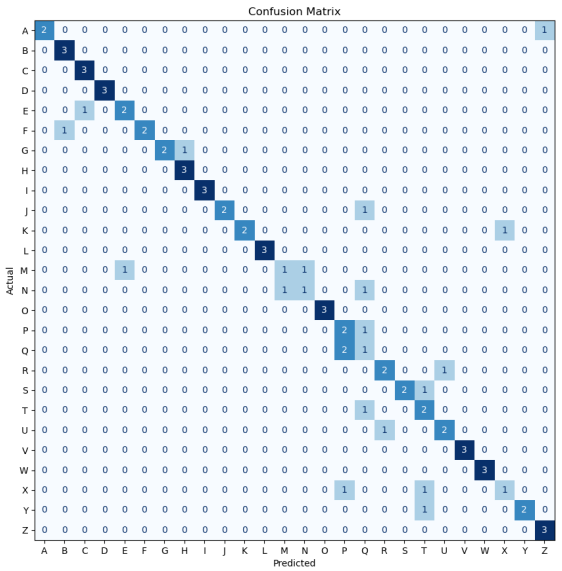


Figure 14: Local Shift-GCN Confusion Matrix

find out whether there is a difference in the results of static letters and dynamic letters (the letter "J" and the letter "Z") can be shown using a confusion matrix from a poor accuracy model because the difference can't be seen if the good model is used. Modified Non-Local Shift-GCN with $b = 2$ can be an example because it has poor model accuracy from all proposed methods in the evaluation on test data. The confusion matrix of the Modified Non-Local Shift-GCN with $b = 2$ can be seen in Figure 12. Based on the confusion matrix, there is no difference between static letters and dynamic letters, even the error is in the static letters which are wrongly predicted to other static letters.

3.3 Evaluation on Different Subjects

The evaluation scenarios on different subjects are intended to test the performance of the model in the hands of different people other than the author's hands. There are three different subjects tested in this

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

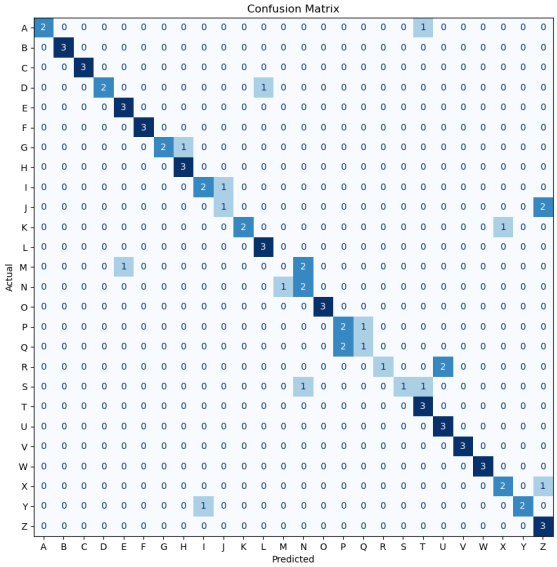


Figure 15: Non-Local Shift-GCN Confusion Matrix

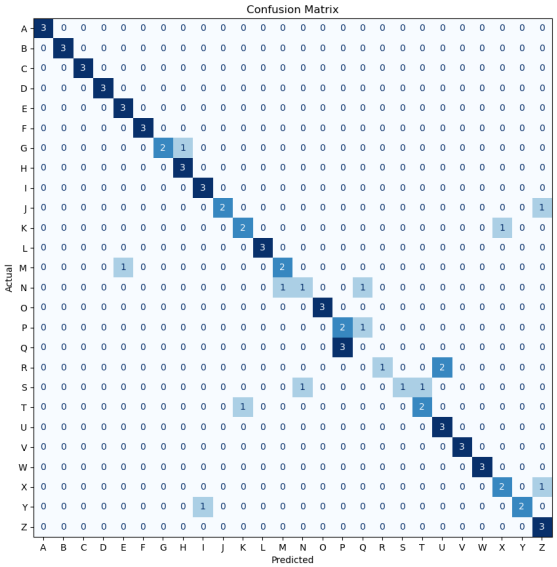


Figure 17: Modified Non-Local Shift-GCN $b = 3$ Confusion Matrix

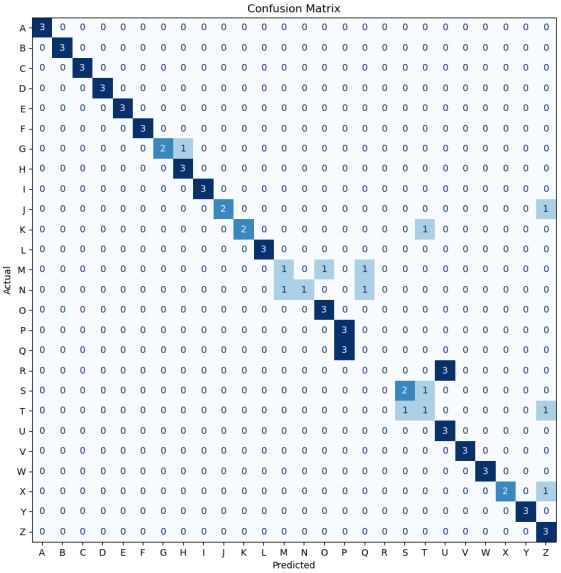


Figure 16: Modified Non-Local Shift-GCN $b = 2$ Confusion Matrix

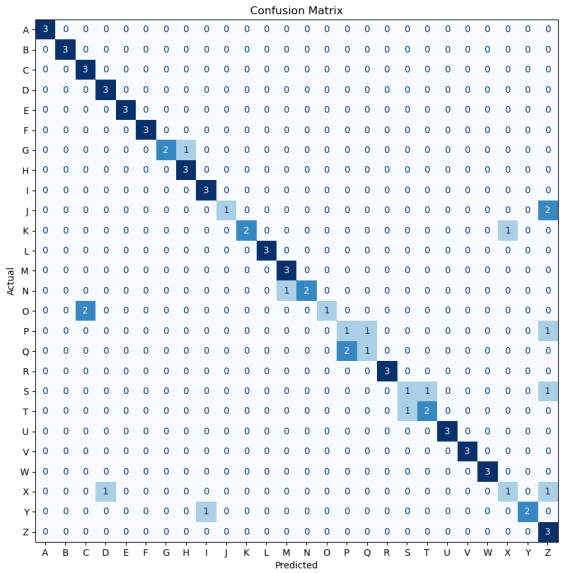


Figure 18: Modified Non-Local Shift-GCN $b = 4$ Confusion Matrix

pilot scenario. The three subjects performed sign language movements from the letter "A" to the letter "Z" with different hand shapes. Examples of hands from these three subjects can be seen in Figure 13.

The model used in this scenario is the fold 6 model for each method. The model from fold 6 was chosen because each method produces the highest accuracy in the testing data. The results of evaluation on different subjects from each method can be seen in Table 12. The confusion matrix of Local Shift-GCN can be seen in Figure 14, Non-Local Shift-GCN can be seen in Figure 15, Modified Non-Local Shift-GCN with $b = 2$ can be seen in Figure 16, Modified Non-Local Shift-GCN with $b = 3$ can be seen in Figure 17, and the Modified Non-Local Shift-GCN with $b = 4$ can be seen in Figure 18.

The evaluation on different subjects shows that the proposed method produces 78.21% accuracy which is better than Local Shift-GCN and Non-Local Shift-GCN. This result is lower than when testing using the dataset collected by the author because of the smaller number of tests. In addition, testing was also carried out with different hand shapes so that it could affect the hand keypoint extraction by MediaPipe Hands.

Journal of Theoretical and Applied Information Technology
30th September 2021. Vol.99. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

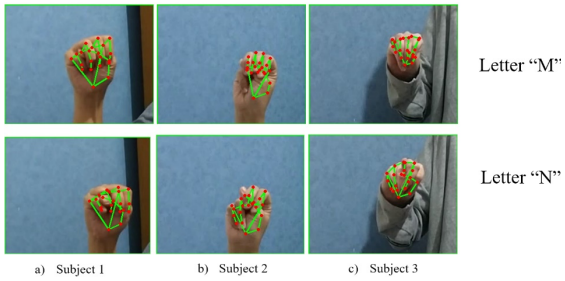


Figure 19: Comparison of Letter "M" and Letter "N"

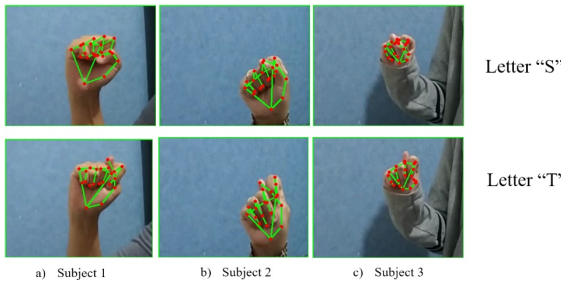


Figure 20: Comparison of Letter "S" and Letter "T"

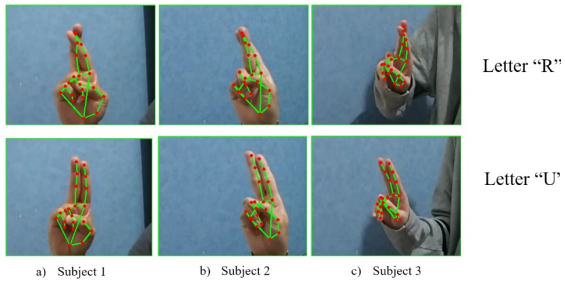


Figure 21: Comparison of Letter "R" and Letter "U"

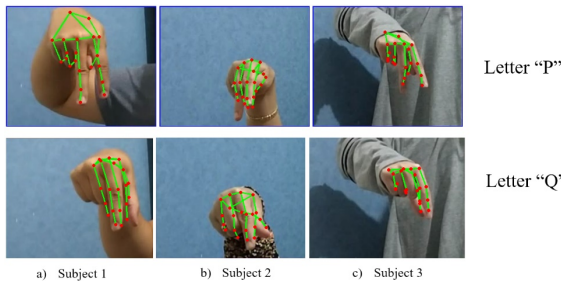


Figure 22: Comparison of Letter "P" and Letter "Q"

The results of the confusion matrix in Figure 14 to Figure 18 produce different letter predictions from one model to another. Based on the confusion matrix, many prediction errors occur in the letter “J”, “M”, “N”, “Q”, “R”, and “S”. The letter "J" which is a dynamic letter is incorrectly predicted to be the letter "I" or "Z" because the letter "J" is a letter “I” with movement. The difference between the letter “J” and the letter “I” can be seen in Figure 4(b) and Figure 5(a). After that, many letters other than the letter "J" were incorrectly predicted as the letter "Z" due to the hand moving while shooting the video, so the model was more dominant in predicting the letter "Z" than the letter "J".

The letter "M" is sometimes wrongly predicted as the letter "N" and the letter "N" is sometimes wrongly predicted as the letter "M". This is due to the similarity between the letter “M” and the letter “N” as well as the shape of the hand and the position of the camera. An example of the letters “N” and “M” can be seen in Figure 19. Cases of letters that look similar also occur in the letter "S" which is sometimes also wrongly predicted as the letter "T". An example of the letters “S” and “T” can be seen in Figure 20. The letter “R” is also sometimes wrongly predicted as the letter “U” with an example that can be seen in Figure 21.

In addition, the letter “Q” was incorrectly predicted as the letter “P” due to the imperfect

Table 13: Word Prediction Results on Local Shift-GCN

Actual Word	Predicted Word	Levenshtein Distance
THE	AHE	1
QUICK	QUIOCK	1
BROWN	BROWM	1
FOX	FOX	0
JUMPS	JUMNPQ	2
OVER	OUER	1
LAZY	LAXZY	1
DOG	DOG	0
JIJIK	JIK	2
ZEBRA	ZEBRA	0
Total Levenshtein Distance		9

Table 14: Word Prediction Results on Non-Local Shift-GCN

Actual Word	Predicted Word	Levenshtein Distance
THE	THE	0
QUICK	QUICK	0
BROWN	BROWN	0
FOX	FOX	0
JUMPS	IJUMPT	2
OVER	OVQER	1
LAZY	LAXZLY	2
DOG	DOG	0
JIJIK	IZJIK	2
ZEBRA	ZEBRA	0
Total Levenshtein Distance		7

Table 15: Word Prediction Results on Modified Non-Local Shift-GCN $b = 2$

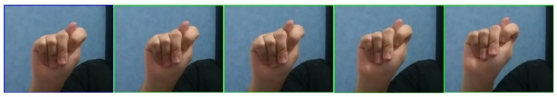
Actual Word	Predicted Word	Levenshtein Distance
THE	THE	0
QUICK	QUICK	0
BROWN	BROWN	0
FOX	FOX	0
JUMPS	JUMPT	1
OVER	OVER	0
LAZY	LAZY	0
DOG	DOG	0
JIJIK	JIJIK	0
ZEBRA	ZEBRA	0
Total Levenshtein Distance		1

Table 16: Word Prediction Results on Modified Non-Local Shift-GCN $b = 3$

Actual Word	Predicted Word	Levenshtein Distance
THE	THE	0
QUICK	QUICK	0
BROWN	BROWN	0
FOX	FOX	0
JUMPS	IJUMPS	1
OVER	OVER	0
LAZY	LAXZY	1
DOG	DOG	0
JIJIK	JIJIK	0
ZEBRA	ZEBRA	0
Total Levenshtein Distance		2

Table 17: Word Prediction Results on Modified Non-Local Shift-GCN $b = 4$

Actual Word	Predicted Word	Levenshtein Distance
THE	TKHE	1
QUICK	QUICK	0
BROWN	BROWN	0
FOX	FOX	0
JUMPS	JUMPT	1
OVER	OVER	0
LAZY	LAZY	0
DOG	DOG	0
JIJIK	JIOIK	1
ZEBRA	ZEBRA	0
Total Levenshtein Distance		3



a) The Example of Letter "T" Video Frames

b) The Transition from Letter "P" to Letter "S"

Figure 23: Comparison of (a) The Letter "T" and (b) The Transition from Letter "P" to Letter "S"

capture of the hand keypoints by MediaPipe Hands. The letters "P" and "Q" also look similar, the difference is that the middle finger is closed, and the thumb is behind the index finger on the letter "Q". This similarity can be seen in Figure 22. Based on this test, the sign language recognition system built in this study relies heavily on hand keypoints extracted with MediaPipe Hands.

3.4 Word Prediction Test

The word prediction on video data test scenario is intended to find the most robust method in predicting words with various letters in one video. This scenario is measured by the Levenshtein distance which can measure the level of difference between the predicted word and the actual word.

The Levenshtein distance between two words is the minimum amount of editing of a single character by insertion, deletion, or replacement, required to convert one word into another. The Levenshtein lev distance of the word a and the word b with i as the index of the character of the word a and j as the index of the character of the word b can be formulated as in equation (1). Suppose there are the words "Saturday" and "Sunday", then the Levenshtein distance is 3 because there are three operations that make the word "Saturday" become "Sunday" namely removing the letter "a", deleting the letter "t" and replacing the letter "r" into a letter "n". The smaller the Levenshtein distance, the more similar one word is to another.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

The results of word prediction of Local Shift-GCN can be seen in Table 13, Non-Local Shift-GCN can be seen in Table 14, Modified Non-Local Shift-GCN with $b = 2$ can be seen in Table 15, Modified Non-Local Shift-GCN with $b = 3$ can be seen in Table 16, and Modified Non-Local Shift-GCN with $b = 4$ can be seen in Table 17. The word prediction test on video data shows that the Modified Non-Local Shift-GCN with $b = 2$ has the lowest total Levenshtein distance of 1 which makes the proposed method able to recognize words with various letters in one video in real environmental conditions better than Local Shift-GCN and Non-Local Shift-GCN.

Modified Non-Local Shift-GCN with $b = 2$ managed to predict 9 words correctly out of 10, with the word "JUMPS" incorrectly predicted as "JUMPT". This is because there is a moment of transition between the letter "P" and the letter "S" so that the model predicts the sequence of the transition frame between letters as the letter "T". Sequence of frame transfer between letters can be processed by the model because every 30 consecutive frames are processed to predict the letters contained in the sequence of frames regardless of the sequence of frames there are letters or are in transition from one letter to another. The comparison between the letter "T" and the transition from letter "P" to letter "S" can be seen in Figure 23.

4. CONCLUSIONS

This study proposed a modified non-local shift graph operation in the Shift-GCN for ASL alphabet recognition. The modification is carried out by weighting the number of shifts based on its adjacency. Directly neighboring vertices have a higher number of shifting features than other vertices, which causes the relationship between

vertices that are directly neighboring to be stronger than other vertices. Based on the experiment results, the Modified Non-Local Shift-GCN with $b = 3$ produces the best accuracy from the average best accuracy of each fold of 99.962% in evaluation on test data which is better than Local Shift-GCN and Non-Local Shift-GCN. Modified Non-Local Shift-GCN with $b = 4$ produces 100% accuracy from the best model and 99.96% from the worst model in evaluation on one dataset. Furthermore, the Modified Non-Local Shift-GCN with $b = 2$, $b = 3$, and $b = 4$, resulted in an accuracy of 78.21% in evaluation on different subjects. At last, Modified Non-Local Shift-GCN with $b = 2$ succeeded in predicting 9 words correctly out of 10 words in the word prediction test. For future work, the development of hand sign language datasets with more diverse hand shapes is needed and further exploration on shift weighting could be performed on a domain involving a large number of skeletal data such as human activity recognition.

REFERENCES:

[1] M. J. Cheok, · Zaid Omar, · Mohamed, and H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Mach. Learn. Cybern.*, vol. 10, pp. 131–153, 2019, doi: 10.1007/s13042-017-0705-5.

[2] V. Adithya and R. Rajesh, "A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2353–2361, 2020, doi: 10.1016/j.procs.2020.04.255.

[3] A. Aljabar and Suharjito, "BISINDO (Bahasa isyarat indonesia) sign language recognition using CNN and LSTM," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 5, pp. 282–287, 2020, doi: 10.25046/AJ050535.

[4] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 12018–12027, Dec. 2019, doi: 10.1109/TIP.2020.3028207.

[5] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 7444–7452, 2018.

[6] Y. Li, Z. He, X. Ye, Z. He, and K. Han, "Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, p. 78. Dec. 2019, doi: 10.1186/s13640-019-0476-x.

[7] C. C. de Amorim, D. Macêdo, and C. Zanchettin, "Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11731 LNCS, pp. 646–657, 2019, doi: 10.1007/978-3-030-30493-5_59.

[8] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 180–189, 2020, doi: 10.1109/CVPR42600.2020.00026.

[9] F. Li, A. Zhu, Y. Xu, R. Cui, and G. Hua, "Multi-Stream and Enhanced Spatial-Temporal Graph Convolution Network for Skeleton-Based Action Recognition," *IEEE Access*, vol. 8, pp. 97757–97770, 2020, doi: 10.1109/ACCESS.2020.2996779.

[10] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition." Accessed: Jun. 16, 2021. [Online]. Available: https://github.com/lshiwjx/2s-AGCN.

[11] H. Xia and X. Gao, "Multi-scale Mixed Dense Graph Convolution Network for Skeleton-based Action Recognition," *IEEE Access*, vol. 4, pp. 1–10, 2021, doi: 10.1109/ACCESS.2020.3049029.

[12] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.10214.

[13] D. Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, 2018, pp. 542–545.

[14] W. Zhu, N. Zeng, and N. Wang, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS® implementations.," *Northeast SAS Users Gr. 2010 Heal. Care Life Sci.*, pp. 1–9, 2010.