IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.17, No.3, July 2023, pp. 313~324
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.76555

Error Action Recognition on Playing The Erhu Musical Instrument Using Hybrid Classification Method with 3D-CNN and LSTM

Aditya Permana[1], Timothy K. Shih[2], Aina Musdholifah[3], Anny Kartika Sari[4]
[1] Master Program of Computer Science; FMIPA UGM, Yogyakarta Indonesia
[2] Department Computer Science and Information Engineering, Collage of Electrical Engineering and Computer Science, National Central University, Taiwan – ROC
[3] Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta Indonesia
[4] Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta Indonesia
e-mail: **[1] aditya.permana@mail.ugm.ac.id**, [2] tshih@g.ncu.edu.tw, [3] aina_m@ugm.ac.id, [4] a_kartikasari@ugm.ac.id

Abstrak

Erhu adalah alat musik petik yang berasal dari China. Dalam memainkan alat musik ini terdapat aturan bagaimana memposisikan tubuh pemain dan memegang alat musik dengan benar. Oleh karena itu, diperlukan suatu sistem yang dapat mendeteksi setiap gerakan dari pemain Erhu. Penelitian ini akan membahas pengenalan aksi pada video menggunakan metode 3DCNN dan LSTM. Metode 3D Convolutional Neural Network merupakan metode yang memiliki basis CNN. Untuk meningkatkan kemampuan menangkap setiap informasi yang tersimpan dalam setiap gerakan, diperlukan kombinasi lapisan LSTM dalam model 3D-CNN. LSTM mampu menangani masalah gradien hilang yang dihadapi oleh RNN. Penelitian ini menggunakan video RGB sebagai dataset, dan terdapat tiga bagian utama dalam preprocessing dan ekstraksi fitur. Tiga bagian utama adalah badan, tiang erhu, dan busur. Untuk melakukan preprocessing dan ekstraksi ciri, penelitian ini menggunakan body landmark untuk melakukan preprocessing dan ekstraksi ciri pada segmen body. Sebaliknya, segmen erhu dan busur menggunakan algoritma Hough Lines. Selanjutnya untuk proses klasifikasi, kami mengusulkan dua algoritma, yaitu algoritma tradisional dan algoritma deep learning. Algoritma dua klasifikasi ini akan menghasilkan output pesan error dari setiap gerakan erhu player.

Kata kunci**–** *Action Recognition, Convolutional Neural Network, 3D-CNN, LSTM*

Abstract

Erhu is a stringed instrument originating from China. In playing this instrument, there are rules on how to position the player's body and hold the instrument correctly. Therefore, a system is needed that can detect every movement of the Erhu player. This study will discuss action recognition on video using the 3DCNN and LSTM methods. The 3D Convolutional Neural Network method is a method that has a CNN base. To improve the ability to capture every information stored in every movement, combining an LSTM layer in the 3D-CNN model is necessary. LSTM is capable of handling the vanishing gradient problem faced by RNN. This research uses RGB video as a dataset, and there are three main parts in preprocessing and feature extraction. The three main parts are the body, erhu pole, and bow. To perform preprocessing and feature extraction, this study uses a body landmark to perform preprocessing and feature extraction on the body segment. In contrast, the erhu and bow segments use the Hough Lines algorithm. Furthermore, for the classification process, we propose two algorithms, namely, traditional algorithm and deep learning algorithm. These two-classification algorithms will produce an error message output from every movement of the erhu player.

Keywords**—** *Action Recognition, Convolutional Neural Network, 3D-CNN, LSTM*

Received July 25th,2022; Revised July 30th, 2023; Accepted July 31th, 2023

1. INTRODUCTION

Erhu is musical instrument was invented at the time of the Tang Dynasty, since thousands of years of development history, with a beautiful, soft tone and rich expression [1]. Erhu musical instrument does not have a fretboard. Erhu also has no taste and various complex parameters that can directly affect intonation control in Erhu playing, so intonation is the main key that must be mastered by Erhu players [2]. So, the position and body gestures of the erhu player greatly affect the sound produced. Starting from the position of the fingers, hands, body, and head, that must be considered in playing the erhu musical instrument. Through the rules that have been determined in playing the Erhu musical instrument, this research will combine two classification methods, namely the traditional classification algorithm and the deep learning classification algorithm. This traditional classification algorithm includes the positions of the head, body, shoulders, Erhu, and bow; therefore, the next process is to calculate every angle formed from each of the parts studied. The second method is to use a deep learning algorithm. The research [3] shows one example of the application of deep learning technology. This technology is used in object detection in CCTV security camera systems. In his research, the method used is the CNN method. CNN method is a method commonly used in several studies on action recognition or object detection. Video classification applies moving object detection technology. In this study, the moving object in question is the body of the erhu player and the erhu musical instrument, namely the Erhu and bow.

Video recognition also to be a concern in [6] research. That research is on dense convolutional networks for efficient video analysis, explained that action classification and gesture recognition are a new stage in the convolution neural network technique. The approach taken in the video analysis research is based on feature extraction per frame using 2D-CNN and utilizing 3D-CNN or LSTM to merge spatiotemporal information. Video classification will perform image processing in each extracted frame. 2D-CNN is considered very successful in feature extraction and classification in image recognition. However, such information will be lost for feature extraction on video when there is movement in each frame. Therefore, to overcome these problems, in research [7], they use 3D-CNN to capture information in each frame and the movement between successive frames. 3D-CNN combines information from both time and space to extract features, making the model more rational [5]. Other research related to action recognition was also conducted [10] aims to segment and recognize continuous gestures. The recognition network is constructed based on the 3D Convolutional neural network, the convolutional long short-term memory called ConvLSTM, and 2D Convolution Neural Network or 2D-CNN for isolated gesture recognition.

Several deep learning methods are now used in image or video classification. One of the methods used by Rui Huo et al. in performing object or action classification in the video is the 3D Convolution Neural Network or 3D-CNN. The dataset used by the 3D-CNN method is a 2-dimensional image; therefore, a special method is needed to generalize from 2 to 3 dimensions first. The process of generalizing 2-dimensional datasets into 3 dimensions is fairly complex. The process does not just add one dimension but must consider the asymmetry between space and time; therefore, the convolutional neural network training process for 3 dimensions is heavier than 2 dimensions because it has more features and parameters [4].

Video recognition is usually used to recognize a human gesture. There are many goals in utilizing video recognition technology, one of which is the classification of human gestures. Human gestures are usually a continuous activity. Continuous gesture recognition aims to recognize continuous and sequential movements and have meanings such as scenarios in the gesture movements. The research conducted by Guangming Zhu et al. aims to segment and recognize continuous gestures. The researcher proposes an effective deep architecture for continuous gesture recognition through this paper. The first step is to segment the continuous gesture sequence into isolated gesture instances using the Res3D Network. Furthermore, the recognition network is constructed based on the 3D Convolutional neural network, the convolutional long short-term memory called ConvLSTM, and 2D Convolution Neural Network or 2D-CNN for isolated gesture recognition. Merging several methods into one architecture, according to the researcher, is more effective in learning long term and deep spatiotemporal features [8].

Deep Learning architecture is currently developing rapidly. Many new architectural models have been experimented with by several researchers. Therefore, it is necessary to have a comparative analysis of several deep learning architecture models that have been tested. Mohit Pandiya et al. have analyzed the deep learning architecture related to object detection [9]. Object detection is one technology that is often classified in computer vision techniques. These technologies include image processing or video processing. The main purpose of object detection is to determine or conclude what objects are in the image or video by considering the accuracy value of these conclusions. Several architectural models are competing to get high accuracy results.

However, according to the researcher, some problems still exist in deep convolutional networks for action recognition. This problem is the first that many previous studies in motion recognition only use a single video segment as the object of research which can provide problems in representing the linkage of the action meaning in the entire video due to lack of information. The second problem is that many studies do not focus on enhancing the ability to characterize movements as time passes. They just put temporal information in the same class as spatial information [12]. Regarding these two problems, the researcher proposes an architectural model for a Long-term 3D Convolutional Fusion Network LT3D-CFN. According to the researcher, this architectural model can represent two aspects: the 3D-CNN method, which replaces the CNN in the two-stream network method, which functions as an extractor feature from both spatial and temporal dimensions of a video. The second aspect is related to the long-term associations between clips of one motion video using the deep LSTM Network method.

Recently, gesture recognition, motion, or object detection has grown widely. This is because the interaction between humans and computers or electronic devices has entered the touchless era. Therefore, one example of gesture recognition is hand gesture recognition. Researchers Muneer Al-Hammadi et al. take advantage of these technological developments by applying the hand gesture recognition method for sign language applications. Sign Language is a sign language used by persons with disabilities to communicate with others. Through this research, the hand gesture recognition application for sign language utilizes the 3D-CNN method in making a gesture recognition model. Hand gesture recognition should consider both spatial and temporal features, and unfortunately, finding discriminative spatiotemporal descriptors for a hand gesture sequence is not a trivial task [10]. Therefore, the researchers in this study proposed a deep convolutional neural network approach for hand gesture recognition, and the researchers chose the 3D-CNN method. To overcome the scarcity of a large labeled hand gesture dataset, they created an approach by applying transfer learning is needed. There are three datasets of RGB video used in the evaluation, namely 40, 23, and 10 classes, and using the previously described approach resulted in recognition rates of 98.12%, 100%, and 76.67% for each class.

Hand gesture recognition is currently a very interesting research object. Specifically related to sign language, which is very dependent on hand movements, the use of deep learning to create a sign language recognition system is very much needed. Other research related to sign language recognition was also conducted by Suharjito et al. Sign language recognition has complex and varied problems. Variations and combinations of words to form sentences make recognizing sign language even more difficult. Many researchers have done research related to this. Therefore, there have been many new methods and architectures to get good recognition results. Because of the similarities between sign language recognition and action recognition, researchers tried to apply a model often used in previous research, namely I3D inception [11]. I3D inception used in this study uses 3D-CNN architecture because invention v1 has been modified to 3D-CNN. The reason for using the 3D-CNN method is because, according to previous researchers, there was an improvement in the results of the ResNet-50, C3D Ensemble, and Two Stream Fusion + IDT methods. From the results of experiments conducted by researchers, 100% accuracy of training using 10 words and 10 people with 100 classes resulted, but the validation results of this experiment were still low.

2. METHODS

This chapter will explain the method implemented in this research. Several methods were used in this research, such as experiment architecture, data collection, data preparation and deep learning model architecture. For data collection this research will use MMPose library to extract the body landmark. MMPose will give each body landmark coordinate to use for preprocessing the dataset such as head, shoulder, left hand, right hand and knees. So that, using that coordinate we can crop each body part for the input dataset to the deep learning model. Then for the Erhu part and bow part, we use the Hough Line Algorithm to get the horizontal and vertical line. This algorithm is already included in Tensorflow so that we can use and do some image processing process to get the line. From the bow line and erhu line, we can get the angle degree for some classifier that we use in this research.

This study will use several classifiers to classify erhu musical instrument players. An erhu instrument player must fulfil several rules in the process of playing the instrument in order to produce perfect sound quality. Table 1 is the list of the class that used in this research.

Table 1 The classifier.

Code	Class Description	Class Threshold
E-11	Head position not normal (too Left or Right)	If head center line > K or < K degrees then E-11
E-12	Left shoulder too high	If left shoulder higher than right shoulder for X% then E-12
E-13	Right shoulder too high	If left shoulder lower than right shoulder for X% then E-13
E-14	Need to seat straight (too Left or Right)	If body center line > K or < K degrees then E-14
E-15	Put knees in normal positions	If absolute (distance of knees – distance of shoulders) > Y% then E-15
E-41	Pole tilt too left – Bow hair and string must be orthogonal	If pole line and bow line is not orthogonal and pole line > N degrees then E-41
E-42	Pole tilt too right – Bow hair and string must be orthogonal	If pole line and bow line is not orthogonal and pole line < N degrees then E-42
E-43	Trace of bow must be in straight line	If horizontal angle of bow changes > M or < M degrees then E-43
E-21	Left elbow too high	If probability > P then E-21
E-22	Left elbow too low	If probability > P then E-22
E-23	Left elbow and wrist in a line	If probability > P then E-23
E-31	Wrong right hand thumb position	If probability > P then E-31
E-32	Wrong right hand index finger position	If probability > P then E-32
E-33	Wrong right hand middle or ring finger position	If probability > P then E-33
E3N	Normal	-

Table 2 is the dictionary of threshold variables that used in each classifier.

Table 2 Threshold variable dictionary

Variable	Value	Description
P	0.8	Percentage value for prediction result threshold
M	5	Degrees value for horizontal bow angle threshold
N	10	Degrees value for vertical angle threshold
K	10	Degrees value for body angle threshold
Y	10	Percentage value for shoulder and knees differentiation threshold
X	15	Percentage value for shoulder angle differentiation threshold

2.1 Data Collection

The MMPose library was used in this experiment to extract full-body landmarks from each frame in the video data set. Body landmarks help extract feature input from body segments such as the left arm and right hand. This feature will be the input for the combination of 3DCNN and LSTM model.

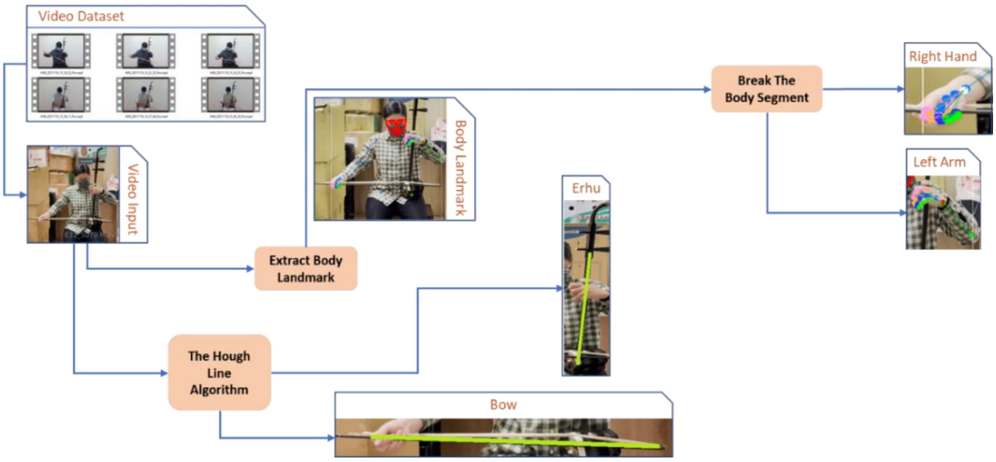

Figure 2.1 Data Collection Architecture

2.2 Data Preprocessing

The break process that shown in both approach architecture means that there is process to cropping the image based on body landmark coordinate to produce left hand and right-hand image for the input to the deep learning model. Then from Hough Line algorithm, we do some image processing to get one horizontal line and one vertical line. The following that shown in figure 2.2 is the results from the line detection process using the Hough Transform. It can be seen that many irregular straight lines are detected.

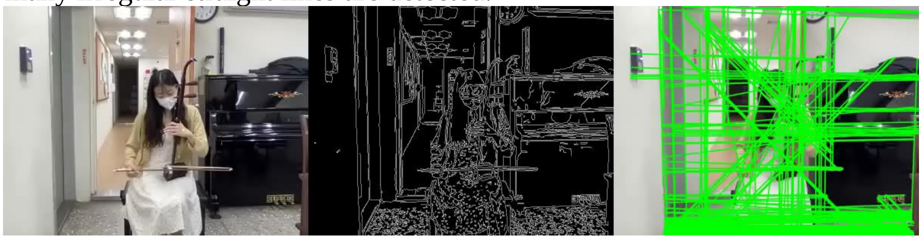

Figure 2.2 The Hough Line Algorithm Line Detection Result Before Image Processing

So that we use image processing in this line detection process. In this research we reduce the noise and do image smoothing using gaussian blur. We also do background subtractor to make the background of the erhu player cleaner so that we can focus only on the player. The concept of background subtractor is will remove the object that not moving frame by frame in the video, so it can improve the result and make the Hough Line algorithm to focus on the erhu player only. The result after do gaussian blur and background subtractor is shown in figure 2.4.

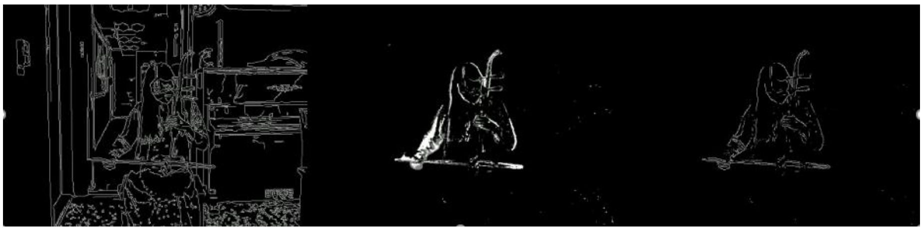

Figure 2.3The Hough Line Algorithm Line Detection Result After Image Processing

After do gaussian and background subtractor then we get few lines detected as shown in the left image from figure 2.4. Then from some horizontal lines and vertical lines we do average process to get only one horizontal line and one vertical line like shown in the right image in figure 2.4.

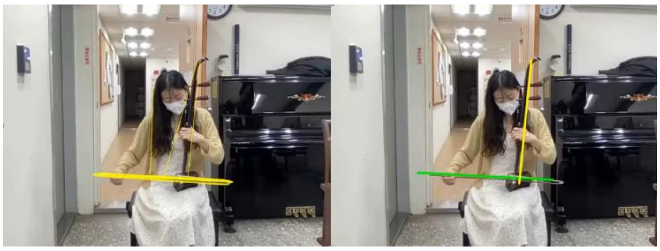

Figure 2.4 The bow line and erhu line

After we get the all body landmark, bow and erhu line coordinate, then we will separate it. First for deep learning model inputs, we will use RGB images as an input and then for traditional algorithm we will use body landmark coordinate and bow and erhu coordinate to be as an input. The figure 2.5 below is the result of data collecting and preprocessing.

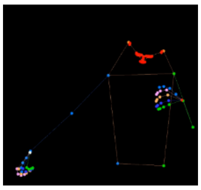

Figure 2.4 MMPose Result

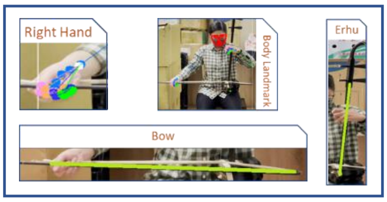

Figure 2.5 Data Collection Result

2.3 Experiment Design

The following is the experimental design applied in this study. This experiment starts with video input which will then be processed every 30 frames using the 3DCNN+LSTM model that has been formed previously through a deep learning training process. In addition to making predictions using deep learning models, several predefined classifiers will be processed using traditional algorithms.

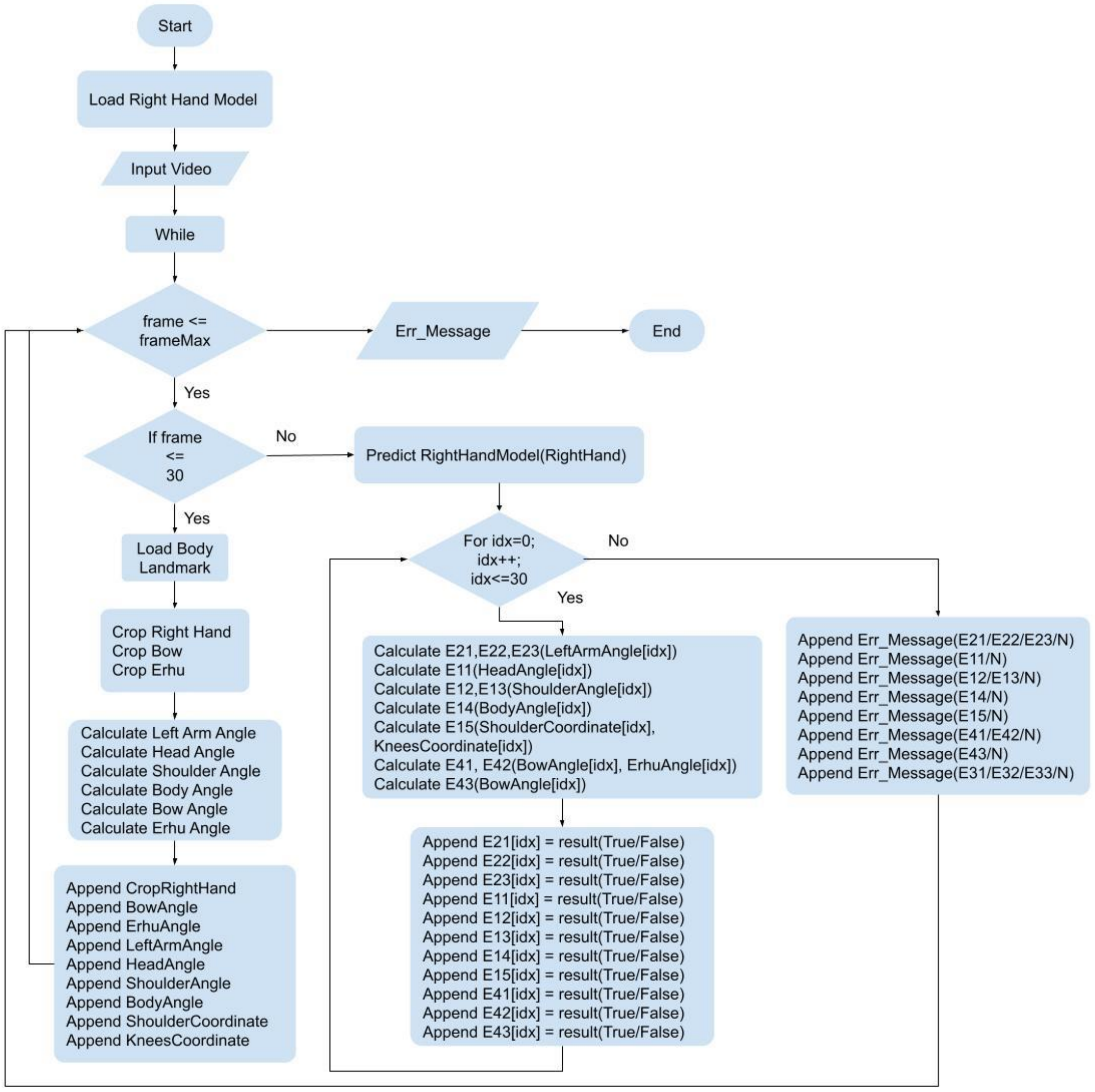
Start
Load Right Hand Model
Input Video
While
frame <= frameMax
Err_Message
End
Yes
If frame <= 30
No
Predict RightHandModel(RightHand)
Yes
Load Body Landmark
For idx=0; idx++; idx<=30
No
Yes
Crop Right Hand
Crop Bow
Crop Erhu
Calculate E21,E22,E23(LeftArmAngle[idx])
Calculate E11(HeadAngle[idx])
Calculate E12,E13(ShoulderAngle[idx])
Calculate E14(BodyAngle[idx])
Calculate E15(ShoulderCoordinate[idx], KneesCoordinate[idx])
Calculate E41, E42(BowAngle[idx], ErhuAngle[idx])
Calculate E43(BowAngle[idx])
Append Err_Message(E21/E22/E23/N)
Append Err_Message(E11/N)
Append Err_Message(E12/E13/N)
Append Err_Message(E14/N)
Append Err_Message(E15/N)
Append Err_Message(E41/E42/N)
Append Err_Message(E43/N)
Append Err_Message(E31/E32/E33/N)
Calculate Left Arm Angle
Calculate Head Angle
Calculate Shoulder Angle
Calculate Body Angle
Calculate Bow Angle
Calculate Erhu Angle
Append E21[idx] = result(True/False)
Append E22[idx] = result(True/False)
Append E23[idx] = result(True/False)
Append E11[idx] = result(True/False)
Append E12[idx] = result(True/False)
Append E13[idx] = result(True/False)
Append E14[idx] = result(True/False)
Append E15[idx] = result(True/False)
Append E41[idx] = result(True/False)
Append E42[idx] = result(True/False)
Append E43[idx] = result(True/False)
Append CropRightHand
Append BowAngle
Append ErhuAngle
Append LeftArmAngle
Append HeadAngle
Append ShoulderAngle
Append BodyAngle
Append ShoulderCoordinate
Append KneesCoordinate

Figure 2.6 Experiment Design

Figure 2.6 is the flowchart of this experiment design. Every 30 frames, an error code will be generated according to the specified classifier. The part that used deep learning method is right hand part. This part is for right hand classifier. The output of this experimental design is a collection of error codes for each 30 frames. This research uses a deep learning 3D Convolutional Neural Network or 3D-CNN model. Figure 2.7 it the model training of the deep learning method that implemented in this experiment.

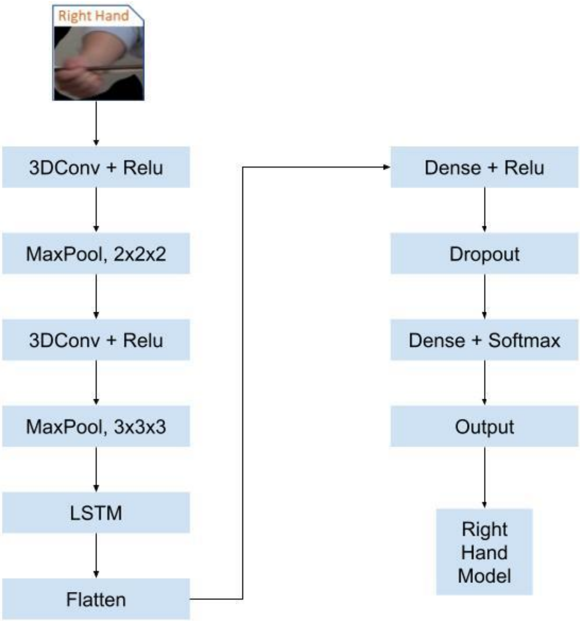

Figure 2.7 3DCNN+LSTM Model Architecture

2.4 Dataset

In this research, all the datasets used were collected and recorded independently that shown in figure 2.8. The dataset used is a video with an average duration of 1 minute. In the dataset, there are ten different erhu players. The total number of videos in the dataset is 60 videos. To increase the number of datasets, each video will be split into several videos with 3 seconds per clip length. There are some of the videos dataset that used in this research.

1.mp4
2.mp4
E21&E31_5096_clip_0.mp4
E21&E31_5105_clip_0.mp4
E21&E32_5076_clip_0.mp4
E21&E35_5123_clip_0.mp4
E21,E31.mp4
E21_5077_clip_0.mp4
E22,E32.mp4
E23,E33.mp4
E23,E34.mp4
E23,E35(2).mp4
Erhu Video Segmentation 640 480 15.mp4
Incomplete bow in right hand 1_0.mp4
Left hand forward 3_18.mp4
Left hand too high 4_14.mp4
Right hand tigers mouth clip violin bow
WIN_20211116_15_34_05_Pro.mp4
WIN_20211116_15_38_11_Pro.mp4

Figure 2.8 Videos Dataset

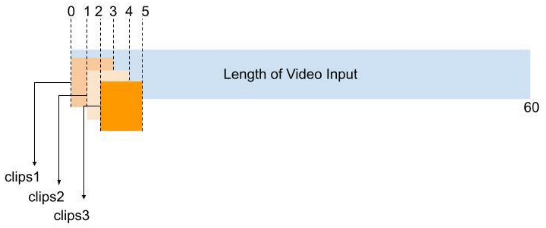

Figure 2.9 Splitting Video Dataset Technique

Technically, splitting the video dataset is done every three seconds by cutting every single frame and then on a video. For example, one video is cut for three seconds starting from the 0th frame to the third frame, then cutting for the next three seconds starts from the first frame to the fourth frame and so on. Figure 2.9 illustrates the process of splitting the video dataset into several clips. In this experiment, one clip lasts 3 seconds with a stride value of 1 for each clip in the video. This process will be repeated for all 60 video datasets resulting in 925 video clips. We then break down that dataset into three parts, namely 65% for the training dataset, 30% for the validation dataset and 5% for the test dataset.

2.5 Evaluation Matrix

In this study, the evaluation used is precision and recall. This is the standard evaluation used in classification experiments. The classification prediction result is a binary classifier that produces a true or false classification value. Prediction results are divided into four quantitative statistical categories: TP or the number of positive samples predicted as positive. The second category is FN, namely the number of positive samples predicted as negative samples. The third category is FP, namely the number of negative samples predicted as positive samples, and the last is TN, the number of negative samples predicted as negative samples.

Table 3.2 Confusion Matrix

	True	False
True (Positive)	TP (True Positive) / Correct Result	FP (False Positive) / Unexpected Result
False (Negative)	TN (True Negative) / Correct Absence of Result	FN (False Negative) / Missing Result

$$precission = \frac{TP}{TP + FP}\ ;\ \ recall = \frac{TP}{TP + FN}\ ;\ accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.6 Experiment Setting

The deep learning model used in this research is 3D Convolutional Neural Network combined with Long Short-Term Memory. The 3DCNN and LSTM models use the following parameter settings:

Table 3.3 Model Training Parameters Setting Table

Algorithm	Value
3DCNN + LSTM	Epoch = 100
	Batch size = 4
	Initial Learning Rate = 0.0001
	Shuffle = True
	Callback = True
	Patience = 5
	Restore best weight = True
	Input: RGB (214 x 214)

3. RESULTS AND DISCUSSION

After doing training for all models used in this system. The following are the results of training using the 3DCNN and LSTM models. Figure 3.1 is the result training loss and accuracy for the 3D-CNN and LSTM model.

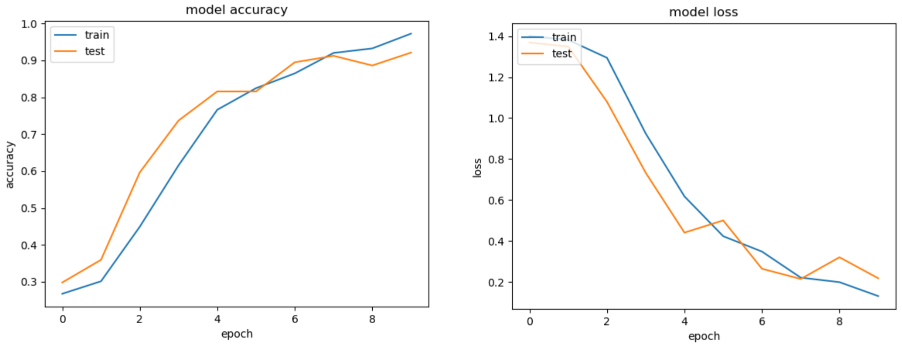

Figure 3.1 Training accuracy and loss 3D-CNN and LSTM model

The result of the training 3D-CNN and LSTM model is looks good. The accuracy of the train dataset is 0.972 and test dataset accuracy is 0.921. Then for loss value of train dataset is 0.13 and loss value of test dataset is 0.22.

3.1 Experiment Evaluation

The following are the results of the evaluation of the experiments that have been carried out. The evaluations used for classes that use deep learning models are precision and recall, as described in the previous chapter. The videos that used for testing in this evaluation is from testing dataset and each class using 46 different datasets. Below is the result of the confusion matrix from class E31 (Wrong right-hand thumb position).

Table 3.1 Class E31 Confusion Matrix

	True	False
Positive	29	17
Negative	33	13

$$precision = \frac{29}{29+17} = \frac{29}{46} = 0.63 = 63\%$$

$$recall = \frac{29}{29+13} = \frac{29}{42} = 0.46 = 69\%$$

$$accuracy = \frac{29+33}{29+33+17+13} = \frac{62}{92} = 0.71 = 71\%$$

Furthermore, the result of the confusion matrix from class E32 (Wrong right-hand index finger position) is shown in table 3.2.

Table 3.2 Class E32 Confusion Matrix

	True	False
Positive	34	12
Negative	36	10

$$precision = \frac{34}{34+12} = \frac{34}{46} = 0.73 = 73\%$$

$$recall = \frac{34}{34+10} = \frac{34}{44} = 0.77 = 77\%$$

$$accuracy = \frac{34+36}{34+36+12+10} = \frac{70}{92} = 0.76 = 76\%$$

Then table 3.3 show the result of the confusion matrix for E33 (Wrong right-hand middle or ring finger position).

Table 3.3 Class E33 Confusion Matrix

	True	False
Positive	37	9
Negative	38	8

$$precision = \frac{37}{37+9} = \frac{37}{46} = 0.80 = 80\%$$

$$recall = \frac{37}{37+8} = \frac{37}{45} = 0.82 = 82\%$$

$$accuracy = \frac{37+38}{37+38+9+8} = \frac{75}{92} = 0.81 = 81\%$$

Then table 3.4 show the result of the confusion matrix for E3N (right hand normal).

Table 3.4 Class E3N Confusion Matrix

	True	False
Positive	38	8
Negative	36	10

$$precision = \frac{38}{38+8} = \frac{38}{46} = 0.82 = 82\%$$

$$recall = \frac{38}{38+10} = \frac{38}{48} = 0.79 = 79\%$$

$$accuracy = \frac{38+36}{38+36+8+10} = \frac{74}{92} = 0.80 = 80\%$$

3.2 Discussion

Through several experiments carried out using hybrid classification method, namely using traditional algorithm and deep learning algorithm that using 3D-CNN and LSTM, we can see the confusion matrix table for precision, recall and accuracy results. First, the average accuracy value of the 4 classes (E31, E32, E33, E3N) is 77%. Second, the precision average value of the 4 classes is 74.5%. Then third, the average recall value of the 4 classes is 76.75%. The ambient conditions in video dataset that used when recording also vary. Some have a simple background or a complex background. The clothes used by erhu players also have no rules, so several players wear clothes with similar and straightforward colors and motifs in the dataset. However, erhu players also wear clothes with many colors and complex motifs..

4. CONCLUSIONS

After an experiment using hybrid classification method, there is several factors influence this result. The first factor is the variation of the room, environment and the clothing used in recording the dataset, which is not much. Furthermore, it can also be concluded that several factors that affect the results is some classifier that require distance and height classification, such as classes for knees distance, left arm to be high or low, left shoulder to high or low than right shoulder, body position slope to left or right, bow not in a straight line, erhu to left or right and head position to left or right is more accurate by using the calculation of the coordinates generated from the body landmark and for right hand that require specific gesture in the each finger position is quite better using deep learning model with average accuracy in all right hand classes is 77%.

REFERENCES

[1] Z. Lian, "*A Reflection On The Intonation In Erhu Performance*", in Journal Bulletin Social Economic and Humanitarian Research, pp 2658-5561, doi:10.5281/zenodo.3593861, 2019

[2] Li Zusheng, "*On the flavor beauty of Erhu Art*", Journal of Wuhan Conservatory of Music No 9, 2003

[3] T. Bing, L. Li, Y. Qu, L. Yan., *"Video Object Detection for Tractability with Deep Learning Method"*. Fifth International Conference on Advanced Cloud and Big Data, doi:10.1109/CBD.2017.75, 2017

[4] Hou, Rui, Chen & Sukthankar, Rahul & Shah, Mubarak., *"An Efficient 3D CNN for Action/Object Segmentation in Video"*, 2019

[5] L. Ma, F. Xu, T. Li and H. Zhang, *"A Moving Object Detection Method Based on 3D Convolution Neural Network"*, 7th International Conference on Information Science and Control Engineering (ICISCE), pp. 55-59, doi:10.1109/ICISCE50968.2020.00022, 2020

[6] T. Jin, Z. He, A. Basu, J. Soraghan, G. Di Caterina and L. Petropoulakis, *"Dense Convolutional Networks for Efficient Video Analysis"*, 5th International Conference on Control, Automation and Robotics (ICCAR), pp. 550-554, doi: 10.1109/ICCAR.2019.8813408, 2019

[7] M. -W. Lin, S. -J. Ruan and Y. -W. Tu, *"A 3DCNN-LSTM Hybrid Framework for sEMG-Based Noises Recognition in Exercise"*, IEEE Access, vol. 8, pp. 162982-162988, doi:10.1109/ACCESS.2020.3021344, 2020

[8] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah and M. Bennamoun, *"Continuous Gesture Segmentation and Recognition Using 3DCNN and Convolutional LSTM"*, IEEE Transactions on Multimedia, vol. 21, no. 4, pp. 1011-1021, doi: 10.1109/TMM.2018.2869278, April 2019

[9] M. Pandiya, S. Dassani and P. Mangalraj, *"Analysis of Deep Learning Architectures for Object Detection - A Critical Review"*, IEEE-HYDCON, pp. 1-6, doi: 10.1109/HYDCON48903.2020.9242776, 2020

[10] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, *"Hand Gesture Recognition for Sign Language Using 3DCNN"*, IEEE Access, vol. 8, pp. 79491-79509, doi: 10.1109/ACCESS.2020.2990434, 2020

[11] Suharjito, H. Gunawan, N. Thiracitta and A. Nugroho, *"Sign Language Recognition Using Modified Convolutional Neural Network Model"*, Indonesian Association for Pattern Recognition International Conference (INAPR), pp. 1-5, doi: 10.1109/INAPR.2018.8627014, 2018

[12] Y. Zhang, K. Hao, X. Tang, B. Wei and L. Ren, "*Long-term 3D Convolutional Fusion Network for Action Recognition*" IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 216-220, doi: 10.1109/ICAICA.2019.8873471, 2019