

Recognition and Assessment of Tae Kwon Do Moves using Kinect Camera Sensor

Ahmad Ashari
Computer Science and
Electronics Department
Universitas Gadjah Mada,
Yogyakarta, Indonesia

M. Idham Ananta Timur
Computer Science and
Electronics Department
Universitas Gadjah Mada,
Yogyakarta, Indonesia

Y. Fendryan Diastomo
Electronics and
Instrumentation Study Program
Universitas Gadjah Mada,
Yogyakarta, Indonesia

ABSTRACT

Tae Kwon Do is a martial art originating from Korea. The number of Tae Kwon Do's students is not proportional to the number of instructors (Sabeum) making the learning process quite hampered especially for beginners. This study offers an alternative system to assess Tae Kwon Do moves. Microsoft Kinect has a technology where it can recognize the movement that is being done by someone and then assess the movement, whether good or need more practice. The AdaBoost (Adaptive Boosting) algorithm is used in this research because it can get the best final hypothesis for data received by Kinect of its input, in order to be further processed. Movement that is used as data in this research only movement apkubi momtong baro jireugi, ap kubi eolgol arae makki, and ap seogi momtong ap chagi. The result of movements recognition system has accuracy 84,5%, precision 81,4%, sensitivity 76,7%, specificity 88,3%, and F-score 76,7% then the assessment has root mean square error value for ap kubi momtong baro jireugi is 35,3%, ap kubi eolgol makki is 19,3%, and ap seogi eolgol ap chagi is 29,6%. The optimal distance for digital frame recording is 3 m.

General Terms

AdaBoost, Tae Kwon Do, Kinect

Keywords

AdaBoost, Tae Kwon Do, Kinect

1. INTRODUCTION

Tae Kwon Do consists of 3 words: tae means foot or fighting with kicking techniques, kwon means hand or punching and self-defense with hand techniques, and do means art or discipline. In a simple way, Tae Kwon Do means art of self-defense using bare hands and feet techniques [1]. Some Indonesian are learning Tae Kwon Do especially teenagers and little kids. The problem is their will to continue study the movements in their house and there is no one who can assess them directly. Based on the problem, this research wants to give an alternative solution by using a system that can recognize and assess the movement from its object. The system needs a device that could do motion capturing. Microsoft Kinect is one of motion capturing devices, not only popular in its era but also has depth camera sensor for the accuracy made this research using Kinect. The system should be good for home education or helping Sabeum (Tae Kwon Do instructor/teacher) in their training place to improve the students' skills especially beginners. AdaBoost algorithm is used in this research because of the ability to classify and assess movements in the form of final hypothesis from data boosting process [2]. Wen-Chang Cheng [3], Phillipe Saade et al. [4], Wei-Chia Kao et al. [5], and Yixia Yun et al. [6] want to show us that AdaBoost have a good result for classification and assessment even when the user wants to do it in real-time.

Support Vector Machine doesn't have the ability to do live scoring because the data processing will be slower when the data is too much [7]. Based on their research, make this paper wants to use AdaBoost for a complex movement especially Tae Kwon Do.

2. ADABOOST ALGORITHM

AdaBoost or adaptive boosting is an algorithm to get the strong classifier by gather the weak classifiers. AdaBoost works like decision tree algorithm where the roots (decision stump) can be one or more levels. The process begins with data training as many as m $\{(x_1, y_1), \dots, (x_m, y_m)\}$ where $x_i \in X, y_i \in Y = \{-1, +1\}$ [8]. Next, weights are initialized as $D_1(i) = \frac{1}{m}$ distribute in D_t (dengan $= 1, \dots, T$) to train weak learner get weak hypothesis ($h_t : X \rightarrow \{-1, +1\}$) and how good or bad the results depend on error value, the smaller value means good result

$$\epsilon_t = \sum_{i=0}^n D_t(i) h_t(x_i) y_i \quad (1)$$

After having error value and train weak learner, next process has to get strong classifier

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad (2)$$

α_t is used for normalization factor Z_t through updating weights distribution D_{t+1}

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \epsilon^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ \epsilon^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (3)$$

The result from weights distribution update and normalization factor is final hypothesis as a sign of boosting process with AdaBoost algorithm is done

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (4)$$

3. MICROSOFT KINECT PROGRAM

Microsoft Kinect have three sensors [8]: RGB camera, depth sensor which contains infrared, and multiarray microphone. These sensors can help the device to have a feature called skeletal tracking. Skeletal tracking can detect the human body up to 2 people [9] and convert it into digital form as shown in Figure 1.

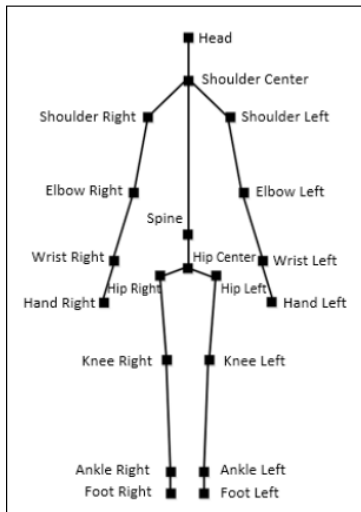


Figure 1. Digital skeletal form

Kinect should get the best angle of the object in its sensor range area about 1.2 - 3.5 meters vertically and maximum 1.8 meters height, it should be placed in a tripod that have 1.03 meters height to have a good result and data as shown in Figure 2.



Figure 2. Kinect position with a tripod

When the system wants to recognize and assess the object, Kinect should detect it first. The position of the object should be facing Microsoft Kinect so every sensor can read it correctly. Multiarray microphone can be ignored here because the system only needs to process the image or video not the voice of the object. Laptop or computer being used here to create a database file and programming so it needs Kinect for Windows Hub to connect with Microsoft Kinect. The connection in the system can be seen at Figure 3.



Figure 3. System connection and object position

The process of this research can be seen at Figure 4 that starting with recording the movement of Sabeum Tae Kwon Do as the reference for the system of the correct techniques and get the skeletal data. Each movement will have their own label here. After that, Kinect will be trained with the help of some software that contribute in this research. While the modelling done, it is time to test the system by using new data from the respondents and compare them with the Sabeum Tae Kwon Do's data to recognize first their movement by knowing the name of the movement, if it is match then the score result will be shown.

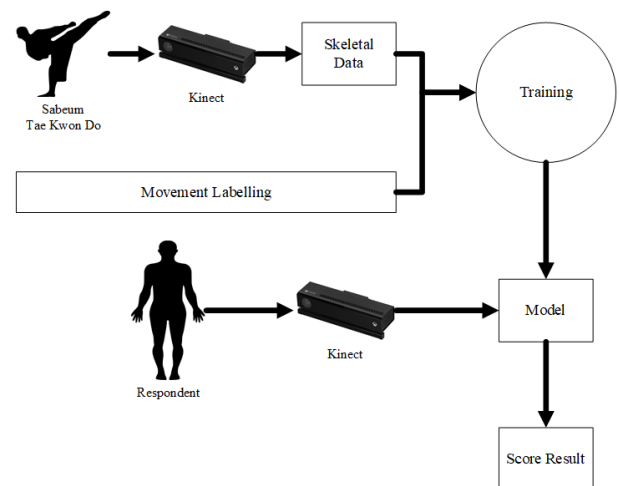


Figure 4. Process of the system

Microsoft provide SDK (Software Development Kit) [10] that contain Visual Gesture Builder to classify every movements recorded by Kinect and build a database file of it with .gbd format. The classification based on the skeletal tracked by Kinect and should choose what body parts that dominate in the movements to make AdaBoost algorithm check it easier like when the object doing punch it means the hands should be dominate in the movement or kicking that feet play a role in this movement. Talking about database, the system should have a right condition where it will be compared with a new data so by recording the instructor of Tae Kwon Do and make it as the data training should help it give a good result.

The whole training process can be seen in Figure 5 where it is using sample clips in .xef format with the help of Visual Gesture Builder to build a database.

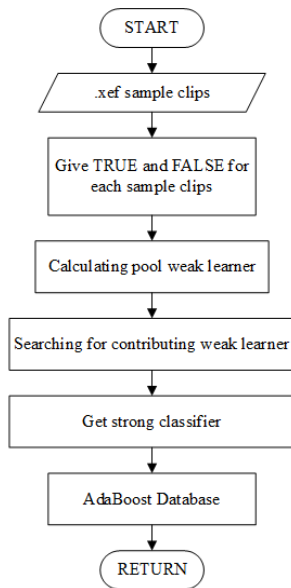


Figure 5. Training Process Flowchart

Beside training process, there will be testing process in every machine learning step to recognize and assess the movement automatically. The whole testing process can be seen in Figure 6 that the point is comparing the new data with the training data.

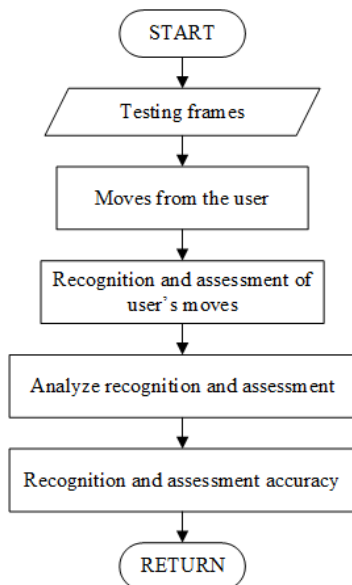


Figure 6. Testing Process Flowchart

Not only Visual Gesture Builder but also Visual Studio that provided by Microsoft to program and make the interface for this research. Visual Studio also use for accessing the database file that saved in .gbd format to separate functions and terms or condition for each movements. Language programming in this system is using C# because it is the most reliable language for database file in .gbd format.

From Visual Studio, variables declaration should need to be in the code for each function and movement. Accessing the database is the key for the system to learning as seen in Figure 7.

```

1. private readonly string
gestureDatabase=@"datahaha/ACUAN1.gbd";
2. private readonly string jir = "jirah_Right";
3. private readonly string eol = "eolah_Left";
4. private readonly string chagi = "chagiah_Right";
5. bool Jireugi = false;
6. bool Eolgol = false;
7. bool Apchagi = false;
8. public static float publicConfidence = 0;
9. public static float nilaijir;
10. public static float nilaieol;
11. public static float nilaichagi;
12. public static string namaDetector = "Belum bergerak";
13. private VisualGestureBuilderFrameSource vgbFrameSource
= null;
14. private VisualGestureBuilderFrameReader vgbFrameReader
= null;
    
```

Figure 7. Declaration & Database Access

After finishing the declaration, let's move into programming for frame detection and adding the movement data inside .gbd file as the new database as seen in Figure 8.

```

1. private void Reader_GestureFrameArrived(object sender,
VisualGestureBuilderFrameArrivedEventArgs e)
2. {
3. VisualGestureBuilderFrameReference frameReference =
e.FrameReference;
4. using (VisualGestureBuilderFrame frame =
frameReference.AcquireFrame())
5. {
6. if (frame != null)
7. {
8. Debug.Write("\n Frame nya gak null");
9. IReadOnlyDictionary<Gesture, DiscreteGestureResult>
discreteResults = frame.DiscreteGestureResults;
10. if (discreteResults != null)
11. {
12. foreach (Gesture gesture in
this.vgbFrameSource.Gestures)
13. {
14. if (gesture.GestureType == GestureType.Discrete)
15. {
16. Debug.Write("\n Gesture Discrete");
17. DiscreteGestureResult result = null;
18. discreteResults.TryGetValue(gesture, out result);
    
```

Figure 8. Adding Each Movements as the New Database

```

1. public TKD(KinectSensor kinectSensor)
2. {
3. if (kinectSensor == null)
4. {
5. throw new ArgumentNullException("kinectSensor");
6. }
7. this.vgbFrameSource = new
VisualGestureBuilderFrameSource(kinectSensor, 0);
8. this.vgbFrameSource.TrackingIdLost +=
this.Source_TrackingIdLost;
9. // Membaca frame yang ada pada file Visual Gesture
Builder
10. this.vgbFrameReader =
this.vgbFrameSource.OpenReader();
11. if (this.vgbFrameReader != null)
12. {
13. this.vgbFrameReader.IsPaused = true;
14. this.vgbFrameReader.FrameArrived +=
this.Reader_GestureFrameArrived;
15. }
16. // Menambahkan seluruh gerakan yang ada di dalam
database
17. using (VisualGestureBuilderDatabase database = new
VisualGestureBuilderDatabase(this.gestureDatabase))
18. {
19. this.vgbFrameSource.AddGestures(database.AvailableGestures
);
20. //this.vgbFrameSource.AddGestures(database.AvailableGes
tures);
21. foreach (var vgb in vgbFrameSource.Gestures)
22. {
23. //Memastikan isi gerakan sesuai dengan yang tersimpan
dalam file .gbd
24. Debug.Write("\n available gesture name = " +
vgb.Name.ToString());
25. }
26. }
    
```

Figure 9. Frame Detection Process

If new database already created, next step is to do frame detection. Here, AdaBoost algorithm provide result.Detected to give true or false condition from the user's movement and result.Confidence to give the score by comparing the new data with the saved database. That information can be put in the program as seen in Figure 9.

Every movement have their own characteristics and different value, so the program should separate the requirements for each movement using result.Detected and result.Confidence as seen in Figure 10. Make sure that each condition has their own variable name correctly like in the declaration program because if it is mis location and doesn't match, the result will be different and both the recognition and assessment system.

```

1. if (result != null)
2. {
3.   Debug.WriteLine("\n Result !=null");
4.   if (gesture.Name.Equals(this.jir))
5.   {
6.     Debug.WriteLine("\n Kebaca Jireugi");
7.     Jireugi = result.Detected;
8.     nilaijir = result.Confidence;
9.   }
10.  if (gesture.Name.Equals(this.chagi))
11.  {
12.    Debug.WriteLine("\n Kebaca Ap Chagi");
13.    Apchagi = result.Detected;
14.    nilaichagi = result.Confidence;
15.  }
16.  if (gesture.Name.Equals(this.eol))
17.  {
18.    Debug.WriteLine("\n Kebaca Eol");
19.    Eolgol = result.Detected;
20.    nilaieol = result.Confidence;
21.  }
22. }

```

Figure 10. Separating Each Movement Requirements

Next step is to make a program for recognition process so Kinect's sensors can differentiate the name of each movement when the user's skeletal detected by the system. Before that, confidence value has bigger role to do the recognition system because every movement have their own starting point value based on the user's power. So, the weak value should be declared in the program for each movement. The confidence value obtained by using result.Confidence. Make sure each movement have the correct function too and must be put in the same place as the confidence value condition. Both will be the parameter to get the name of the movement and score result as seen in Figure 11.

```

1. if (Jireugi && nilaijir > 0.1)
2. {
3.   Debug.WriteLine("\n Jireugi");
4.   namaDetector = "Ap Kubi Momtong Jireugi";
5.   publicConfidence = nilaijir;
6. }
7. if (Apchagi && nilaichagi > 0.2)
8. {
9.   Debug.WriteLine("\n Apchagi");
10.  namaDetector = "Ap Seogi Eolgol Ap Chagi";
11.  publicConfidence = nilaichagi;
12.  Debug.WriteLine("\n Masuk tidak public confidence " +
13.  nilaichagi.ToString());
14. }
15. if (Eolgol && nilaieol > 0.3)
16. {
17.   Debug.WriteLine("\n Eolgol");
18.   namaDetector = "Ap Kubi Eolgol Arae Makki";
19.   publicConfidence = nilaieol;
20.   Debug.WriteLine("\n Masuk tidak public confidence " +
21.   nilaieol.ToString());
22. }

```

Figure 11. Recognition Process

Unfortunately, because this is a complex movement, confidence value changing rapidly based on user's speed make the system sometimes confused on what the user is doing. The solution for this problem by making a segmentation for confidence value from the highest range, middle range, until the lowest range as seen in Figure 12. The

system is now ready to recognize and assess the user's movement without confusing anymore.

```

1. if (TKD.publicConfidence >= 0)
2. {
3.   this.detek.Content = TKD.namaDetector;
4.   Hasil = TKD.publicConfidence * 100;
5.   if (Hasil >= 90 && Hasil <= 100)
6.   {
7.     this.nilai.Content = "A";
8.   }
9.   if (Hasil >= 55 && Hasil < 80)
10.  {
11.    this.nilai.Content = "B";
12.  }
13.  if (Hasil < 45)
14.  {
15.    this.nilai.Content = "C";
16.  }

```

Figure 12. Confidence Value Segmentation

4. RECOGNITION & ASSESSMENT

Before the system give the score for the object's movement, it should know what they are doing and the name of it like the explanation in previous part. This research only using 3 basic movements for Tae Kwon Do beginners, there are Ap Kubi Momtong Baro Jireugi, Ap Kubi Eolgol Makki, and Ap Seogi Eolgol Ap Chagi. AdaBoost algorithm has the same work like decision tree algorithm where it searching for the match condition and if nothing match with the condition then the system won't recognize the object's movement. Starting with the first name of the movement and check the condition is match or not because if it is not match, it will be moving to the next condition and so on until the system find a match.

If the system already finds a matching condition with the data, it will give score result based on the name of movement and user's behavior. Otherwise, the system will give nothing score result when the user's movement can be detected as seen in Figure 13.

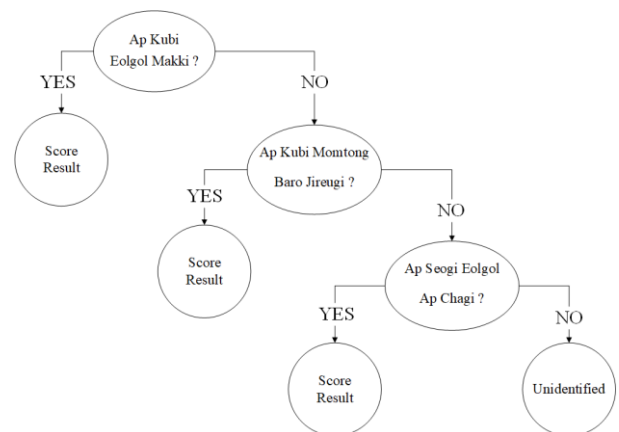


Figure 13. Recognition in Decision Tree

The whole process of how the recognition and assessment work can be seen in Figure 14 where the system should read the database and going through frame detection that the user's skeletal should be detected. The score result will be appear if the recognition process is done, when it is not recognized anything then the score will be none.

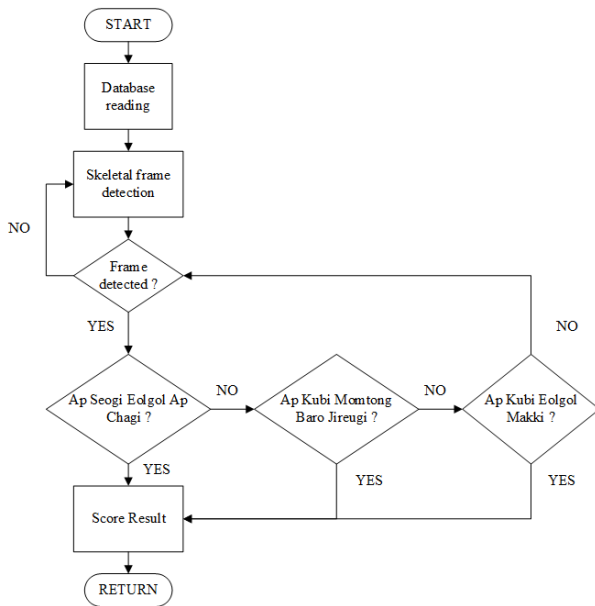


Figure 14. Recognition & Assessment Flowchart

5. SYSTEM INTERFACE

The system needs the interface for this application so the user knows and understand what to do and how good their score. The user can see the technique for each movement from the images and must choose to do between one of the three basic movements such as Ap Kubi Momtong Baro Jireugi, Ap Kubi Eolgol Makki, or Ap Seogi Eolgol Ap Chagi displayed in the monitor. The system can detect the user's movement correctly when he or she doing it once at a time, not doing it all at one time. They also can see what is the name of the movement in the monitor if their data match with the training data saved in the database which contain the correct technique from Sabeum Tae Kwon Do. After the data match and the name shown up, the score result will be shown up too, whether the user get an A, B or C for their technique as seen in Figure 15.

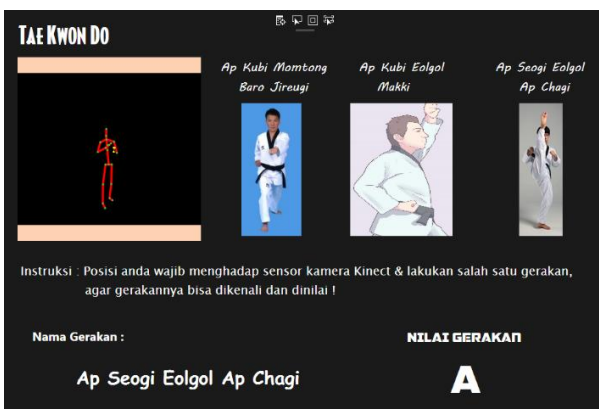


Figure 15. System Interface

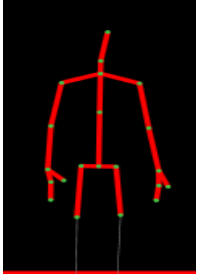
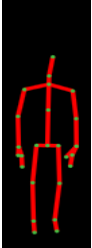
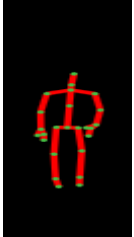
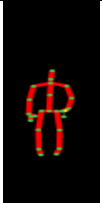
6. RESULT

6.1 User's Optimal Position

Kinect has disadvantage for its sensor range area detection, so to get the optimal position must be between in the range area like explained before. Calibration should be done here by comparing from the sensor reading with measuring device. The three movements just have 2 position of stances, between Ap Seogi (standing position) or Ap Kubi (one leg should be in front of the other one). This step is useful for the system

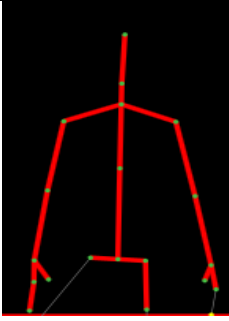
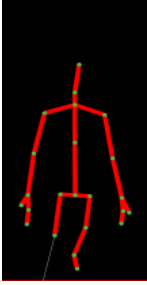
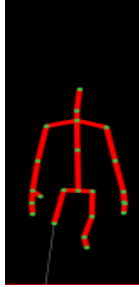
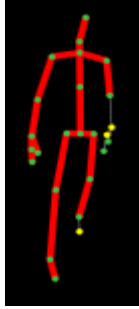
because it can detect the skeleton of the user perfectly without losing the ability of recognition and assessment.

Table 1. Ap Seogi Stances Calibration

No.	Ap Seogi		Ap Seogi Skeleton
	Measuring Device	Sensor	
1.	1,5 m	1,50 m	
2.	2 m	2,01 m	
3.	3 m	3,02 m	
4.	3,5 m	3,50 m	

As seen in Table 1 that it is a calibration process for Ap Seogi stances. Finding the optimal position is by looking at the digital skeleton detected by Kinect from 1.5 until 3.5 meters range area. The difference between the measuring device with Kinect's sensors is not that much so the conclusion is the system can do its job correctly for the Ap Seogi position. Based on the table, the optimal position for this stance when the user is in 3 meters because the sensors can read the full body perfectly and still left some spaces, in case the user have more than 1.8 meters of height so it won't be a problem.

Table 2. Ap Kubi Stances Calibration

No.	Ap Kubi		Ap Kubi Skeleton
	Measuring Device	Sensor	
1.	1 m	1,03 m	
2.	1,5 m	1,58 m	
3.	2 m	2,02 m	
4.	2,4 m	2,43 m	

The other stances, Ap Kubi, can be seen in Figure 17 for the calibration process. Following Ap Seogi's range area, the table give us the information that the difference is not that much too between measuring device and sensor, just like Ap Seogi so the system should have no problem too with these stances.

6.2 Movement Recognition

The system can detect perfectly for each movement with the correct names. Figure 18 shows the system can detect Ap Kubi Momtong Baro Jireugi.

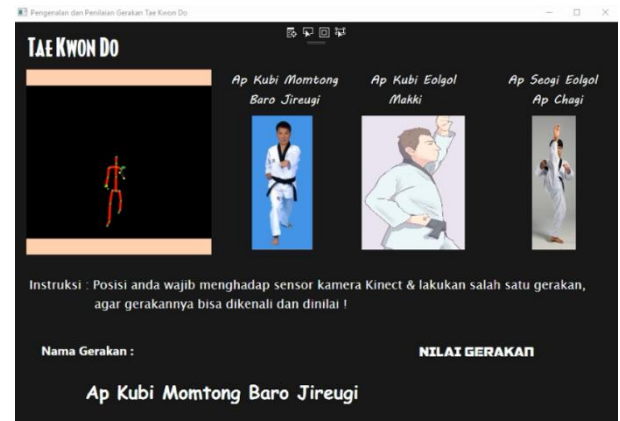


Figure 16. Ap Kubi Momtong Baro Jireugi

The picture shows that the user's movement having a match with the saved database or training data of the correct technique from Sabeum Tae Kwon Do. The user is literally doing a punch named Ap Kubi Momtong Baro Jireugi so the system doing well with this technique. The score result is not already shown there because this is still in recognition process. When this is done, then score result will be shown in the monitor.

Not only Ap Kubi Momtong Baro Jireugi but also Ap Kubi Eolgol Makki can be detected by the system. The difference between Ap Kubi Momtong Baro Jireugi with Ap Kubi Eolgol Makki is the technique. Ap Kubi Eolgol Makki is for defending the head of the user even though Ap Kubi Momtong Baro Jireugi and Ap Kubi Eolgol Makki have the same stances that is Ap Kubi. While the user doing this technique in front of Kinect and have a match with the database, it shows the name of the technique correctly as seen in Figure 17.

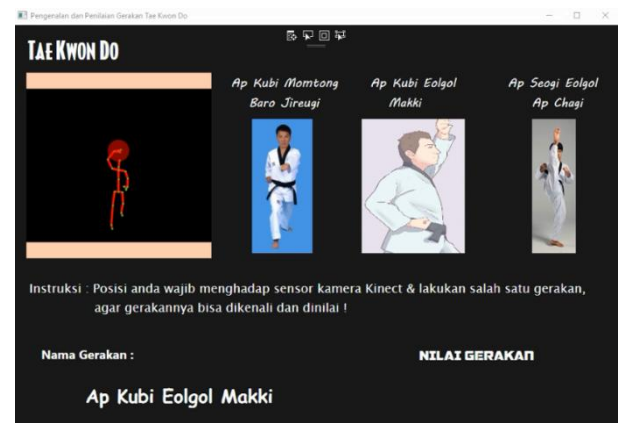


Figure 17. Ap Kubi Eolgol Makki

Last technique called Ap Seogi Eolgol Ap Chagi can be detected correctly too when the user movement have a match with the training data as seen in Figure 18.

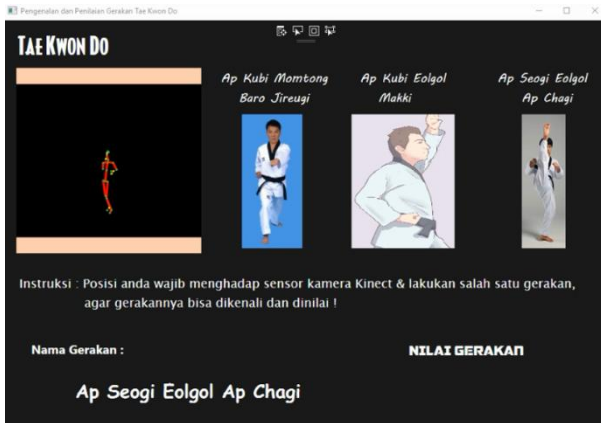


Figure 18. Ap Seogi Eolgol Ap Chagi

Accuracy of the recognition system for each system is different. Ten attempts were made to obtain it. First, let's see from the recognition process for all moves as seen in Table 3.

Table 3. Recognition Process Result

Predicted \ Supposed	Ap Kubi Momtong Baro Jireugi	Ap Kubi Eolgol Makki	Ap Seogi Eolgol Ap Chagi
	Ap Kubi Momtong Baro Jireugi	8	1
Ap Kubi Eolgol Makki	2	9	3
Ap Seogi Eolgol Ap Chagi	0	0	6

The system sometimes not working well to do its job based on the table. What it supposed to be become detected to another movement or technique. At least, when the user doing Ap Kubi Momtong Baro Jireugi and Ap Kubi Eolgol Makki are not detected as Ap Seogi Eolgol Ap Chagi because they are not using legs to kick.

Confusion matrix is useful enough to measure how good the system is and here is the formula for accuracy, precision, sensitivity, specificity, and F-score.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$PPV = \frac{TP}{TP+FP} \quad (6)$$

$$TPR = \frac{TP}{TP+FN} \quad (7)$$

$$TNR = \frac{TN}{TN+FP} \quad (8)$$

$$Fscore = \frac{2TP}{2TP+FP+FN} \quad (9)$$

Based on those formulas and Table 4, the accuracy is 86,7%, the precision is 80%, sensitivity is 80%, specificity is 90%, and F-score is 80% show that the system can detect Ap Kubi Momtong Baro Jireugi good enough.

Table 4. Ap Kubi Momtong Baro Jireugi's Confusion Matrix

Predicted \ Supposed	Ap Kubi Momtong Baro Jireugi	Not Ap Kubi Momtong Baro Jireugi
	Ap Kubi Momtong Baro Jireugi	8 (TP)
Bukan Ap Kubi Momtong Baro Jireugi	2 (FN)	18 (TN)

Based on the formulas and Table 5, the accuracy is 80%, the precision is 64.3%, sensitivity is 90%, specificity is 75%, dan

F-score is 75% show that the system can detect Ap Kubi Eolgol Makki good enough too.

Table 5. Ap Kubi Eolgol Makki's Confusion Matrix

Predicted \ Supposed	Ap Kubi Eolgol Makki	Not Ap Kubi Eolgol Makki
	Ap Kubi Eolgol Makki	9 (TP)
Bukan Ap Kubi Eolgol Makki	1 (FN)	15 (TN)

Based on the formulas and Table 6, the accuracy is 86.7%, the precision is 100%, sensitivity is 60%, specificity is 100%, and F-score is 75% show that the system can detect Ap Seogi Eolgol Ap Chagi good enough too.

Table 6. Ap Seogi Eolgol Ap Chagi's Confusion Matrix

Predicted \ Supposed	Ap Seogi Eolgol Ap Chagi	Not Ap Seogi Eolgol Ap Chagi
	Ap Seogi Eolgol Ap Chagi	6 (TP)
Bukan Ap Seogi Eolgol Ap Chagi	4 (FN)	20 (TN)

6.3 Movement Assessment

After going through recognition process, here comes the assessment process. The system is tested to assess 4 respondents. The score will be showed up how good the user's technique. The method to see the assessment is good or not by using Root Mean Squared Error (RMSE). The smaller error value is a sign that the system can do its job well. Here is the formula.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}} \quad (10)$$

where P = Predicted value, O = Actual value, n = Number of respondents

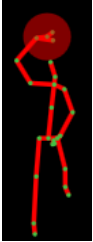
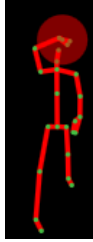
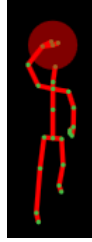
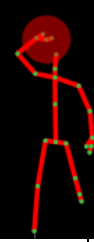
As a comparison, Sabeum or instructor of Tae Kwon Do will be asked for their point of view in assessing the respondent's technique. Their score will be used as a reference to the system's assessment.

Table 7. Ap Kubi Momtong Baro Jireugi Assessment

Respondent	1	2	3	4
Position				
System	0	40	66	81
Sabeum	60	75	78	76

While the system is assessing Ap Kubi Momtong Baro Jireugi as seen in Table 7, the error value for this technique or movement is 35.3 using RMSE formula.

Table 8. Ap Kubi Eolgol Makki Assessment

Respondent	1	2	3	4
Position				
System	89	83	100	81
Sabeum	60	70	85	65

While the system is assessing Ap Kubi Momtong Baro Jireugi as seen in Table 8, the error value for this technique or movement is 19.3 using RMSE formula.

Table 9. Ap Seogi Eolgol Ap Chagi Assessment

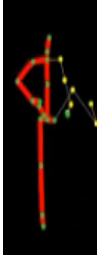

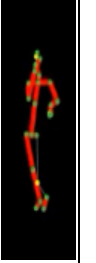

Respondent	1	2	3	4
Position				
System	64	100	67	21
Sabeum	60	68	90	65

Figure 27. Ap Seogi Eolgol Ap Chagi Assessment

While the system is assessing Ap Seogi Eolgol Ap Chagi as seen in Table 9, the error value for this technique or movement is 29.6 using RMSE formula.

7. CONCLUSION

The result of the system recognition and assessment for three basic movements of Tae Kwon Do can concluded that: the optimal range from Kinect to the user is 3 meters planted in a tripod with 1.03 meters of height; the recognition system is already good enough with total accuracy is 84,5%, total

precision is 81,4%, total sensitivity is 76,7%, total specificity is 88,3%, and total F-score is 76,7%; the assessment has error value for each movement from Ap Kubi Momtong Baro Jireugi is 35.3%, Ap Kubi Eolgol Makki is 19.3%, and Ap Seogi Eolgol Ap Chagi is 29.6%.

8. ACKNOWLEDGMENTS

This paper is supported by computer science postgraduate program department of computer science and electronics for the research funding of this work.

9. REFERENCES

- [1] Suryadi, V. Y. (2003). taekwondo: poomse tae geuk. Jakarta, Gramedia Pustaka Utama.
- [2] Schapire, R.E. (1999). A Brief Introduction to Boosting Analyzing the training error. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. doi:citeulike-article-id:765005.
- [3] Cheng, W. (2016). Pedestrian Detection Using an RGB-Depth Camera. 2016 International Conference on Fuzzy Theory and Its Applications (iFuzzy).
- [4] Saade, P., Joly, P. and Awada, A. (2013). Simulating actions for learning. Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM), 2013 IEEE 11th International Workshop of.
- [5] Kao, W.C., Hsu, S.C. and Huang, C.L. (2015). Human upper-body motion capturing using Kinect. ICALIP 2014 - 2014 International Conference on Audio, Language and Image Processing, Proceedings. p 245–250. doi:10.1109/ICALIP.2014.7009794.
- [6] Yun, Y., Changrampadi, M.H. and Gu, I.Y.H. (2014). Head pose classification by multi-class AdaBoost with fusion of RGB and depth images. 2014 International Conference on Signal Processing and Integrated Networks, SPIN 2014. p174–177. doi:10.1109/SPIN.2014.6776943.
- [7] Salazar, K.A., Garc, E.S. and Percybrooks, W.S. (2017). *Autonomous Recognition of Martial Arts Forms using RGB-D Cameras*. p1–5.
- [8] Kangtanto. (2012). Cara Kerja Kinect. <http://kangtanto.com/technology/cara-kerja-kinect/>
- [9] Microsoft. (n.d.). *Skeletal Tracking*. Accessed 13 May 2018, <https://msdn.microsoft.com/en-us/library/hh973074.aspx/>
- [10] Jana, A. (2012) Kinect for Windows SDK Programming Guide.