Multiple Classifier Learning of New Facial Extraction Approach for Facial Expression Recognition using Depth Sensor

Nattawat Chanthaphan[1], Keiichi Uchimura[1], Takami Satonaka[2] and Tsuyoshi Makioka[2]
[1]*Graduate School of Science and Technology, Kumamoto University, Kumamoto, Japan*
[2]*Electronic Systems Tech., Information Systems Tech., Kumamoto Prefectural College of Technology, Kumamoto, Japan*

Keywords: Emotion Recognition, Feature Extraction, Structured Streaming Skeleton, Depth Sensor.

Abstract: In this paper, we are justifying the next step experiment of our novel feature extraction approach for facial expressions recognition. In our previous work, we proposed extracting the facial features from 3D facial wire-frame generated by depth camera (Kinect V.2). We introduced the facial movement streams, which were derived from the distance measurement between each pair of the nodes located on human facial wire-frame flowing through each frame of the movement. The experiment was conducted by using two classifiers, K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM), with fixed values of *k* parameter and kernel. 15-people data set collected by our software was used for the evaluation of the system. The experiment resulted promising accuracy and performance of our approach in the last experiment. Consequently, we were anticipating to know the best parameters that would reflect the best performance of our approach. This time experiment, we try tuning the parameter values of K-NN as well as kernel of SVM. We measure both accuracy and execution time. On the one hand, K-NN overcomes all other classifiers by getting 90.33% of accuracy, but on the other hand, SVM consumes much time and gets just 67% of accuracy.

1 INTRODUCTION

Due to the growing of the technology industry, the researchers and the inventors are trying to develop the system that could think in the same way or has the intelligence as human being. The intelligent systems that could deal with human properly have been being developed for several decades and there were numerous valuable applications or systems were exposed to this world so far. One of the most important fields is computer vision which tries to make computer to be able to comprehend the visualization of human.

In this paper, we will concentrate on one of the most popular computer vision techniques called human facial emotion recognition. In our previous works (Chanthaphan et al., 2015-2016), we introduced a novel approach for extracting facial features from the moving facial skeleton for facial emotions recognition by using depth camera instead of using color/grayscale camera (2D) or 3D structure. In the experiment, we have used the data set that was collected by our designed software since there was no open data set (the moving sequences of coordinates on facial wireframe) for our approach distributed on any distributor websites. The result from the experiment showed a promising consequence, nevertheless, there were several parameters to be concerned, for instance, number of clusters of template dictionary, *k* parameter of K-NN, kernel function of SVM, structure of facial skeleton and etc.

Since there are many researches that are related to this field, we would like to sample some of well-known algorithms as follows. 2D based approaches consist of Active Shape Models (ASM) (Cootes et al., 1992), Active Appearance Models (AAM) (Cootes et al, 1998), Constrained Local Models (CLM) (Cristinacce, 2006) and etc. For 3D image, 3D Morphable Models (3DMM) (Blanz, 1999) is the famous one. Additionally, there was a new approach of Baltrusaitis et al. (2012) that introduced the use of depth camera combining with CLM. All those mentioned approaches indeed worked very well and they are also renowned. However, they still have some variations and conditions to be concerned such as head pose, head orientation, distance from camera, light condition, skin tone and so forth.

Chanthaphan, N., Uchimura, K., Satonaka, T. and Makioka, T.
Multiple Classifier Learning of New Facial Extraction Approach for Facial Expressions Recognition using Depth Sensor.
DOI: 10.5220/0005948000190027
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 5: SIGMAP*, pages 19-27
ISBN: 978-989-758-196-0

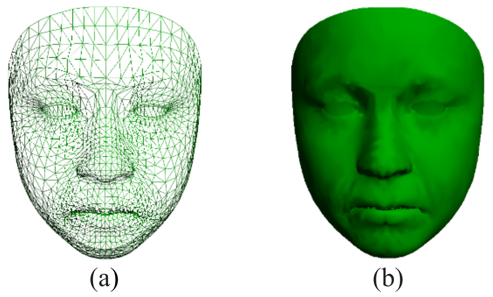

Figure 1: Facial skeleton generated by Kinect HD face API. This skeleton consists of 1347 vertices: (a) Wire-frame mode, (b) Solid mode.

In contrast, our based approaches could reduce those mentioned constraints by employing the approach of human gesture recognition of Zhao et al. (2014) which indicated about how to address the intra-class variations. Intra-class variations consist of four variations as follows.

- Viewpoint variation: This variation describes about the relation between human body and viewpoint of the camera.
- Anthropometry variation: This variation is related to the difference between human body sizes which do not affect the human movement.
- Execution rate variation: This variation indicates the problem with different frame rate of the camera or the moving speed of human.
- Personal style variation: This variation is about the difference of human performing their action differently.

The new sensory device, namely Kinect V2 has been used in order to generate the motion stream of human face. Figure 1 illustrates the facial skeleton (wire-frame) generated by using Kinect HD face API. In this paper, we have used the same method as in the previous works, but this time, we concentrated on parameter tuning of classifiers, for example, *k* parameter of K-Nearest Neighbors (K-NN) as well as the kernel function of Support Vector Machine (SVM). At the end, the most suitable parameter and kernel that suit our feature extraction approach will be summarized.

2 RELATED WORKS

As we have mentioned about various variations in the introduction, image pre-processing is conducted prior to the feature extraction phase. One of pre-processing technique is head pose estimation. In the book of Baggio et al. (2012), it stated a well-known method to do 3D head pose estimation by using AAM and POSIT (Pose from Orthography and Scaling with Iterations) (Dementhon et al., 1995). The principal component analysis (PCA) was also used to reduce the number of parameters of model and the Delaunay Triangulation (DT) (Lawson, 1972) was used to create the statistical texture of the AAM. In the paper of Zhu et al. (2012), it indicated that RGB camera-based approach had very expensive computation time. Anyhow, new sensory device was released in 2010 named Microsoft Kinect, the abilities of this device were described in this article of Zhang (2012) and we could reduce our effort on pre-processing using Microsoft Kinect.

Piana et al. (2014) also made use of Kinect to recognize human emotion. They proposed using 3D human body skeleton generated by Kinect to distinguish human emotions, whereas the overall accuracy was just 61.3%. The result of this research showed that human body gestures were not a good description for emotions.

Mao et al. (2015) described that because human faces were 3D object, 2D images which were captured by RGB camera were insufficient to represent the geometrical feature. Therefore, they decided to use two sorts of feature generated by Kinect. The first one was Animation Units (AUs) which was the value that could be obtained by using Kinect face tracking software development kit (SDK). There were six AUs provided by this SDK (brow raiser, brow lower, lip raiser, lip stretcher, lip corner depressor and jaw lower) which had value between -1 and 1. The other one, they called Feature point positions (FPPs) which were 45 coordinates of 45 points obtained by Kinect face tracking SDK as well. The result of classification was computed by fusing result of 30 consequent frames, both from AUs and FPPs. So, this was pre-segmentation based approach which could not handle execution rate variation. For the evaluation, they have collected their own database named UJS Kinect emotion database (USJ-KED) from ten actors with variations in five poses (-30∘, -15∘, 0∘, 15∘ and 30∘) and seven emotions (anger, disgust, fear, happiness, neutral, sadness and surprise). Multi-pose was requiring in their database because their approach was still not able to deal with viewpoint variation and anthropometry variation. As they were using Kinect as the sensor, we have compared our approach with them. However, it was just a rough comparison due to several different factors, like database and number of emotional classes they used.

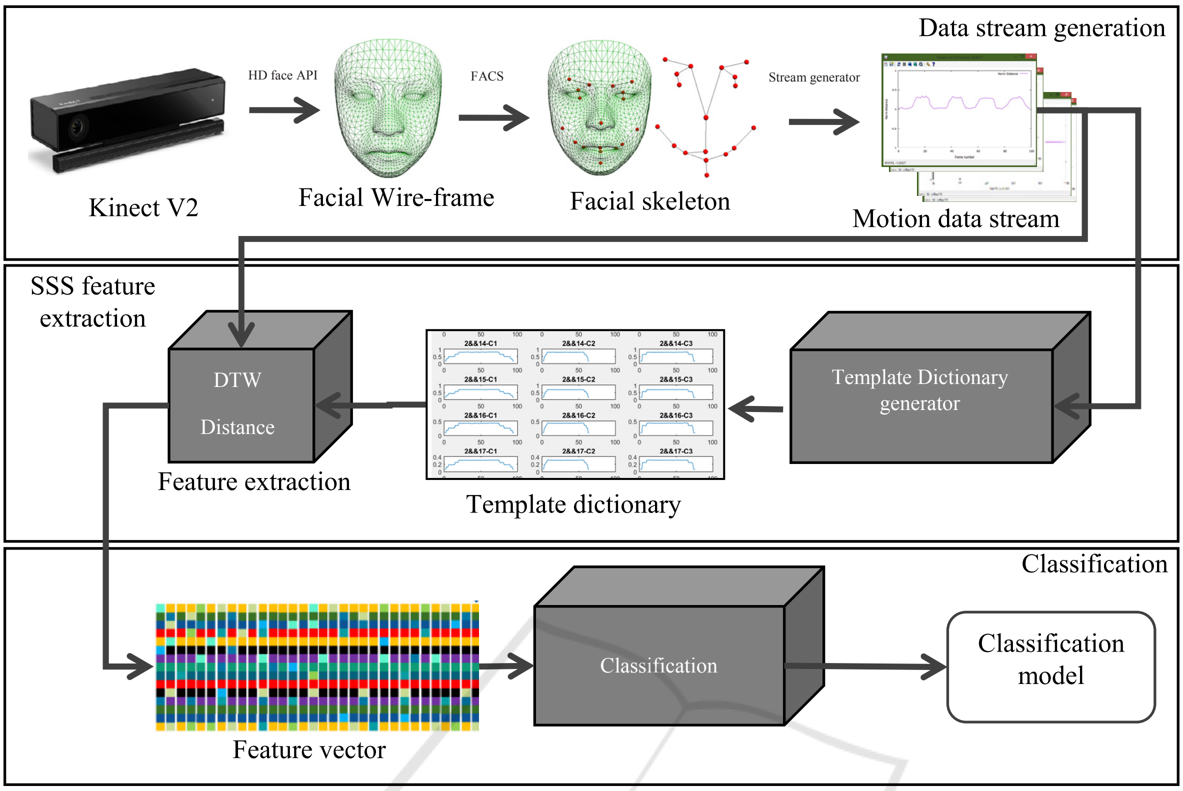

Figure 2: The Overall of frame work of the proposed approach.

In the work of Zhao et al. (2014), they also introduced a novel approach called Structured Streaming Skeleton (SSS). This approach succeeded on handling all those variations. So, we have decided to adapt SSS feature extraction approach to our approach in order to distinguish human facial emotion rather than human body gesture.

The other related works besides SSS feature extraction were Facial Action Coding System (FACS) of Ekman et al. (1976) the points of the interesting for feature extraction process. Dynamic Time Warping (DTW) distance as well-known technique to find minimal alignment between two sequences from work of Sakoe et al. (1978) was used to find value representing each attribute of the feature vector and finally work of Sakurai et al. (2007) to find optimal end frame of scanning.

3 PROPOSED APPROACH

Our approach is based on SSS feature extraction as described in the related work. Therefore, the framework of our system could be shown as in Figure 2.

3.1 Data Stream Generation

By the help of Kinect working together with HD face Application Programming Interface (API), we can get the facial wire-frame which responds to the facial movement instantly. The wire-frame consists of 1347 vertices. To select the proper vertices that we are going to use for generating the stream, we must refer to the paper of Ekman et al. (1976) that describes about FACS then we can summarize the points that represent each emotion as in Table 1. Then, we will get the points that could represent all those emotions as shown in Figure 3. Next step, we will generate the motion streams by calculating the distance between each pairwise joint and normalizing them by the path between each joint.

Equations 1 and 2 are used for the calculation. nC2 combination equation 3 is used to calculate the number of pairs (rows of streams).

$$S_{ij}(t)=\frac{E\ (p_i(t),p_j(t))}{Path_{ij}(t)} \quad (1)$$

$$Path_{ij}(t)=\sum_{m=1}^{\#Node\_ij-1} E\left(p_{L_m}(t),p_{L_{m+1}}(t)\right) \quad (2)$$

$$Rows\_of\_Streams = \frac{N(N\text{-}1)}{2} \tag{3}$$

Where, the definition of each symbol in equations 1, 2 and 3 is justified in Table 2.

In this case, we have 18 feature points (N = 18). Therefore, the number of rows will be 153.

Finally, we will have the normalized distances of all 153 rows for all frames and the motion data streams are generated.

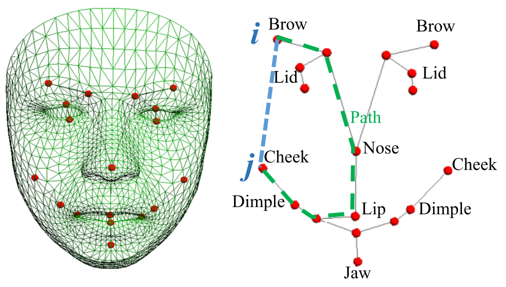

Figure 3: Facial skeleton consists of 18 feature points. The blue dotted line indicates direct distance between points *i* and *j*. The green dotted line indicates path distance between points *i* and *j*.

Table 1: FACS (Facial Action Coding System).

Emotion	FACS
Anger	Brow Lowerer + Upper Lid Raiser + Lid Tightener + Lip Tightener
Contempt	Lip Corner Puller + Dimpler
Disgust	Nose Wrinkler + Lip Corner Depressor + Lower Lip Depressor
Fear	Inner Brow Raiser + Outer Brow Raiser + Brow Lowerer + Upper Lid Raiser + Lid Tightener + Lip Stretcher + Jaw Drop
Happiness	Cheek Raiser + Lip Corner Puller
Sadness	Inner Brow Raiser + Brow Lowerer + Lip Corner Depressor
Surprise	Inner Brow Raiser + Outer Brow Raiser + Upper Lid Raiser + Jaw Drop

Table 2: Notations of equations 1, 2 and 3.

Symbol	Description
$S_{ij}(t)$	Normalized distance
i,j	Point index
t	Frame index
$p_i(t)$, $p_j(t)$	Coordinate (x,y and z) of points
L_m	Sorted point indices list of particular $Path_{ij}$ indexed by m
$E(p_i(t),p_j(t))$	Euclidean distance
$Path_{ij}(t)$	Path distance, see Figure 3 for more detail.
N	Number of points on facial skeleton
$\#Node\_ij$	Number of points between i and j

3.2 SSS Feature Extraction

We could separate this phase into two steps; Template dictionary generation and feature extraction.

(1) Template Dictionary Generation

Firstly, we have to manually split the motion streams to several gesture instances.

Secondly, we have to pick the instance that has the highest frame number to be the reference instance, and then compute DTW distance between the reference sequence and the rest of sequences.

After receiving DTW distances for all pairs of gesture instances, we have to do ascending sorting (quick sort). Please be informed that DTW distance can indicate the similarity between two sequences. After sorting, the gesture instances those resemble each other - DTW distance values are close - will be located close together.

Next, we have to group the gesture instances that have DTW distance close to each other to be in the same cluster, and then average all gesture instances in the same cluster to be just one sequence per each cluster. To average the gesture instances that have different frame numbers, we need to shift the shorter sequence with the difference of frame number divided by two. The equations and algorithm to calculate motion template dictionary are as follows.

Where, the definition of each symbol in equations 4, 5 and 6 is justified in Table 3.

Then, we will have the template dictionary generated from all gesture instances as shown in Figure 4. This time, the number of clusters per one row is fixed to five.

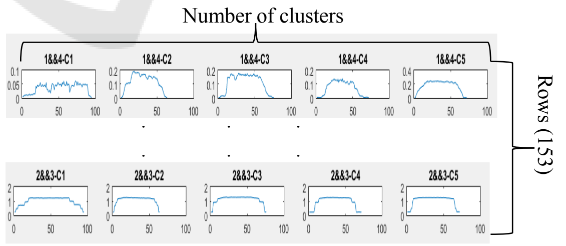

Figure 4: Template dictionary (Rows * Number of clusters).

```
Input: Clusters of gesture instances
Output: Template dictionary
for $\forall d \in \{0,\ldots,D\text{-}1\}$ do
    for $\forall g \in \{0,\ldots,G\text{-}1\}$ do
```

$$A_d^g(0{:}F_d^g) = \frac{M_d^g(0{:}F_d^g)}{N_d^g} \tag{4}$$

for $\forall k \in \{0,\ldots,N_d^g\text{-}1\}$ **do**

$$Fs_{dk}^g = \frac{F_d^g\text{-}f_{dk}^g}{2} \tag{5}$$

for $\forall t \in \{Fs_{dk}^g,\ldots,f_{dk}^g+Fs_{dk}^g\}$ **do**

$$A_d^g(t) = A_d^g(t) + \frac{S_{dk}^g\left(t\text{-}Fs_{dk}^g\right)}{N_d^g} \tag{6}$$

(2) Feature Extraction

After receiving template dictionary which is generated from the gesture instances of human face, we will use it to extract the feature vectors from the stream. Therefore, all streams will be scanned again to calculate DTW distance starting from current frame with each sequence inside template dictionary. The minimum DTW distance will be used as one attribute of feature vector. Equations 7, 8, 9 and 10 are the equations to calculate DTW distance.

Given sequences X and Y

$$f(0,0) = 0,\ f(i,0) = f(0,j) = \infty \tag{7}$$

$$f(i,j) = (x_i\text{-}\ y_j)^2 + \min\begin{cases} f(i,j\text{-}1) \\ f(i\text{-}1,j) \\ f(i\text{-}1,j\text{-}1) \end{cases} \tag{8}$$
$$(i = 1,\ldots,n;\ j = 1,\ldots,m)$$

$$C = \begin{bmatrix} 0 & \infty & \cdots & \infty \\ \infty & \ddots & & \\ \vdots & & \ddots & \\ \infty & & & f(n,m) \end{bmatrix};C \in R^{n\times m} \tag{9}$$

$$D(X,Y) = f(n,m) \tag{10}$$

Where, the definition of each symbol in equations 7, 8, 9 and 10 is justified in Table 3.

The stream monitoring technique of (Sakurai et al., 2007) is adapted to our approach to determine the number of frame of sequence Y by finding the optimal ending frame. The process of scanning can be shown in Figure 5. Each row of streams, starting from current frame to the optimal ending point, will be scanned with every sequence in the same row of template dictionary to get DTW distance. Each row will produce G attributes of feature vector. Therefore, after this step, the feature vector for one frame will consist of 153*5 = 765 attributes (in case of G = 5) and be ready for the classification process.

Table 3: Notations of equations 4 to 10.

Symbol	Description
$A_d^g(0\text{:}F_d^g)$	Averaged sequence of cluster g in row d, Frame starts from 0 to F_d^g
g	Cluster index
G	Number of clusters
d	Row d
D	Number of rows
F_d^g	Maximum frame number of cluster g in row d
N_d^g	Number of sequences to be averaged in cluster g row d
Fs_{dk}^g	Number of frame to be shifted of sequence k in cluster g of row d
f_{dk}^g	Number of frames of sequence k in cluster g of row d
k	Sequence index
t	Frame index
$S_{dk}^g(0\text{:}f_{dk}^g)$	Sequence k of cluster g in row d, Frame starts from 0 to f_{dk}^g
$M_d^g(0\text{:}F_d^g)$	The gesture instance whose frame number is maximum of cluster g in row d
$f(i,j)$	The element i,j inside C
C	DTW cost matrix
n	Maximum frame of sequence X
m	Maximum frame of sequence Y
$D(X,Y)$	DTW distance between X and Y
∞	Very high value
X	Sequence in template dictionary
Y	Sequence starting from current frame to optimal ending frame

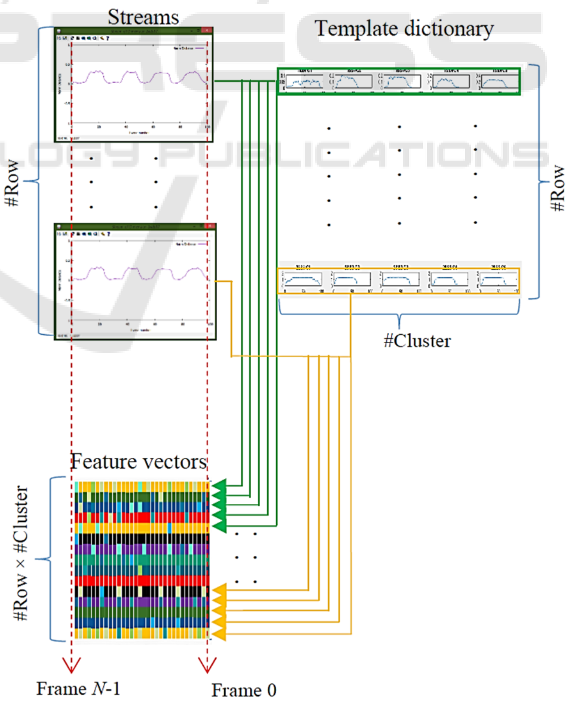

Figure 5: Overall process of feature extraction step, each row of streams will produce G attributes of feature vector. After the process of scanning, the feature vector for one frame will consist of G (number of clusters of template dictionary) × number of rows attributes.

4 EXPERIMENTS

Since there was no data set available for our proposed approach, we have presented the method to construct the data set by ourselves. We designed and developed the software to collect the data set and the Graphical User Interface (GUI) of our software is shown in Figure 6. Data set collector software was developed by C# .net with Kinect API (all are packed with Kinect for Windows SDK).

The software was able to collect eight emotions including happiness, sadness, surprise, fear, anger, disgust, contempt and neutral. The actors had 15 seconds (325 frames) for expressing each emotion. Each frame consisted of 1347 coordinates and it would be reduced after FACS selection process.

This time, we have collected data set from fifteen actors for our experiment (eight emotions per each). In each emotion, they have been asked to freely act three times of emotions in according with the emotion label shown on the screen because we wanted to get the data that intuitively represented their emotions. Therefore we had 325*8*15 = 39,000 frames which could be separated into 3*8*15 = 360 gesture instances for template dictionary generation phase. The system environment for our experiment was Intel® core™ i5-4570 3.20GHz, 4 GB DDR2, Windows 8.1x64, GPU NVIDIA GeForce GTX 750 Ti. We decided to use native C++ in order to be able to use CUDA (Yang et al., 2008) architecture provided by NVIDIA GeForce GPU which could activate multi-thread processing on GPU. Therefore our feature extraction was executed in parallel processing.

Due to the constraints of time and system environment, we have fixed the number of clusters in template dictionary to five clusters ($G = 5$).

As we have mentioned in an introduction, the goal of our current work was to find the best fitting parameter of classifier. From our previous work, we have used K-NN and SVM. So, we continued using these classifiers with various values and kernel functions. 10-fold cross validation was used for the performance and accuracy evaluation. An open source software named RapidMiner which was popular, convenient and reliable described in paper of Jović et al. (2014) was used in classification phase.

Figure 7(a) shows the result of K-NN classifier working with various values of k parameter. As you might see, the accuracy linearly decreases when the value of k is raised. Consequently, k equals to one is the best fitting value which reflects the best accuracy of 90.33%.

The execution time of K-NN is quite linear. 1,200 frames were sampled from 39,000 frames for training and testing. As shown in Figure 7(b), the execution time swings between 1.7 to 1.8 seconds, whether increasing or decreasing k.

Figure 7(c) manifests the result of classifying with SVM. The kernels we used for the experiment consisted of five kernels - Dot, Radial, Linear, Polynomial degree 2 and degree 3 respectively. The result seems to be worse than K-NN, especially with polynomial degree three kernel. It shows the worst accuracy which is just 18.08%, nevertheless, if we change to polynomial degree two, it reveals an outstanding result among those kernel functions of SVM. The execution time of SVM is severely high comparing to K-NN as shown in Figure 7(d).

The reason why K-NN distinctly beats SVM is the problem of our feature vectors having very high dimension or attribute number. Even though SVM has many kernel functions to modify the dimension of input space, there seems to be no kernel function that could make a good feature space linearly separable by the hyperplane.

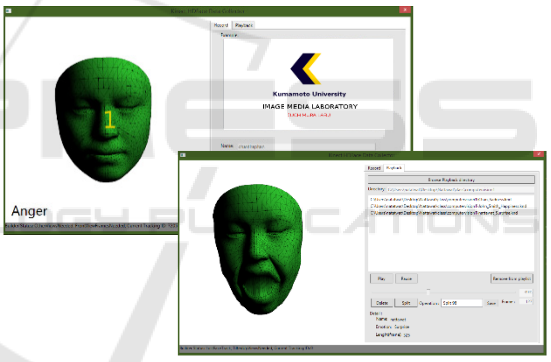

Figure 6: Data set collector software.

Tables 4 and 5 are the confusion matrices of K-NN with k = 1 and SVM with Polynomial degree two respectively. The accuracy of detecting fear is the worst. In contrast, the detection accuracy of neutral overcomes all other emotions. It is more confusion in fear because the expression is ambiguous and rather similar to surprise. Therefore, there is a high probability to get confused with surprise, whereas the motion of neutral is clearer. So, we could get the best accuracy in this case.

Table 6 shows the accuracy comparison between proposed approach and state-of-the-art approach (Mao et al., 2015). Since there are several different factors, such as data set, system environment, number of classes, classifier etc. mentioned in related works section, the information in this table is

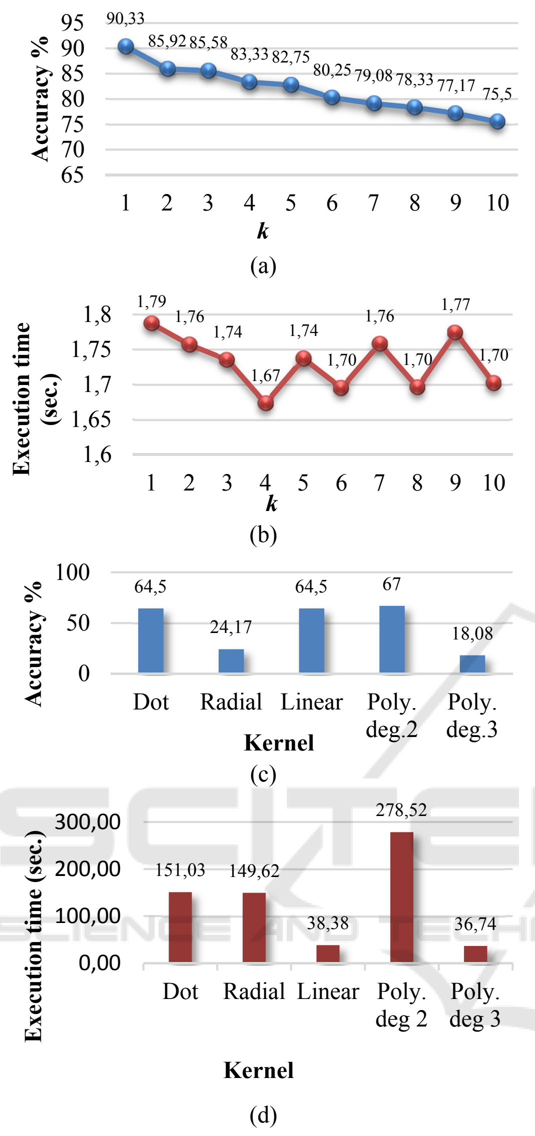

Figure 7: Experimental result. (a) Accuracy of K-NN, (b) Execution time of K-NN, (c) Accuracy of SVM, (d) Execution time of SVM.

just a rough comparison. We want to manifest the accuracy comparison between approaches that use contemporary depth sensor, particularly Kinect, in this table. The average accuracy of the proposed approach is 66.66% for SVM with polynomial degree two and 90.19% with $k = 1$ for K-NN which is higher than the average accuracy of the state-of-the-art approach which is 80.57%. Furthermore, our approach could eliminate intra-class variation occurring in their approach which could help reducing some constraints in their work, for example, reducing number of data set (viewpoint, anthropometry and personal style variation are eliminated), and avoiding pre-segmentation before classification (execution rate variation is eliminated).

5 CONCLUSIONS

By the helps of several high efficient techniques and sensors, we could achieve all those goals we have defined in the first place.

Firstly, our proposed approach could reduce intra-class variations from human facial emotion recognition system.

The most suitable classifier for our feature extraction approach was K-NN with k = 1, which resulted the best accuracy and speed.

Our approach succeeded on adapting SSS approach to facial emotion detection. Moreover, the accuracy was 90.33% +/- 1.91% for K-NN and 67% +/- 4% for SVM.

The accuracy of our proposed approach (K-NN) outperformed the state-of-the-art approach (Mao et al., 2015) around 9.62% higher.

From our study, we have learnt that intra-class variation was one of the most extreme factors which could decrease the accuracy of facial recognition system and this was the evidence to proof our hypothesis about the persistence of intra-class variation in facial emotion expression. SSS feature extraction approach might not be the best approach to solve the intra-class variations, but it showed a promising consequence of utilizing facial motion stream and depth sensor and it proofed that we could use depth sensor to classify human emotions.

In the future, we planned to study the rest of parameters, for example, number of clusters, number of points on the facial skeleton and etc. in order to find the most appropriate value which could result better accuracy and performance. Furthermore, we planned to implement real-time classification with SSS feature extraction approach.

Table 4: SSS feature vector with K-NN, $k = 1$.

Actual class↓	Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Happiness	**94.67**	0.67	0.00	1.33	0.00	0.00	0.67	2.67
Sadness	2.00	**88.67**	2.00	1.33	1.33	1.33	2.00	1.33
Surprise	1.33	2.67	**88.67**	4.00	0.00	1.33	0.67	1.33
Fear	5.33	2.67	3.33	**82.67**	0.67	3.33	0.67	1.33
Anger	0.00	2.67	4.00	0.00	**86.00**	2.67	2.00	2.67
Disgust	0.00	0.00	0.67	0.67	0.67	**94.67**	3.33	0.00
Contempt	1.33	1.33	0.00	1.33	2.67	0.67	**91.33**	1.33
Neutral	1.33	0.67	0.00	0.67	0.67	0.00	0.67	**96.00**

Table 5: SSS feature vector with SVM, Polynomial degree 2.

Actual class↓	Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Happiness	**78.00**	4.67	0.00	2.67	2.67	4.00	0.00	8.00
Sadness	2.67	**69.33**	4.00	5.33	3.33	1.33	4.00	10.00
Surprise	4.67	6.00	**67.33**	10.00	1.33	2.00	3.33	5.33
Fear	15.33	7.33	9.33	**50.00**	3.33	3.33	4.00	7.33
Anger	2.00	4.00	0.67	2.00	**63.33**	2.67	14.00	11.33
Disgust	6.00	7.33	1.33	4.67	12.00	**55.33**	8.67	4.67
Contempt	6.00	1.33	0.00	2.67	2.67	7.33	**69.33**	10.67
Neutral	5.33	4.00	0.67	2.67	1.33	2.67	0.00	**83.33**

Table 6: Accuracy comparison between each approach.

Approach↓	Happiness	Sadness	Surprise	Fear	Anger	Disgust	Neutral	Average
Mao et al., (2015)	75.58	73.74	96.40	80.00	79.27	79.54	79.52	**80.57**
Proposed approach [*1]	94.67	88.67	88.67	82.67	86.00	94.67	96.00	**90.19**
Proposed approach [*2]	78.00	69.33	67.33	50.00	63.33	55.33	83.33	**66.66**

*1 K-NN with $k = 1$, *2 SVM with Polynomial degree 2

REFERENCES

Chanthaphan, N., Uchimura, K., Satonaka, T., Makioka, T., 2015. Facial emotion recognition based on facial motion stream generated by Kinect. In *11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. pp. 117-124.

Chanthaphan, N., Uchimura, K., Satonaka, T., Makioka, T., 2016. New feature extraction method for facial emotion recognition by using Kinect. In *The Korea-Japan joint workshop on Frontiers of Computer Vision (FCV)*. pp. 200-205.

Cootes, T. F., Taylor, C. J., 1992. Active shape models - Smart snakes. In *Proc. British Machine Vision Conference (BMVC)*. pp. 266-275.

Cootes, T. F., Edwards, G. J., Taylor, C. J., 1998. Active appearance models. In *5th European Conference on Computer Vision (ECCV)*. vol. 2, no. 1, pp. 484-498.

Cristinacce, D., Cootes, T. F., 2006. Feature detection and tracking with constrained local models. In *Proc. British Machine Vision Conference (BMVC)*. vol. 3, no. 1, pp. 929-938.

Blanz, V., Vetter, T., 1999. A morphable model for the synthesis of 3D faces. In *Proc. the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. pp. 187-194.

Baltrusaitis, T., Robinson, P., Morency, LP., 2012. 3D constrained local model for rigid and non-rigid facial tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2610-2617.

Zhao, X., Li, X., Pang, C., Sheng, Q. Z., Wang, S., Ye, M., 2014. Structured streaming skeleton - a new feature for online human gesture recognition. In *ACM Trans. Multimedia Comput. Commun. Appl.*. vol. 11, no. 1, pp. 1-18.

Baggio, D. L., Emami, S., Escrivá, D. M., Ievgen, K., Mahmood, N., Saragih, J., Shilkrot, R., 2012. *Mastering OpenCV with practical computer vision projects*. Packt Publishing Ltd. Birmingham. pp. 235-260.

Dementhon, D. F., Davis, L. S., 1995. Model-based object pose in 25 lines of code, In *International Journal of Computer Vision*. vol. 15, no. 1-2, pp. 123-141.

Lawson, C. L., 1972. Transforming triangulations, In *Discrete Mathematics*. vol. 3, no. 1, pp. 365-372.

Zhu, X., Ramanan, D., 2012. Face detection, pose estimation, and land mark localization in the wild, In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*. pp. 2879-2886.

Zhang, Z., 2012. Microsoft Kinect sensor and its effect. In *IEEE Computer Society*. vol. 19, no. 2, pp. 4-12.

Piana, S., Staglianò, A., Odone, F., Verri, A., Camurri, A., 2014. *Real-time automatic emotion recognition from body gestures*. Cornell university library: Computing Research Repository (CoRR), pp. 1-7.

Mao, Q. R., Pan, X. Y., Zhan, Y. Z., Shen, X. J., 2015. Using Kinect for real-time emotion recognition via facial expressions. In *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 4, pp. 272-282.

Ekman, P., Friesen, W., 1976. Measuring facial movement. In *Environmental psychology and nonverbal behaviour*. pp. 56-75.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Trans. on Acoustics, Speech and Signal Processing*. vol. 26, no. 1, pp. 43-49.

Sakurai, Y., Faloutsos, C., Yamamuro, M., 2007. Stream monitoring under the time warping distance. In *Proc. IEEE International Conference on Data Engineering (ICDE)*. pp. 1046-1055.

Yang, Z., Zhu, Y., Pu, Y., 2008. Parallel Image Processing Based on CUDA. In *Computer Science and Software Engineering*. vol. 3, no. 1, pp.198-201.

Jovic, A., Brkic, K., Bogunovic, N., 2014. An overview of free software tools for general data mining. In *Proc. Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. pp. 1112-1117.