Integrated Internet of Things (IoT) technology device on smart home system with human posture recognition using kNN method

M I Siddiq¹, I P D Wibawa¹ and M Kallista²

¹ Electrical Engineering, School of Electrical Engineering, Telkom University, Bandung, Indonesia
² Computer Engineering, School of Electrical Engineering, Telkom University, Bandung, Indonesia

*idrissiddiq@student.telkomuniversity.ac.id

Abstract. IoT device technology is currently developing rapidly, for example in smart home systems that have several features including lighting, surveillance security, temperature control, water sensors, and smart electricity. IoT device consists of smart electricity integrated with human action recognition using sensor vision are developed in this work. In smart electricity system, we build some relays controlled by smartphone applications and web-based platforms. We can control the relays and monitor the voltage, current, and power used from electricity appliances that are connected to our IoT device. In human action recognition, we use a single RGB camera to capture some human poses into spatiotemporal sequences to get data for training. There are six poses for testing scenario, these poses will be clustered using kNN (k-Nearest Neighbor) method. Each human action that is recognized will be connected to an IoT device for controlling the switching mode on the relays in smart electricity system. The result in this experiment shows that the system successfully reads every single posture with quite good accuracy performance using confusion matrix.

1. Introduction

Pose recognition aims to recognize expressions based on human body poses [1]. These recognizable expressions usually involve the hands, arms, head, and other limb positions. Poses can be used to represent certain messages because each pose can be distinguished from one another. This introduction is usually used to facilitate human and computer interfaces [2]. Poses can be recognized by a computer through the camera. The computer sends a message to the server to carry out certain commands. After that, messages obtained by the server are forwarded to various other devices that are connected by the internet, one of which is the IoT device. Therefore, pose recognition can be used to give commands to smart sockets while connected to the internet. Smart socket is a socket that using internet of things technology so that it can be controlled and power monitored through the application. The human action recognition process is carried out in two stages, first reading the joint skeleton and then learning the posture classification. In the skeleton reading process some researchers developed the following methods i.e. Stacked hourglass was introduced in 2016 [3], Vnect [4], OpenPose [5], VICON [6], PoseNet [7] were introduced in 2017; and in 2019 there is Pose Regression [8]. Stacked hourglass used bottom-up and top-down process but had relatively long computing time for training. Vnect is applied
to virtual reality for gaming and character control by constructing 2D images into 3D coordinates using the skeleton fitting method, and using smoothness filtering. OpenPose has full-featured full body, finger and face detection that is reliable enough for multi-person detection. The PAF (part affinity field) method can accommodate occlusion and overlapping cases. VICON can detect unusual (complex) diverse poses but the training data required is relatively long (21 days, image training approximate 10,000 images). PoseNet uses the Generative Adversarial Networks (GANs) architecture; in the case of backlight where the background is too bright, the skeleton detection process reduces performance accuracy. Pose Regression can estimate poses with high fps (frame per second) using the map regression method.

After the skeleton detection process, the next step is the classification of movements. Some of the movement recognition classification methods are kNN (k-Nearest Neighbor) [9], HMM (Hidden Markov Model) [10], RF (Random Forest) [11], SVM (Support Vector Machine) [12], Template Matching [13], Graph and ELM (Extreme Learning Machine) [14]. For the classification model using deep learning, here are some methods that can be applied by CNN (Convolutional Neural Network) [15], RNN (Recurrent Neural Network) [16], LSTM (Long-Short Term Memory) [17] and YOLO (You Only Look Once) [18]. The deep learning model and the other models have good generalization performance in terms of accuracy for the classification of human action recognition.

Smart socket has been built as IoT device that use Android applications as user and device interfaces. The android application can give commands in the form of turning on or off the power on the socket. IoT is a concept that aims to expand the benefits of internet connectivity that is connected continuously in terms of sharing data, remote control, and various other things. IoT is used to support the MQTT communication protocol that can connect a PC with a microcontroller. In this final project, IoT is used as a medium for sending topics from a PC to a microcontroller. MQTT is used as interface between camera and smart socket. MQTT is a publish and subscribe-based lightweight message protocol used above the Transmission Control Protocol or Internet Protocol (TCP/IP) protocol [19]. TCP/IP is an interconnection protocol through hosts, networks, and the internet [20]. MQTT has a small data packet size so that one server can handle thousands of clients. This makes MQTT suitable for Machine to Machine (M2M) communications that require small footprint codes or limited networks. The publish-subscribe message pattern on MQTT requires a broker who is responsible for distributing message topics to interested clients. The users only need to make certain poses in front of the camera that has been installed in home to turn on/off the smart socket. The novelty in this paper is to build Internet of Things (IoT) device in smart home using vision sensor to recognize human pose using kNN method.

2. Design of System

In this paper, a socket system is connected to the camera using MQTT broker. The camera sensor detects human poses and then calculate the key point position using PoseNet. After getting the key point position, the poses will be labeled as target for training process. Moreover, for the testing process, the key points will be obtained and compared with training data in order to classify the human pose, as labelled in previous step, by using the kNN method. kNN determines the classification based on the k-nearest objects from the target or testing point, given data cluster (labelled poses). To determine the distance value, a calculation using Euclidean is performed [9].

\[
D(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - x_t)^2 + (y_i - y_t)^2}
\]

(1)

where \(D(x, y)\) is Euclidean distance between target pose and labelled pose, \((x_i, y_i)\) is cluster position of labelled pose, and \((x_t, y_t)\) is testing coordinate.

The output of kNN will processed by MQTT broker as a control input to turn on/off the smart socket via internet network that connected with appliances in smart home system. Furthermore, we can monitor the sensor readings from the smart socket such as voltage, current and active power by using smartphone or web-based platform.
Figure 1. Integrated IoT device with vision sensor for automatic switch control in smart socket.

2.1. Recognizing human body joints (key points)

PoseNet is a CNN based package that contains a standalone model to execute pose recognition/estimation for running real-time in the browser using TensorFlow.js. The output of this algorithm is to recognize and estimate the position of key body joints in $xy$-coordinates. PoseNet currently detect 17 key points which is illustrated in Figure 2. Besides the key point coordinates, this algorithm also can estimate confidence scores. Confidence score determines the overall confidence value of pose estimation which ranges from 0.0 to 1.0. Key point contains the estimated poses of person such as nose, right knee, left arm, etc. It contains the position and confidence score of each node of joint skeleton. The key point is trained bay using mobilenet (as shown in Figure 3) deep learning architecture because of its fast computation process.

Figure 2. Key points joint skeleton node using posenet model.

Figure 3. MobileNet training.

PoseNet has 2 algorithm versions which can detect only one person pose in a picture/video (Single Pose Estimation/SPE) and can detect multiple persons poses in a picture/video (Multiple Pose Estimation/MPE). Instead of using SPE, we use MPE for pose recognition even it is more complex and slower than SPE. However, if there are several persons appearing in an image, the detected key points are less likely to be associated with wrong poses. Even there is just single person, MPE algorithm commonly used instead of SPE. In order to avoid error in SPE that is conflated of different human pose, we choose MPE. The procedure of MPE algorithm is explained as below:

- Input Image Element, is html elements that contain images to predict poses, such as videos or images. Importantly, the elements of the picture or video inserted must be square.
- Image Scale Factor, is scale for image or video, the smaller the image scale factor will reduce the image quality and sacrificing accuracy but increasing speed. a number between 0.2 and 1 with default use is 0.5.
- Flip horizontal, to flip the image or videos between vertical and horizontal. For default use it is false.
• Output Stride, this effect accuracy and speed of the estimated pose. The higher value of output stride, the better accuracy but slower speed and vice versa.
• Maximum pose detection, this value contains maximum pose that can be detect using integer value with defaults value is 0.5.
• Pose confidence score threshold, this value to control minimum confidence score from pose. The value is between 0.0 to 1.0 with defaults value is 0.5.
• Non-maximum suppression (NMS) radius, this value controlling return of minimum distance between pose. This value contain number in pixel with defaults value is 20.

The output of PoseNet can be seen in Figure 4, where $x$ and $y$ represent the coordinate of joint skeleton and confidence score is also shown in the result.

2.2. Smart socket
We built smart socket as can be seen in Figure 5 that is connected through internet network in smart home. Users can control and monitor smart socket by using smartphone or web-based platform. In order to calculate active power, we need to find power factor ($\cos \varphi$) using this formula

$$P = VI \cos \varphi$$

2. Experiment result
The pose recognition using the k-Nearest Neighbor (kNN) method will be applied to test and classify human pose. Hardware design for experiment is shown in Figure 7. The recognition test of human pose
is divided into 6 different key poses as shown in Figure 8. For training data taken from skeleton coordinate readings with 10 training data scenarios, 20 training data, and 30 training data for each labelled pose A - F. To determine the accuracy value, a calculation using confusion matrix is performed

\[ Acc = \frac{TP + TN}{TP + TN + FP + FN} \]  

(2)

where \( Acc \) is accuracy (%), \( TP \) is true positive, \( TN \) is true negative, \( FP \) is false positive, and \( FN \) is false negative. While for testing tested with 30 poses according to each labelled pose, then the results of the classification of human pose are represented on the confusion matrix which can be seen in Figure 9.

**Figure 8.** Illustration of human pose recognition test divided into six labelled poses, A - F poses (using PoseNet).

| Actual pose | Prediction pose |
|-------------|-----------------|
| A           | 90.0            |
| B           | 10.0            |
| C           | 80.0            |
| D           | 13.3            |
| E           | 6.7             |
| F           | 23.3            |

(a) 60 training data samples

| Actual pose | Prediction pose |
|-------------|-----------------|
| A           | 96.7            |
| B           | 6.7             |
| C           | 93.3            |
| D           | 3.3             |
| E           | 6.7             |
| F           | 3.3             |

(b) 120 training data samples

| Actual pose | Prediction pose |
|-------------|-----------------|
| A           | 90.0            |
| B           | 6.7             |
| C           | 86.7            |
| D           | 3.3             |
| E           | 6.7             |
| F           | 6.7             |

(c) 180 training data samples

**Figure 9.** The test results are written in a confusion matrix related to the performance of accuracy for pose recognition using the kNN approach, for the classification of pose poses A - F.

Tests were done by using hardware include GPU NVIDIA GeForce GTX 1050 Ti, and a single RGB camera Logitech C922 Pro Stream HD Webcam with computing processing only at 8-11 FPS (frames per second). Mean accuracy precision (mAP) was obtained by the kNN approach for pose recognition obtained an average accuracy of 80% (N = 60), 92.8% (N = 120), and 93.9% (N = 180). From the test
results, it can be seen that improvements in accuracy performance can be done by increasing the amount of training data.

4. Conclusion
The recognition of human pose based on vision sensor was done out in two stages, the first step is the detection joint of skeleton and the second step is classification process using kNN method. The performance generalization of system had been evaluated and the result for classification accuracy of human pose is 93.9% for 180 dataset training samples. The experiment shows the accuracy can be improved by adding more dataset samples. The integration camera sensor with IoT device, i.e. smart socket, was successfully done for control and monitor home appliances by using MQTT broker. For further works, we try to recognize human gesture or activity recognition and compare the result with others classification methods.

Acknowledgement
The second and third author would like to acknowledge support from Telkom University on funding for this project, Penelitian Dasar dan Terapan, 2019.

References
[1] Paraskevopoulos G, Spyrou E, Sgouropoulos D, Giannakopoulos T and Mylonas P 2019 Real-time arm gesture recognition using 3D skeleton joint data Algorithms 12 5 1–17
[2] Utami D B and Ichwan M 2017 Pengenalan Pose Tangan Menggunakan HuMoment J Infotel. 9 1 100
[3] Newell A, Yang K and Deng J 2016 Stacked Hourglass Networks for Human Pose Estimation Eur Conf Comput Vis. VIII LNCS 9912 483–499
[4] Mehta D, Sridhar S, Sotnychenko O, Rhodin H, Shafiei M and Seidel H P 2017 VNect: Real-time 3D human pose estimation with a single RGB camera ACM Trans Graph 36 4
[5] Cao Z, Simon T, Wei S E and Sheikh Y 2017 Realtime multi-person 2D pose estimation using part affinity fields Proc - 30th IEEE Conf Comput Vis Pattern Recognition, CVPR 1302–1310
[6] Nishi K and Miura J 2017 Generation of human depth images with body part labels for complex human pose recognition Pattern Recognit 71 402–413
[7] Chen Y, Shen C, Wei XS, Liu L and Yang J 2017 Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation Proc IEEE Int Conf Comput Vis. 1221–1230
[8] Luvizon D C, Tabia H and Picard D 2019 Human pose regression by combining indirect part detection and contextual information Comput Graph 85 15–22
[9] Devanne M, Wannous H, Berretti S, Pala P, Daoudi M and Del Bimbo A 2015 3-D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold IEEE Trans Cybern. 45 7 1340–1352
[10] Mallick T, Das P P and Majumdar A K 2019 Posture and sequence recognition for Bharatanatyam dance performances using machine learning approach arXiv preprint arXiv:1909.11023 1–20
[11] Wu Q, Xu G, Li M, Chen L, Zhang X and Xie J 2018 Human pose estimation method based on single depth image IET Comput Vis. 12 6 919–924
[12] Ding W, Liu K, Cheng F and Zhang J 2015 STFC: Spatio-temporal feature chain for skeleton-based human action recognition J Vis Commun Image Represent 26 329–337
[13] Guo Y, Li Y and Shao Z 2017 DSRF: A flexible trajectory descriptor for articulated human action recognition Pattern Recognit 76 137–148
[14] Davis S E, Cremaschi S and Eden M R 2018 Efficient Surrogate Model Development: Impact of Sample Size and Underlying Model Dimensions Computer Aided Chemical Engineering. Elsevier Masson SAS 44 979–984
[15] Rahmani H and Mian A 2016 3D action recognition from novel viewpoints Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 1506–1515
[16] Liu J, Shahroudy A, Xu D and Wang G 2016 Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition *ECCV* *9907* 816–833

[17] Núñez J C, Cabido R, Pantrigo J J, Montemayor A S and Vélez J F 2018 Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition *Pattern Recognit.* *76* 80–94

[18] Shinde S, Kothari A and Gupta V 2018 YOLO based Human Action Recognition and Localization *Procedia Comput Sci* *133* 831–838

[19] Saputra G Y, Afrizal A D, Mahfud F K R, Pribadi F A and Pamungkas F J 2017 Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya) *Inform Mulawarman J Ilm Ilmu Komput.* *12* 2 69

[20] Brit D T and Matthews C 2006 Front cover *TCP/IP Tutorial and Contract* *38*