Sitting Posture Prediction and Correction System using Arduino-Based Chair and Deep Learning Model

Haeyoon Cho
Department of Computer Science & Engineering
Ewha Womans University
Seoul, South Korea
chohy0555@gmail.com

Hee-Joe Choi
Department of Computer Science & Engineering
Ewha Womans University
Seoul, South Korea
heejo05@hanmail.net

Chae-Eun Lee
Department of Computer Science & Engineering
Ewha Womans University
Seoul, South Korea
nuguziii@naver.com

Choo-Won Sir
Department of Computer Science & Engineering
Ewha Womans University
Seoul, South Korea
rebeccachoosir@gmail.com

***Abstract*—In this paper, we propose a system that applies deep learning to classify different postures of the users of our specially designed chair composed of Arduino hardware. The system provides users with information of their real-time posture and analytics for a given period of time, hence users can figure out their sitting habits. Moreover, the system suggests guidance videos of stretching body parts that help the users to correct their sitting postures.**

***Index Terms*—Arduino-based chair, sitting posture correction, deep learning model, android application, customized system**

I. Introduction

As society advances, people experience increased work burdens of using computer compared to that of the past days. In other words, modern people spend a lot of time sitting on a chair, working on a desk. It is very evident that people who work in the office spend their time sitting on a chair more than time sleeping at night. For the people who spend much time on the desk, the importance of maintaining a correct posture became critical. For one reason, sitting in the correct posture helps people to work efficiently and effectively. Another reason is that it could also bring psychological comfort [1] [2]. However, it is very challenging for an individual to find out one's inappropriate posture. Since their inappropriate sitting posture has become a habit for a long time and it is challenging to observe their posture in an objective way, they need assistance. To solve these problems, various related studies on posture monitoring system have been done. Specifically, they focused on prediction of the sitting posture, by using computer vision technologies such as interpolation [3] or using near-optimal sensor placement strategy to place the pressure sensors effectively to get the posture data [4]. In these papers, writers implement a system using pressure sensors to group postures into 3 types of improper postures [3], or implement a system using pressure sensors to focus on reducing hardware and computational costs [4].

In recent years, deep convolutional neural networks (CNNs) have been very popular in solving many vision problems. Many studies show improvement of performance by replacing existing algorithms with CNNs, for example in low-level vision tasks such as image denoising [5]. Since related works use the posture data with the images and apply their algorithms to predict the postures, it is expected to have better performance when we substitute their algorithms by CNNs.

In this paper, we propose a system that uses a specialized Arduino-based chair to predict and analyze the sitting posture of the user and provides appropriate videos to help them correct their posture by analyzing the user statistics on their overall posture data. We achieved performance improvement in predicting the user posture by using deep CNNs and LBCNet (Lower-Balanced Check Network). We introduce a new, highly varied dataset of human postures. Moreover, the system distinguishes itself from the previous works in that it uses both pressure sensors and ultrasonic sensors to check the balance of both upper and lower part of the body. Lastly, the comments produced by the system, which guide the users on how to correct their sitting posture by using the keyword matching algorithm.

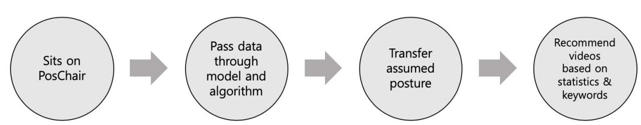

Fig. 1. Flow of the system.

All authors contributed equally to this work.

978-1-7281-5411-4/19/$31.00 ©2019 IEEE
DOI 10.1109/SOCA.2019.00022

II. System Design

The system consists of 3 main components: 1. the Arduino-based chair named PosChair, 2. server system which provides computation for functionalities that include deep learning, databases, and rule-based learning and 3. the mobile application based on Android OS. Users sit on the PosChair which uses its pressure sensors and ultrasonic sensors to get the data needed for the posture prediction. The data is sent to the server system that processes the data and passes it to its deep learning model to predict the lower-balance, specifically focused on pelvis and to predict the upper-balance by using rule-based learning algorithm. Using the two predictions on the posture of the user, we combine them and predict the comprehensive posture of the users at last. This final prediction is saved in the database of the server system and sent to the Android application for the users to monitor the results.

The simple demonstration of the flow of the system is shown in "Fig. 1". Once the user sits on PosChair, the data are sent to server and are processed with the deep learning model along with algorithms to provide users with the following information:

- Detect the posture of the user and alarm the user with the application push notification.
- Provide statistics based on the data of the user posture.
- Provide user-customized keywords for stretching-needed body parts and related videos by analyzing the posture data of the user.

III. System Algorithm

"Fig. 2" presents the overall algorithm that the system uses to predict and analyze user posture to provide a guideline. First, we acquire sensor data using pressure sensors and ultrasonic sensors. Next, we use the Balance Check Algorithm to determine which posture the data is representing. The final approximation of the posture is used in the Keyword & Video Matching Algorithm to provide the videos and analytics to be shown to the user.

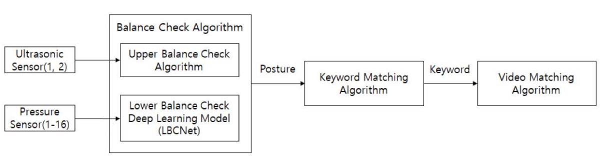

Fig. 2. Flow of data on the algorithm.

A. *Posture Prediction Algorithm*

The posture prediction algorithm consists of an upper-balance check algorithm and a lower-balance check network (LBCNet). In the upper-balance check algorithm, we use two sensors to get two distinct data, which are placed around the shoulder and neck of the chair. After comparing the data with that of correct posture that the user set initially and obtaining the relative difference, the value is used in rule-based learning algorithm to detect a user's upper body position which is determined in one of three cases: the correct posture, the turtle-neck posture, or the leaning-forward posture. LBCNet uses 16 pressure sensors to acquire data. "Fig. 3" shows there are several steps to convert sensor data into images. We define 5*8 array with 16 values based on the chair's actual sensor position. For empty spots, we filled in them with the average of the surrounding values. Then, using bicubic interpolation [11], 5*8 array is interpolated to produce an array of 40*40 (range from 0 to 1050). After RGB mapping, we get 40*40*3 color image at the end. This image is used to predict the user's posture on the lower part of the body. By combining the two postures on upper part and lower part, the system predicts the final posture of the user.

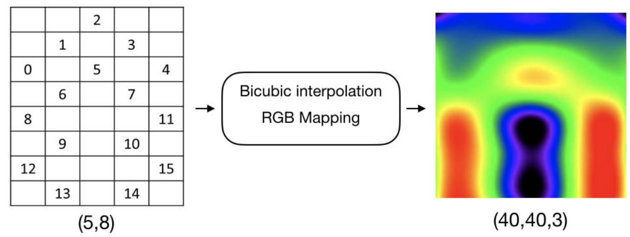

Fig. 3. Generating 40*40*3 resolution image.

LBCNet, which is a deep convolutional neural network, has an architecture as shown in "Fig. 4". It takes two images as an input and it has an initial image that represents the *correct* posture that the user set initially and the image that represents the current posture of the user. This network predicts the lower-balance posture by using the difference between the two images, the initial image and the current image. As shown in a related work [12], we also use transfer-learning algorithm to solve the problem of insufficient training data. We utilized VGG19 pre-trained model to perform feature extraction for each image respectively. We extract the feature before passing the pooling layer of conv2_3 to get the edge level information on the image. Each of the features that passed the VGG19 model is concatenated to pass our own model to predict the difference between the two features. For the output, it has a list of 4 binary values, which are multi-class and multi-label. Each class determines whether the user is crossing his or her leg or slouching his or her back or in other types of posture. Since it is challenging to categorize postures into classes due to the diversity of postures, we categorized postures based on balance to predict the final posture. By using this approach, we extracted about 40 kinds of posture to predict when combined with the upper balance check algorithm results. This makes our system robust to the error that occurs when there is no matching class for the given data result.

B. *Posture Correction Algorithm - Keyword Matching and Video Recommendation*

Keyword matching is done using posture prediction data that are saved in the database. The corresponding keyword

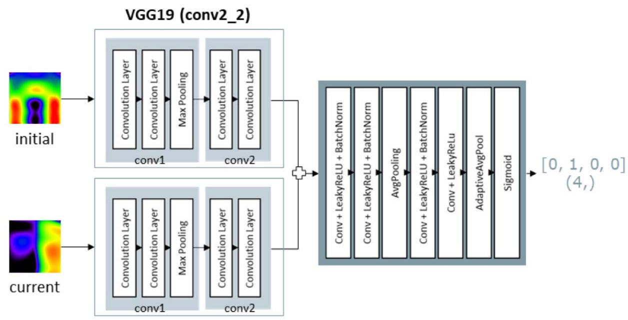

Fig. 4. LBCNet model architecture.

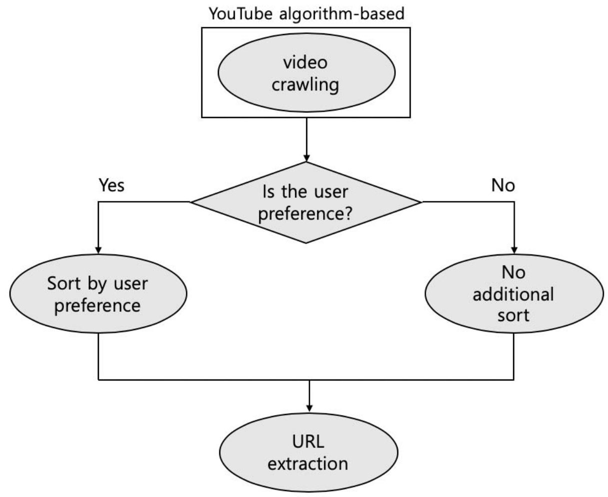

Fig. 5. Video recommendation process.

for each posture prediction data is shown in “TABLE I”. The keywords are used to search guidance videos of stretching body parts on video sharing platforms such as YouTube. The video searching and selection process is as shown in “Fig. 5”. URLs of video are saved in the database of server system beforehand carried out by web-crawling on Youtube, rather than searching video list every single time when user sends a request. The list of videos is verified to be highly related to the keywords since the videos are from smart search algorithm provided by YouTube. The number of videos is limited up to 10 for each of the keyword. When Android application sends request to view the recommended videos, the most common keyword in the user’s database is selected and fetched within the highest priority. In terms of the priority, if some videos are categorized with same keywords, a video preferred by the user comes up first. User can mark a video that she prefers with *like* button in the Android application.

TABLE I
CORRESPONDING KEYWORDS FOR EACH POSTURE.

Posture	Keyword
Crouching	Cervical disc
Turtle neck	Turtle neck
Bent shoulder	Back/shoulder bent
Slouching	Back/shoulder bent
Pelvis/spine unbalance	Pelvis/spine unbalance
Stress on the back	Back/shoulder bent
Stress on the joints	Pelvis/spine unbalance, joints pain
Crossed legs	Pelvis/spine unbalance, blood circulation problem

IV. IMPLEMENTATION

A. System Development Environment

For data input, through the Arduino Mega2560 board attached to the main body of the chair, the value of the pressure sensors and the ultrasonic sensors are delivered. Since Arduino Mega 2560 board supports more concurrent processing ports than a basic Arduino board, 2560 board is chosen, that is, supporting 16 analog ports and 40 digital ports. The sensor values delivered by the Arduino board are transferred to the server built on Naver Cloud Platform using the Arduino wifi module. At this point, the Flask [7] application implemented on the server must be running and ready to receive the data transmission of the Arduino. SQLite [10] is used for the database system, and PyTorch [8] along with SciKit-Learn [9] are libraries for deep learning algorithm. Operations of Android mobile application are run by communicating in conjunction with the server.

B. System Implementation

The hardware system is implemented using 16 pressure sensors and 2 ultrasonic sensors. 16 pressure sensors are placed as shown in “Fig. 6” and installed on the seat plate. The placement of these pressure sensors was the most effective experimental result since the image obtained when seated in the correct posture was intuitively matched to the actual posture while using this arrangement. Ultrasonic sensors are attached to the neck part of the chair as shown in “Fig. 6”.

Fig. 6. PosChair with the sensors.

The main screen of the Android application consists of four tabs: *Main*, *Report*, *Video* and *Settings* tabs. On the *Main* tab, the current posture of user on the chair can be identified by an image or brief description, as shown in “Fig. 7-1”. Also as shown in “Fig. 7-2”, the *Report* tab provides an analysis of user postures by everyday, every week and month. If user sits improperly, the user can focus on which body part is put in an improper way, especially about the pelvis that is important for balancing. Next, as shown in “Fig. 7-3”, the *Video* tab shows keywords sorted by the statistics about the user’s posture, a list of videos recommended for those keywords, and a list of videos that the user prefers. The preference on the video of the user is set by clicking the *like* button at the bottom right side of each video. Finally, the *Settings* tab allows users to choose whether to receive push notifications or to receive notifications for specific body parts the user wants.

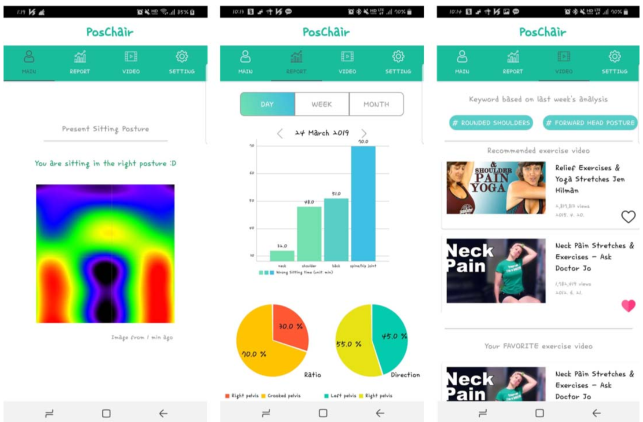

Fig. 7. PosChair Android application.

For database system, SQLite is used and data is expressed by total of 7 tables. The core table *Posture_label* contains the date of the data saved and user’s ID which is the email address. The key, which is the ID, of the table *Posture_label* is also the foreign key referenced by the table *User* which stores the user’s information and the table *Median* which stores the average of the real-time data of postures. When the user first starts the application, ID, name, password and initial posture data are inserted into the table *User*. After a certain period of time, collected posture data by the sensors are inserted as the upper and lower position information in *Posture_data* table. Table *Median* stores the average of the data collected over a set period of time, which is then grouped together to store the averages once again. The keywords extracted from this are compared with the table *Keyword*, leading to incrementation of the count of the corresponding keywords. Data are stored daily in *dayChart*. Measures on how many hours the user is sitting for a day and how correct his or her posture are stored with the date as a primary key along with the count of each keyword on a daily basis. In this work, videos are recommended using the video recommending algorithm. *Youtube_Video* table, which stores information on recommended videos, also stores whether users prefer the videos recommended. Also, it stores basic information of videos: URL, video title, keyword, posted date, number of views, and number of likes.

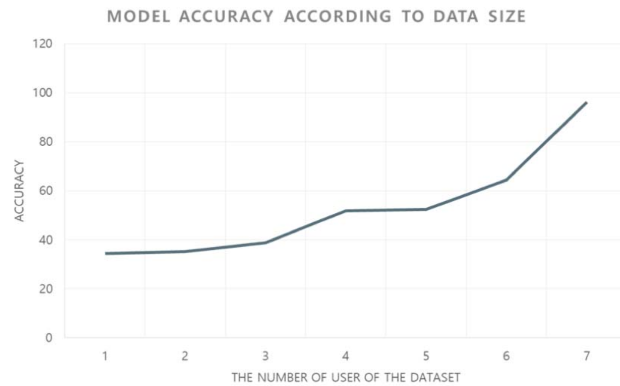

Fig. 8. Graph on accuracy of the model.

V. Experimental Results

“Fig. 8” shows the accuracy results on the LBCNet model. For the training dataset, we followed a related work [4] to define 10 typical sitting postures and add 5 different postures: crossed-legged, left/right leg-on-the-chair, crossed-legged with left or right leg up. We collected 15 different postures from 8 random users, approximately 10,000 images of training data. The dataset image is shown in “Fig. 9”. For the hyper-parameters, we used adam optimizer and L2 loss. The batch size is 128 and the epoch is 150. Learning rate is set to 0.001 and weight decay set to 0.00001. As a result, the final accuracy showed approximately 96%. And as provided in “Fig. 8”, as the dataset size increasesthe number of user signifies the dataset of the random users accumulated the accuracy goes up. However, there is a limit that we should collect more and various datasets since the datasets show variations depending on the user characteristics.

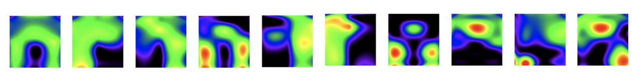

Fig. 9. Dataset image example.

VI. Conclusion

In this paper, we proposed a system which is composed of a chair device equipped with Arduino sensors, an Android application that can receive feedback on the posture and a server, and can assume the posture of the sitting user and allow him or her to monitor his or her own posture. Firstly, the pressure sensor and ultrasonic sensor values were used to accurately identify the user’s posture by considering both the upper and lower body. Moreover, by giving users push notifications and recommending videos based on the measured posture, users can correct their misformed posture habits. The proposed smart chair system is expected to enhance the efficiency of work and have a positive effect on health for the people of our times, who are seated most of the day. In the future, adding more user posture data and the number of sensors will help to develop a more sophisticated and accurate system. In addition, strengthening security will be necessary to safely process and manage user data. While revising and supplementing existing research, it is planned to continue our research by using deep learning servers so that the servers can both learn and store the posture-judging model at the same time. As the number of users increases in the future, Apache Flink or Apache Spark Streaming can be used to handle vast amounts of real-time streaming data from users.

REFERENCES

[1] Sun-young Kang, "A Comparison of Pelvic, Spine Angle and Buttock Pressure in Various Cross-legged Sitting Posture," KAUTPT, Vol. 19, No. 1, pp. 1-9, Feb 2012.

[2] Ji-Hea Choi, "Comparisons of Postural Habits, Body Image, and Peer Attachment for Adolescents with Idoipathic Scoliosis and Healthy Adolescents," J. Korean Acad Child Health Nurs, Vol. 17, No. 3, pp. 167-173, July 2011.

[3] Hojin Ha, "Design of Algorithm for Guidance of Sitting Posture Correction Using Pressure Sensor and Image Processing Interpolation Technique, Journal of Korean Institute of Information Technology 14(1), 37-44, 2016.

[4] Mutlu Bilge, et al., "Robust, low-cost, non-intrusive sensing and recognition of seated postures," Proceedings of the 20th annual ACM symposium on User interface software and technology. ACM, 2007.

[5] V. Jain and S. Seung, "Natural image denoising with convolutional networks," Conference on Neural Information Processing Systems, 2009

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations, 2015

[7] Flask, https://palletsprojects.com/p/flask/

[8] PyTorch, https://pytorch.org/

[9] Scikit-learn, https://scikit-learn.org/stable/

[10] SQLite, https://www.sqlite.org/index.html

[11] R. Keys, "Cubic convolution interpolation for digital image processing," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1153-1160, December 1981.

[12] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," https://arxiv.org/abs/1808.01974, 2018.