https://doi.org/10.5607/en.2019.28.1.54
Exp Neurobiol. 2019 Feb;28(1):54-61.
pISSN 1226-2560 • eISSN 2093-8144

Technologue

en
Experimental Neurobiology

Check for updates

Machine-Learning Based Automatic and Real-time Detection of Mouse Scratching Behaviors

Ingyu Park[1], Kyeongho Lee[2], Kausik Bishayee[3],
Hong Jin Jeon[4], Hyosang Lee[2]* and Unjoo Lee[1]*

[1]Department of Electrical Engineering, Hallym University, Chuncheon 24252,
[2]Department of Brain and Cognitive Sciences, DGIST, Daegu 42988, [3]Department of Pharmacology, College of Medicine,
Hallym University, Chuncheon 24252, [4]Department of Psychiatry, Depression Center, Samsung Medical Center,
Sungkyunkwan University School of Medicine, Seoul 06351, Korea

Scratching is a main behavioral response accompanied by acute and chronic itch conditions, and has been quantified as an objective correlate to assess itch in studies using laboratory animals. Scratching has been counted mostly by human annotators, which is a time-consuming and laborious process. It has been attempted to develop automated scoring methods using various strategies, but they often require specialized equipment, costly software, or implantation of device which may disturb animal behaviors. To complement limitations of those methods, we have adapted machine learning-based strategy to develop a novel automated and real-time method detecting mouse scratching from experimental movies captured using monochrome cameras such as a webcam. Scratching is identified by characteristic changes in pixels, body position, and body size by frame as well as the size of body. To build a training model, a novel two-step J48 decision tree-inducing algorithm along with a C4.5 post-pruning algorithm was applied to three 30-min video recordings in which a mouse exhibits scratching following an intradermal injection of a pruritogen, and the resultant frames were then used for the next round of training. The trained method exhibited, on average, a sensitivity and specificity of 95.19% and 92.96%, respectively, in a performance test with five new recordings. This result suggests that it can be used as a non-invasive, automated and objective tool to measure mouse scratching from video recordings captured in general experimental settings, permitting rapid and accurate analysis of scratching for preclinical studies and high throughput drug screening.

Key words: Machine learning, Decision tree, Mouse, Scratching, Pruritus, Itch

INTRODUCTION

Itch is a devastating symptom accompanied by various disease conditions originating from dermatological, systemic, neurological or psychogenic problems [1]. During the last decade, a great advance has been made in our understanding of the molecular and cellular mechanisms of itch [2]. A number of itch-transducing molecules and neurotransmitters are identified in the primary sensory neurons and spinal cord, and the brain circuits mediating itch have begun to be revealed using animal models such as rodents and monkeys. Scratching is a main behavioral consequence elicited by acute treatment of pruritogens or by chronic itch conditions, and has been quantified in the animal and human studies

Received November 20, 2018, Revised January 28, 2019,
Accepted January 30, 2019

*To whom correspondence should be addressed.
Unjoo Lee, TEL: 82-33-248-2354, FAX: 82-33-242-2524
e-mail: ejlee@hallym.ac.kr
Hyosang Lee, TEL: 82-53-785-6147, FAX: 82-53-785-6109
e-mail: hyosang22@dgist.ac.kr

as a correlate representing the subjective experience of itch. In many previous studies involving laboratory animals such as mice, scratching was counted manually as in scratching bouts by human annotators by playing video recordings back and forth, which is a labor-intensive and time-consuming process often hinders the progress of the study and may also subject to human errors. Thus, it has attempted to develop automated detecting methods for mouse scratching based on various strategies, including the detection of vibration and sound generated when a mouse scratches the skin with the hind paw, motional detection of a metal ring or magnet implanted to the hind limb, the use of force platform detecting repetitive events, and also computer vision-based analysis [3-12]. Few studies have involved a machine-learning algorithm. Although those methods permit automated and decent detection of scratching, they are often limited by invasive and potentially disturbing handling of animals as well as costly software and testing equipment, such as high-speed and high-resolution cameras and depth-sensing cameras.

Here, we propose a new machine learning-based method for automatic and real-time detection of mouse scratching behavior in its homecage from videos recorded with an inexpensive monochrome camera such as webcam. By taking advantage of characteristic frame-dependent changes occurring when a mouse exhibits scratching, the novel method can distinguish scratching from other homecage activities such as walking, rearing, grooming, and digging. After two rounds of training with a two-step decision tree-inducing algorithm, it exhibited an ability in detecting scratching behavior with a sensitivity and specificity of 95.19% and 92.96%, respectively. In the following performance test, the new method could detect scratching from a separate set of recordings taken at various pixel resolutions with a sensitivity and specificity of 97.3% and 79.3%, respectively, and outperformed other machine-learning algorithms, such as support vector machine (SVM), k-nearest neighbor (kNN), convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM). Given that the proposed method requires neither expensive recording equipment nor accessary devices attached to an animal, it is very useful and reliable over existing automated scoring system measuring mouse scratching.

MATERIALS AND METHODS

Recording of mouse scratching behavior

The animal experiments were approved by DGIST Institutional Animal Care and Use Committee (IACUC) and were conducted in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals.

C57BL6/J wild-type male mice were housed singly and were maintained on a 12-h light/dark schedule with ad lib access to food and water. After a 10-min baseline recording in a home cage, a mouse was intradermally injected at the nape of the neck with a pruritogen, chloroquine (0.2 mg in 50 μl), using a 31G insulin syringe (BD Ultra-Fine II). The injected mouse was returned to the home cage in a room with 300~450 lux illumination and was recorded for 30 min using a Security Monitor Pro and a webcam (Logitech HD Pro webcam) installed 66 cm above from the bedding of cage with dimensions (mm) of W403×L165×H174, using a tripod. The recording was performed at 29.0±0.01 frames per second (fps) in the resolution of 960×720 pixels. Manual scoring was performed by two different human annotators by playing videos back and forth.

Feature extraction

Mouse scratching occurs in small sets of very fast movements, which could arise at about 10 times per second or more. Detecting individual scratches is not facilitated from data recorded using a customized monochrome camera with a frame rate of less than 30 fps. Therefore, in this study, we paid attention to short-term patterns generated when a mouse scratches with respect to body size S_b, frame-to-frame body size difference ΔS_b, frame-to-frame body position difference ΔP_b and frame-to-frame pixel difference ΔN_p. Body size S_b is the area in pixels of a mouse body in a frame. Frame-to-frame body size difference ΔS_b is the difference in pixels of body size S_b in two consecutive frames. Frame-to-frame body position difference ΔP_b is the distance in pixels of the center position of the mouse body in two consecutive frames. Frame-to-frame pixel difference ΔN_p is the number of pixels with difference values in two consecutive frames. When a mouse scratches, it usually stands on one of its hind paws and uses another one of its hind paws to scratch its ear, back, or side, resulting in lateral bending and flection of its body. Its body shape in the recording changes from the lateral bending and flection position in one frame to the standing back position in the next frame. Therefore, frame-to-frame body size difference ΔS_b, frame-to-frame body position difference ΔP_b and frame-to-frame pixel difference ΔN_p could be specific to patterns of body movement during scratching. Body size S_b is also specific to the patterns of body movement during scratching.

Fig. 1a illustrates the overall flowchart of the detection method for a mice scratching and the sample images for each step in Fig. 1b. As shown in the figure, first the input frame image $I(x, y, t_i)$ is binarized to segment an edge image $B(x, y, t_i)$ of a mouse from the background using the equation (1) where t_i is the time stamp of the i^{th} frame image. The threshold θ_{bg} is adjusted to reduce noise due to sawdust in the cage.

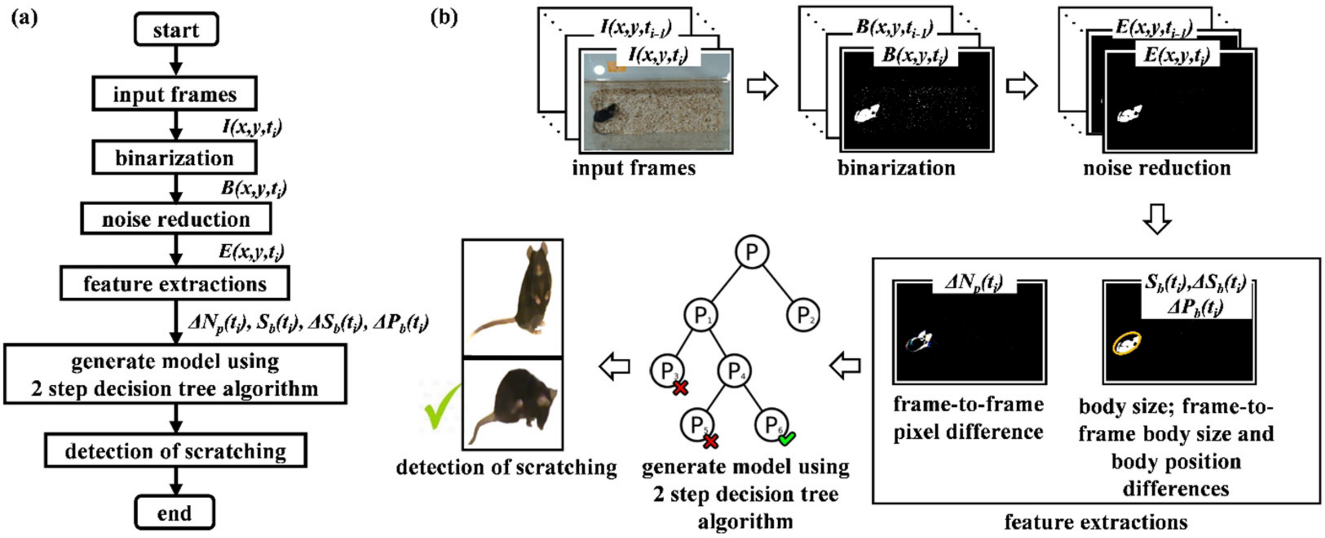

Fig. 1. (a) Overall flowchart (a) and sample image (b) showing the procedures to develop a novel automatic method detecting scratching. Experimental movies recoded following an intradermal injection of an itch-causing compound into the nape of the neck of a mouse were registered and binarized for the frame-to-frame analysis. Following a process of noise reduction, four characteristic changes accompanying scratching, changes in body size as well as changes in pixels, body size and body position by frame, were extracted from the movies. These features were utilized to determine scratching from other homecage behaviors by a two-step decision tree-inducing algorithm during two rounds of training procedures to build a novel detection method.

$$B(x,y,t_i)=\begin{cases}1 & I(x,y,t_i)>\theta_{bg}\\ 0 & otherwise\end{cases} \tag{1}$$

Subsequently, an erosion image $E(x, y, t_i)$ of the binary image $B(x, y, t_i)$ is produced by a 2×2 square structuring element in order to reduce edge or isolated noises as shown below in (2).

$$E(x,y,t_i)=B(x,y,t_i)\cap B(x-1,y,t_i)\cap B(x,y-1,t_i) \tag{2}$$

Next, parameters of body size S_b, frame-to-frame body size difference ΔS_b, frame-to-frame body position difference ΔP_b and frame-to-frame pixel difference ΔN_p are obtained from the erosion image $E(x, y, t_i)$ using an object fitting method such as OpenCV function findContours: $E(x, y, t_i) \rightarrow C(x_c, y_c, t_i)$as follows.

$$S_b(t_i)=\#\text{ of pixels} \in C(x_c,y_c,t_i) \tag{3}$$

$$\Delta S_b(t_i)=S_b(t_i)\text{-}S_b(t_{i\text{-}1}) \tag{4}$$

$$\Delta P_b(t_i)=\sqrt{(\bar{x}_i-\bar{x}_{i-1})^2+(\bar{y}_i-\bar{y}_{i-1})^2} \tag{5}$$

$\bar{x}_i$ and $\bar{y}_i$ are the coordinates of the midpoint pixel of the contour $C(x_c, y_c, t_i)$ and estimated by the zero-order spatial moments: $\bar{x}_i=\frac{m_{10}}{m_{00}}$, $\bar{y}_i=\frac{m_{01}}{m_{00}}$. Here, the moments m_{pq} for a contour $f(x, y)$ are given by:

$$m_{pq}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x^p y^q f(x,y)dxdy \text{ with } p,q=0,1,2,... \tag{6}$$

The frame-to-frame pixel difference ΔN_p is calculated as the number of non-zero pixels with the absolute value of the frame-to-frame differential images between time t_i and t_{i-1} to extract the motion from the silhouette image, as shown in (7).

$$\Delta N_p(t_i)=\sum_x\sum_y\left|E(x,y,t_i)-E(x,y,t_{i-1})\right| \tag{7}$$

Finally, a two-step decision tree algorithm is applied to create a training model of a mouse scratching behavior by using the features of body size S_b, frame-to-frame body size difference ΔS_b, frame-to-frame body position difference ΔP_b, and frame-to-frame pixel difference ΔN_p. The training model is then used to detect the mouse scratching behavior.

Two-step decision tree algorithm

A decision tree, one of the most popular machine-learning algorithms, is a tree where each node, link, and leaf represents a feature, a decision, and an outcome, respectively. We used the J48 decision tree-inducing algorithm using information gain. It is based on the concept of entropy and information content to decide which feature to split on at each step in building the tree. We applied the J48 decision tree-inducing algorithm in two steps to build a training model of a mouse scratching behavior by using the four features, the frame-to-frame pixel difference, the frame-to-frame body position difference, the frame-to-frame body size difference and the body size. In other words, all the frames of the training data were used in the first step to build a model of a mouse scratching behavior and then the resultant frames detected as scratching in the first step were used as the training data in the second step to build a fine modulated version of the model. The C4.5 post-pruning algorithm was used at both steps to optimize the computational efficiency and classification accuracy of the training model.

RESULTS AND DISCUSSION

Statistical analysis of the significance of the extracted four features

Fig. 2 presents the comparisons of the normalized histograms (bars) and distributions (lines) between behaviors of scratching (Scr; red bars and lines) and no scratching (NoScr; blue bars and lines) in the features, the frame-to-frame pixel difference (Fig. 2a), the frame-to-frame body position difference (Fig. 2b), the frame-to-frame body size difference (Fig. 2c) and the body size (Fig. 2d), extracted from the training data. Fig. 2 also shows the results of ANOVA analysis between the behaviors of scratching and no scratching in the features. The ANOVA analysis indicated that there were significant differences between the behaviors of scratching and no scratching (p-value<0.0001) in all the features of the frame-to-frame pixel difference, the frame-to-frame body position difference, the frame-to-frame body size difference and the body size. The averages and standard deviations of each of the features for various behaviors including scratching are summarized in Table 1 where the unit is the number of pixels and the superscripts

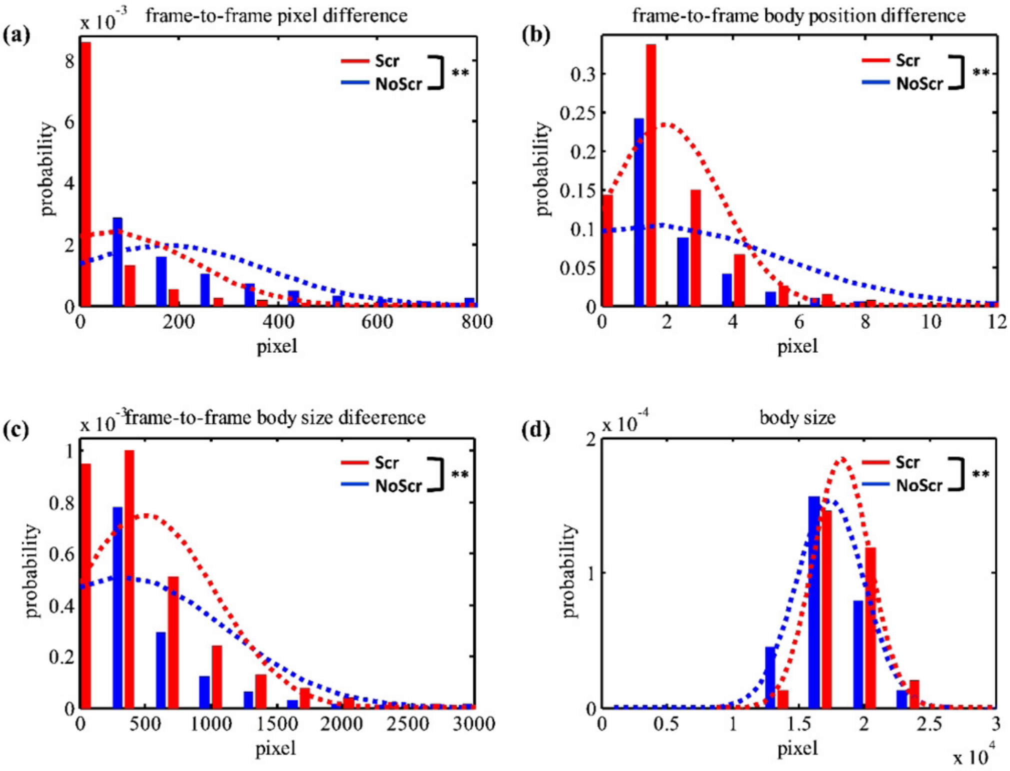

Fig. 2. Normalized histograms and distributions and ANOVA analysis results for scratching (Scr, red bars and lines) and no scratching (NoScr, blue bars and lines) behaviors in (a) frame-to-frame pixel difference, (b) frame-to-frame body position difference, (c) frame-to-frame body size difference and (d) body size. (**p-value<0.0001).

Table 1. Averages and standard deviations of the features for various behaviors including scratching (unit: no. of pixels)

Behavior	Frame-to-frame pixel difference	Frame-to-frame body position difference	Frame-to-frame body size difference	Body size
scratching	173.3±204.6	1.92±1.68	496.6±530.9	18724.2±2168.1
grooming	54.0±120.7**	1.47±2.56**	315.8±628.5**	16561.9±2401.0**
digging	26.3±90.5**	1.42±2.77**	335.5±591.5**	18321.3±2145.1
rearing	108.6±214.6**	2.69±6.84**	531.5±1335.0*	18660.0±3150.2**
walking	597.8±591.1**	7.42±11.79**	900.9±1877.7**	21386.0±3110.3**

p-value<0.0001**, p-value<0.01* in one-way ANOVA for comparing scratching with each behavior on each feature, Data are mean±SD.

** and * indicate p-value<0.0001 and p-value<0.01, respectively obtained in one-way ANOVA for comparing scratching with each behavior on each feature. As shown in Table 1, the frame-to-frame pixel difference of scratching is significantly smaller than that of walking while it is significantly bigger than those of grooming and digging in the averages and the standard deviations. On the other hand, the frame-to-frame body position difference the frame-to-frame body size difference of scratching are significantly smaller than those of rearing and walking in the averages and the standard deviations. Therefore, the frame-to-frame pixel difference, the frame-to-frame body position difference and the frame-to-frame body size difference parameters of scratching could be the effective features in the classification of scratching behavior in relation to other behaviors.

Performances of the training model built by the two-step decision tree algorithm

J48 decision tree-inducing algorithm (Weka implementation of C4.5) was used to build a training model of a mouse scratching behavior by using the features of the frame-to-frame pixel difference, the frame-to-frame body position difference, the frame-to-frame body size difference and the body size. Eight recording data, recordings No. 1 through 8, were used. The duration in seconds (frame rate per second) of each recording were 1,903 (29.98), 1,866 (29.99), 1,846 (30.00), 1,835 (29.94), 1,835 (29.99), 1,835 (30.00), 1,839 (29.96), and 1,837 (29.89), respectively, in the ascending order of the recording number. In the first step, all the frames of the recordings number 1 to 3 were used as the training data to build a first version of the model. Once the first version of the model was obtained, in the second step, the resulted frames detected as scratching in the first step were used as the training data to build a fine modulated second version of the model. The C4.5 post-pruning algorithm was used at both steps to optimize the computational efficiency and the classification accuracy of the training model where the confidence factor was set to 0.5. Fig. 3 shows the decision tree of the fine modulated model obtained in the second step, where features, decisions, and outcomes are displayed in the shapes of ellipse, line, and square, respectively. Features are depicted in differently colored ellipses with letters enclosed, where 'fd', 'pd', 'sd' or 'sz' refer the frame-to-frame pixel difference (fdiff), the frame-to-frame body position difference (pdiff), the frame-to-frame body size difference (sdiff) or the body size (size), respectively. Decisions are displayed in different patterns of lines, in which solid or dotted lines are cases for the corresponding feature not to be or to be greater than the number specified beside the feature, respectively. Outcomes are displayed as a letter 'N' for non-scratching or 'S' for scratching on a colored square with white or grey, respectively. Fig. 3 clearly shows that the features of the frame-to-frame pixel difference and the frame-to-frame body size difference had dominant roles in the detection of the scratching behavior; this was also the

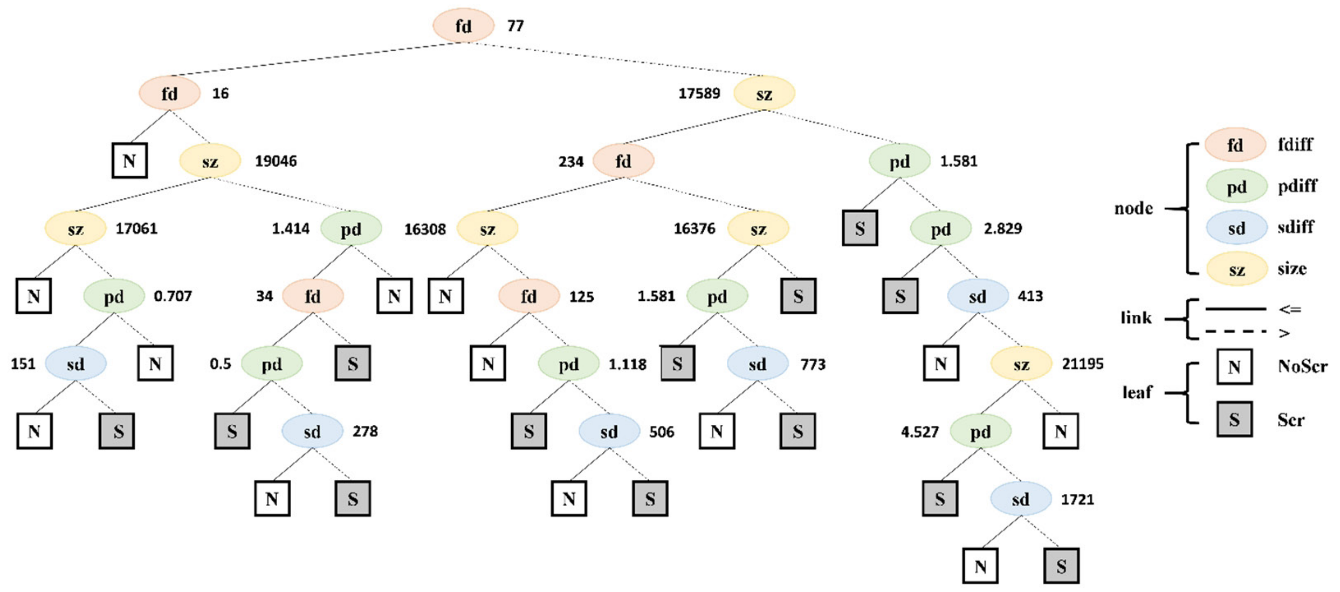

Fig. 3. Decision tree of the fine modulated model obtained in the second step, where features, decisions, and outcomes are expressed in shapes of ellipse, line, and square, respectively. Features are depicted in colored ellipses with letters enclosed, where 'fd', 'pd', 'sd' or 'sz' refer the frame-to-frame pixel difference, the frame-to-frame body position difference, the frame-to-frame body size difference or the body size, respectively. Decisions are displayed in different patterns of lines, in which solid or dotted lines are cases for the corresponding feature not to be or to be greater than the number specified beside the feature, respectively. Outcomes are displayed as a letter 'N' for non-scratching or 'S' for scratching on a colored square with white or grey, respectively.

conclusion of Fig. 2. We implemented the algorithm of the decision tree in Python 3.7 and tested the performance with the testing data of the recordings number 4 to 8 as well as the training data.

Table 2 shows the sensitivity and specificity values of the performance test for each recording by using the models obtained from the first and the second steps for comparison. The sensitivity was estimated from the rate of the number of frames detected as scratching (true positives) to the total number of frames of scratching (true positives and false negatives). The specificity was estimated from the rate of the number of frames detected as non-scratching (true negatives) to the total number of frames of non-scratching (true negatives and false positives). The results indicate an average sensitivity and a specificity of 95.19% and 92.96%, respectively in the model of the second step where the average ratios of true positives, true negatives, false positives and false negatives were 1.71, 91.32, 6.88, and 0.09, respectively. On the other hand, the results indicate an average sensitivity and a specificity of 98.23% and 86.65%, respectively in the model of the first step. In this instance, the sensitivity was a little less than before (by 3.09%) but the specificity had much more improved (by 7.28%) in the model obtained from the second step compared with that from the first step. All the true and the false were verified manually in which the effect of human errors were minimized by cross-checking more than 10 times. The performance of the proposed method was also verified after a preliminary screening by comparing to other machine learning algorithms such as classification and regression tree, support vector machine, k-nearest neighbor, convolutional neural network, recurrent neural network, and long short term memory [unpublished data].

Table 2. Calculated sensitivity and specificity values of the performance test for each recording by using the models obtained from the first and the second steps for comparison

Recording Number	Sensitivity (%)		Specificity (%)	
	1st step model	2nd step model	1st step model	2nd step model
1	98.7	94.6	84.4	91.2
2	98.9	95.9	91.0	94.2
3	100.0	97.7	88.8	94.1
4	98.2	98.0	84.4	92.6
5	100.0	91.4	85.1	92.4
6	96.2	95.5	83.6	91.3
7	96.4	95.3	85.8	92.9
8	97.4	93.1	90.1	95.0
Average	98.23	95.19	86.65	92.96

Validation of the performances of the training model built by the two-step decision tree algorithm using standardized input datasets

To validate the performances of the two-step decision tree algorithm, three recordings with various pixel resolutions of 300×340, 800×600, and 1260×960 were used after resizing their pixel resolution to 960×720 and the frame rate per second to 30. The duration in seconds (frame rate per second) of each pixel resolution of 300×340, 800×600, and 1260×960 was 1,809 (30.00), 1,804 (29.99), and 1,940 (21.26), respectively. The distances from camera and the intensities of illumination were varying. Table 3 shows the sensitivity and specificity values of the performance test for each of the three recordings, which indicate an average sensitivity and a specificity of 97.3% and 79.3%, respectively in the model of the second step. On the other hand, the results indicate an average sensitivity and a specificity of 97.4% and 62.7%, respectively in the model of the first step. In this instance, the sensitivity was similar to before but the specificity had much more improved (by 16.6%) in the model obtained from the second step compared with that from the first step.

Validation of the outperformance of the two-step DT compared to other machine-learning algorithms in identifying scratching

The performance of the two-step decision tree algorithm was compared with that of other machine-learning algorithms, SVM, kNN, CNN, RNN, and LSTM, using a separate set of three movies recorded at the pixel resolution of 300×340, 800×600, or 1260×960. All movies were processed to the pixel resolutions of 960×720 and the frame rate of 30 f/s. LinerSVC and KNeighborsClassifier of Scikit-Learn library in Python were used as a SVM and s kNN algorithms, respectively. Hinge function was used as the loss function, and 100.0 was set to the value of the parameter C for optimizing the performances of LinearSVC. The parameter k was set to 25 for minimizing the error rate and

Table 3. Calculated sensitivity and specificity values of the performance test for three recordings with various pixel resolutions and recording distances by using the models obtained from the first and the second steps for comparison

Pixel resolutions	Sensitivity (%)		Specificity (%)	
	1st step model	2nd step model	1st step model	2nd step model
320×240	94.9	94.8	69.6	85.1
800×600	99.1	99.1	72.2	84.4
1260×960	98.3	98.2	46.4	68.6
Average	97.4	97.3	62.7	79.3

Table 4. The comparison of sensitivity and specificity of machine-learning algorithms. The analyzed movies were recored at indicated pixel resolution

Pixel resolutions	Sensitivity (%)					
	2DT	SVM	kNN	CNN	RNN	LSTM
320×240	94.8	68.0	91.0	92.4	84.8	90.7
800×600	99.1	63.3	93.5	97.4	96.5	96.5
1260×960	98.2	43.1	94.6	92.6	76.2	95.9
Average	97.3	58.1	93.0	94.1	85.8	94.3
Recording No.	**Specificity (%)**					
	2DT	**SVM**	**kNN**	**CNN**	**RNN**	**LSTM**
320×240	85.1	94.4	72.2	90.0	92.7	84.2
800×600	84.4	96.4	76.5	81.5	90.0	85.4
1260×960	68.6	97.5	50.9	66.4	90.6	59.1
Average	79.3	96.1	66.5	79.3	91.1	76.2
Sensitivity×Specificity (%)	77.15	55.83	61.84	74.62	78.16	71.85

optimizing the performances of KNeighborsClassifier. Models of CNN, RNN and LSTM were implemented in Python with Keras. The architecture of the CNN model was designed with 7 layers: one input layer, two convolutional layers, two globalmaxpooling layers, a fully connected layer, and a sigmoid output layer. Binary_crossentropy and adam were used as the loss function and the optimizer, respectively. SimpleRNN with 4 input layers, 32 hidden layers and 1 output layer was designed with batch size of 100 and epochs of 100. LSTM was also designed to have the same structure and parameters. As summarized in Table 4, the sensitivity of the two-step DT was superior to that of other algorithms (two-step DT 97.3% vs. SVM 58.1%, kNN 93.0%, CNN 94.1%, RNN 85.8%, LSTM 94.3%). The specificity of the two-step DT was less than SVM and RNN, but similar to or slightly better than CNN, LSTM, and kNN (two-step DT 79.3% vs. SVM 96.1%, kNN 66.5%, CNN 79.3%, RNN 91.1%, LSTM 76.2%). The combined performance of two-step DT, as calculated by sensitivity×specificity, is better than SVM, kNN, CNN, and LSTM and comparable to RNN.

CONCLUSIONS

Herein, we propose an automated and real-time method developed by the machine learning-based approach, which can be applicable for the detection and quantification of mouse scratching produced under general experimental settings and recorded with commercially available monochrome cameras such as a webcam. It is a non-invasive and inexpensive methods suitable for objective counting of scratching. The advantages of this method are several folds. First, the proposed method can detect scratching with excellent and reliable accuracy. The sensitivity and specificity measured in the performance test are similar or better than existing automated counting methods. Second, it requires neither an attachment nor a surgical implantation of a device to a subject animal which may not only cause inflammatory reactions in the skin but also affect mouse behaviors. Third, experiments can be performed in a homecage and require no additional specialized observation chamber or expensive equipment. Finally, the graphical user interface developed by this study would allow one to easily analyze scratching behavior by providing manual reviewing functions, and generating a raster plot and a table with quantified results. The performance of the proposed method could be further improved by recording movies with a high speed and resolution camera and by painting identifiable markers on the head and toe of the subject animal to facilitate the detect of characteristic scratching posture. The GUI of this method is available at: http://github.com/Park-Ingyu/MouseBehav (also, see Supplementary Material 1).

ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research (2017R1A2B1009234 to U.L.), the Brain Research Program (2016M3C7A1947307, 2017M3A9F1030063) of the National Research Foundation (NRF) of Korea, funded by the Ministry of Education, the Ministry of Science and ICT (2017R1A2B4003351 to H.L.) and Hallym University Research Fund (HRF-201707-011 to U.L.).

REFERENCES

1. Dong X, Dong X (2018) Peripheral and central mechanisms of itch. Neuron 98:482-494.
2. Lee JS, Han JS, Lee K, Bang J, Lee H (2016) The peripheral and central mechanisms underlying itch. BMB Rep 49:474-487.
3. Elliott GR, Vanwersch RA, Bruijnzeel PL (2000) An automated method for registering and quantifying scratching activity in mice: use for drug evaluation. J Pharmacol Toxicol Methods 44:453-459.
4. Inagaki N, Igeta K, Shiraishi N, Kim JF, Nagao M, Nakamura N, Nagai H (2003) Evaluation and characterization of mouse scratching behavior by a new apparatus, MicroAct. Skin Pharmacol Appl Skin Physiol 16:165-175.
5. Elliott P, G'Sell M, Snyder LM, Ross SE, Ventura V (2017) Automated acoustic detection of mouse scratching. PLoS One 12:e0179662.
6. Umeda K, Noro Y, Murakami T, Tokime K, Sugisaki H, Yamanaka K, Kurokawa I, Kuno K, Tsutsui H, Nakanishi K,

Mizutani H (2006) A novel acoustic evaluation system of scratching in mouse dermatitis: rapid and specific detection of invisibly rapid scratch in an atopic dermatitis model mouse. Life Sci 79:2144-2150.

7. Orito K, Chida Y, Fujisawa C, Arkwright PD, Matsuda H (2004) A new analytical system for quantification scratching behaviour in mice. Br J Dermatol 150:33-38.
8. Brash HM, McQueen DS, Christie D, Bell JK, Bond SM, Rees JL (2005) A repetitive movement detector used for automatic monitoring and quantification of scratching in mice. J Neurosci Methods 142:107-114.
9. Ishii I, Kurozumi S, Orito K, Matsuda H (2008) Automatic scratching pattern detection for laboratory mice using high-speed video images. IEEE Trans Autom Sci Eng 5:176-182.
10. Nie Y, Ishii I, Yamamoto K, Orito K, Matsuda H (2009) Real-time scratching behavior quantification system for laboratory mice using high-speed vision. J Real Time Image Process 4:181-190.
11. Tarrasón G, Carcasona C, Eichhorn P, Pérez B, Gavaldà A, Godessart N (2017) Characterization of the chloroquine-induced mouse model of pruritus using an automated behavioural system. Exp Dermatol 26:1105-1111.
12. Salem G, Krynitsky J, Pohida T, Hayes M, Burgos-Artizzu X (2016) Three dimensional pose estimation of mouse from monocular images in compact systems. In: 2016 23rd International Conference on Pattern Recognition (ICPR 2016): Cancun, Mexico, 4–8 December 2016 (Institute of Electrical and Electronics Engineers, eds), pp 1750-1755. IEEE, Piscataway, NJ.