國立臺北科技大學

電機工程研究所

碩士學位論文

利用機器學習演算法偵測與量化小鼠的疼痛

Detection and Quantification of Mice in Pain Utilizing Machine Learning Algorithms

研究生：陳毓峯

指導教授：吳昭正 博士

中華民國一百零八年一月

國立臺北科技大學
電機工程研究所
碩士學位論文

利用機器學習演算法偵測與量化小鼠的疼痛
Detection and Quantification of Mice in Pain Utilizing Machine Learning Algorithms

研究生：陳毓峯

指導教授：吳昭正 博士

中華民國一百零八年一月

國立臺北科技大學
研究所碩士學位論文口試委員會審定書

本校 電資學院電機工程 研究所 陳毓峯 君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委　　員： 李玟翰

陳志明

林建和

高立人

指導教授： 吳昭正

所　　長： 張陽郎

中 華 民 國 一百零八年 1 月 8 日

摘　要

論文名稱: 利用機器學習演算法偵測與量化小鼠的疼痛

頁數: 三十一 頁

校所別: 國立台北科技大學 電機工程 研究所

畢業時間: 一百零七學年度 第一學期　　　學位: 碩士

研究生: 陳毓峯　　　　　　　　　　　指導教授: 吳昭正

關鍵詞: 臉部特徵識別; 疼痛偵測與量化; 監督式分類學習法; 支援向量機; 哈爾濾波器; Mask RCNN;

動物試驗為現行測試藥物的重要流程之一，以止痛藥為例，動物實驗中透過將實驗動物施打藥物或手術等方式使其產生疼痛的感覺，之後再施打測試藥物用以測試該藥物的有效性。然而，實驗動物無法透過言語的方式表達其感受，因此目前以侵入式的測試方式為主，例如透過觸碰其疼痛處並觀察反應用以推測有效性。近年來，相關領域的研究開始探討以小鼠臉部表情偵測疼痛的可行性，文獻中雖然透過實驗設計證實此方法的可行性，但是實驗方法依然以小鼠專家的人工方式進行判讀，人工判讀方式容易因為個人的經驗與精神狀態而有分歧，並且難以量化疼痛的反應，因此無法用於長期疼痛的監測。本論文建立於相關文獻的基礎之上，開發自動化的系統透過小鼠的臉部表情進行自動辨識，論文中將利用機器學習的演算法自動學習小鼠臉部表情代表疼痛的特徵，取代人工判讀的部分，自動化整個流程以達到長期監測疼痛的目的。

本論文共分兩階段，第一階段著重在偵測疼痛的部分，此階段利用監督式與非監督式機器學習演算法，學習小鼠臉部疼痛以及非疼痛時的臉部特徵變化量，臉部特徵又分為手動標示與自動學習的部份，比較兩者的偵測率與效能之差異。第二階段延伸第一階段的結果，加入線性回歸的特性針對疼痛進行量化。實驗部分為了評估機器學習算法的準確性，本研究使用高速攝影機錄製不同實驗設定的小鼠影片，實驗分別使用相同劑量的辣椒素和弗飾佐劑，和不同劑量的雙氯芬酸，前者用以蒐集疼痛時小鼠的影像訓練分

類器，後者以不同劑量的止痛劑評估演算法的有效性。最後將機器判斷的預估結果和小鼠專家進行比較，實驗結果驗證本論文提出架構之可行性與效能，期待本論文的貢獻未來可以用以突破監測疼痛的技術瓶頸。

Abstract

Title: Detection and Quantification of Mice in Pain Utilizing Machine Learning Algorithms

Pages: 31

School: National Taipei University of Technology

Department: Electrical Engineering

Time: January, 2019 Degree: Master

Researcher: Yu-Feng Chen Advisor: Chao-Cheng Wu

Keywords: pain face; facial pattern; detection; quantification; supervised classification; support vector machines; haar cascade; mask RCNN;

Animal experiments have been a critical process for drug test. If painkillers were taken for example, some operations would be performed on animals to produce the feeling of pain, and then various dosage of painkillers would be injected for evaluation. Unfortunately, it is impossible for animals to communicate in the same way human being does, so the current evaluations mainly relied on their behaviors. Recently, facial expressions have been explored and demonstrated for detection of pain in literature. Facial expression approach was so-called mouse grimace scale (MGS) method. Unfortunately, the proposed methods were semi-quantitative and still had to rely on human interpretation for evaluation of pain. Therefore, the thesis utilized machining learning algorithms to quantify pain level via facial expressions, and developed a segmentation tool to extract facial expressions for automatic determination of the mouse grimace scale (MGS).

The thesis consisted of two parts. The first part aimed to develop a classification model for pain detection. The facial expressions were simplified to a set of features, including angles and distances among the locations of eyes, ears, and nose. Then, machine-learning algorithms were applied to train a classification model. The second part further quantified level of pain and validated performance of machine learning algorithms compared with the human interpretation of experts. The experimental studies took advantage of complete Freund's adjuvant(CFA) to simulate spontaneous pain in clinical, and diclofenac was used as the painkiller. In order to evaluate level of pain, different amount of diclofenac were injected, and the facial expressions of mice were recorded as several videos by a high-speed camera. The experimental results demonstrated that the proposed models are applicable to detect the pain via facial features with low false alarm. Besides, the machine learning algorithms are also effective for estimation of level of pain.

致謝

本論文得以順利完成，首要感謝指導老師吳昭正教授，在碩士學程期間給予我許多實質上的幫助與指導，每周的開會討論研究，都有嶄新的觀點與實驗想法，引導我們用更深層的思維邏輯來看待問題，這在我對研究剖析時有深刻的影響，受益良多。課後之餘，老師的關係與我們亦師亦友，常和我們分享許多處事經歷，並且鼓勵我們步足國際。在此再度特別感謝老師在大學和碩士期間，推薦和鼓勵我到日本和法國實習，參與國際研討會。再擁有廣闊的國際視野的同時，也讓我大幅成長。此外，也特別感謝中央研究院生物醫學科學研究所的陳志成老師與李政翰博士，提供專業知識、資料和生醫上的知識，讓我有機會能在碩論期間獲得跨領域合作的機會，以及感謝所有蒞臨指導的口試委員。

再來感謝實驗室的學長紹安、彥均、祐綸、晉瑋、浩軒及宗祐，在我剛進研究所時帶我了解許多實驗室的相關研究並且讓我逐漸熟悉實驗室的環境，以及浩民、鉉宗、東樺、冠儒、秉宏、懿賢、沐海、佳婷與宇眠，在撰寫論文和實驗期間給予鼓勵與支持。讓我在念碩士期間的熬夜時光特別有精神和動力。最後也要感謝實驗室所有的學弟妹，願意讓我有試教的機會，讓我能有機會把我所理解的專業知識，依照學習人的程度不同去選擇用不同的話語描述。這兩年的碩士期間無疑是我人生裡色彩斑斕的兩年。

最後感謝我的父母，一路支持我完成學業，放手讓我去追求我所熱愛的領域。 特別是當我要去法國和日本實習的時候，時時刻刻關心在海外的我是否吃得溫飽。我想我能夠順利完成碩士學業，除了感謝自己認真對待自己熱愛的事物以外，要感謝的人真的太多了。僅僅一頁的篇幅並不能寫下我所有的感激，所以我只能寫到此，關閉我的文檔去和路途上曾幫助我的老師與朋友表達感激。

Content

List of Tables

List of Figures

Chapter 1 INTRODUCTION

1.1 Motivation and Background

Machine learning is a subset of artificial intelligence that allows software to mimic the cognitive functions of human without explicitly programming. With advances of machine learning in computer vision, many algorithms have been reliable and robust, which were capable to predict correctly by observing patterns from image data. Because it continues to flourish in these years, image-based analyses had been widely applied in biological fields for pain quantification.

In animal experiments, drug tests were designed for evaluation of painkillers. Unfortunately, effective measurement of pain was not available. In fact, to precisely evaluate pain was a hot topic in biological and medical fields. Thus, the thesis tried to engage the machine learning and image processing techniques to develop a standard process for pain measurement via facial expressions. Based on this hypothesis, the SVM classifier was utilized for pain classification and compared its performance with the current methodology, which was manually confirmed by the mice experts. Then, regression functions were applied for estimation of pain level. However, to grab frames with the full set of facial features from videos was extremely time consuming. At completion of the proposed work, this thesis expected to achieve: (1) an automatic annotation method to grab frames with the full set of facial features from videos. (2) A pain scale method without human interpretation, which implemented support vector machine and linear regression for classification and estimation. (3) Validation of machine learning algorithms for pain scale. Since the proposed methods need biological experiments for validation, the drug experiments and background knowledge were provided by the experts from Academia Sinica. Capsaicin and CFA were chosen and injected at nape skin and left hand paw respectively[1]. The diclofenac was used as analgesic to relief the pain in order to validate the proposed method.

In the literature, the facial expressions had been exploited as an important indicator of pain. The facial coding scales were originally proposed by Ekman and Friesen [2] in 1978, and were further applied on non-verbal human populations in 2002[4]. Moreover, Darwin predicted that nonhuman animals exhibit similar facial expressions for emotional states as humans do in 1872 and 1998 [5][6]. Based on those proposed theories, this thesis made a reasonable assumption that pain in mice affected their facial expression. It also extended the principle of human face detection to develop an annotation tool. The general uses of rapid object detection was first proposed by Viola [13] in 2001, and was further investigated on human face by Wilson [15] in 2006. However, Kaiming-He proposed a robust and high accuracy technique for object detection in 2017 [19], which was Mask R-CNN for instance segmentation. A comparison was done in the experimental studies to analyze advantages and disadvantages of two different techniques. There were two major objectives to develop an automatic annotator. One was to collect the data for training pain classification models and validating effectiveness of models. Since the thesis was based on image analysis, and the time cost of annotation was extreme high. Thus, to develop an algorithm for the facial patterns would help to reduce the cost for sample collection; the other was to automate the process. Sotocinal proposed a partially semi-automated method for quantifying pain for the laboratory rat in 2011 [1]. In the paper, Haar cascade was utilized for semi-annotation of mouse images, but it can not be completely automatic because Haar Cascade was hard to detect the subtle patterns, which contained orbital tightening, cheek flattening, whisker and ears changing. Moreover, the level of pain still had to rely on human interpretation, which might be inconsistent and subjective. To improve the process, the thesis simplified facial expressions to a set of features for estimation of pain by a pre-train model.

1.2 Objective

The long-term goal of the thesis was to better understand effectiveness of pain classification by utilizing machine learning algorithms so that its contribution can improved accuracy for measurement of pain. The objective was to develop a standard procedure for pain measurement and an automatic annotation tool for data collection. Due to different facial expressions on pain, the proposed approach had been developed to provide a robust measurement, and the following experimental studies had proved its effectiveness. It included three steps: automatic annotation of facial patterns, supervised classification algorithms, and frame-based regression.

The target algorithms included Mask Region-Convolution Neural Network (Mask R-CNN), Haar Cascade, support vector machine (SVM), and regression. To balance between speed and accuracy while executing annotation of facial patterns, the thesis compared performance between Mask R-CNN and Haar Cascade. They took advantage of characteristics of facial expressions for classification of pain. Then, SVM was utilized to form a model for classification, and the experimental results demonstrated that the proposed approach could be an accurate and consistent diagnosis tool.

1.3 Organization

This thesis was organized as follows. Chapter 2 briefly introduced the history of algorithms and image processing techniques. Chapter 3 described the proposed methodologies and compared its performance with the groundtruth manually confirm by mice experts. Chapter 4 demonstrated utilization of the automatic annotation tool and pain classifiers and illustrated the facial expressions that classifiers utilized as features to detection pain. Chapter 5 summarized the conclusions and future work.

Chapter 2 OVERVIEW

This chapter briefly introduced the development history and methodology for implementation of machine learning algorithms, which included Haar classifier, Adaptive Boosting, Mask R-CNN, and support vector machine.

2.1 Haar Classifier Algorithms

Haar cascade algorithm was based on weak classifiers and boosting algorithms. Each weak classifier contained a Haar feature, which calculated subtraction of the pixel area as shown in Figure 2-1. A weak classifier was considered as a single vote, and classification problem would be determined based on the number of votes multiple weighting. AdaBoost algorithm was an algorithm used to modify the weights to improve accuracy in a loop. Overall, Haar cascade used the contrast values to determine relative light and dark area in a sub-window and to classify the mouse face by cascading a series of weak classifiers.

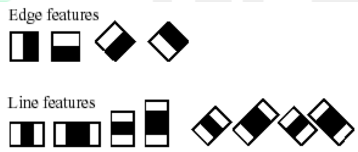


Figure 2-1. Illustration of standard Haar features from [13].

Recalled that there were over 180,000 rectangles features associated with each sub-window of a mouse face image, a number far larger than the number of pixels [13]. The general solution of the expensive process was to reduce the number of features computation and classifier learning [13][14][15]. Therefore, an inequality function was frequently used to simplify classifiers; In the meantime, an integral image was used to optimize computation of Haar features. The details of two solutions were described as follows.

In the first solution, a weak learning classifier determined the optimal threshold in order to minimize the number of misclassified samples. It consisted a Haar feature x (rectangle), a threshold θ_j, and a polarity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1, & if \ \ p_j f_j(x) < p_j \theta_j \\ 0, & otherwise \end{cases} \tag{2.1}$$

where j represented the number of weak classifiers in a completed set of sub window.

The second solution exploited a simple idea of integral image. Each pixel in an integral image was a sum of a specific region in the original image, so a Haar feature could efficiently be computed by in an operation as shown in Figure 2-2. The function of an integral image was shown as follows:

$$ii(\mathrm{x}, \mathrm{y}) = \sum_{x' \leq x, y^i \leq y} i(x', y') \tag{2.2}$$

Where i and ii represented the original and integral images respectively.

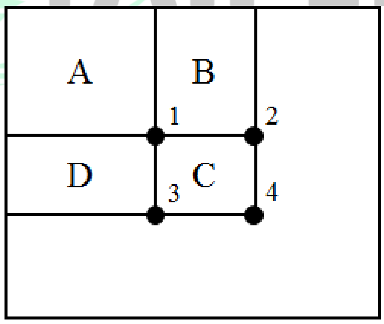


Figure 2-2. Illustration of computing Haar features on integral image. To sum up the rectangle C, the operation could be easily computed as 4 + 1 – (2 + 3) from [13].

Although the above methods speed up the computation time, the accuracy of each weak classifier was not good enough for application. Moreover, it still needed to compute 180,000 features in each sub-window, and some of them were irrelevant. Therefore, the adaptive boosting algorithm was proposed to select meaningful features and boost performance.

2.2 Adaptive Boosting Algorithm and Cascade Method

AdaBoost initialized the weighting of each weak classifier by the uniform distribution, and the α_t in the final function would be determined based on the scale element β_t, which controlled the trend of weighting according to the error rate ϵ_t in each iteration. In the end, the cumulative function would be used to classify objects of interest in a higher detection rate. For reducing the false alarm rate, a serious of pre-trained models was cascaded to form a robust classifier as shown in Figure 2-3. Normally, the cascaded order implemented by the thesis moved the stages with high rejection rate to the front in order to reduce the process time.

- Given example images $(x_1, y_1), \ldots (x_n, y_n)$, where $y_i = 0,1$ for negative and positive examples respectively.
- Initialize weights $\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights $\omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,j}}$, so that ω_t is a probability distribution.
  2. For each feature, j train a classifier h_j, which is retricted to using a single feature. The error is evaluated with respect to ω_t, $\epsilon_j = \sum_i \omega_i \left| h_j(x_i) - y_i \right|$
  3. Choose the classifier h_t with the lowest error ϵ_t.
  4. Update the weights: $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$

     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & otherwise \end{cases}$$

where $\alpha_t = log^{\frac{1}{\beta_t}}$

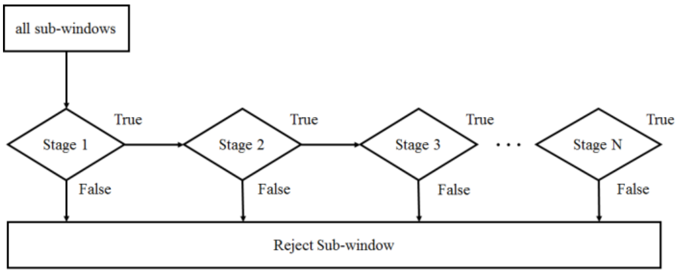


Figure 2-3. Schematic depiction of a detection cascade; a redraw version from [13].

2.3 Mask Region Convolution Neural Network

Kaiming-He presented a conceptually simple, flexible, and general framework for object instance segmentation in [19]. Then, Mask R-CNN becomes a state-of-the-art and robust technique for object segmentation since 2017. The core idea of Mask RCNN was to decouple mask and class prediction; meanwhile, it used bilinear interpolation for better maximum pooling in Faster R-CNN for accurate pixel alignment and object bounding box recognition as shown in Figure 2-4. The structure of Mask R-CNN contained a two-stage procedure. One stage was Region Proposal Network (RPN [24]) and predication of class, and the other was box offset and a binary mask. In contrast with traditional methods, which classification depended on mask predictions, the mask prediction was performed in parallel with classification.

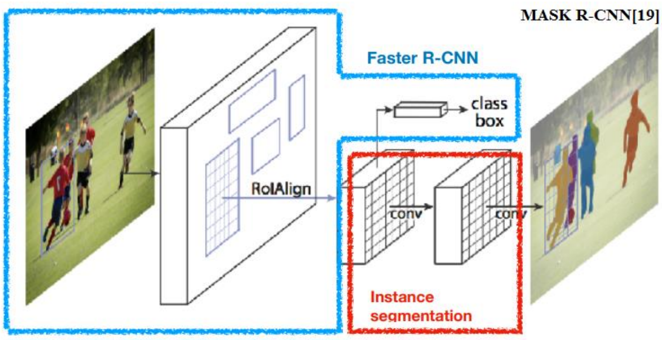


Figure 2-4. Illustration of Mask R-CNN framework from [19].

Formally, the loss function was defined as:

$$L = L_{cls} + L_{box} + L_{mask} \tag{2.3}$$

The classification loss L_{cls} and bounding-box loss L_{box} were followed the definition in [23]. The mask loss L_{mask} was defined as the average binary cross-entropy loss and cooperated with the per-pixel sigmoid function, which allowed the network to generate masks for every class without competition among classes. This loss function L was verified as a key function in [19].

2.4 Linear-Based Support Vector Machine

Support Vector Machine algorithm was implemented because it could generate a reliable classification model based on limited number of training samples. Moreover, Support Vector Machine has shown excellent performance on classification in these years. Its core principals was described as follows. It mapped the p-dimensional data samples into $(p-1)$-dimensional feature space through a linear mapping function. Then a linear decision function was constructed in the feature space as shown in Figure 2-5.

Let $\{(x_i, y_i)|x_i \epsilon R^d, i = 1,2, \dots, n \epsilon S\}$ denoted n samples of an image, where S was the set of pixels, d was the number of facial patterns and $y_i \epsilon \{-1,1\}$ indicating the class to which the sample x_i belongs. Since all the samples had been perpendicular to the hyperplane, a linear decision function in feature space is of the form:

$$f(x_i) = \ w^T x_i + b \tag{2.4}$$

where w was an unit vector and b was a scalar. The outcome of the linear function for each x_i would be $f(x_i) \geq 0$ if $y_i = 1$ and $f(x_i) < 0$ if $y_i = -1$. In order to find the *optimal hyperplane* with the largest margin between two classes, the SVM algorithm selected the samples that are closest to the separating hyperplane, so called support vectors. The hyperplane determined by support vectors can be written as follow:

$$f(\mathrm{x}) = \begin{cases} w^T x_i + \mathrm{b} \geq +1, & \text{THEN } \textit{positive} \\ w^T x_i + b \leq -1, & \text{THEN } \textit{negative} \end{cases} \tag{2.5}$$

Then, the above inequality was re-wrote as follow:

$$y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, n \tag{2.6}$$

The width of margin was $\frac{2}{\|w\|}$ to maximum the margin to reduce the error. In 1995, Cortes and Vapnik proposed a maximum margin method, so called *soft margin* [18]. The soft margin SVM introduced slack variable to solve the minimum problem as follow:

$$\min J(w, \xi) = \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_{i=1}^{n} \xi_i \tag{2.7}$$

where ξ_i was a slack variable, and it always satisfies $\xi_i \geq 0$. C was a penalty constant to control the model complexity. To estimate the pain, this thesis introduced a regression-based estimator, which extended the results of Support Vector Machine algorithm in order to quantify the level of pain.

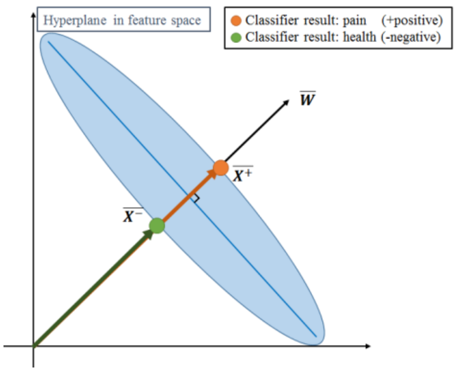


Figure 2-5. Illustration of Support Vector Machine algorithm; Sample x was be classified to pain and health by hyperplane in feature space.

Chapter 3 SUPERVISED PAIN FACE QUANTIFICATION

This thesis utilized the machine learning algorithms to quantify pain level via facial expressions and developed an automatic annotation tool to annotate facial expressions for the mouse grimace scale (MGS). The proposed methodologies included three steps: automatic annotation of patterns, a supervised classification model, and a frame-based estimator for pain quantification. The methodologies were explained in this chapter, and the flowchart was shown in Figure 3-1. The comparison conducted by experimental studies and the detail of database were described in next chapter.

3.1 Flowchart of Proposed Methodologies

Figure 3-1 illustrated the general ideas of the designed experiments and the comprehensive view of the proposed framework. To simulate the environment, the experts in Academia Sinica of Taiwan injected the algesic and analgesic substance on mice and recorded the videos by a high-speed camera for implementation of machine learning algorithms. Since the experts and the proposed methods were relied on facial patterns to detect pain, the developed annotation tool was able to automatically annotate the facial features from frames. Then, the features were used to train a SVM classifier for detection of pain, and a statistical based estimator was applied at the end for quantification as shown in Figure 3-1.

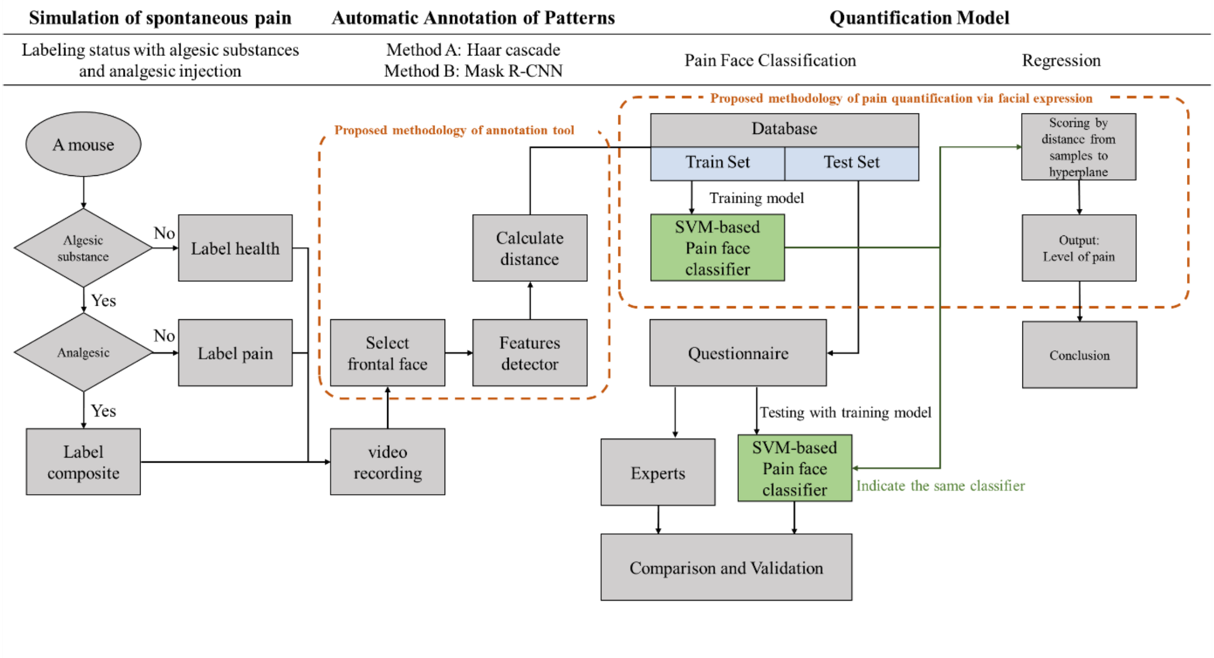
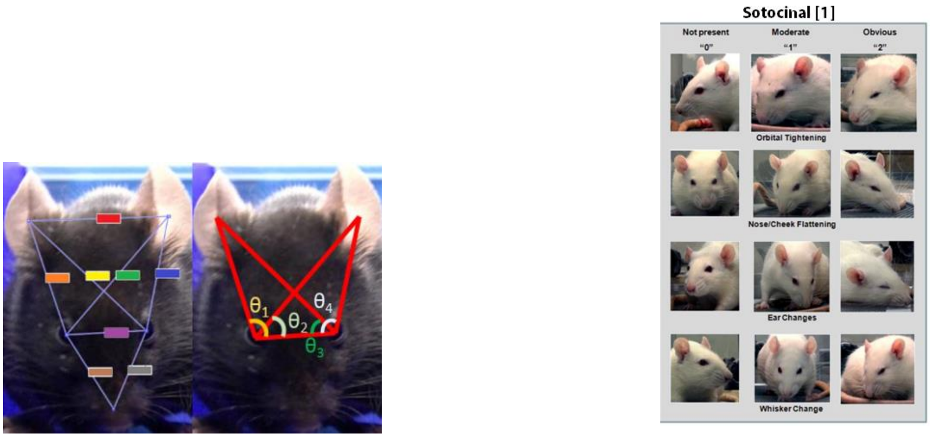


Figure 3-1. The flowchart of methodology for pain quantification and validation

3.2 Automatic Annotation of Facial Patterns

The goal of annotation tool was to automatically recognize the facial patterns from frames. This thesis proposed to simplify facial expressions to a set of features, including angles and distances among locations of eyes, ears, and nose indicated in Figure 3-2. In contrast with previous method [1], where depended on subtle features, the proposed features were able to provide more information for quantification. The outputs of annotation tool would be fed into the classification models [8]-[24] for object recognition and algorithm selection. Finally, these two candidates were chosen for object annotation. One was Haar Cascade, which was trained with gray scale images; another was Mask R-CNN, which was trained with color images. In order to evaluate performance of the candidates, five videos were recorded from five different mice for comparison. Then, one of five videos was chosen for training the models as shown in Figure 3-3. The performance was evaluated by the other four videos. The learning features of Haar Cascade was shown in Figure 3-4 as a rectangle, which mixed with white and black area.



(a) (b)

Figure 3-2. (a) Proposed patterns: A set of features including distances and angles among the eyes, ears and nose; (b) In method [1], the experts were required to score pain via subtle facial patterns.

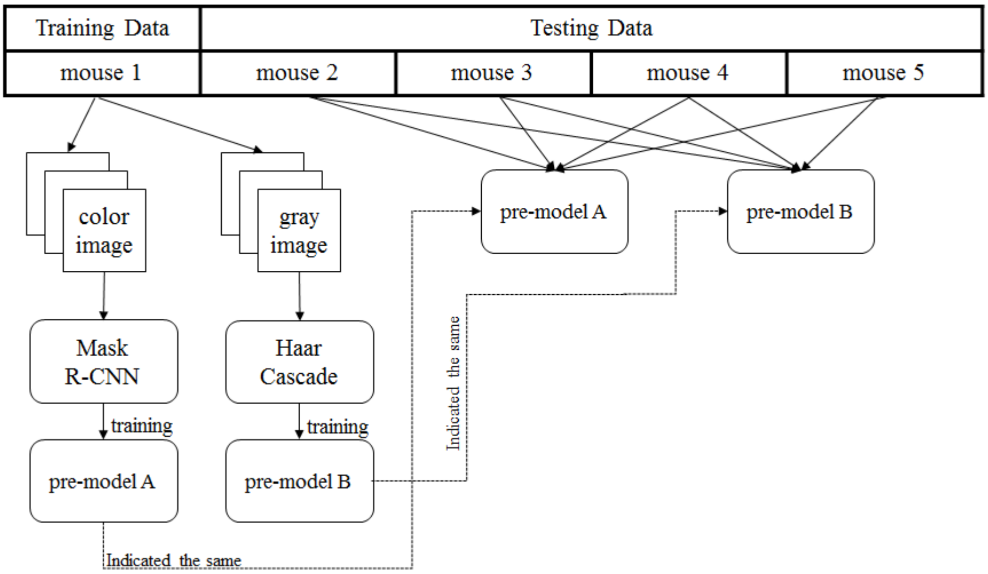


Figure 3-3. Illustration of performance evaluation. Pre-model "A" was trained by Mask R-CNN algorithms; pre-model "B" was trained by Haar Cascade.

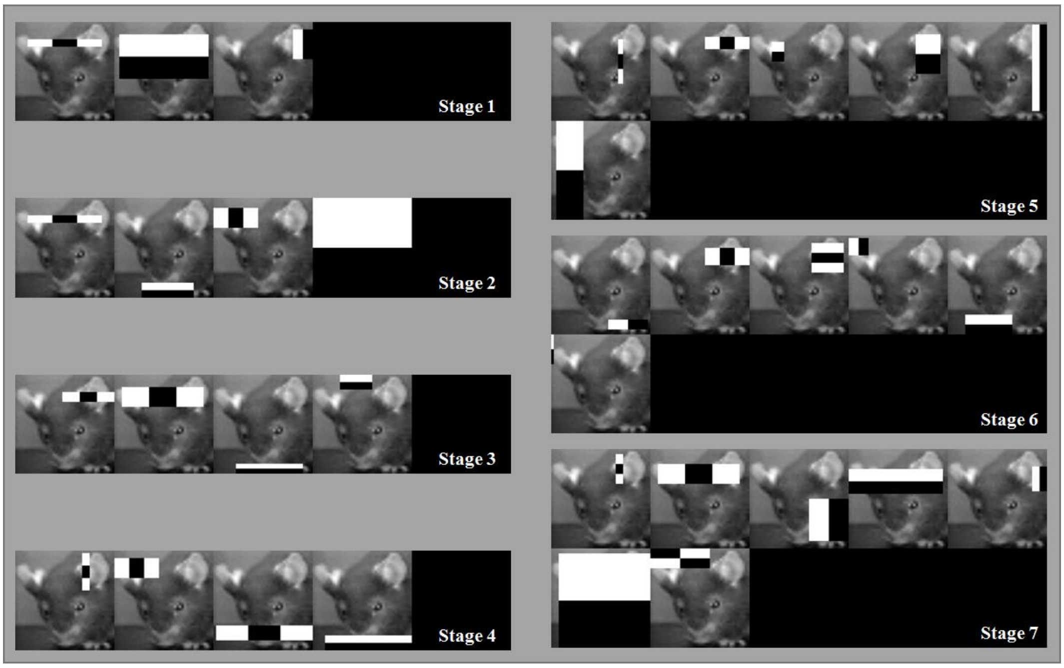


Figure 3-4. Visualization of Haar Cascade in a series of eight stages. Each stage contained some selected weak classifiers, and the final output of a strong classifier must pass through all stages.

3.3 Support Vector Machine Classifier

Two different training strategies of a pain classifier were introduced in this section and evaluated on the same mice. One was to train the pain classifiers separately for each mouse; another was to train a single classifier by a mix of six mice at once. In fact, the pain reaction on facial expressions was slightly different. Hence, the difference play a critical role on performance of two strategies as shown in Figure 3-5 and Figure 3-6. The experiment recorded ten minutes videos for six mice separately, and the facial patterns were automatic annotated by Mask-RCNN. Then, the patterns were fed to linear SVM for training and quantifying pain. SVM was applied to train classifiers due to limited training samples. The next section would further introduce a method for estimation of pain scale.

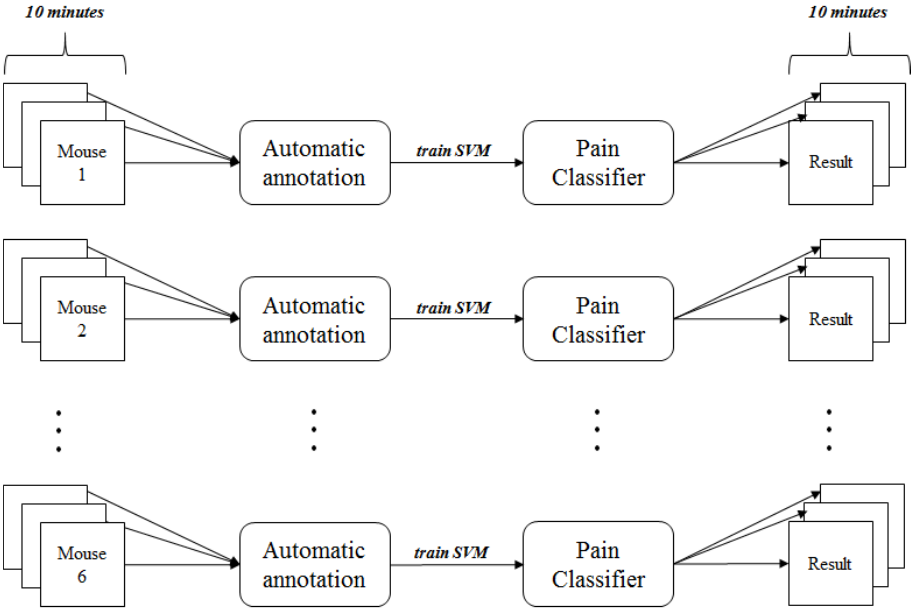


Figure 3-5. First strategy: train the pain classifiers separately for each of experimental mice.

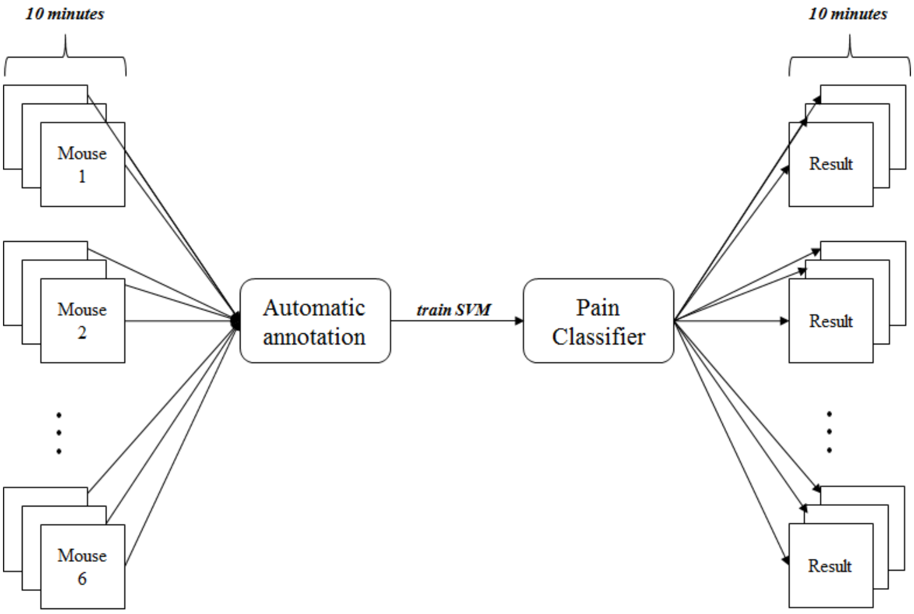


Figure 3-6. Second strategy: train a single model with all the dataset.

3.4 Frame-Based Regression

A regression based estimator was placed at the end stage for estimation of pain scale as shown in Figure 3-8. The regression function calculated the distance from samples to hyperplane in the feature space. In order to reduce the overfitting issue while regression, a tolerance was added to reduce the number of support vectors in SVM algorithm.

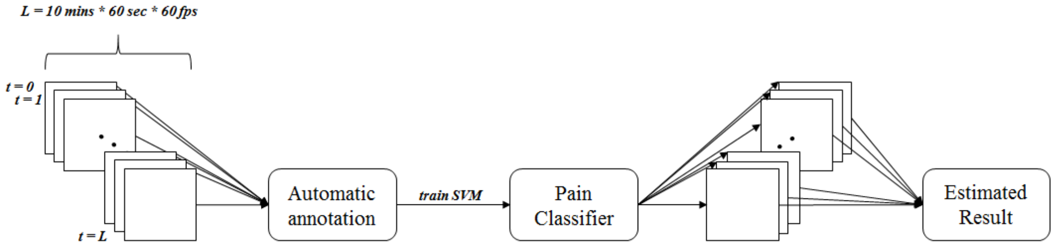


Figure 3-7. Flow chart of pain estimator.

The tolerance was added to the equation (2.6) as shown in equation (3.1).

$$y_i(w^T x_i + b) \geq 1 + \varepsilon, i = 1,2, \ldots, n \tag{3.1}$$

where ε controlled the width of the insensitive zone, and always satisfies $\varepsilon \geq 0$. The tolerance solved the overfitting problem by reducing the number of support vectors.

Continued the equation (2.4), the unit vector w was given when solving the equation (2.7). Then, the estimated function of pain level was simplified in the following equation:

$$Pain\ \ scale\text{:}\ f(z_i) = \sum\nolimits_{i=1\ldots N} \frac{w_i x_i + b}{N} \tag{3.2}$$

where i represented the i^{th} frame and z_i was the pain scale of a frame. The distance from sample X to hyperplane were calculated by $(\omega_i x_i + b)$ since the dot product projected the vector x on w.

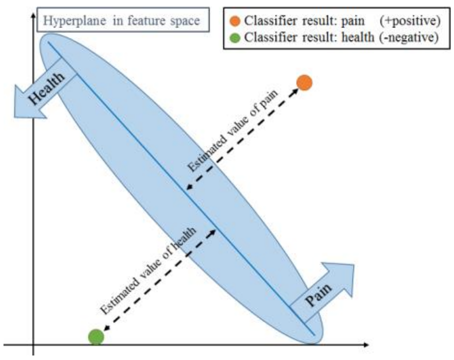


Figure 3-8. Illustration of the level of pain. The estimated value calculated distance from sample to hyperplane in feature space.

Chapter 4 EXPERIMENT RESULTS

Performance of pain scale was compared with the ground truth manually confined by mouse experts, which was the present method to classify and measure the pain via facial expressions. However, performance of developed annotation tool was evaluated by the ground truth manually annotated through open-source software, Viper and VGG annotators. Overall, the evaluation was based on two criteria, which were injection quantity of analgesics and algestic substance and overlapping areas between manual and automatic annotated bounding box.

4.1 Database

The thesis had established two images database, which were the CFA dataset and Capsaicin dataset as shown in Table 4-1 and Table 4-2. Each mouse was enclosed in a box to capture its facial expressions in videos as shown in Figure 4-1. The length of videos was set to ten minutes, and the groundtruth of each frame was labeled as pain if the video was recorded after the injection of algesic substance. In order to train machine learning algorithms, six mice were injected capsaicin, which was one of the strongest algesic substance to induce pain. The other six mice were injected CFA and diclofenac to evaluate effectiveness of machine learning algorithms.

Figure 4-1. Illustration of the hardware equipment.

Table 4-1. An image dataset under the condition of before and after capsaicin injection at nape skin.

Capsaicin-dataset (frames)		
Mouse	*pre-capsaicin*	*post-capsaicin*
male2-1	311	352
male3-1	328	416
male3-2	364	345
female2-1	390	315
female3-1	513	349
female3-2	310	306
Total	2529	2426

Table 4-2. An image dataset under the condition of before and after CFA injection at left hand paw, and were given different treatment in settle time.

CFA-dataset						
Mouse	*pre-CFA*	*post-CFA*	*diclofenac 1hr*	*diclofenac 2hr*	*diclofenac 3hr*	*Diclofenac quantity*
BK	36135	36105	36015	36000	36060	10mg kg
LW	36105	36090	36030	36045	36015	10mg kg
dMW	36570	36075	36015	36015	36015	3 mg kg
RK	36045	36060	36015	36030	36015	3 mg kg
MW	37545	36180	36030	36000	36000	None
SW	36150	36060	36000	36000	36030	None

4.2 Efficiency of Automatic Annotation

Performance of the facial patterns annotator was evaluated on the CFA dataset. Both methodologies utilized one of six videos for training of models and the others for evaluation. Since the error of classification might occur, a common description of the error rate was used by the terms as follows:

True Positive *(TP): The facial patterns were correctly circled by the recognized bounding boxes.*

True Negative (TN): The background was correctly not circled by the recognized bounding boxes.

False Positive *(FP): The background was incorrectly circled by the recognized bounding boxes.*

False Negative *(FN): The facial patterns were incorrectly not circled by the recognized bounding boxes.*

Figure 4-2 and Figure 4-3 demonstrated the face detection by implementing Haar Cascade and Mask R-CNN. In fact, Haar Cascade was easily affected by the quality of images, such as contrast and saturation. Therefore, lots of false classified bounding boxes were shown in Figure 4-2-(d). Moreover, the eyes and nose are the subtle features, which caused the classifier hardly to detect by using the spatial or pixel information. Unfortunately, Haar features relied on the contrast of neighbor pixels. Hence, its performance was worse to look for facial patterns.

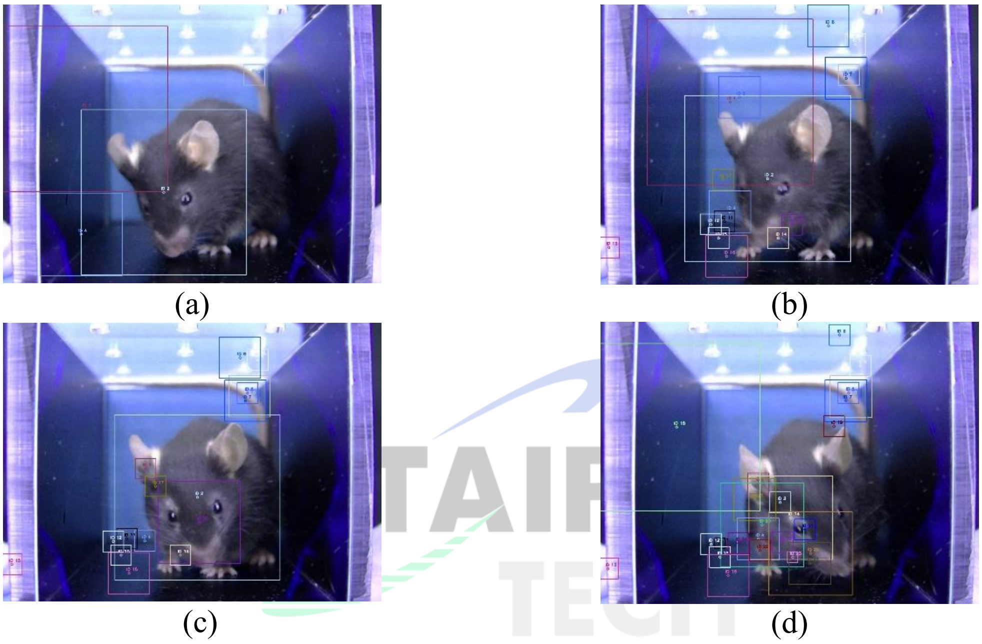


Figure 4-2. Face detection by Haar Cascade in different front view.

Comparison between two strategies for face detection was shown in Table 4-3. The results indicated that both methodologies could correctly circle the faces when the mouse faced the camera as shown in Figure 4-2-(a) and Figure 4-3-(a). However, many false positive bounding boxes occurred on Haar Cascade. This might be because of the quality of image and the algorithms itself. Normally, Haar Cascade resized the region of interest to a 24 by 24 mask as shown in Figure 3-4. Then, the decision boundary was trained to learn how to distinguish mouse faces from background by analyzing the pixel and spatial information on that mask. Unfortunately, some of the information were lost while reducing the spatial resolution. This step would make Haar Cascade learn patterns from blur images. Therefore, regions would be falsely detected if it had similar contrast as the training data. Moreover, the false positive cases would decrease performance of the following pain scale. After all, Mask R-CNN was chosen for annotation of facial patterns since its accuracy was much better. The recognized results of each facial pattern were shown in Table 4-4. The result indicated that deep learning algorithm was able to detect the eyes, which are hard for Haar Cascade.

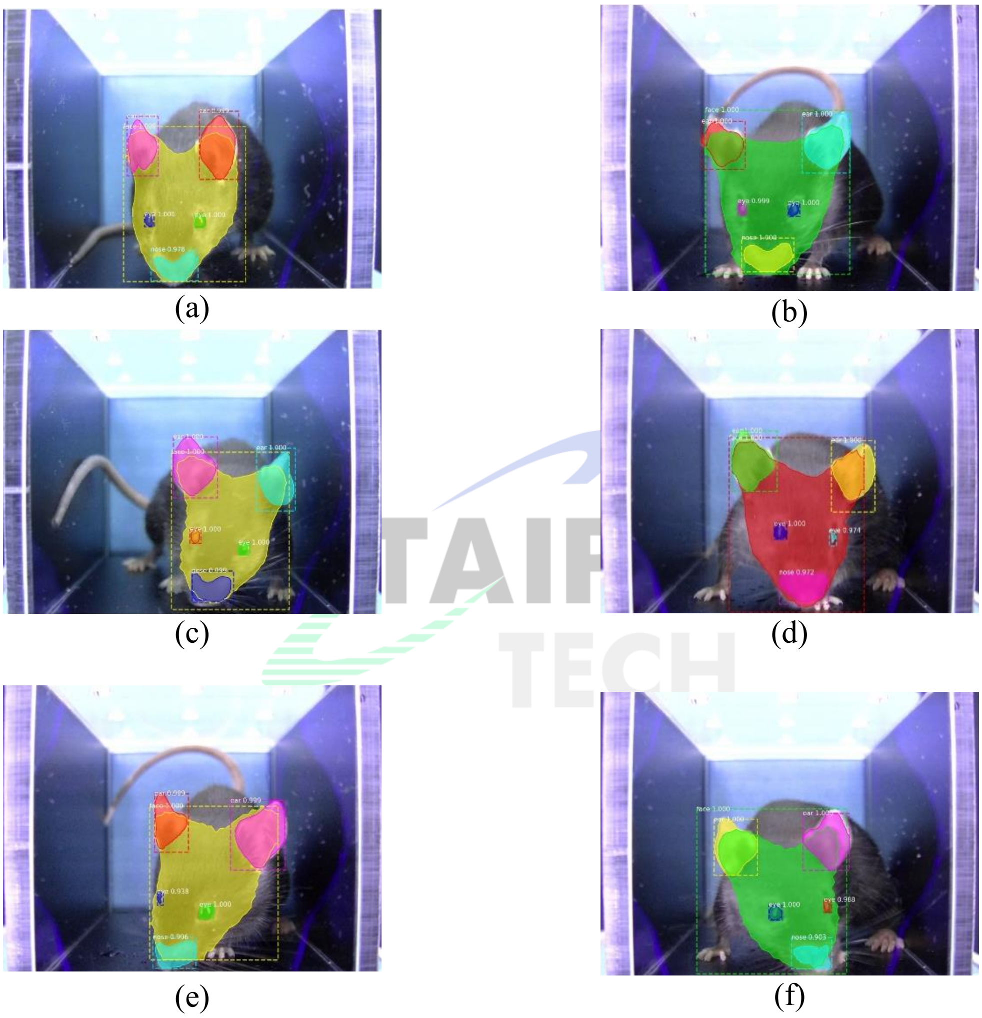


Figure 4-3. Face detection by Mask R-CNN in different front view.

Table 4-3. Classification results of two methodologies on face detection.

Methodologies	Videos	TP	FP	TN	FN	Detection Power	False Alarm	Accuracy
Haar Cascade (adaBoost)	1	49604	1350817	0	0	100	100	3.54
	2	66441	1118573	0	0	100	100	5.60
	3	16167	1126797	0	0	100	100	1.41
	4	23792	1216051	0	0	100	100	1.91
	5	42085	974988	0	0	100	100	4.13
Mask R-CNN (ResNet-101-FPN)	1	1669	95	34075	300	**100**	**0.278**	**98.90**
	2	2885	70	32025	1125	**100**	**0.218**	**96.69**
	3	612	1684	33662	60	**91.07**	**4.76**	**95.15**
	4	878	5324	29738	178	**83.14**	**15.18**	**84.76**
	5	1695	3452	29906	1007	**62.73**	**10.34**	**87.63**

Table 4-4. Automatic annotation result of facial patterns by implementing Mask R-CNN.

Methodologies	Videos	Face	Right eye	Left eye	Right ear	Left ear	Nose
Mask R-CNN *(ResNet-101-FPN)*	1	98.9	**98.9**	**98.9**	98.9	98.8	97.6
	2	96.6	**96.6**	**96.6**	96.6	96.6	92.5
	3	95.1	**95.1**	**95.1**	95.1	95.1	95.1
	4	84.7	**84.9**	**82.4**	84.9	84.9	90.8
	5	87.6	**86.4**	**85.6**	86.4	86.4	87.6
Average Performance		92.5±5.4	**92.3±5.6**	**91.7±6.4**	92.3±5.6	92.3±5.6	92.7±3.4

4.3 Effectiveness of Facial Patterns and Pain Classifier

Performance of proposed features and pain classifiers was evaluated on the capsaicin-dataset because it was one of the strongest algestic substance to cause spontaneous pain. The experimental studies compared two different strategies of training methods, and their results indicated that the model could be confused by individual difference among mice. To evaluate the performance, a common description of the error rate was used by the term as follows:

True Positive *(TP): the frame was correctly classified as pain, and the ground truth is pain.*

True Negative (TN): the frame was correctly classified as health, and the ground truth is health.

False Positive *(FP): the frame was incorrectly classified as pain, but the ground truth is health.*

False Negative *(FN): the frame was incorrectly classified as health, but the ground truth is pain.*

Table 4-5 demonstrated that the performance would be improved when the number of features increase. It indicated that the proposed features are complementary and provide specific information for classifiers. In addition, Figure 4-4 showed that the facial reactions of each mouse is different when feeling pain. For example, the inconsistent performance presented in Figure 4-4 indicated difference of facial expressions between male 3-1 and 3-2. If both mice had the similar facial expressions, the performance should be similar to each other. Figure 4-5 was the schematic diagram of Figure 4-4 to illustrate the rank of facial patterns for each experimental mouse. The color of dots represented separability of each pattern, and the location referred as its corresponding performance for pain classification. If the point was located at the upper-left side, it showed that the corresponding pattern was an important index for classifiers to observe the pain face of mouse. However, none of the dots represented that there was a robust and consistent pattern for all experimental mice. In another words, six mice had different facial patterns during pain. Moreover, the results in Table 4-5 also showed that it is better to train the SVM model for each of them. The performance was bad when a single model was trained by a combination of facial expressions from all mice. Therefore, the experimental results demonstrated that it is better to train a SVM model by using all the proposed features, and the corresponding model should be trained for each of mice to accommodate their individual difference.

In the next step, the outcomes of classifiers were evaluated and validated by human interpretation of experts tabulated in Table 4-6. A questionnaire was designed for this purpose. The questionnaire randomly selected 10 images from each mouse in the capsaicin dataset, so in total there were 60 image frames from six mice. The results in Table 4-6 indicated that machine learning algorithm is able to provide better and robust results than human interpretation of experts based on facial expressions of mice.

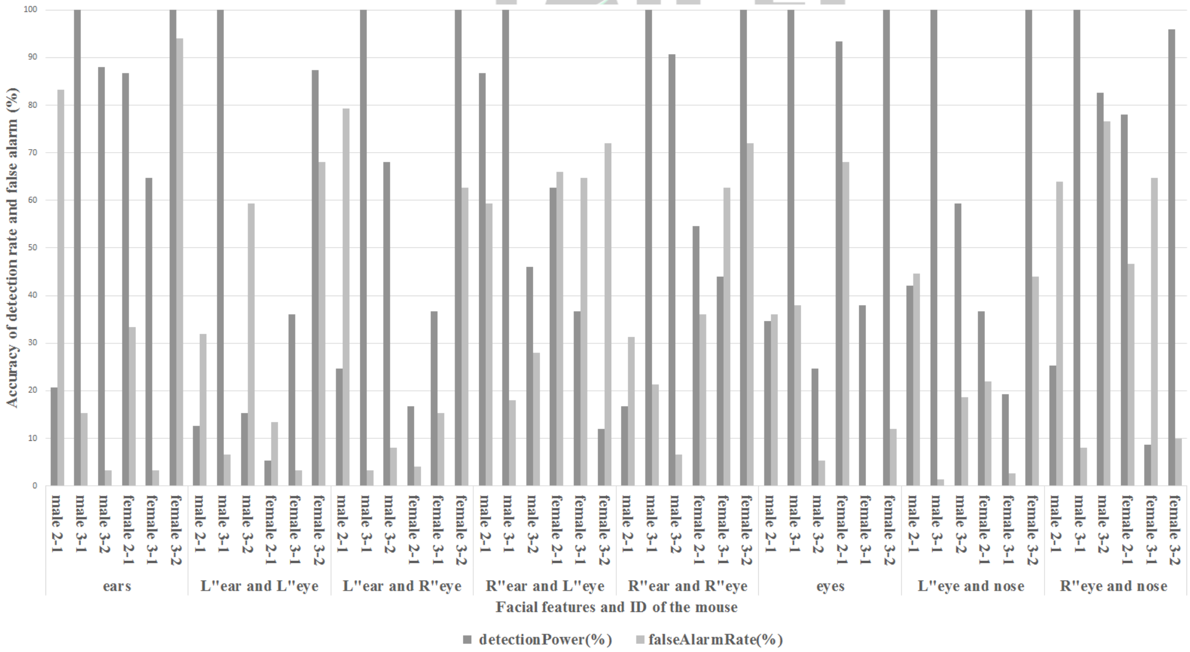


Figure 4-4. Performance of using different facial patterns to classify pain. Capital "R" and "L" in the features list are an abbreviation of "right" and "left".

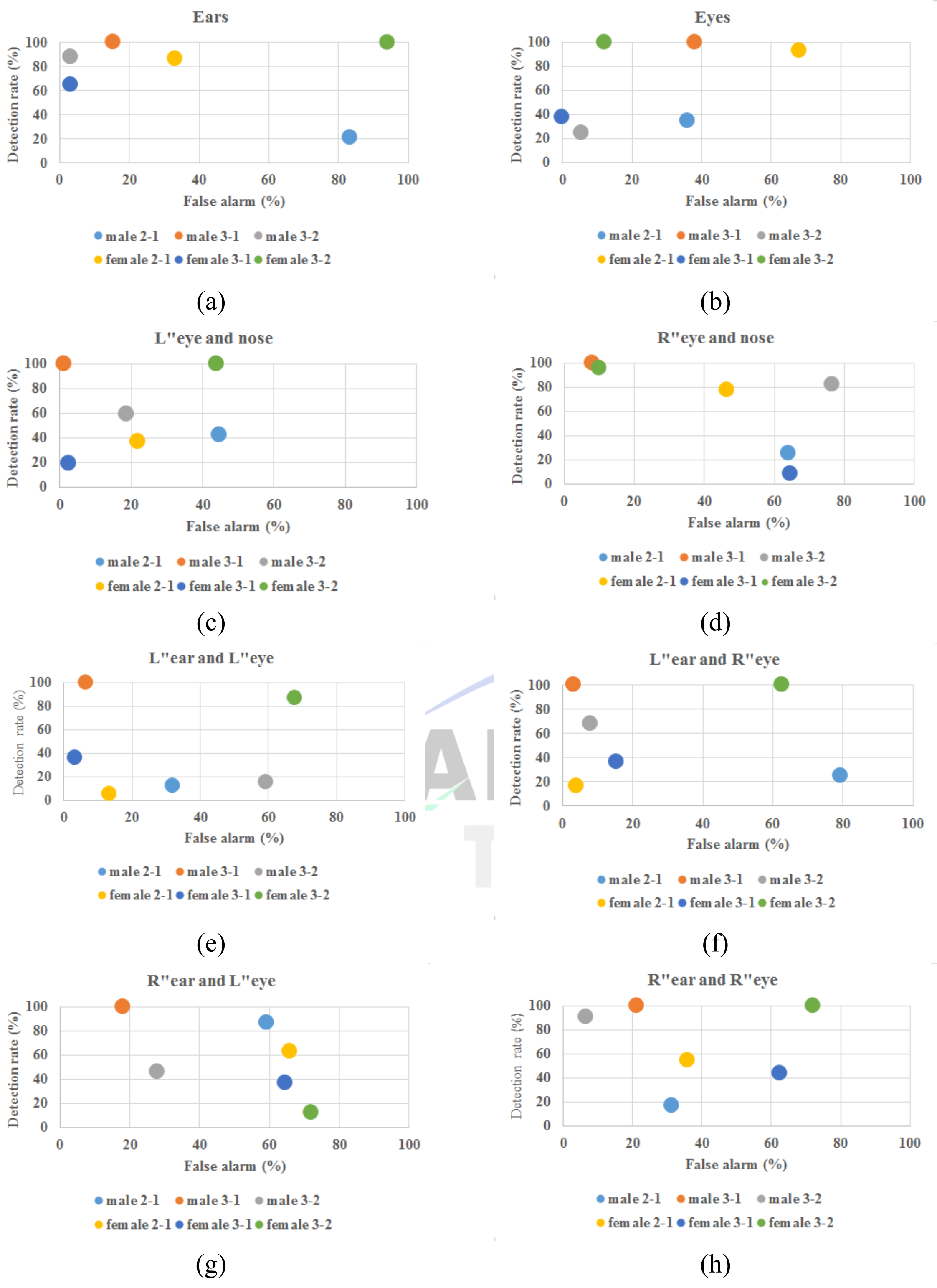


Figure 4-5. The schematic diagram of *Figure 4-4*. (a)-(h).

Table 4-5. Performance of SVM classifier by two different training strategies.

Strategies	Performance	Amount of using features							
		1	2	3	4	5	6	7	8
One by One	Detection Power	59.2	62.6	72.5	71.5	60.8	72.3	79.1	84
	False Alarm	4.4	5.8	6.5	6.5	5.3	10.8	18.4	28.5
All in one	Detection Power	51.3	53.3	54.8	55.1	55.1	55.1	55.1	55.1
	False Alarm	0	1.1	1.1	1.1	1.1	1.1	1.1	1.1

Table 4-6. Performance of pain classified by different methodologies.

Methodologies		TP	TN	FP	FN	Detection Power	False Alarm
Human	Mouse expert A	20	19	11	10	66	36
	Mouse expert B	18	22	8	12	60	26
	Mouse expert C	5	27	3	25	16	10
	Mouse expert D	22	19	11	8	73	36
Machine learning	SVM classifier	27	30	0	3	90	0

4.4 Performance of Pain Quantification

Performance of pain quantification was validated on the CFA dataset. The mice in the CFA dataset were injected CFA to induce pain. After injecting CFA for a period, diclofenac was given as analgesic to reduce the pain caused by CFA. After onset of analgesic, three more videos were recorded after injection for 1, 2, and 3 hours respectively as shown in Table 4-2. Figure 4-6 was the schematic diagram of pain quantification by implementing SVM and regression. Then, the pain scales estimated by the regression model would be compared with the period of time after injection for validation in Figure 4-7. The pain scale increased from 4% to 88% after CFA injection and started to decline after injection of 10 milligrams diclofenac. Later, the estimated value started to rise up slowly when the effect of diclofenac started to reduce by time. As shown in the previous section, the bias among mice was also shown in Figure 4-7 and Figure 4-8 for estimation of pain scales. Each mouse was identified by a different color. If the difference between 1st and 2nd videos was considered as pain sensitivity, and the 3rd, 4th, and 5th videos demonstrated the trend of pain scale after injection of analgesic. It was obvious to observe from Figure 4-8 that the pain sensitivity was slightly different for each experimental mouse, and the pain sensitivity could be considered as an important indicator for the following trend of pain scales. For example, the mice labeled as “LW” and “dMW” had similar pain sensitivity, which showed that their pain reactions were similar, but the trend of pain scales was different because of the dose of diclofenac. Overall, the results indicated that the proposed method is effective for pain quantification.

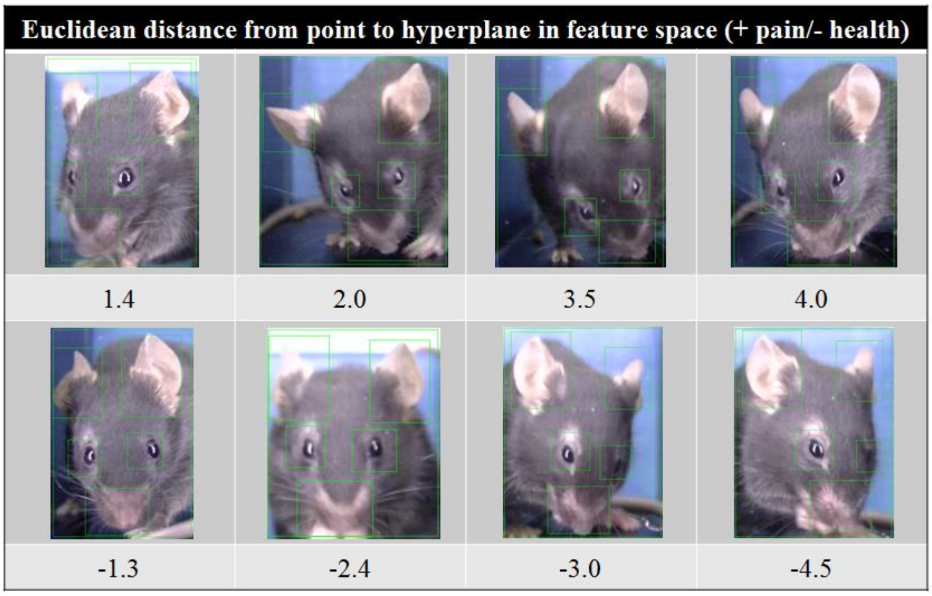


Figure 4-6. A quantitative pain scale to evaluate level of pain caused by CFA.

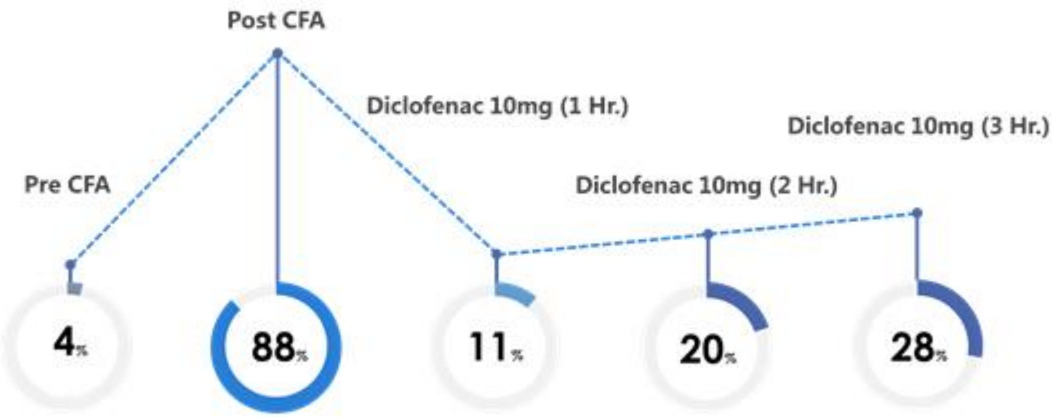


Figure 4-7. Estimation of pain scale for the mouse "BK" in the CFA dataset.

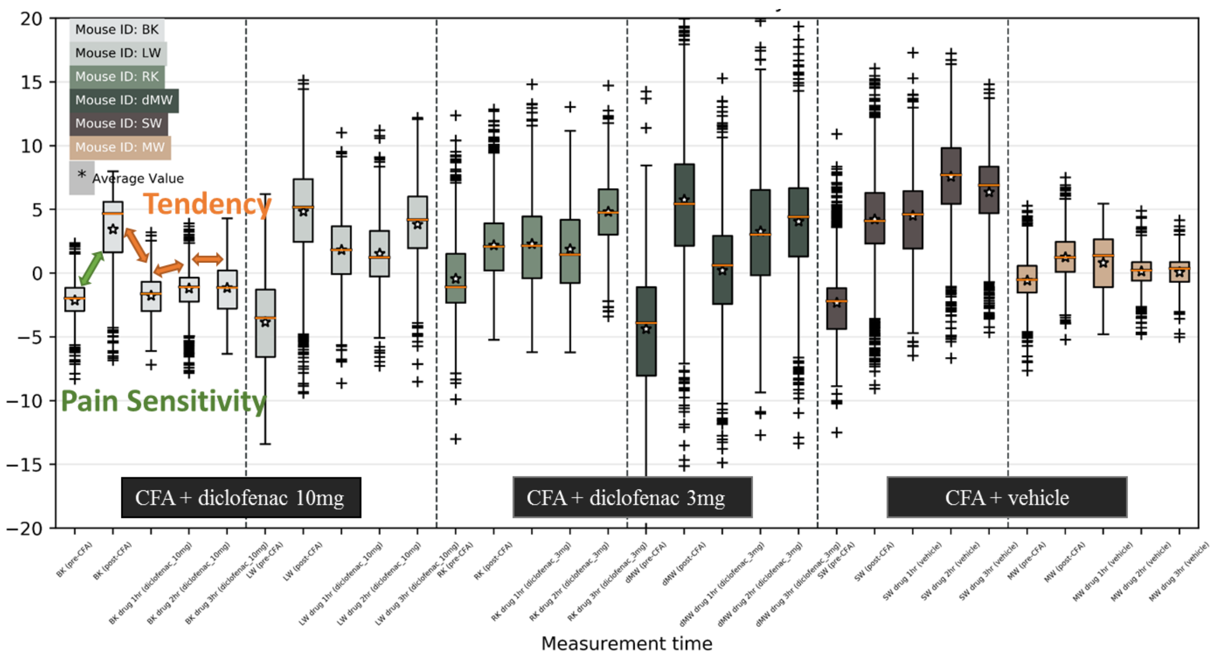


Figure 4-8. Performance of proposed method on six mice.

Chapter 5 CONCLUSION AND FUTURE WORK

The thesis proposed a solution to quantify the pain via facial expressions. Without relying on subjective interpretations of experts, the proposed methods can be utilized for measuring an ongoing pain for 24 hours. Several machine learning algorithms were investigated and applied to provide robust classification and quantification. In addition, the contribution of automatic annotator can reduce time consuming process to label facial patterns for the machine learning process. The advantages of the proposed methods could be summarized as follows. First, a methodology was proposed and evaluated to train a machine learning model without human intervention, and the accuracy was better than human interpretation of experts. Second, the proposed method combined a regression method to quantify pain via facial expression. Third, an automatic process was developed by using Mask R-CNN for extraction of facial expressions, which help to improve the time consuming process. Overall, the experimental results showed the ability of SVM for detection of pain, but also pointed out the limitation. First, performance of SVM model was affected by the learning facial patterns, which means that models need to customize for each experimental mouse. Second, the accuracy of Mask R-CNN could be improved by increasing the number of training data. Third, the period of the front-view was not predictable. Since the facial expressions have to be collected from the front-view images, the stability play a critical role in the proposed methods.

REFERENCE

[1] Sotocinal, Susana G., et al. "The Rat Grimace Scale: a partially automated method for quantifying pain in the laboratory rat via facial expressions." Molecular pain 7.1 (2011): 55.

[2] Ekman, Paul, and Wallace V. Friesen. Manual for the facial action coding system. Consulting Psychologists Press, 1978.

[3] Ekman, Paul, and Wallace V. Friesen. Facial Action Coding System: Investigatoris Guide. Consulting Psychologists Press, 1978.

[4] Williams, Amanda C. de C. "Facial expression of pain: an evolutionary account." Behavioral and brain sciences 25.4 (2002): 439-455.

[5] Murray, London, and C. Darwin. "The expression of the emotions in man and animals." (1872).

[6] Darwin, Charles, and Phillip Prodger. The expression of the emotions in man and animals. Oxford University Press, USA, 1998.

[7] Langford, Dale J., et al. "Coding of facial expressions of pain in the laboratory mouse." Nature methods 7.6 (2010): 447.

[8] Vapnik, Vladimir. "Pattern recognition using generalized portrait method." Automation and remote control 24 (1963): 774-780.

[9] Hsieh, Tsung-Hsun, et al. "Time-course gait analysis of hemiparkinsonian rats following 6-hydroxydopamine lesion." Behavioural brain research 222.1 (2011): 1-9.

[10] Wu, Ting-Fan, Chih-Jen Lin, and Ruby C. Weng. "Probability estimates for multi-class classification by pairwise coupling." Journal of Machine Learning Research 5.Aug (2004): 975-1005.

[11] Basak, Debasish, Srimanta Pal, and Dipak Chandra Patranabis. "Support vector regression." Neural Information Processing-Letters and Reviews 11.10 (2007): 203-224.

[12] Mogil, Jeffrey S., and Mona Lisa Chanda. "The case for the inclusion of female subjects in basic science studies of pain." Pain 117.1 (2005): 1-5.

[13] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001.

[14] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55.1 (1997): 119-139.

[15] Wilson, Phillip Ian, and John Fernandez. "Facial feature detection using Haar classifiers." Journal of Computing Sciences in Colleges 21.4 (2006): 127-133.

[16] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM transactions on intelligent systems and technology (TIST) 2.3 (2011): 27.

[17] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[18] C.Cortes and V.Vapnik, “Support-Vector Networks”, Machine Learning, 20, 1995.

[19] He, Kaiming, et al. "Mask r-cnn." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.

[20] Nasrabadi, Nasser M. "Pattern recognition and machine learning." Journal of electronic imaging 16.4 (2007): 049901.

[21] Uijlings, Jasper RR, et al. "Selective search for object recognition." International journal of computer vision 104.2 (2013): 154-171.

[22] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[23] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[24] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).