Research Article

Pain Expression Recognition Based on pLSA Model

Shaoping Zhu

Department of Information Management, Hunan University of Finance and Economics, Changsha 410205, China

Correspondence should be addressed to Shaoping Zhu; zenglegen@163.com

Received 9 January 2014; Accepted 19 February 2014; Published 27 March 2014

Academic Editors: Y.-B. Yuan and S. Zhao

Copyright © 2014 Shaoping Zhu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new approach to automatically recognize the pain expression from video sequences, which categorize pain as 4 levels: “no pain,” “slight pain,” “moderate pain,” and “severe pain.” First of all, facial velocity information, which is used to characterize pain, is determined using optical flow technique. Then visual words based on facial velocity are used to represent pain expression using bag of words. Final pLSA model is used for pain expression recognition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model. Experiments were performed on a pain expression dataset built by ourselves to test and evaluate the proposed method, the experiment results show that the average recognition accuracy is over 92%, which validates its effectiveness.

1. Introduction

In recent years, tremendous amounts of researches have been carried out in the field of automatic expressions (such as pain, anger, and sadness) recognition from video sequence. Pain is a subjective and personal experience, and pain recognition is still difficult. There are numerous potential applications for pain recognition. Doctors can recognize pain when patients are experiencing genuine pain so that their pains are taken seriously, like young children who could not self-report pain measures, or many patients in postoperative care or transient states of consciousness, and with severe disorders requiring assisted breathing, among other conditions [1, 2]. Real-time automatic system can be trained which could potentially provide significant advantage in patient care and cost reduction.

Measuring or monitoring pain is normally conducted via self-report as it is convenient and requires no special skill or staffing. However, self-report measures cannot be used when patients cannot communicate verbally. Many researchers have pursued the goal of obtaining a continuous objective measure of pain through analyses of tissue pathology, neurological “signatures,” imaging procedures, testing of muscle strength, and so on [3]. These approaches have been fraught with difficulty because they are often inconsistent with other

evidence of pain [3], in addition to being highly invasive and constraining to the patient.

The experience of pain is often represented by changes in facial expression. So, facial expression is considered to be the most reliable source of information when judging the pain intensity experienced by another. In the past several years, significant efforts have been made to identify reliable and valid facial indicators of pain [4–14]. In [4, 5], an approach was developed to automatically recognize acute pain; active appearance models (AAM) were used to decouple shape and appearance parameters from face images; based on AAM, three pain representations were derived. And then SVM were used to classify pain. In [6–10], Prkachin and Solomon validated a facial action coding system (FACS) based measure of pain that can be applied on a frame-by-frame basis. But these methods require manual labeling of facial action units or other observational measurements by highly trained observers [15, 16], which is both timely and costly. Most must be performed offline, which makes them ill-suited for real-time applications in clinical settings. In [11], a robust approach for pain expression recognition was presented using video sequences. An automatic face detector is employed which uses skin color modeling to detect human face in the video sequence. The pain affected portions of the face are obtained by using a mask image.

The obtained face images are then projected onto a feature space, defined by Eigenfaces, to produce the biometric template. Pain recognition is performed by projecting a new image onto the feature spaces spanned by the Eigenfaces and then classifying the painful face by comparing its position in the feature spaces with the positions of known individuals. Zhang and Xia [12] used supervised locality preserving projections (SLPP) to extract feature of pain expression, and multiple kernels support vector machines (MK SVM) are employed for recognizing pain expression. Methods described above used static features to character pain expression, but these static features cannot fully represent pain.

In this paper, we propose a method for automatically inferring pain form video sequences. This approach includes two steps: extracting feature of pain expression and classifying pain expression. In the extracting feature, features of pain expression are extracted by motion descriptor based on optical flow. Then we convert facial velocity information to visual words using “bag-of-words” models, and pain expression is represented by a number of visual words; final pLSA model is used for pain expression recognition. In addition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model.

The paper is structured as follows. After reviewing related work in this section, we describe the pain feature extraction based on optical flow technique and “bag-of-words” models in Section 2. Section 3 gives details of pLSA model for recognizing pain expression. Section 4 shows experiment result, also comparing our approach with three state-of-the-art methods, and the conclusions are given in the final section.

2. Pain Expression Representation

2.1. Facial Velocity Feature. According to the physiology, the experience of pain is often represented by changes in facial expression and the expression is a dynamic event; it is must be represented by the motion information of the face. So, we use facial velocity features to characterize pain. The facial velocity features (optical flow vector) are estimated by optical flow model, and each pain expression was coded on a 4-level intensity dimension (A-D): “no pain,” “slight pain,” “moderate pain,” and “severe pain.”

Given a stabilized video sequence in which the face of a person appears in the center of the field of view, we compute the facial velocity (optical flow vector) $\mathbf{u} = (u_x, u_y)$ at each frame using optical flow equation, which is expressed as

$$I_x u_x + I_y u_y + I_t = 0, \quad (1)$$

where

$$\begin{aligned} I_x &= \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t}, \\ u_x &= \frac{dx}{dt}, \quad u_y = \frac{dy}{dt}, \end{aligned} \quad (2)$$

where (x, y, t) is the image in pixel (x, y) at time t , where $I(x, y, t)$ is the intensity at pixel (x, y) and time t , u_x, u_y are the horizontal and vertical velocities in pixel (x, y) .

We can obtain $\mathbf{u} = (u_x, u_y)$ by minimizing the objective function:

$$C = \int_D [\lambda^2 \|\nabla \mathbf{u}\|^2 + (\nabla I \cdot \mathbf{u} + I_t)^2] dx dy. \quad (3)$$

There are many methods to solve the optical flow equation. We use the iterative algorithm [17] to compute the optical flow velocity:

$$\begin{aligned} u_x^{k+1} &= \bar{u}_x^k - \frac{I_x [I_x \bar{u}_x^k + I_y \bar{u}_y^k + I_t]}{\lambda + I_x^2 + I_y^2}, \\ u_y^{k+1} &= \bar{u}_y^k - \frac{I_y [I_x \bar{u}_x^k + I_y \bar{u}_y^k + I_t]}{\lambda + I_x^2 + I_y^2}, \end{aligned} \quad (4)$$

where k is the number of iterations, initial value of velocity $u_x^0 = u_y^0 = 0$, and $\bar{u}_x^k, \bar{u}_y^k$ is the average velocity of the neighborhood of point (x, y) .

The optical flow vector field $\mathbf{u}$ is then split into two scalar fields u_x and u_y , corresponding to the x and y components of $\mathbf{u}$ [18]. u_x and u_y are further half-wave rectified into four nonnegative channels u_x^+, u_x^-, u_y^+ , and u_y^- , so that $u_x = u_x^+ - u_x^-$ and $u_y = u_y^+ - u_y^-$. These four nonnegative channels are then blurred with a Gaussian kernel and normalized to obtain the final four channels $ub_{x^+}, ub_{x^-}, ub_{y^+}$, and ub_{y^-} .

Facial pain expression is represented by velocity features that are composed of the channels $ub_{x^+}, ub_{x^-}, ub_{y^+}$, and ub_{y^-} of all pixels in facial image. Because pain expression can be regarded as facial motion, the velocity features can describe pain effectively, in addition to the velocity features having been shown to perform reliably with noisy image sequences [18], and have been applied in various tasks, such as action classification and motion synthesis. But the dimension of these velocity features is too high ($4 \times N \times N$, where $N \times N$ is image size) to be used directly for recognition and, so, we convert these velocity features into visual words using “bag of words” [19, 20].

2.2. Visual Words for Characterizing Pain. The “bag-of-words” model was originally proposed for analyzing text documents, where a document is represented as a histogram over word counts.

In this paper, each facial image is divided into blocks whose size is $L \times L$, and each image block is represented by optical flow vector of all pixels in the block. On this basis, pain is represented by visual words using the method of BoW (bag of words).

To construct the codebook, we randomly select a subset from all image blocks; then, we use k -means clustering algorithms to obtain V clusters. Codewords are then defined as the centers of the obtained clusters, namely, visual words. In the end, each face image is converted to the “bag-of-words” representation by appearance times of each codeword in the image that is used to represent the image, namely, BoW histogram.

The step for characterizing pain is as follows.

Step 1. Optical flow channels ub_{x^+} , ub_{x^-} , ub_{y^+} , and ub_{y^-} are computed.

Step 2. Each facial image is divided into $n \times n$ blocks, which is represented by optical flow vector of all pixels in the block.

Step 3. Vision words are obtained using k -means clustering algorithms.

Step 4. Pain expression is represented by BoW histogram d :

$$d = \{n(I, w_1), \dots, n(I, w_j), \dots, n(I, w_M)\}, \quad (5)$$

where $n(I, w_j)$ is the number of visual word w_j included in image and M is the number of vision words in word sets.

Figure 1 shows an example of our “bag-of-words” representation.

3. pLSA-Based Pain Expression Recognition

We use the pLSA models [21] to learn and recognize human pain. Our approach is directly inspired by a body of work on using generative topic models for visual recognition based on the “bag-of-words” paradigm. The pLSA models have been applied to various computer vision applications, such as scene recognition, object recognition, action recognition, and human detection [22–26].

3.1. Probabilistic Latent Semantic Analysis (pLSA). pLSA is a statistical generative model that associates documents and words via the latent topic variables, which represents each document as a mixture of topics. We briefly outline the principle of the pLSA in this subsection. The model of pLSA is shown in Figure 2.

Suppose document, word, and topic are represented by d_i , w_j , and z_k , respectively. The joint probability of document d_i , topic z_k , and word w_j can be expressed as

$$p(d_i, z_k, w_j) = p(w_j | z_k) p(z_k | d_i) p(d_i), \quad (6)$$

where $p(w_j | z_k)$ is the probability of word w_j occurring in pain category z_k , $p(z_k | d_i)$ is the probability of topic z_k occurring in image d_i , and $p(d_i)$ can be considered as the prior probability of d_i . The conditional probability of $p(w_j | d_i)$ can be obtained by marginalizing over all the topic variables z_k :

$$p(w_j | d_i) = \sum_k p(z_k | d_i) p(w_j | z_k). \quad (7)$$

Denote $n(d_i, w_j)$ as the occurrence of word w_j in image d_i ; the prior probability $p(d_i)$ can be modeled as

$$p(d_i) \propto \sum_j n(d_i, w_j). \quad (8)$$

A maximum likelihood estimation of $p(w_j | z_k)$ and $p(z_k | d_i)$ is obtained by maximizing the function using

the expectation maximization (EM) algorithm. The objective likelihood function of the EM algorithm is

$$l = \prod_i \prod_j p(w_j | d_i)^{n(w_j, d_i)} \quad (9)$$

or

$$l = \sum_i \sum_j n(d_i, w_j) \log p(w_j | d_i). \quad (10)$$

The EM algorithm consists of two steps: an expectation (E) step computes the posterior probability of the latent variables, and a maximization (M) step maximizes the completed data likelihood computed based on the posterior probabilities obtained from E-step. Both steps of the EM algorithm for pLSA parameter estimate are listed below.

E-Step. Given $p(w_j | z_k)$ and $p(z_k | d_i)$, estimate $p(z_k | d_i, w_j)$:

$$p(z_k | d_i, w_j) = \frac{p(w_j | z_k) p(z_k | d_i)}{\sum_l p(w_j | z_l) p(z_l | d_i)}. \quad (11)$$

M-Step. Given the estimated $p(z_k | d_i, w_j)$ in E-Step and $n(d_i, w_j)$, estimate $p(w_j | z_k)$ and $p(z_k | d_i)$:

$$p(w_j | z_k) = \frac{\sum_i n(d_i, w_j) p(z_k | d_i, w_j)}{\sum_i \sum_h n(d_i, w_h) p(z_k | d_i, w_h)}, \quad (12)$$

$$p(z_k | d_i) = \frac{\sum_j n(d_i, w_j) p(z_k | d_i, w_j)}{n(d_i)}, \quad (13)$$

where $n(d_i) = \sum_j n(d_i, w_j)$ is the length of document d_i .

Given a new document, the conditional probability distribution over aspect $p(z | d_{\text{new}})$ can be inferred by maximizing the likelihood of d_{new} using a fixed word-aspect distribution $p(w_j | z_k)$ learned from the observed data [21]. The iteration of inferring $p(z | d_{\text{new}})$ is the same as the learning process except that the word-topic distribution $p(w_j | z_k)$ in (12) is a fixed value, that is, learned from training data.

3.2. pLSA-Based Pain Expression Recognition. In this paper, we treat each block in an image as a single word w_j , an image as a document d_i , and a pain category as a topic variable z_k . For the task of pain classification, our goal is to classify a new face image to a specific pain class. During the inference stage, given a testing face image and the document specific coefficients $p(z_k | d_{\text{test}})$, we can treat each aspect in the pLSA model as one class of pains. So, the pain categorization is determined by the aspect corresponding to the highest $p(z_k | d_{\text{test}})$. The pain category k of d_{test} is determined as

$$k = \arg \max_k p(z_k | d_{\text{test}}). \quad (14)$$

For pain recognition with large amount of training data, this would result in long training time. In this paper, we adopt a supervised algorithm to train pLSA, which is similar to [27].

FIGURE 1: The processing pipeline of the “bag-of-words” representation: (a) give an image, (b) divide into $L \times L$ blocks, (c) represent each block by a “visual word,” and (d) ignore the ordering of words and represent the facial image as a histogram over “visual words.”

FIGURE 2: Graph model of pLSA. Nodes represent random variables. Shaded nodes are observed variables and unshaded ones are unseen variables. The plates stand for repetitions.

Each image has class label information in the training images, which is important for the classification task. Here, we make use of this class label information in the training images for the learning of the pLSA model, since each image directly corresponds to a certain pain class on train sets; the image

FIGURE 3: Graph model of SpLSA. Nodes represent random variables. Shaded nodes are observed variables and unshaded ones are unseen variables. The plates stand for repetitions.

for training data becomes observable. This model is called supervised pLSA (SpLSA). The graphical model of SpLSA is shown in Figure 3.

The parameter $p(w_j|z_k)$ in the training step defines the probability of a word w_j drawing from a topic z_k . Letting each

topic in pLSA correspond to a pain category, the distribution $p(w_j | z_k)$ in the training can be simply estimated as

$$p(w_j | z_k) = \frac{n_{j,k}}{n_k}, \quad (15)$$

where n_k is the number of the images corresponding to the k th pain class and $n_{j,k}$ is the number of the j th word (block) in the images corresponding to the k th pain class. This means that the $p(w_j | z_k)$ calculated by this way can be used to initialize the $p(w | z)$ in the EM algorithm for model learning, which makes the EM algorithm converge more quickly. The supervised training of pLSA is summarized in Algorithm 1. Once the distribution $p(w | z)$ is computed by the EM algorithm, for a testing face image d_{test} , the posterior distribution $p(z_k | d_{\text{test}})$ can be calculated the same as in original pLSA. The training of pLSA for classification is summarized in Algorithm 2.

Algorithm 1. Supervised training of the pLSA.

Step 1. For all k and j , calculate

$$p(w_j | z_k) = \frac{n_{j,k}}{n_k}, \quad (16)$$

as the initialization of the $p(w | z)$ and random initialization of the $p(z | d)$.

Step 2. E-Step: for all (d_i, w_j) pairs, calculate

$$p(z_k | d_i, w_j) = \frac{p(w_j | z_k) p(z_k | d_i)}{\sum_l p(w_j | z_l) p(z_l | d_i)}. \quad (17)$$

Step 3. M-Step: substitute $p(z_k | d_i, w_j)$ as calculated in Step 2; for all k and j , calculate

$$p(w_j | z_k) = \frac{\sum_i n(d_i, w_j) p(z_k | d_i, w_j)}{\sum_m \sum_i n(d_i, w_m) p(z_k | d_i, w_m)}. \quad (18)$$

Step 4. M-Step: substitute $p(z_k | d_i, w_j)$ as calculated in Step 2; for all i and k , calculate

$$p(z_k | d_i) = \frac{\sum_j n(d_i, w_j) p(z_k | d_i, w_j)}{n(d_i)}. \quad (19)$$

Step 5. Repeat Steps 2–4 until the convergence condition is met.

The supervised training algorithm not only makes the training more efficient, but also improves the overall recognition accuracy significantly.

Algorithm 2. Training of the pLSA for classification

Step 1. For all k and j , calculate

$$p(w_j | z_k) = \frac{n_{j,k}}{n_k}. \quad (20)$$

Step 2. E-Step: for all (d_{test}, w_j) pairs, calculate

$$p(z_k | d_{\text{test}}, w_j) = \frac{p(w_j | z_k) p(z_k | d_{\text{test}})}{\sum_l p(w_j | z_l) p(z_l | d_{\text{test}})}. \quad (21)$$

Step 3. Partial M-Step: fix $p(w_j | z_k)$ as calculated in Step 1; for all k , calculate

$$p(z_k | d_{\text{test}}) = \frac{\sum_j n(d_{\text{test}}, w_j) p(z_k | d_{\text{test}}, w_j)}{n(d_{\text{test}})} \quad (22)$$

Step 4. Repeat Steps 2 and 3 until the convergence condition is met.

Step 5. Calculate pain class

$$k = \arg \max_k p(z_k | d_{\text{test}}). \quad (23)$$

4. Experimental Results and Analysis

The effectiveness of the proposed algorithm was verified by using C++ and MATLAB hybrid implementation on a PC with Pentium 3.2 GHz processor and 4 G RAM.

We have built a database of painful and normal face images. In this database, there are four groups of images (“no pain,” “slight pain,” “moderate pain,” and “severe pain”), and each group includes 20 males and 20 females. The face images were taken under various laboratory-controlled lighting conditions, and each face image was normalized to a size of 64×64 . Some sample images are shown in Figure 4.

In experiments, 30 face images per class are randomly chosen for training, while the remaining images are used for testing. We preprocessed these images by aligning and scaling them so that the distances between the eyes were the same for all images and also ensuring that the eyes occurred in the same coordinates of the image. We run the system 5 times and obtain 5 different training and testing sample sets. The recognition rates were found by averaging the recognition rate of each run.

Each facial image is divided into blocks whose size is $L \times L$. First, we studied the effect of the size of image block on the recognition accuracy. Figure 5 represents the recognition accuracy curve with different block sizes L . It can be concluded that the accuracy peaks when the block sizes L is 8. Therefore L is set as 8.

In order to determine the value of M , that is, the number of the visual word set, the relation between M and recognition accuracy was observed, which is displayed in Figure 6. It is revealed in Figure 4 that the recognition accuracy is risen up at the beginning with the increasing of M recognition and if M is larger than or equal to 60, the recognition accuracy is stabilized to 0.922. As a result, M is set as 60.

To examine the accuracy of our proposed pain recognition approach, we compare our method to three state-of-the-art approaches for pain recognition using the same data. The first method is “AAM + SVM” [4], which used active appearance models (AAM) to extract face features and SVM to classify pain. The second method is “Eigenimage”

FIGURE 4: Examples of recognizing pain expression from facial videos. (a) No pain, (b) slight pain, (c) moderate pain, and (d) severe pain.

FIGURE 5: Recognition accuracy curve with different block sizes.

[11], which used Eigenface for pain recognition. The third method is “SLPP + MKSVM” [12], which used SLPP to extract feature of pain expression and multiple kernels support vector machines (MKSVM) for recognizing. 200 different expression images are used for this experiment. Some images contain the same person but in different moods. The recognition results are presented in the confusion matrices shown in Table 1. Each cell in the confusion matrix is the average results; our results are at the upper left; the results of “AAM

TABLE 1: Confusion matrix for pain recognition.

(a)				
	A	B	C	D
A	0.95	0.04	0.01	0.00
B	0.04	0.91	0.05	0.00
C	0.01	0.03	0.91	0.05
D	0.00	0.02	0.05	0.93
(b)				
	A	B	C	D
A	0.84	0.10	0.03	0.02
B	0.11	0.78	0.08	0.03
C	0.03	0.09	0.79	0.08
D	0.01	0.05	0.12	0.83
(c)				
	A	B	C	D
A	0.85	0.11	0.03	0.01
B	0.10	0.80	0.08	0.02
C	0.02	0.08	0.79	0.08
D	0.01	0.04	0.12	0.84
(d)				
	A	B	C	D
A	0.88	0.09	0.02	0.01
B	0.09	0.82	0.07	0.02
C	0.02	0.08	0.83	0.07
D	0.00	0.02	0.08	0.90

TABLE 2: Comparison of different reported results on pain dataset.

Method	Accuracy (%)
Our method	92.20
“AAM + SVM” [4]	81.20
“Eigenimage” [11]	82.50
“SLPP + MKSVM” [13]	86.50

+ SVM,” “Eigenimage,” and “SLPP + MKSVM” are presented in the upper right, the lower left, and the lower right, respectively. where A, B, C, and D indicate no pain, slight pain, moderate pain, and severe pain, respectively. As Table 1 shows, our method improves the recognition accuracies in all categories. It achieves 92.2% average recognition rate, whereas “AAM + SVM” obtain 81.2%, “Eigenimage” gets 82.5%, and “SLPP + MKSVM” attains 86.5%, as shown in Table 2. The reason is that we improve the recognition accuracy in the two stages of pain feature extraction and pain expression recognition. In the stage of pain feature extraction, we use motion features that are reliable with noisy image sequences and describe pain effectively, while other methods used static features, which cannot effectively describe the expression of pain. In the stage of expression recognition, we use bag-of-words framework and pLAS model to classify expression images. In addition, we make use of this class label information in

FIGURE 6: Relation curve between M and accuracy.

the training images for the learning of the pLSA model, which can improve the overall recognition accuracy significantly.

5. Conclusion

Pain recognition can provide significant advantage in patient care and cost reduction. In this paper, we present a novel method to recognize the pain expression and give the pain level at the same time. The main contribution can be concluded as follows.

- (1) Visual words are used for pain expression. Optical flow model is used for extracting facial velocity features; then we convert facial velocity features into visual words using “bag-of-words” models.
- (2) We use pLSA topic models for pain expression recognition. In our models the “latent topics” directly correspond to different pain expression categories. In addition, in order to improve the recognition accuracy, the class label information was used for the learning of the pLSA model.
- (3) Experiments were performed on a pain expression dataset built by ourselves and evaluate the proposed method. Experimental results reveal that the proposed method performs better than previous ones.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Research Foundation for Science & Technology Office of Hunan Province under Grant no. 2014FJ3057, by Hunan Provincial Education Science and “Twelve Five” planning issues (no. XJK012CGD022), by the Teaching Reform Foundation of Hunan Province Ordinary

College under Grant no. 2012401544, and by the Foundation for Key Constructive Discipline of Hunan Province.

References

- [1] D. L. Wong and C. M. Baker, “Pain in children: comparison of assessment scales,” *Pediatric Nursing*, vol. 14, no. 1, pp. 9–17, 1988.
- [2] K. D. Craig, K. M. Prkachin, and R. V. E. Grunau, “The facial expression of pain,” in *Handbook of Pain Assessment*, D. C. Turk and R. Melzack, Eds., pp. 153–169, Guilford Press, New York, NY, USA, 2nd edition, 2001.
- [3] D. Turk and R. Melzack, “The measurement of pain and the assessment of people experiencing pain,” in *Handbook of Pain Assessment*, D. C. Turk and R. Melzack, Eds., pp. 1–11, Guilford Press, New York, NY, USA, 2nd edition, 2001.
- [4] A. B. Ashraf, S. Lucey, T. Chen et al., “The painful face—pain expression recognition using active appearance models,” in *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI '07)*, pp. 9–14, ACM, Nagoya, Japan, November 2007.
- [5] A. B. Ashraf, S. Lucey, J. F. Cohn et al., “The painful face—pain expression recognition using active appearance models,” *Image and Vision Computing*, vol. 27, no. 12, pp. 1788–1796, 2009.
- [6] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, “Painful monitoring: automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database,” *Image and Vision Computing*, vol. 30, no. 3, pp. 197–205, 2012.
- [7] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin, “Automatically detecting pain using facial actions,” in *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII '09)*, pp. 1–8, Amsterdam, The Netherlands, September 2009.
- [8] P. Lucey, J. F. Cohn, I. Matthews et al., “Automatically detecting pain in video through facial action units,” *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 41, no. 3, pp. 664–674, 2011.
- [9] K. M. Prkachin, S. Berzinzs, and R. S. Mercer, “Encoding and decoding of pain expressions: a judgment study,” *Pain*, vol. 58, no. 2, pp. 253–259, 1994.
- [10] K. M. Prkachin and P. E. Solomon, “The structure, reliability and validity of pain expression: evidence from patients with shoulder pain,” *Pain*, vol. 139, no. 2, pp. 267–274, 2008.
- [11] M. Monwar, S. Rezaei, and K. Prkachin, “Eigenimage based pain expression recognition,” *IAENG International Journal of Applied Mathematics*, vol. 36, no. 2, pp. 1–6, 2007.
- [12] W. Zhang and L. M. Xia, “Pain expression recognition based on SLPP and MKSVM,” *International Journal of Engineering and Manufacturing*, vol. 3, pp. 69–74, 2011.
- [13] K. M. Prkachin, “The consistency of facial expressions of pain: a comparison across modalities,” *Pain*, vol. 51, no. 3, pp. 297–306, 1992.
- [14] K. M. Prkachin and S. R. Mercer, “Pain expression in patients with shoulder pathology: validity, properties and relationship to sickness impact,” *Pain*, vol. 39, no. 3, pp. 257–265, 1989.
- [15] J. F. Cohn, Z. Ambadar, and P. Ekman, “Observer-based measurement of facial expression with the facial action coding system,” in *The Handbook of Emotion Elicitation and Assessment*, pp. 203–221, Oxford University Press, New York, NY, USA, 2007.
- [16] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial Action Coding System: Research Nexus*, Network Research Information, Salt Lake City, Utah, USA, 2002.

- [17] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–204, 1981.
- [18] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 726–733, IEEE, Nice, France, October 2003.
- [19] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, pp. 43–52, 2010.
- [20] T. Li, T. Mei, I. S. Kweon, and X. Hua, "Contextual bag-of-words for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 381–392, 2011.
- [21] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–204, 1981.
- [22] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, ACM, New York, NY, USA, 1999.
- [23] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Computer Vision—ECCV: Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Part IV*, vol. 4, pp. 517–530, Springer, Berlin, Germany, 2006.
- [24] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1816–1823, Beijing, China, October 2005.
- [25] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1605–1614, June 2006.
- [26] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 370–377, Beijing, China, October 2005.
- [27] J. Wang, P. Liu, M. F. H. She, A. Kouzani, and S. Nahavandi, "Supervised learning probabilistic latent semantic analysis for human motion analysis," *Neurocomputing*, vol. 100, pp. 134–143, 2013.