PAPER

Pain Intensity Estimation Using Deep Spatiotemporal and Handcrafted Features

Jinwei WANG[†], ***Member and*** **Huazhi SUN**[†a)], ***Nonmember***

SUMMARY Automatically recognizing pain and estimating pain intensity is an emerging research area that has promising applications in the medical and healthcare field, and this task possesses a crucial role in the diagnosis and treatment of patients who have limited ability to communicate verbally and remains a challenge in pattern recognition. Recently, deep learning has achieved impressive results in many domains. However, deep architectures require a significant amount of labeled data for training, and they may fail to outperform conventional handcrafted features due to insufficient data, which is also the problem faced by pain detection. Furthermore, the latest studies show that handcrafted features may provide complementary information to deep-learned features; hence, combining these features may result in improved performance. Motived by the above considerations, in this paper, we propose an innovative method based on the combination of deep spatiotemporal and handcrafted features for pain intensity estimation. We use C3D, a deep 3-dimensional convolutional network that takes a continuous sequence of video frames as input, to extract spatiotemporal facial features. C3D models the appearance and motion of videos simultaneously. For handcrafted features, we propose extracting the geometric information by computing the distance between normalized facial landmarks per frame and the ones of the mean face shape, and we extract the appearance information using the histogram of oriented gradients (HOG) features around normalized facial landmarks per frame. Two levels of SVRs are trained using spatiotemporal, geometric and appearance features to obtain estimation results. We tested our proposed method on the UNBC-McMaster shoulder pain expression archive database and obtained experimental results that outperform the current state-of-the-art.

key words: *pain intensity estimation, 3D convolutional network, histogram of oriented gradients, feature fusion*

1. Introduction

In the field of modern medicine and healthcare, monitoring pain is essential for correct diagnosis and symptomatic treatment. At present, the clinical assessment of pain is generally performed through patient's self-report. Some pain assessment scales have been developed to capture patient's self-report of pain intensity, e.g., the numerical rating scale (NRS) [1] and the visual analogue scale (VAS) [2]. However, this method is limited because some pain sufferers cannot communicate verbally or express their pain accurately, such as infants, the elderly, patients in intensive care units (ICUs) and patients with impaired cognitive function. Alternatively, pain assessment through a manual examination by a medical physician or nurse appears to be a better approach. However, there are also some problems with such an approach, as follows. 1) Manual assessment is subjective, either regarding the assessors or the assessed. 2) Manual assessment cannot be performed continuously over time. Therefore, it is difficult to observe whether the pain is increasing, decreasing or spiking. 3) Due to the high workload that medical staff in hospitals currently experience, manual assessment of pain consistently and reliably has been difficult to achieve. Given the above problems, a reliable automatic pain recognition and pain intensity estimation method that implements a more objective, efficient and economical solution for pain assessment is essential.

Many studies have explored automatic pain assessment from pain indicators, such as physiological signals and crying sounds. Some researchers recognize pain using brain activity data, i.e., functional magnetic resonance imaging (fMRI) [3] and electroencephalography (EEG) [4]. Galvanic skin response (GSR), electromyography (EMG) and electrocardiogram (ECG) are also used for pain assessment [5]. However, these methods are generally highly invasive and constraining to the patient. Studies on crying are most common in infants, whose primary expression is crying. The authors of [6] conducted an acoustic analysis of babies' cries to distinguish between pain-induced and normal crying.

Another efficient and non-invasive solution is to use facial expressions. In the past, significant efforts have been devoted to identifying reliable and valid facial indicators of pain [7], [8]. These efforts revealed how to distinguish between pain and no pain through facial action units (AUs) defined by the facial action coding system (FACS) [9], which makes researchers aware of the possibility of automatic pain recognition through facial expressions using computer vision and pattern recognition techniques. However, as with most pattern recognition problems, an abundance of training and testing data that are representative of the target application are needed to construct robust models with reliable performance.

To facilitate automatic pain assessment, researchers from McMaster University and the University of Northern British Columbia (UNBC) released the UNBC-McMaster shoulder pain expression archive database [10]. Over the past few years, many studies have used this database to validate their methods for automatic pain assessment. Most of the initial studies used appearance and geometric features extracted using active appearance models (AAMs) [11] to detect AUs and the PSPI for the presence of pain [12]–[14]. With an in-depth study, many new handcrafted fea-

Manuscript received September 29, 2017.
Manuscript revised January 22, 2018.
Manuscript publicized March 12, 2018.

[†]The authors are with the College of Computer and Information Engineering, Tianjin Normal University, Tianjin, 300387, P.R. China.

a) E-mail: huazhisun@protonmail.com (Corresponding author)

DOI: 10.1587/transinf.2017EDP7318

tures are used for this purpose, such as DCT, LBP, Gabor, and PHOG. In general, two or three features are used together [15]–[17]. Recently, deep learning has successfully been applied to the computer vision domain and achieved impressive results [18], [19]. Accordingly, two deep models, the convolutional neural network (CNN) and the recurrent convolutional neural network (RCNN) [20], have been explored for automatic pain intensity estimation [21]. One problem with deep learning is that deep structures typically need large amounts of data for training to achieve good performance. However, facial expression data of pain are particularly difficult to obtain. For example, the UNBC McMaster shoulder pain database contains 48,398 frames, but 82.71% (40,029) of these frames are 'no-pain' frames, and only 17.29% (8,369) are 'pain' frames. Consequently, deep methods may fail to achieve desirable performance due to insufficient data.

The latest studies (e.g., [22]) show that handcrafted features may provide complementary information with deep nets. Motivated by this finding, in this work, we use both handcrafted and deep-learned features for continuous pain intensity estimation when confronted with limited 'pain' sample images. For handcrafted features, we use geometric and appearance features simultaneously. The geometric features are extracted through calculating the distance between normalized facial landmarks tracked by an AAM on each frame and the ones of the normalized mean face shape, and the appearance features are the histogram of oriented gradients (HOG) [23] around normalized facial landmarks on each frame.

For deep features, frame-level static facial expression features are not sufficient. Previous studies for expression recognition [24], [25] show that sequence-level dynamic spatiotemporal features of facial expressions significantly improve the recognition performance. Therefore, we use the deep 3-dimensional convolutional network (C3D) [26], which takes a continuous sequence of video frames as input, to extract spatiotemporal facial features. C3D has made outstanding progress in handling various video analysis tasks, particularly action recognition. C3D models appearance and motion information simultaneously and achieves good performance on different video benchmarks [26]. To the best of our knowledge, our work is the first to use C3D structures for continuous pain intensity estimation.

Our method is trained and evaluated on the UNBC-McMaster shoulder pain database. The experimental results show that for limited samples, the combination of deep-learned features with handcrafted features yields a significant improvement relative to only one of these features. Compared with previous works, our results reveal high performance and outperform the current state-of-the-art.

The remainder of this paper is structured as follows. In Sect. 2, we present a review of previous work in automatic pain recognition and assessment. Section 3 describes our proposed methodology based on the combination of handcrafted geometric, appearance and deep spatiotemporal features. In Sect. 4, we describe the experiments on the UNBC-McMaster shoulder pain database and compare the results to previous studies. Section 5 concludes the paper and presents directions for future work.

2. Related Work

Over the past decade, many studies have evaluated their proposed approaches on the UNBC-McMaster shoulder pain dataset. Early studies focused on distinguishing whether the subject per frame is in pain. Initially, Ashraf et al. in [12] released a baseline recognition accuracy using geometric and appearance features named the S-PTS and the C-APP extracted by active appearance models (AAMs). They later [13] proposed a frame-by-frame pain detection method by recognizing action units (AUs) and calculating the Prkachin and Solomon pain intensity (PSPI) [12] score, which is defined as

$$\begin{aligned} Pain &= AU4 + max(AU6, AU7) \\ &\quad + max(AU9, AU10) + AU43 \end{aligned} \tag{1}$$

Any frame with a PSPI score greater than or equal to 1 is considered to contain pain. An SVM followed by a linear logistical regression (LLR) outputs pain scores from fusing the best AUs together. Based on previous studies, paper [27] extracted 3D information from an AAM to track the expression and head movement caused by pain. All of these studies have shown that using both geometric and appearance features can significantly improve performance. Following this direction, Khan et al. [16] also simultaneously used pyramid histogram of oriented gradients (PHOG) and pyramid local binary patterns (PLBP) as geometric and appearance features for recognizing pain/no pain. Four classifiers were tested: SVM, 2 nearest neighbor (2NN), decision tree (DT) and random forest (RF), and the simplest 2NN provided excellent results. In [28], geometric features were computed using 22 facial landmarks, and a k-NN classifier was trained for classifying AUs to calculate the PSPI score, which was used to recognize pain. Pedersen [29] proposed a discriminative feature extractor based on an autoencoder that can learn discriminative pain-related appearance features because it is trained using a loss function that can adjust the trade-off between reconstruction error and classification error.

Recent studies have shifted the focus from distinguishing between pain and no pain to estimating the intensity of pain. Some studies categorize custom pain levels based on PSPI scores. For example, Hammal et al. [30] classify the PSPI score into four levels of pain intensity (none, trace, weak, and strong) and use a combination of AAM, log-normal filters, and SVMs to measure pain intensity. Rudovic et al. [31] discretized the PSPI score into six pain levels and proposed a heteroscedastic conditional ordinal random field (CORF) model to recognize these levels. This model can adapt to the variability of the pain expressions from different subjects. Later, Zhao et al. [17] proposed an algorithm based on the alternating direction method of multipliers (ADMM) to solve the optimization problem for the

Table 1 Summary of previous studies.

Study	Pain levels	Features	Classifier or Regressor	Mode	Validation
Ashraf et al.[12]	2	SPTS, SAPP, CAPP from AAM	SVM	Direct	Leave one subject out
Lucey et al.[13]	2	SPTS, CAPP from AAM	SVM+LLR	Direct, AUs	Leave one subject out
Lucey et al.[27]	2	SPTS, SAPP, CAPP from AAM	SVM+LLR	AUs	Leave one subject out
Khan et al.[16]	2	PHOG, PLBP	SVM, 2NN, DT, RF	Direct	10-fold cross validation
Zafar et al.[28]	2	Custom features from 22 FCPs	k-NN	AUs	Leave one subject out
Pedersen [29]	2	Features from Autoencoder	SVM	Direct	Leave one subject out
Hammal et al.[30]	4	CAPP, Log-Normal filters	SVM	AUs	Leave one subject out
Rudovic et al.[31]	6	LBP	KCORF	Direct	Leave one subject out
Zhao et al.[17]	6	LBP, Gabor, PCA	OSVR learning by ADMM	Direct	Leave one subject out
Irani et al.[32]	3	Spatiotemporal oriented energy	Compute from features	Direct	Leave one subject out
Kaltwang et al.[15]	16	PTS, DCT, LBP	RVR	AUs, Direct	Leave one subject out
Florea et al.[33]	16	HoT	SVR	Direct	Leave one subject out
Neshov et al.[34]	16	SIFT, PCA	Linear SVR	Direct	Leave one subject out
Hong et al.[35]	16	2Standmap	SVM	Direct	Leave one subject out
Zhou et al.[21]	16	Features from RCNN	Linear function of RCNN	Direct	Leave one subject out

This table is a summary of previous studies of automatic pain assessment on the UNBC-McMaster shoulder pain database. The 2nd column is the level number of pain recognition or estimation. Two represents recognition of no pain (PSPI=0) and pain (PSPI>0). Four represents no pain (PSPI=0), trace (PSPI=1), weak (PSPI=2), and strong (PSPI: 3-15). Six represents none (PSPI=0), mild (PSPI=1), discomforting (PSPI=2), distressing (PSPI=3), intense (PSPI: 4-5), and excruciating (PSPI: 6-15). Three represents no pain (PSPI=0), weak (PSPI: 1-2) and strong (PSPI: 3-15). The 5th column is the mode of pain assessment. There are two types of modes: direct represents assessing pain directly from face images of subjects, and AUs represents assessing pain through calculating the PSPI score from the intensity of AUs recognized from face images of subjects.

model of ordinal support vector regression (OSVR) learning, and they achieved competitive performance in the fully supervised method for estimating pain. Irani et al. [32] divided the PSPI score into three levels (no pain, weak and strong) and proposed a method that uses steerable and separable filters that can measure the energies released by the facial muscles to extract spatiotemporal features of pain.

At present, most of the pain intensity estimation studies identify 16 pain levels (0-15) of PSPI scores using classification or regression. Kaltwang et al. [15] proposed a continuous pain intensity estimation method through fusing geometric (facial landmarks) and appearance (DCT and LBP) features. A relevance vector regression (RVR) is trained for classification. The paper shows that direct pain estimation from the image can be more accurate than the calculation from the AUs. Florea et al. [33] introduced the histogram of topographic features (HoT) that is composed of Hessian and gradient-based histograms to identify pain intensity levels. Neshov et al. [34] first utilized the supervised descent method (SDM) to detect facial landmarks and then capture local scale-invariant feature transform (SIFT) features to describe facial muscle deformation. Finally, a linear SVR was trained for the estimation of pain level. Hong et al. [35] applied the second-order standardized moment average pooling (2Standmap) technique to pain intensity estimation and found that the result is better than all approaches that only rely on a single descriptor. Zhou et al. [21] used a regression RCNN that can extract contextual information from image sequences to conduct pain intensity estimation. A method based on VGG-face features [36] and SVR was also proposed as the baseline. With the input of AAM-warped facial images, the RCNN achieves competitive performance.

Table 1 presents a summary of previous studies of automatic pain assessment on the UNBC-McMaster shoulder pain dataset. This table consists of three parts: recognition of pain and on pain, estimation of custom pain level and estimation of PSPI score. Each part indicates the feature descriptors, the classifier or regressor, the number of pain levels, the assessment mode and the validation method.

3. Methodology

In this section, we will describe our proposed method for pain intensity estimation in detail, which is based on the combination of handcrafted geometric, appearance and deep spatiotemporal features. Figure 1 summarizes the pipeline of our method, which consists of the extraction process of three features. In the following subsections, first, we describe the pre-processing. Next, we explain the properties of the C3D network, which can be directly used as the model to extract spatiotemporal features. Then, we present the details of extracting geometric and appearance features. Finally, we introduce the regression model of pain intensity based on SVR and the fusion strategy of three features. In this work, we perform pain intensity estimation using the 16-level PSPI scores.

3.1 Preprocessing

When calculating the facial features, the position and size of the face in different videos or even the same video will change due to the movement of the human body and the change in the focal length of the camera. To enhance the robustness of the pain intensity estimation algorithm, we preprocess the original image to achieve invariance to different face poses. In the first step, we compute the mean facial shape using the 66 facial landmarks extracted by the AAM and provided by the database. The mean facial shape is scaled to a 64x64 pixel image. In the second step, we perform Delaunay triangulation on the mean facial shape and the facial shape in each frame. Finally, based on the Delaunay triangular mesh, the facial pixels in each frame are pro-

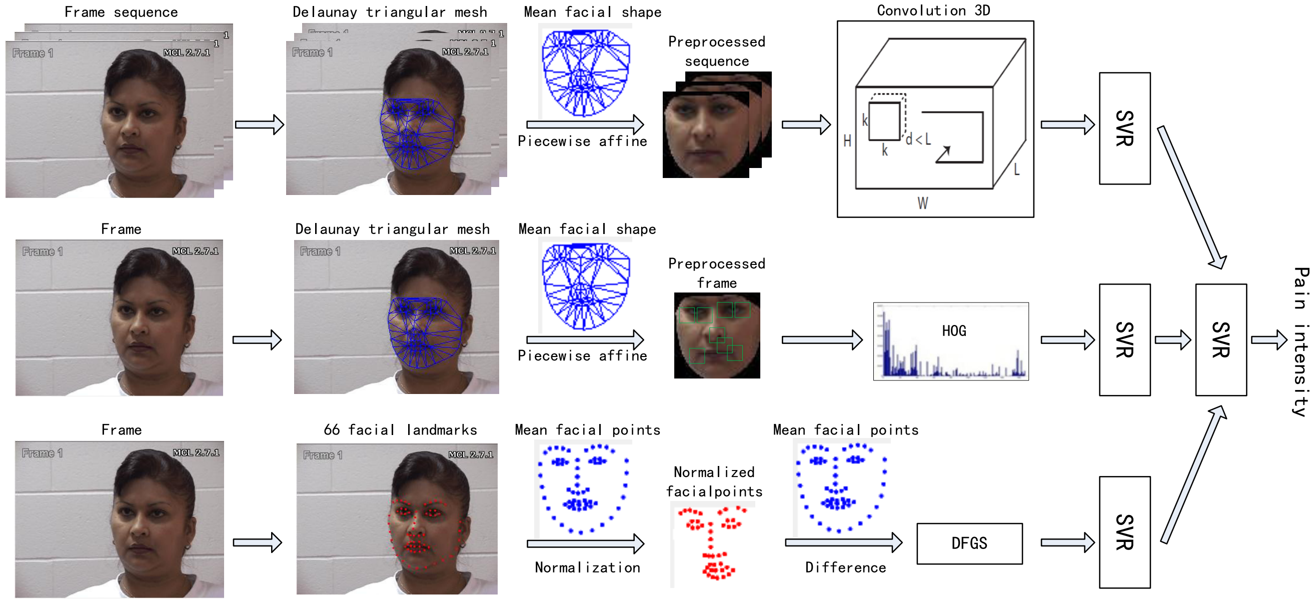


Fig. 1 The pipeline of our proposed method for pain intensity estimation.

jected onto the mean facial shape by piecewise affine transformation (see Fig. 1). After preprocessing, all images have a size of 64x64 pixels and will be used as the input of the following C3D network and to obtain HOG features. We experimented with many image sizes, but the 64x64 pixel size is ideal for training and testing the C3D network, with the best results in both efficiency and performance.

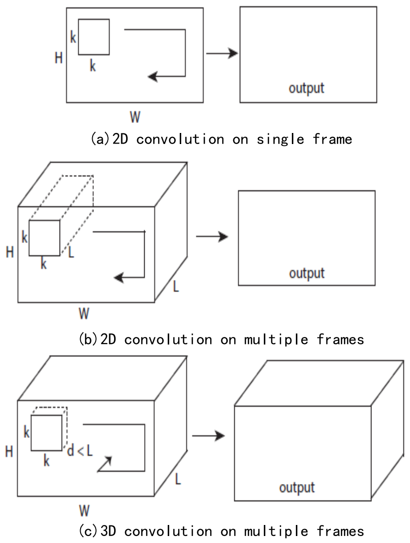
3.2 C3D Networks

C3D is a straightforward and efficient approach for spatiotemporal feature learning using deep 3-dimensional convolutional networks (3D ConvNets). It performs convolution on three channels. In this way, the outputs of C3D can be used as features for serving multiple tasks.

2D ConvNets apply convolution and pooling operations spatially only to 2D static images. The 2D convolution with an image as input will output an image. The 2D convolution with multiple images as input will also output an image. Thus, 2D ConvNets will lose temporal information after each convolution operation. 3D convolution will preserve the temporal information of the input signals, resulting in an output volume. In C3D, the operations are performed spatiotemporally by adding the time dimension. The 2D and 3D convolutional frameworks are shown in Fig. 2.

The C3D network takes a sequence of frames with fixed length as the input video volume and outputs spatiotemporal features with the specified length. Internally, the C3D network captures appearance for the first few frames and tracks the salient motion in the subsequent frames. Thus C3D differs from standard 2D ConvNets in that it selectively attends to both motion and appearance. After being trained, C3D can be used as a feature extractor for video analysis tasks.

In this work, we use C3D to model the appearance and motion of the face simultaneously. As shown in the upper part of Fig. 1, we use preprocessed consecutive facial images of size 64x64 pixels as the input to train the C3D network and use the output of the network as the spatiotemporal facial features to train the regression model for pain intensity estimation.



(a) 2D convolution on single frame

(b) 2D convolution on multiple frames

(c) 3D convolution on multiple frames

Fig. 2 2D and 3D convolution operations. a) 2D convolution applied on an image results in an image. b) 2D convolution applied on a video volume also results in an image. c) 3D convolution applied on a video volume results in another volume, preserving the temporal information of the input data.

If C3D is trained with a large-scale dataset, its network structure can be designed as deep as possible, but the depth is limited by the GPU memory capacity. Therefore, with the memory capacity (8 GB) of our GPU, we select the C3D network structure similar to [26] which has 8 convolutional, 5

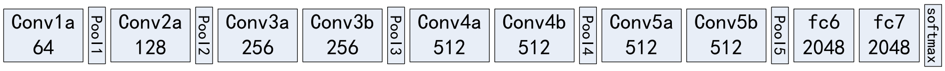


Fig. 3 C3D network architecture. The C3D network has 8 convolutional, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are 3x3x3 with a stride of 1 in both the spatial and temporal dimensions. The number of filters is denoted in each rectangle. The pooling layers are denoted from Pool 1 to Pool 5. Each pooling kernel is 2x2x2, except that Pool 1 is 1x2x2. All fully connected layers have 2048 output units.

pooling, and 2 fully connected layers, followed by a softmax output layer. The structure is presented in Fig. 3. Where Conv1a to Conv5b denote 8 convolutional layers. The number of filters is denoted under the name of each convolutional layer. All 3D convolution filter kernels are 3x3x3 with stride 1x1x1, which means that the kernel temporal depth and spatial size are both 3, and the spatial and temporal stride are both 1. The authors of [26] had proved by experiments that 3x3x3 with stride 1x1x1 is the best kernel choice for the C3D network. The pooling layers are denoted from Pool 1 to Pool 5. All pooling layers are max pooling. Each pooling kernel is 2x2x2, except that Pool 1 is 1x2x2 with the intention of not to merge the temporal signal too early. Fc6 and fc7 denote fully connected layers. Each of them has 2048 output units, which can produce a 2048D vector for use as spatiotemporal facial features.

3.3 Extraction of Geometric and Appearance Features

Geometric and appearance features have been successfully applied to automatic pain recognition, particularly when used in combination, because they separately represent the unique facial features and complement each other. Geometric features describe changes in the position and shape of facial components, such as the mouth, eyes and eyebrows. Appearance features represent small facial deformations caused by pain, such as nasolabial furrow and frown.

To extract geometric features, we normalize the facial shape in each frame image by aligning facial landmarks to the mean facial shape using the affine transformation of the facial points corresponding the eye corners and nose tip. Then, we take the difference between the normalized and the mean facial shape and use the result as the geometric facial features. We call this feature the delta of facial geometric shape or DFGS for short (see Fig. 1). Here, the 49 landmarks around the eyes, eyebrows, nose, and mouth are selected to perform the difference operation, thereby generating a 98D feature vector.

For extracting appearance features, we first preprocess each image according to the method in Sect. 3.1. Then, HOG features are extracted based on the 49 landmarks as mentioned above in the preprocessed images (see Fig. 1). We extract a block of 16x16 pixels around each facial landmark. This block contains 2x2 cells with a size of 8x8 pixels. The number of overlapping cells between neighboring blocks is 8 pixels. The number of orientation histogram bins is 9. In this way, we obtain a 1404D feature vector for the 49 facial landmarks. The block size of 16x16 pixels provided better results for the input image size than the other block sizes that we investigated.

3.4 Pain Intensity Estimation

We use regression models to perform continuous pain intensity estimation. In this work, support vector regression (SVR) is used to map the features to the corresponding pain intensity. We select SVR because it can handle large representation spaces, is easy to train, and generalizes well. SVR has been applied in various fields. The idea of SVR is based on calculating linear regression functions in high-dimensional feature space, where the input data are mapped to nonlinear functions. We use L2-regularized L2-loss SVR, which means that given a set of feature-label pairs $(x_i, y_i), i = 1, \ldots, l, x_i \in R^n, y_i \in R$, the SVR solves the following unconstrained optimization problem:

$$\min_{\omega} \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{l} (\max(0, \left|y_i - \omega^T x_i\right| - \epsilon))^2 \tag{2}$$

where $(\max(0, \left|y_i - \omega^T x_i\right| - \epsilon))^2$ is the loss function, $C > 0$ is a penalty parameter, and ϵ is a parameter to specify the sensitivity of the loss.

For feature fusion, there are two general strategies: early fusion and late fusion [37]. Early fusion is performed at the feature level by concatenating different feature vectors into a high-dimensional vector, which is then used for classification or regression. Late fusion is performed at the score level, combining different scores obtained through supervised learning on different features into one vector and then used for classification or regression. In this work, we tried both early and late fusion strategies. For the early fusion, we concatenate the C3D, geometric and HOG features into a feature vector, and use it to train an SVR model. For the late fusion, we use a two-level SVRs model. First, we train an SVR model separately on each type of feature (C3D, geometric and HOG features). The PSPI scores of the input frames are the labels for training three SVR models. Second, we combine the prediction score f_i output from each SVR model with feature set F and utilize it to train a second-level SVR model, where F is $[f_1, f_2, \ldots f_n]$ and n is the number of the first-level SVRs to be combined. Since the PSPI scores output by the SVR can be less than the minimum pain level 0 or above the maximum pain level 15, we set all predictions below 0 to 0 and above 15 to 15. When comparing the results of early and late fusion strategies, we find that late fusion performs better than early fusion. Therefore in our method, we choose late feature fusion strategy, the two-level SVRs model (see Fig. 1). We use linear SVRs in both early and late fusion strategies. The detailed results will be analyzed in Sect. 4.4.

4. Experiments and Results

4.1 Database Description

We conducted experiments on the UNBC-McMaster shoulder pain expression archive database [10] to evaluate our proposed methodology. This database is the most common data set for assessing pain detection or pain intensity estimation methods. It contains 200 face videos of 25 adult patients with rotator cuff and other shoulder injuries. The resolution of each video is 320x240 pixels. Spontaneous expression of pain from these subjects is recorded using digital cameras in a laboratory room during performing range-of-motion tests of their affected and unaffected shoulder. Active (the subject moves the arm himself) and passive (a physiotherapist moves the subject's arm) movements are recorded. The FACS code and PSPI score (from 0 to 15) are provided for all frames. The dataset also provides 66 facial landmarks tracked by the AAM for each frame. It is a very challenging database, and in some videos, distinguishing between pain and no pain becomes a difficult task, even for clinical professionals. The database is also imbalanced because it contains a total of 48,398 frames, but 82.71% (40,029) are 'no-pain' frames, and only 17.29% (8,369) are 'pain' frames. Because of the lack of pain frames, any model is likely to become biased toward the prediction of no pain.

4.2 Sample Strategy

To overcome the problem of the imbalanced data distribution, we combine class-aware sampling [38] with under-sampling during training our models. The class-aware sampling strategy has been successfully applied to image classification and obtained an approximately 0.6% accuracy improvement. Specifically, we use two types of lists: one is the PSPI score list, and the other is the list of frames for each PSPI score. For every training iteration, we first randomly sample a score (e.g., 3) in the PSPI score list, then randomly sample a frame in this score, and finally sample another score and its frame. When the end of a frame list for PSPI score is reached, a shuffle operation is performed. A shuffle operation is also performed at the end of the PSPI score list. In this work, the class-aware sampling strategy reduces the average RMSE by approximately 1% and increases the average PCC by approximately 0.8%. When performing class-aware sampling, we under-sample the no-pain class (PSPI=0) such that both pain and no-pain categories have the same probability of being sampled by this strategy. We find that the combination of class-aware sampling and under-sampling is effective in both preserving intensity pattern and reducing redundant samples, particularly the ones with a PSPI score of 0.

4.3 Measurement

In our experiments, we evaluate the proposed method using the same strategy as previous studies (such as [15], [33]–[35] and [21]), namely the leave-one-subject-out 25-fold cross-validation strategy. We leave all sequences of one chosen subject as the testing set and the remaining sequences of 24 subjects as the training set at the same time. We use the average Pearson correlation coefficient (PCC) and the average root mean square error (RMSE) as evaluation metrics. PCC measures how well the prediction can capture the trend of pain intensity change. PCC is calculated as follows:

$$PCC = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y} - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

and RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{4}$$

where n is the total number of frames of testing sequences. $\hat{y}_i$ and y_i are the pain intensity estimation and the ground truth of the ith frame, respectively. $\bar{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i$ (the sample mean), and analogously for $\bar{y}$.

4.4 Experimental Details

For C3D networks, we try to use different lengths of preprocessed face image sequences as input to find the optimum length that obtains the best performance. Due to the limitations of memory capacity and computing affordability of our GPU, we select four lengths of the image sequence for testing, namely 4, 8, 16 and 32. We adopt 25-fold cross-validation and use the average RMSE and average PCC as evaluation metrics. Figure 4 presents experimental results of training and testing the C3D network using different lengths

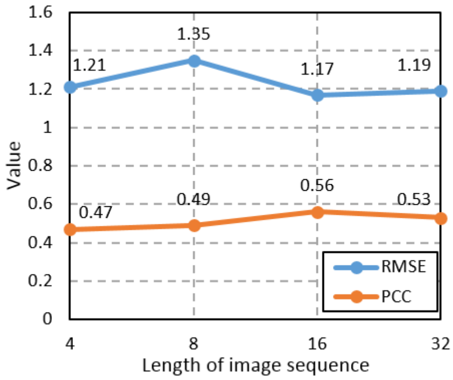


Fig. 4 Experimental results of training and testing the C3D network using different lengths of image sequences. The results are measured by the average RMSE and average PCC obtained by 25-fold cross-validation. The blue line denotes the average RMSE and the orange line denotes the average PCC. C3D performs best when the length of the input image sequence is 16.

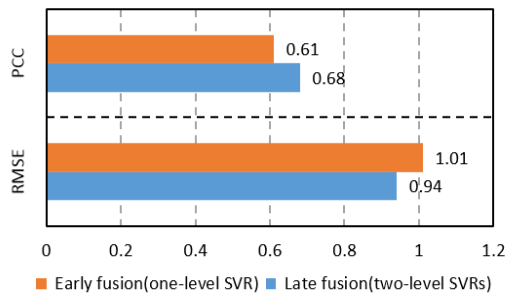


Fig. 5 Experimental results of using early and late feature fusion. The results are measured by the average RMSE and average PCC obtained by 25-fold cross-validation. The orange bar denotes early fusion (one-level SVR), and the blue bar denotes late fusion (two-level SVRs). Late fusion performs better than early fusion.

of image sequences, where the blue line denotes the average RMSE and the orange line denotes the average PCC. As shown in this figure, when using image sequences of length 16, we obtained the best performance, which is the lowest RMSE and the highest PCC. Therefore, we choose 16 as the length of the image sequence that is input to the C3D network.

To allow each frame to be processed by the C3D network, we copy the first frame of each video 15 times, and then we place the copied image in front of the first frame and renumber them such that the original first frame becomes the sixteenth frame. For each fold validation, we sample 20,000 sequences of 16 frames as the training set according to the strategy presented in Sect. 4.2. We train the C3D networks from scratch. When training the networks, we set the initial learning rate as 0.001, the momentum as 0.9, the attenuation coefficient of the learning rate as 0.1, and set decreasing learning rate strategy as 'step', which means that when executed a certain number of steps, the learning rate will be decreased to its 1/10.

For feature fusion, we used early and late fusion strategies described in Sect. 3.4 to experiment on the UNBC-McMaster database and compare the results. The results are measured by the average RMSE and PCC obtained by 25-fold cross-validation. Figure 5 presents the results, where the orange bar denotes early fusion (one-level SVR), and the blue bar denotes late fusion (two-level SVRs). Obviously, late fusion performs better than early fusion, because the PCC of late fusion is higher than that of early fusion, while the RMSE of late fusion is lower than that of early fusion. Therefore, in our method, we choose late feature fusion strategy (the two-level SVRs model). All the results analyzed in Sect. 4.5 are based on the late feature fusion strategy.

Our experiments were performed on a workstation with two 2.10 GHz Intel(R) Xeon(R) E5-2620v4 CPUs, 16 GB of RAM, and two NVIDIA GTX1080 GPUs. Each GPU has 8GB memory. The computation time of the our proposed method is 23ms per frame, that is, about 43 frames per second.

Table 2 Comparison of different features and combinations of features. The results for pain intensity estimation, measured by RMSE and PCC. The best results are given in bold letters.

Feature	RMSE	PCC
HOG	1.270	0.564
C3D	1.167	0.555
DFGS	0.990	0.595
C3D+HOG	1.071	0.599
DFGS+HOG	1.028	0.626
C3D+DFGS	0.955	0.626
C3D+DFGS+HOG	**0.942**	**0.676**

4.5 Experimental Results

We compare the performance of different individual features and combined features. Table 2 shows the result of this comparison. As shown in this table, if the PCC value is used as the performance measure, among the individual features, the geometric feature DFGS performs better than all the others, followed by the HOG feature, and the C3D feature has the lowest performance. For the RMSE, the best is also DFGS, followed by C3D, and the lowest is HOG. For the handcrafted features, the geometric features perform better than the appearance features. A possible reason for these results is that the position change of the facial feature points caused by pain intensity can more effectively capture the characteristics of pain and is more easily distinguished by the followed regressor or classifier than the other features. This result also shows that common handcrafted features (e.g., DFGS and HOG) are still very efficient for specific problems and that the deep-learned features do not perform well on small data sets.

Table 2 also shows that combined features perform considerably better than individual features. Specifically, the combination of any two individual features is better than each of these two individual features. The combination of deep-learned (e.g., C3D) and handcrafted (e.g., DFGS) features performs better than the combination of two handcrafted features (e.g., DFGS+HOG). The combination of all three features (C3D+DFGS+HOG) is better than the combination of any two individual features and performs the best. This means that all features have contributed to the improved performance. The results show that although the C3D feature does not perform well by itself, it can provide helpful information that can complement the handcrafted features. Additionally, handcrafted features are high-level abstractions of original face images and are likely to simplify critical information. However, deep-learned features are obtained directly from the image pixels; thus, they have less information loss. Figure 6 presents an example of the pain intensity estimation from one subject image sequence using our best model, i.e., C3D+DFGS+HOG two-level SVRs. As shown, the continuous estimation value of pain intensity is close to the ground truth in most frames.

In our experiments, we also compared our method with the state-of-the-art in the literature on the UNBC-McMaster shoulder pain archive database. The results are shown in

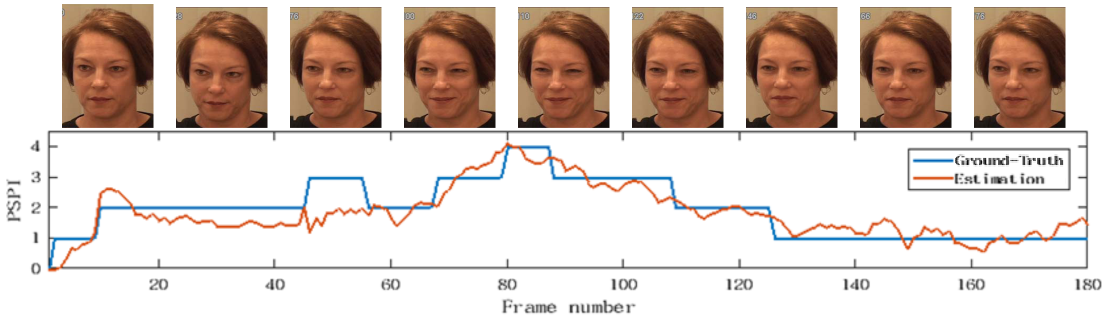


Fig. 6 An example of pain intensity estimation using C3D+DFGS+HOG and two-level SVRs.

Table 3 Comparison of the proposed method with the state-of-the-art in the literature. The results for pain intensity estimation measured by RMSE and PCC. The best results are presented in bold.

Method	RMSE	PCC
DCT+LBP / RVR [15]	1.18	0.59
HoT / SVR [33]	1.08	0.49
SIFT+PCA / SVR [34]	1.13	0.59
2Standmap / SVM [35]	1.19	0.55
RCNN / Regression [21]	1.24	0.65
C3D+DFGS+HOG / SVR	**0.94**	**0.68**

Table 3. We obtain promising results in that the RMSE of our method is 0.94 and lower than that of other methods. Moreover, the PCC of our method is 0.68 and higher than that of the other methods. This result indicates that our method is more effective than current methods. By comparison, the methods from [15] to [35] use only static features, whereas our approach integrates deep dynamic features into the learning process. The method of [21] also uses deep dynamic features, but our method combines static and deep dynamic features and achieves better results. It again proves that combining handcrafted features with deep spatiotemporal features can improve the performance of pain intensity estimation.

5. Conclusion

In this paper, we propose a frame-level automatic pain intensity estimation method based on the combination of handcrafted features and deep-learned spatiotemporal features. First, we project the facial pixels obtained by the AAM in each frame onto the mean facial shape through a piecewise affine transformation to achieve face frontalization. Then, we input consecutive preprocessed images into a C3D network to learn and extract spatiotemporal features. Moreover, the difference between the normalized and the mean facial shapes is computed as the facial geometric feature, and the HOG extracted around the landmarks of the mean facial shape on preprocessed images is used as the facial appearance feature. Finally, we use late feature fusion strategy. We train an SVR on each feature separately and combine the outputs of these three SVRs as the input to train the second-level SVR to predict pain intensity. The experimental results show that our proposed method achieves a higher PCC and lower RMSE than previous studies and outperforms the state-of-the-art. Our future work is to extend the proposed framework for estimating AU intensity.

Acknowledgments

This work is supported by the Tianjin Research Program of Application Foundation and Advanced Technology (no.15JCQNJC00900), the Tianjin Municipal Education Commission Project for Scientific Research Plan (no.JW1701), the Tianjin Key Technology Research and Development Program of the Ministry of Science and Technology (no.14RCGFGX00847), and the Doctor Foundation of Tianjin Normal University (no. 52XB1410).

References

[1] A.C de C Williams, H.T.O. Davies, and Y. Chadury, “Simple pain rating scales hide complex idiosyncratic meanings,” Pain, vol.85, no.3, pp.457–463, 2000.

[2] M. Lynch, “Pain as the fifth vital sign,” J. intravenous nursing :the official publication of the Intravenous Nurses Society, vol.24, no.2, p.8594, 2001.

[3] J.E. Brown, N. Chatterjee, J. Younger, and S. Mackey, “Towards a Physiology-Based Measure of Pain: Patterns of Human Brain Activity Distinguish Painful from Non-Painful Thermal Stimulation,” Plos One vol.6, no.9, pp.e24124, 2011.

[4] E. Schulz, A. Zherdin, L. Tiemann, C. Plant, and M. Ploner, “Decoding an individual’s sensitivity to pain from the multivariate analysis of EEG data,” Cerebral Cortex, vol.22, no.5, pp.1118–1123, 2012.

[5] P. Werner, A. Al-Hamadi, R. Niese, S. Walter, S. Gruss, and H.C. Traue, “Automatic Pain Recognition from Video and Biomedical Signals,” Proc. IEEE Conf. Pattern Recognit., 2014, pp.4582–4587, Stockholm, Schweden, 2014.

[6] JMM. Jam and H. Sadjedi, “A System for Detecting of Infants with Pain from Normal Infants Based on Multi-band Spectral Entropy by Infant’s Cry Analysis,” Proc. 2nd IEEE Conf. ICCEE, pp.72–76, Dubai, UAE, 2009.

[7] K. Craig, K. Prkachin, and R. Grunau, “The facial expression of pain,” Handbook of Pain Assessment, 2nd edition, Guilford, New York, 2001.

[8] 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.

[9] P. Ekman, W. Friesen, and J. Hager, Facial Action Coding System, Research Nexus, Salt Lake City, UT, USA, 2002.

[10] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, and I. Matthews, “Painful data: The UNBC-McMaster shoulder pain expression archive database,” Proc. IEEE Conf. FG, pp.57–64, 2011.

[11] T.F. Cootes, G.J. Edwards, and C.J. Taylor, “Active appearance mod-

els," IEEE Trans. Pattern Anal. Mach. Intell., vol.23, no.6, pp.681–685, 2001.

[12] A.B. Ashraf, S. Lucey, J.F. Cohn, T. Chen, Z. Ambadar, K.M. Prkachin, and P.E. Solomon, "The painful face-pain expression recognition using active appearance models," Image and vision computing, vol.27, no.12, pp.1788–1796, 2009.

[13] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K.M. Prkachin, "Automatically detecting pain using facial actions," Proc. IEEE Conf. ACII 2009, pp.1–8, Amsterdam, The Netherlands, 2009.

[14] P. Lucey, J. Cohn, S. Lucey, S. Sridharan, and K.M. Prkachin, "Automatically detecting action units from faces of pain: Comparing shape and appearance features," Proc. IEEE Conf. CVPR 2009, pp.12–18, Miami, Florida, USA, 2009.

[15] S. Kaltwang, O. Rudovic, and M. Pantic, "Continuous pain intensity estimation from facial expressions," Advances in Visual Computing, vol.2012, pp.368–377, 2012.

[16] R.A. Khan, A. Meyer, H. Konik, and S. Bouakaz, "Pain detection through shape and appearance features," Proc. IEEE Conf. ICME 2013, pp.1–6, San Jose, CA, USA, 2013.

[17] R. Zhao, Q. Gan, S. Wang, and Q. Ji, "Facial Expression Intensity Estimation Using Ordinal Information," Proc. IEEE Conf. CVPR 2016, pp.3466–3474, Las Vegas, NV, USA, 2016.

[18] A. Tatsuma and M. Aono, "Food Image Recognition Using Covariance of Convolutional Layer Feature Maps," IEICE Trans. Inf. & Syst., vol.E99-D, no.6, pp.1711–1715, 2016.

[19] H.D. Nguyen, A.D. Le, and M. Nakagawa, "Recognition of Online Handwritten Math Symbols using Deep Neural Networks," IEICE Trans. Inf. & Syst., vol.E99-D, no.12, pp.3110–3118, 2016.

[20] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," Proc. IEEE Conf. CVPR 2015, pp.3367–3375, Boston, MA, 2015.

[21] J. Zhou, X. Hong, F. Su, and G. Zhao, "Recurrent Convolutional Neural Network Regression for Continuous Pain Intensity Estimation in Video," Proc. IEEE Conf. CVPR 2016, pp.1535–1543, Las Vegas, NV, USA, 2016.

[22] W. Li, S. Manivannan, S. Akbar, J. Zhang, E. Trucco, and S.J. McKenna, "Gland segmentation in colon histology images using hand-crafted features and convolutional neural networks," Proc. 13th IEEE Conf. ISBI, pp.1405–1408, Prague, Czech Republic, 2016.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. IEEE Conf. CVPR 2005, pp.886–893, San Diego, CA, USA, 2005.

[24] P. Lu, W. Zheng, Z. Wang, Q. Li, Y. Zong, M. Xin, and L. Wu, "Micro-Expression Recognition by Regression Model and Group Sparse Spatio-Temporal Feature Learning," IEICE Trans. Inf. & Syst., vol.E99-D, no.6, pp.1694–1697, 2016.

[25] J. Yan, W. Zheng, M. Xin, and J. Yan, "Integrating Facial Expression and Body Gesture in Videos for Emotion Recognition," IEICE Trans. Inf. & Syst., vol.E97-D, vol.3, pp.610–613, 2014.

[26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," Proc. IEEE Conf. ICCV 2015, pp.4489–4497, Santiago, Chile, 2015.

[27] P. Lucey, J.F. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K.M. Prkachin, "Automatically detecting pain in video through facial action units," IEEE TRANS. SYST., MAN, CYBERN., SYST. vol.41, no.3, pp.664–674, 2011.

[28] Z. Zafar and N.A. Khan, "Pain Intensity Evaluation through Facial Action Units," Proc. IEEE Conf ICPR 2014, pp.4696–4701, Stockholm, Sweden, 2014.

[29] H. Pedersen, "Learning Appearance Features for Pain Detection Using the UNBC-McMaster Shoulder Pain Expression Archive Database," Computer Vision Systems, Springer International Publishing, pp.128–136, July 6, 2015.

[30] Z. Hammal and J.F. Cohn, "Automatic detection of pain intensity," Proc. ACM Conf ICMI 2012, pp.47–52, Santa Monica, CA, USA, 2012.

[31] O. Rudovic, V. Pavlovic, and M. Pantic, "Automatic Pain Intensity Estimation with Heteroscedastic Conditional Ordinal Random Fields," Proc. Conf. ISVC 2013, pp.234–243, Rethymnon, Crete, Greece, July 29-31, 2013.

[32] R. Irani, K. Nasrollahi, and T.B. Moeslund, "Pain recognition using spatiotemporal oriented energy of facial muscles," Proc. IEEE Conf. CVPRW 2015, pp.80–87, Boston, MA, USA, June 2015.

[33] C. Florea, L. Florea, and C. Vertan, "Learning Pain from Emotion: Transferred HoT Data Representation for Pain Intensity Estimation," Proc. Conf. ECCVW 2014, pp.778–790, Zurich, Switzerland, Sept. 2014.

[34] N. Neshov and A. Manolova, "Pain detection from facial characteristics using supervised descent method," Proc. IEEE Conf. IDAACS 2015, pp.251–256, Berlin, Germany, 2015.

[35] X. Hong, G. Zhao, S. Zafeiriou, M. Pantic, and M. Pietikäinen, "Capturing correlations of local features for image representation," Neurocomputing, vol.184-C, pp.99–106, 2016.

[36] O.M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," Proc. Conf. 26th BMVC, vol.1 no.3, pp.6, Swansea, UK, 2015.

[37] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders, "Early versus late fusion in semantic video analysis," Proc. ACM Conf. Multimedia 2005. pp.399–402, New York, USA, 2005.

[38] L. Shen, Z. Lin, and Q. Huang, "Learning Deep Convolutional Neural Networks for Places2 Scene Recognition," Computer Science, arXiv:1512.05830, 2015.

Jinwei Wang received the Ph.D. degree in the School of Computer Science and Technology, Tianjin University and the M.S. degree in Computer Software and Theory from Institute of Computing Technology, Chinese Academy of Sciences. Now he is an associate professor at the College of Computer and Information Engineering, Tianjin Normal University. His research work focuses on affective computing and pattern recognition.

Huazhi Sun received the Ph.D. degrees in College of Computer and Communication Engineering, University of Science and Technology Beijing. Now he is a professor at the College of Computer and Information Engineering, Tianjin Normal University. His research work focuses on pattern recognition and operating system.