Incorporating high-level and low-level cues for pain intensity estimation

Ruijing Yang*†, Xiaopeng Hong†, Jinye Peng* §, Xiaoyi Feng‡, Guoying Zhao†*

*School of Information Science and Technology
Northwest University,
Xi'an, P. R. China
Email: ruijingyang9013@gmail.com, pjy@nwu.edu.cn

†Center for Machine Vision and Signal Analysis
University of Oulu,
Oulu, Finland
{xiaopeng.hong, guoying.zhao}@oulu.fi

‡School of Electronics and Information
Northwestern Polytechnical University,
Xi'an, P. R. China
fengxiao@nwpu.edu.cn

***Abstract*—Pain is a transient physical reaction that exhibits on human faces. Automatic pain intensity estimation is of great importance in clinical and health-care applications. Pain expression is identified by a set of deformations of facial features. Hence, features are essential for pain estimation. In this paper, we propose a novel method that encodes low-level descriptors and powerful high-level deep features by a weighting process, to form an efficient representation of facial images. To obtain a powerful and compact low-level representation, we explore the way of using second-order pooling over the local descriptors. Instead of direct concatenation, we develop an efficient fusion approach that unites the low-level local descriptors and the high-level deep features. To the best of our knowledge, this is the first approach that incorporates the low-level local statistics together with the high-level deep features in pain intensity estimation. Experiments are evaluated on the benchmark databases of pain. The results demonstrate that the proposed low-to-high-level representation outperforms other methods and achieves promising results.**

I. Introduction

Pain expression is one of the vital signs in health related condition evaluation. Automatic estimation of pain aims to identify and quantify pains from visual sources such as images or videos, and thus plays a key role in real-time health-care applications. However, pain is difficult to capture and quantify, owing to its transience and subjectivity. Besides, the variations in illumination, head poses, and articulation among different people make the problem much more challenging. Usually the "golden standard" of pain estimation is self-report measurements, i.e., subjects are asked to describe the experienced pain at certain levels. Nevertheless, this method suffers from subjective biases [1] and cannot be applied to the people who are not able to talk, such as infants and unconscious people in ICU. Major advances on the measurement of pain are made by Ekman and Friesen's observational system - the "Facial Action Coding System (FACS)" [2] and Prkachin and Solomon's "Prkachin and Solomon Pain Intensity Metric (PSPI)" [3][1]. Yet this kind of observer rating methods are time-consuming and requires certified experts to annotate face images. Under this circumstance, automatic pain estimation is indispensable.

Over the past decades, with the booming advances in machine learning and computer vision, considerable progress are achieved on automatic recognition of pain from facial images. Since local descriptors are the fashion of capturing the local properties in image patches, a number of researches deploy local descriptors with various classifiers to identify pain from facial images [4], for instance, Discrete Cosine Transform (DCT) [5], Local Binary Pattern (LBP) [6], and Histogram of Oriented Gradient (HOG) [7]. To form the global representation of an image, a pooling stage is commonly used to combine the local features. Most widely used pooling techniques compute the first-order statistics via an average or maximum operation over individual feature dimensions. These methods perform well in practice when it is combined with appropriate coding methods [8]. However, the first-order pooling discards the inter-correlations among the individual local features and processes each dimension of the local features independently. By describing the correlations among different local features, a few works [9]–[15] demonstrate that second-order pooling can outperform the commonly used first-order pooling methods. Hong *et al.* [16] investigate different modes of second-order pooling to describe the global facial features with only a few feature dimensions.

Many works employ fusion methods to build global representations. Fusion methods are commonly categorized into early fusion (a.k.a. feature fusion) and late fusion (a.k.a. decision fusion). Early fusion is usually applied to concatenate multiple features before classification, while late fusion refers to the judgment of the output scores from different classifiers. Lucey *et al.* [17] combine shape features together with appearance features at feature level to achieve improved results for pain monitoring. Kaltwang *et al.* [4] propose a three-stage approach, where they train their classifiers using Relevance Vector Regression (RVR) with three local descriptors separately. In the third step they employ late fusion: the outputs from previous regressors are combined and fed into a new RVR to obtain the final prediction. These fusion methods are simple and easy to implement. However, since different

§Corresponding author

[1]PSPI is currently the only quantification metric of pain intensity from observer view on a frame-by-frame basis.

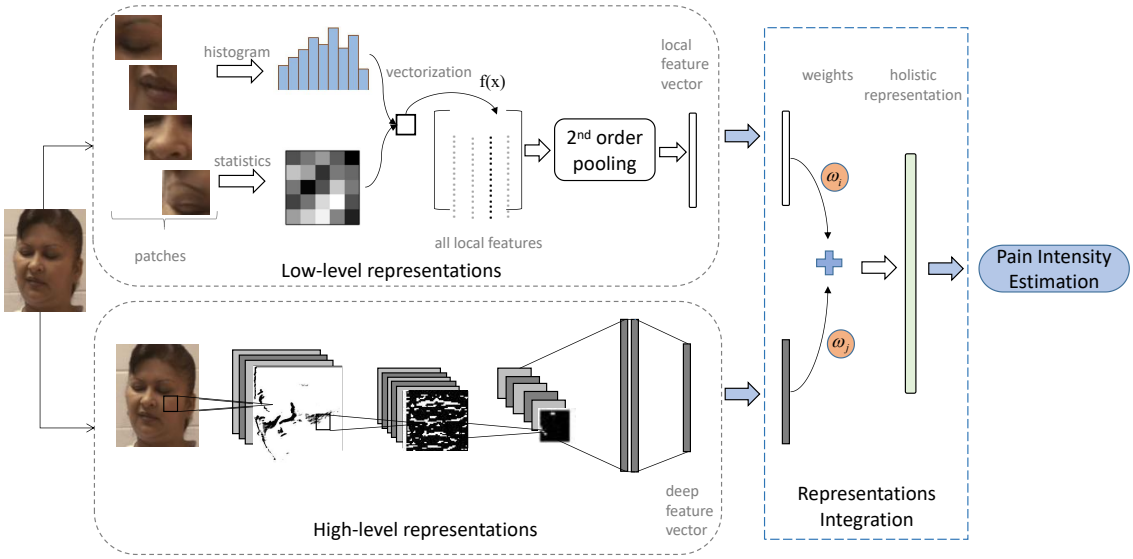

Fig. 1. **Framework of the proposed method**. Our approach contains three main parts: 1) extracting low-level (local) features from image patches and assembling them using second-order pooling method; 2) obtaining high-level representations via deep convolutional neural network; 3) integrating the low-level and high-level representations by a weighting process.

features convey different types of information, for specific tasks, multiple features should not be treated equally in the fusion stage. Recently, research attentions have moved to pain intensity estimation by using deep features from, for example, the recurrent convolutional neural network (RCNN) [18] or the deep convolutional neural network (CNN) [19]. They are semantically high-level features. However, the advantages of the low-level descriptors (local descriptors) and the high-level representations (deep features) have not yet been integrated.

To fill this gap, in this paper, we propose a weighted method to explore and incorporate the second-order local features and the deep features. Our method is applied to the task of continuous pain intensity estimation and achieves promising results.

The contributions of this paper are two-fold: 1) We develop the efficient second-order pooling method for a set of local descriptors extracted from facial images. Particularly, we take advantages of symmetric positive definite matrices that form the Riemannian structure, to assemble sets of local features, while preserving their inter-correlations. 2) We incorporate the low-level descriptors with high-level deep features by an efficient weighting method, in conjunction with linear classifiers, which further enhances the performance of pain intensity estimation and the simple implementation also allows for scalability in the number of features.

II. Related Work

The changes in facial appearances could be useful cues for pain expression recognition [20]. Facial Action Coding System (FACS) [2] pinpoints this correlation. FACS is used to describe the corresponding correlation between different facial muscle movements and facial expressions described by 44 independent action units (AU). Later, Prkachin and Solomon [3] propose the Prkachin and Solomon Pain Intensity Metric (PSPI), where pain is defined as the sum of the intensities of brow lowering, orbital tightening, levator contraction, and eye closure (four core AUs that are most related with pain). This metric is able to discern among 16 discrete pain levels on a frame-level basis. Early researches on pain analysis focus on the recognition from static images [21] and on posed expressions [22] which cast the light on automatic identification of pain. During the past years, the advent of *UNBC-McMaster Shoulder Pain Database Archive* [23] has propelled the research field toward analyzing spontaneous pain expression [4], [17], [24]–[32].

The methodologies used in estimating spontaneous pain vary across studies. Generally, the key to successfully identifying pain is the robust representations, where local descriptors have exhibited its powerful ability on capturing discriminative features from image patches on pain expressions [4], [33].

Local descriptors encode facial features from image pixels into the form such as histogram (HOG), statistics, orthogonal transform (DCT), and have shown their discriminative power in various face analysis tasks. Brahman *et al.* [34], Ashraf *et al.* [28], and Lucey *et al.* [23] use local low-level (DCT) or raw features (AAM shape features) as descriptors to identify pain and no pain from images. Extending pain detection as a multi-level classification problem, Hammal and Cohn [35] extract appearance features to classify pain intensities into a four-level scale. It is worth noting that Kaltwang *et al.* 's work [4] is the first that estimated the continuous pain intensities of the entire 16 levels of PSPI metric, by applying shape features, local descriptors of Local Binary Patterns (LBP), Discrete Cosine

Transform (DCT) and their combinations.

Recently, high-level representations such as the features extracted from Convolution Neural network (CNN) have been implemented successfully on computer vision tasks [36]. CNNs could grasp subtle information within image patches by applying a set of filters and build a global representation via a set of consecutive layers with different activation and pooling functions. Rodriguez *et al.* [19] apply a very deep convolutional neural network (VGG-16) [37], which is pre-trained on millions of face images, together with Long Short-Term Memory (LSTM) [38] to estimate pain intensity. Zhou *et al.* implement the Recurrent Convolution Neural Network (RCNN) as a regressor to predict the frame-level pain intensity.

Many of the existing pain estimation methods utilize features of a particular type. However, each type of representation has its advantage in a speciality. For instance, low-level representations are robust to illumination variations and registration errors, while high-level representations are adept at generating features that are semantically interpretable. Therefore, it is desirable to bond low-level and high-level representations so that they provide complementary information. Currently, there is no such research that attempts to incorporate local descriptors with deep features for pain intensity estimation.

In this paper, we come right from this perspective. To form our low-level representations, we use the combination of several local descriptors. Then through a weighting process, the low-level representations are integrated with the deep features to generate the global representation of the face. Experimental results show that our integrated low-to-high-level representation *2pooldeep* can efficiently exploit discriminative information among local descriptors and deep features, and lead to the boost of performance compared to other fusion methods.

III. The proposed framework

The proposed framework can be summarized in the diagram as depicted in Fig. 1. As we can see in Fig. 1, our pain intensity estimation framework mainly consists of assembling the low-level descriptors, extracting deep features, and integrating the low-level and high-level representations through a weighting procedure.

A. The assembled low-level descriptors

In the case of combining local descriptors, the existing works usually aggregate different descriptors by simple concatenation, which may not be capable of capturing higher correlations between local feature pairs. Encouraged by second-order pooling from [11] and [16], we formulate it into our pain estimation framework to reveal the correlations among local descriptors.

1) Local descriptors: The local descriptors that we use consist of statistics-based and histogram-based descriptors. We assume that an image is represented by I, and R is the region inside I, with the size of $W \times H$. $\mathbf{x} = (x, y)$ is the pixel inside region R, where x and y are the pixel positions.

Statistics-based descriptors: For each pixel $\mathbf{x}$, a set of raw features $\mathbf{f_r}(\mathbf{x})$, $\mathbf{f_r}(\mathbf{x}) \in \mathbb{R}^n$, are first extracted, which describe the properties of pixels, such as the intensity, the first and second-order partial derivatives with respect to the pixel locations. Here, we use the following raw feature sets: $\mathbf{f_r}(\mathbf{x}) = [I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|]^T$, where I_x and I_y are the horizontal and vertical derivatives, I_{xx} and I_{yy} are the second-order partial derivatives. We then choose the statistics-based local descriptor [10] to perform on $\mathbf{f_r}(\mathbf{x})$ in region R, which first computes the covariance matrix across each dimension of $\mathbf{f_r}(\mathbf{x})$. The covariance matrix of $\mathbf{f_r}(\mathbf{x})$ in region R is defined as:

$$\mathbf{G}(\mathrm{R}) = \mathrm{c_r} \sum_{\mathbf{x} \in R} (\mathbf{f_r}(\mathbf{x}) - \mu_\mathrm{r})(\mathbf{f_r}(\mathbf{x}) - \mu_\mathrm{r})^\mathrm{T}, \tag{1}$$

where c_r is the normalization factor, μ_r is the mean value of $\{\mathbf{f_r}(\mathbf{x})\}_{\mathbf{x} \in \mathbb{R}}$. To obtain $\mathbf{G}(\mathrm{R})$, integral image [9] is applied to achieve fast implementation. Due to the symmetric positive define property of covariance matrix, the statistics-based descriptor $\mathbf{f_s}(\mathrm{R})$ is finally obtained by choosing the upper triangular entries of $\mathbf{G}(\mathrm{R})$ into a vector.

Histogram-based descriptors: The Histogram of Image Gradient Orientation [16] is used as histogram-based local descriptor $\mathbf{f_h}(\mathrm{R})$. For each region R, the statistics-based descriptor $\mathbf{f_s}(\mathrm{R})$ and histogram-based descriptor $\mathbf{f_h}(\mathrm{R})$ are concatenated to form the assembled low-level descriptors:

$$\mathbf{f}(\mathrm{R}) = [\mathbf{f_s}(\mathrm{R});\ \mathbf{f_h}(\mathrm{R})]. \tag{2}$$

We then apply the second-order pooling to $\mathbf{f}(\mathrm{R})$ to obtain a global representation. Here we name the second-order pooling feature of an image as *2poolfeat*.

2) Second-order average pooling: We focus on second-order interactions (such as the outer product) with the average operation, which is performed on $\mathbf{f}(\mathrm{R})$. There are two main stages of *second-order average pooling*, namely, the *pre-defined mapping* and the *non-linear mapping*.

Pre-defined mapping: Second-order central moment [39] is first applied on $\mathbf{f}(\mathrm{R})$ to obtain the pre-processed feature vector $\mathbf{p}(\mathrm{R})$. Specifically, central moments are the moments about variable's mean. Therefore, the pre-processed feature vector $\mathbf{p}(\mathrm{R})$ is defined as:

$$\mathbf{p}(\mathrm{R}) = \mathbf{f}(\mathrm{R}) - \mu, \tag{3}$$

where μ is the mean of $\mathbf{f}(\mathrm{R})$. Normalization is then applied on the features, thus the pre-defined mapping here is named as *normalized central moment*.

We then define the *second-order average pooling* $\mathbf{G}_{2Avgp}$ as the matrix:

$$\mathbf{G}_{2Avgp}(I) = c \sum_{\mathrm{R} \subset I} \mathbf{p}(\mathrm{R})\mathbf{p}(\mathrm{R})^\mathrm{T}, \tag{4}$$

where c is the normalization factor.

Non-linear mapping: Logarithmic mapping [22] are applied to second-order average pooling to form non-linear mapping. Since the second-order average pooling results in symmetric

positive definite (SPD) matrices, they form a Riemannian manifold [12]. This could be mapped to an Euclidean tangent space via logarithmic mapping under strong theoretical guarantee [22]. Hence, the non-linear mapped second-order average pooling of image I is defined as:

$$\mathbf{g}_{2pool}(I) = logm(\mathbf{G}_{2Avgp}(I)). \quad (5)$$

As $\mathbf{G}_{2Avgp}(I)$ is symmetric, we form the final assembled low-level descriptor by concatenating the elements in the upper triangle of $\mathbf{G}_{2Avgp}(I)$.

B. High-level descriptors

In parallel, we compute our high-level representations through a type of Convolution Neural Networks (CNN) [40], which are variations of multilayer perceptrons (MLPs) inspired by biological structure of neural connections in brain. Typically, CNN is a sequence of Convolutional Layer, Pooling Layer and Fully-Connected Layer, stacked on top of each other, and every layer transforms a volume of data (in tensor form) of activations to another through a differentiable function. Different from the traditional neural network, CNN connects neurons to a local region of the input volume (spatially sparse connectivity), and share the parameters across the entire visual field among a set of filters (i.e., the same layer). These constraints of the model enable CNNs to achieve faster and better generalization on vision problems.

To obtain the high-level representation, we feed the pain data (frames in each video) into VGG-16 [37], a very deep CNN that has shown its power through a number of visual tasks. Since we want to obtain the representation of the pain frames, the fully connected layer is removed, and the outputs of `fc6` layer are chosen. This process results in a 4096-*d* feature vector $\mathbf{g}_{vgg} \in \mathbb{R}^{4096}$ representing each frame.

C. Integration of low-level and high-level representations

Each type of descriptors has different properties and the power of robustness against the impacts from the variations in subject identity, head pose, illumination conditions. Local representations encode features in patches which are defined in terms of the facial landmark locations. The high-level representations convey facial information through a hierarchical procedure. By integrating multiple types of representations, we can exploit the potential facts from face images.

We employ a weighted method to incorporate the low-level and high-level features. To make the evaluation more reasonable, we normalize each feature by applying L2 norm. Let $\hat{\mathbf{g}}_{2pool}$ and $\hat{\mathbf{g}}_{vgg}$ be the normalized feature for $\mathbf{g}_{2pool}$ and $\mathbf{g}_{vgg}$, and the integrated low-to-high representation of image I is defined as:

$$\mathbf{g} = [\omega_0 \hat{\mathbf{g}}_{2pool};\ \omega_1 \hat{\mathbf{g}}_{vgg}], \quad (6)$$

where $0 < \omega_0 < 1$, $0 < \omega_1 < 1$ are the weighted coefficients of $\hat{\mathbf{g}}_{2pool}$ and $\hat{\mathbf{g}}_{vgg}$, and constrained by $\omega_0 + \omega_1 = 1$. In this research, the parameters ω_0 and ω_1 are obtained empirically: we try different sets of ω_0 and ω_1 and choose the pair that achieves the best performance. Having obtained ω_0 and ω_1, the local and deep features can be effectively combined. Note that in practice, the features using our method are not limited to the two described above. Indeed, other combinations of features can be employed by our weighted integration method.

IV. Experiments

We focus on evaluating our method on the UNBC-McMster Shoulder Pain Archive Database [23] and view pain intensity estimation as a regression process.

A. The UNBC-McMaster Shoulder Pain Archive

The UNBC-McMaster Shoulder Pain Expression Archive Database [23] is a widely used database for pain estimation. It contains in total 200 video sequences of 48,398 FACS coded frames from 129 volunteers (63 males, 66 females). The subjects are of various occupations and age groups. These subjects are self-identified as suffering from shoulder pain and the videos are recorded when they are experiencing a series of active and passive motions of their affected and unaffected limbs. In this database, each frame is AU-coded by certified FACS coders, and the corresponding PSPI scores are computed in 16 discrete levels (0-15), according to the following equation:

$$Pain = AU4 + (AU6||AU7) + (AU9||AU10) + AU43. \quad (7)$$

As the database is unbalanced across the 16 levels, many previous researches manually balance the data by down-sampling the majority class (no pain). In our experiments, we utilize a weighted loss function during training, to learn a better model.

B. Experimental Settings

To mitigate the influence of possible inconsistent colors and poses across the videos included in the database, the first step of our approach is to segment the face region from each video sequence. For that purpose, we employ part of the 66 facial landmarks detected by Active Appearance Model (AAM) [41], which are provided by the database publisher [17]. We fix a set of key landmark points which consist of the positions of left eye, right eye, the outermost points of left and right side of the face, and the points on the bottom of jaw. By utilizing AAM landmarks, it is straightforward to track faces along the video. The facial regions are then cropped from each video frame according to the key landmarks.

We evaluate the performance of the proposed feature integration method on the continuous pain intensity estimation. To achieve this, we learn a regression function which maps the features into corresponding pain intensities. The function is learned by the linear L2-regularized L2-loss Support Vector Regressor (SVR). For fast computing, we adopt the Liblinear package tool [42]. The Mean Squared Error (MSE) and Pearson Correlation Coefficients (PCC) are used as the evaluation measurements.

C. Experimental Results

We perform the proposed weighted integration *2pooldeep* of the two features: *2poolfeat* and *vgg*, and compare the results from each of them to see the enhancement of performances. In all our experiments, a leave-one-subject-out cross validation procedure is applied. To be more specific, we train our model using the data from 24 subjects and test on the left one. The mean MSE and mean PCC are computed across all subjects. For training the SVR, the best c is selected through a loop search on training data.

Following [16], each input frame is divided into 8×8 non-overlapping blocks with a resolution of 16×15. The *2poolfeat* is then computed and vectorized into a feature vector as the low-level representation of the frame. After that, L2-norm is applied on the feature vector. We tried different parameters of HIGO (e.g., 4bin, 8bin), and reported the best in this paper. In addition, we also evaluate the performance of single deep feature *vgg*, together with linear SVR, as the baseline method of high-level representations.

For a fair comparison with the state of the art, we only include the methods that utilize the whole database with similar evaluation methods. However, the evaluation measurements may still differ from researches. For instance, Kaltwang *et al.* [4] mention that their MSE and PCC are computed per subject and per sequence and correspondingly weighted by the number of frames in each sequence. In our experiments, MSE and PCC are computed for each subject, and the average MSE and PCC across subjects are reported. In addition, to better reveal the actual performance on test data, we also introduced a measurement that calculate the MSE and PCC weighted by the ratio of test samples and all the samples, denoted as SMSE, SPCC.

As we can see in Table 1, comparing to the low-level features *2poolfeat* and the deep feature *vgg*, our proposed method *2pooldeep* achieves the best performance[2], showing its power in incorporating the potential information between low-level local descriptors and the high-level deep features. In addition, *vgg* obtains slightly better results than *2poolfeat*, which indicates the power of high-level representations.

In Table 2, we report the results into two parts, the upper rows are the comparisons of the single feature performances with the state of the art, and the bottom three rows present the methods that combine multiple features. As the results indicated in Table 2, among all the low-level single features, *2poolfeat* outperforms other single features, showing the advantage of utilizing second-order information. For feature integration, it can be seen that the proposed *2pooldeep* obtains the promising performance with average MSE and PCC of 1.446 and 0.520, respectively, indicating our weighting method is highly discriminative. Compared to [4], where they fuse the outputs of three local descriptors that are treated equally, we go one step further, to incorporate low-level descriptors with high-level deep features, and consider their different characteristics via a weighting process. As a result, our proposed *2pooldeep* achieves improved performance. With regard to encoding the inter-correlation among local descriptors using higher-order statistics, we also obtain superior results, compared to [31]. Florea *et al.* [31] combine gradient information with Hessian-based histogram, whereas we consider extracting the invariant features from several local descriptors at the same time and uniting them by second-order average pooling.

[2]The reported results of *2poolfeat* [16] is slightly better than the one implemented in this paper. This is caused by the different ways of pre-processing which has showed the importance of face alignment and registration.

TABLE I
THE PERFORMANCE OF THE PROPOSED METHOD

Features	MSE	PCC	SMSE	SPCC
2poolfeat	1.636	0.366	1.453	0.350
vgg	1.553	0.523	1.470	0.495
2pooldeep	**1.446**	**0.520**	**1.325**	**0.504**

TABLE II
PERFORMANCES COMPARED TO THE STATE OF THE ART

Features	MSE	PCC
PTS [4]	2.592	0.363
DCT [4]	1.712	0.528
LBP [4]	1.812	0.483
Hessian Histograms [31]	3.760	0.250
Gradient Histograms [31]	4.760	0.340
vgg	**1.553**	**0.523**
2poolfeat	1.636	0.366
PTS+DCT+LBP(RVR) [4]	1.804	0.502
Hes+Grad [31]	3.350	0.410
2pooldeep (ours)	**1.446**	**0.520**

V. CONCLUSION

Automatic pain intensity estimation is of great importance since it is vital for the intelligent health-care and can be applied in both clinics and home cares. In this paper, we introduced a discriminative weighted approach that integrates second-order pooling based low-level descriptors with high-level deep representations. Different from traditional feature combination methods, our weighting approach is able to capture the most important facial cues from higher-level semantic representations together with the low-level local ones that describe the statistical attributes of the face images. Besides, it is simple to implement, requiring few parameters, and can obtain high estimation performance in conjunction with linear regressors. Furthermore, instead of fusing one specific local descriptor with deep features, we also presented second-order pooling procedures for a set of local descriptors, which can squeeze the redundant information while preserving their pairwise correlations. The experimental results on the benchmark pain database suggest that our method outperforms the previous works that employ fusion methods. Considering the importance of facial dynamics throughout time, and with the encouraging results, we plan to extend the method to fully

exploit the temporal information in video sequences for better performance in future works.

ACKNOWLEDGMENT

This work was supported, in part, by the National Natural Science Foundation of China (No. 61772419 & 61572205), Program for Changjiang Scholars and Innovative Research Team in University of Ministry of Education of China (No. IRT_17R87), Special Research Project of Shaanxi Education Department (No. 16JK1774), the National Key Research and Development Program of China (No. 2017YFB0203104) and the Fund for Integration of Cloud Computing and Big Data, Innovation of Science and Education (No. 2017A1950). This works was also partly supported by the Academy of Finland, Infotech Oulu, and Tekes Fidipro Program. Furthermore, we express deep gratitude to the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

[1] R. R. Cornelius, *The science of emotion: Research and tradition in the psychology of emotions.* Prentice-Hall, Inc, 1996.

[2] P. Ekman and W. Friesen, *Facial Action Coding System: A technique for the measurement of facial movements*. Plato Alto: Consulting Psychologist Press, 1978.

[3] 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.

[4] S. Kaltwang, O. Rudovic, and M. Pantic, "Continuous pain intensity estimation from facial expressions," in *International Symposium on Visual Computing*. Springer, 2012.

[5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[6] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *ECCV*. Springer, 2004.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*. IEEE, 2005.

[8] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *ICCV*. IEEE, 2011.

[9] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *TPAMI*, vol. 30, no. 10, pp. 1713–1727, 2008.

[10] P. Li and Q. Wang, "Local log-euclidean covariance matrix (l 2 ecm) for image representation and its applications," *ECCV*, 2012.

[11] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *ECCV*. Springer, 2012.

[12] X. Hong, H. Chang, S. Shan, X. Chen, and W. Gao, "Sigma set: A small second order statistical region descriptor," in *CVPR*. IEEE, 2009.

[13] X. Hong, G. Zhao, M. Pietikäinen, and X. Chen, "Combining lbp difference and feature correlation for texture description," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2557–2568, 2014.

[14] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with grassmann covariance matrices," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 8, no. 4, p. 14, 2016.

[15] X. Hong, H. Chang, S. Shan, B. Zhong, X. Chen, and W. Gao, "Sigma set based implicit online learning for object tracking," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 807–810, 2010.

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

[17] 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.

[18] J. Zhou, X. Hong, F. Su, and G. Zhao, "Recurrent convolutional neural network regression for continuous pain intensity estimation in video," in *CVPR Workshops*, 2016.

[19] P. Rodriguez, G. Cucurull, J. Gonzàlez, J. M. Gonfaus, K. Nasrollahi, T. B. Moeslund, and F. X. Roca, "Deep pain: Exploiting long short-term memory networks for facial expression classification," *IEEE Transactions on Cybernetics*, 2017.

[20] K. D. Craig, S. A. Hyde, and C. J. Patrick, "Genuine, suppressed and faked facial behavior during exacerbation of chronic low back pain," *Pain*, vol. 46, no. 2, pp. 161–171, 1991.

[21] S. Brahnam, L. Nanni, and R. Sexton, "Introduction to neonatal facial pain detection using common and advanced face classification techniques," in *Advanced Computational Intelligence Paradigms in Healthcare–1*. Springer, 2007.

[22] G. C. Littlewort, M. S. Bartlett, and K. Lee, "Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain," in *International Conference on Multimodal Interfaces*. ACM, 2007.

[23] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, "Painful data: The unbc-mcmaster shoulder pain expression archive database," in *FG*. IEEE, 2011.

[24] Z. Hammal, M. Kunz, M. Arguin, and F. Gosselin, "Spontaneous pain expression recognition in video sequences," in *International Academic Conference on Visions of Computer Society*. BCS, 2008.

[25] 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 Transactions on Cybernetics*, vol. 41, no. 3, pp. 664–674, 2011.

[26] M. Monwar and S. Rezaei, "Pain recognition using artificial neural network," in *IEEE International Symposium on Signal Processing and Information Technology*. IEEE, 2006.

[27] 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.

[28] 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.

[29] P. Lucey, J. Howlett, J. Cohn, S. Lucey, S. Sridharan, and Z. Ambadar, "Improving pain recognition through better utilisation of temporal information," in *International Conference on Auditory-Visual Speech Processing*. NIH Public Access, 2008.

[30] Z. Hammal and J. F. Cohn, "Automatic detection of pain intensity," in *International Conference on Multimodal Interaction*. ACM, 2012.

[31] C. Florea, L. Florea, and C. Vertan, "Learning pain from emotion: transferred hot data representation for pain intensity estimation," in *ECCV*. Springer, 2014.

[32] K. Sikka, A. Dhall, and M. Bartlett, "Weakly supervised pain localization using multiple instance learning," in *FG*. IEEE, 2013.

[33] L. Nanni, S. Brahnam, and A. Lumini, "A local approach based on a local binary patterns variant texture descriptor for classifying pain states," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7888–7894, 2010.

[34] S. Brahnam, C.-F. Chuang, R. S. Sexton, and F. Y. Shih, "Machine assessment of neonatal facial expressions of acute pain," *Decision Support Systems*, vol. 43, no. 4, pp. 1242–1254, 2007.

[35] Z. Hammal and M. Kunz, "Pain monitoring: A dynamic and context-sensitive system," *Pattern Recognition*, vol. 45, no. 4, pp. 1265–1280, 2012.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] W. Feller, *An introduction to probability theory and its applications: volume I*, 1968.

[40] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[41] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *TPAMI*, vol. 23, no. 6, pp. 681–685, 2001.

[42] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1871–1874, 2008.