Pattern Recognition Letters 140 (2020) 26–33

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Self-supervised pain intensity estimation from facial videos via statistical spatiotemporal distillation[☆]

Mohammad Tavakolian[a,*], Miguel Bordallo Lopez[b], Li Liu[a,c]

[a] Center for Machine Vision and Signal Analysis (CMVS), University of Oulu, Finland
[b] VTT Technical Research Centre of Finland Ltd, Oulu, Finland
[c] National University of Defense Technology, China

A R T I C L E I N F O

Article history:
Received 31 July 2020
Revised 6 September 2020
Accepted 13 September 2020
Available online 19 September 2020

MSC:
41A05
41A10
65D05
65D17

Keywords:
Self-supervised learning
Representation learning
Pain assessment
Statistical spatiotemporal distillation

A B S T R A C T

Recently, automatic pain assessment technology, in particular automatically detecting pain from facial expressions, has been developed to improve the quality of pain management, and has attracted increasing attention. In this paper, we propose self-supervised learning for automatic yet efficient pain assessment, in order to reduce the cost of collecting large amount of labeled data. To achieve this, we introduce a novel similarity function to learn generalized representations using a Siamese network in the pretext task. The learned representations are finetuned in the downstream task of pain intensity estimation. To make the method computationally efficient, we propose Statistical Spatiotemporal Distillation (SSD) to encode the spatiotemporal variations underlying the facial video into a single RGB image, enabling the use of less complex 2D deep models for video representation. Experiments on two publicly available pain datasets and cross-dataset evaluation demonstrate promising results, showing the good generalization ability of the learned representations.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

The concept of pain is learned by individuals through their life experiences. Hence, pain is a subjective phenomenon, which is influenced to varying degrees by biological, psychological, and social factors. Formally, pain can be described as an unpleasant sensory and emotional experience typically caused by actual or potential tissue injury. Therefore, pain is considered as an indicator for the health condition, which identifies harmful conditions for the body. Reliable understanding of pain contributes to disease diagnosis and provides useful information for healthcare personnel to choose adequate treatment to avoid long-lasting consequences for individuals' health.

Automatic pain assessment has attracted increasing attention of the research community, as it offers continue and objective pain assessment that improves the clinical outcome. Using visual data, many automatic pain assessment methods have been proposed by analyzing facial expressions which are a reliable indicator of pain, estimating pain intensity, and distinguishing facial expressions caused by pain from other ordinary facial expressions. In automatic pain assessment, the development of powerful feature representations from facial videos plays a very important role and thus has been one main focus of research. Earlier works attempt to address this problem by either ignoring the temporal cue and extracting spatial features from frames, or treating videos as volumetric objects to capture spatiotemporal features. For instance, Lucey et al. [17] used SVM classifiers to classify there levels of pain intensity. Kaltwang et al. [14] computed a facial representation by extracting LBP and DCT features from video frames. Zhao et al. [32] introduced an alternating direction technique of multipliers to solve Ordinal Support Vector Regression. Recent advances in deep learning, in particular Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have brought significant progress for the problem of pain analysis from facial videos. State of the art results in this field are typically supervised deep learning methods. Zhou et al. [33] trained a Recurrent Convolutional Neural Network (RCNN) to solve a regression problem. Similarly, Rodriguez et al. [21] extracted features from frames using a CNN and fed these features to a LSTM. Tavakolian and Hadid [23] proposed a Spatiotemporal Convolutional Neural Network to capture low, mid, and long-range variations in the face for pain intensity estimation. However, data annotation in pain analysis is typically done using

☆ Handle by Associate Editor Li Liu.

* Corresponding author.
E-mail address: mohammad.tavakolian@oulu.fi (M. Tavakolian).

https://doi.org/10.1016/j.patrec.2020.09.012
0167-8655/© 2020 Elsevier B.V. All rights reserved.

the Facial Action Coding System (FACS) [6] which firstly decomposes facial expressions into basic action units and then codes each action unit with onset, offset and intensity. FACS based labeling is costly and time consuming, since it takes around two hours for a FACS expert to code one minute video [16].

In addition, the generalization ability of supervised learning is rather limited, *i.e.* defined largely by the training data. In other words, the performance of existing methods is limited to the data which they are trained on [20,27,29]. The differences in culture, gender and age, as well as imaging conditions such as lighting, pose, low resolution and noise are counted as factors for performance deterioration when generalizing methods across datasets [5]. Othman et al. [20] performed cross-dataset pain recognition using two datasets and their combination. However, they reduced the size of the deep model as well as training data by sampling frames that have high chance of pain occurrence. Currently, to the best of our knowledge, another challenging issue in facial video based pain analysis is the lack of large scale benchmark datasets due to the difficulties in data collection and labeling, while deep learning methods require large scale datasets for training to obtain good performance. Therefore, leveraging knowledge from pre-trained models and using it to advance pain assessment could be a good option. In order to improve the generalization of pain assessment methods, we advocate the test of performance robustness by applying cross-database evaluation and the use of dataset combination to advance pain assessment.

To avoid time-consuming and costly data annotations, self-supervised learning has been put forward to learn visual representations from large-scale unlabeled data ([13]). It is concerned with learning semantically meaningful features from unlabeled data attributes. Hence, self-supervised learning can be considered as an alternative to supervised learning in automatic pain assessment, where collecting large-scale labeled dataset is a tedious task. In this paper, we propose a self-supervised learning method to capture visual features from unlabeled data using a pretext task to improve the generalization of learned representations for pain intensity estimation. To be specific, we learn feature representations based on measuring sample similarity of large-scale unlabeled data with a Siamese network [2,11]. Then, we transfer the learned representations to the downstream task of estimate pain intensity level. We note that deep models usually have large computational complexity for spatiotemporal processing of videos. On the other hand, processing short intervals of videos is not efficient as they contain less information [26]. To circumvent the deficiencies caused by processing videos on short intervals and/or avoiding deep models for temporal processing, we propose a highly simple, efficient yet effective method called Statistical Spatiotemporal Distillation (SSD) to aggregate motion and appearance information underlying a video into one feature map. By temporally dividing the videos into smaller segments, SSD extracts the mutual information between consecutive segments to generate feature maps based on Gaussian Scale Mixture. Using the obtained feature maps (*i.e.* 2D representations of videos) allows employing less complex deep models, which are designed for images, for video processing.

In summary, our main contributions in this paper are summarized as follows.

- To the best of our knowledge, this is the first work that explores the idea of self-supervised representation learning in pain estimation. To achieve this, we adapt a novel similarity function through a Siamese network to learn representations from unlabeled data.
- In order to achieve computational efficiency, we propose a simple, efficient yet effective method (SSD) to encode statistical information of the underlying dynamic and appearance of an arbitrary-length video into a single image map.
- We extensively validate the effectiveness of our proposed method for pain intensity estimation in cross-database settings, using two publicly available datasets.

2. Proposed method for automatic pain assessment

In this section, we present firstly in detail the proposed SSD approach, which enables us to effectively aggregate the appearance and dynamic variations within a video into one single RGB image, and, the use of less complex 2D deep models to analyze the data. Then, we present how to explore self-supervised learning to learn generalized representations from unlabeled data for automatic pain assessment.

2.1. Statistical spatiotemporal distillation

Visual attention is usually given to the regions that have more descriptive information. Inspired by information theory, the local information of an image can be quantified in terms of sequences of bits [10]. We extend this notion to the temporal dimension in order to capture the discriminative spatiotemporal information. To this end, under a Markovian assumption, we delineate a video as an image map by devising a statistical model for a neighboring group of pixels using Gaussian Scale Mixture (GSM).

Firstly, we divide a given video into several nonoverlapping segments. Then, the mutual information between consecutive segments (*i.e.* the total perceptual information content of consecutive segments) is explored to model the variations in dynamics and appearance. Hence, we encode the spatiotemporal variations of two consecutive segments into one image map. Further, we use an aggregation function to combine several image maps that are obtained by processing consecutive segments throughout the video (Fig. 1).

Our approach is formally formulated as follows. A block volume, $\boldsymbol{x}_i$, in a video segment is modeled with GSM: $\boldsymbol{x}_i = \alpha_i \boldsymbol{u}_i$, where $\boldsymbol{u}_i$ is a zero-mean Gaussian vector, α_i is a mixing multiplier. The general form of GSM allows α_i to be random variable that has a certain distribution in a continuous scale. For computation simplicity, we assume that α_i only takes a fixed value at each block volume (*i.e.* it has different values in each block volume). For notation simplicity, we omit the subscript *i*. In practical scenarios, the noise effect needs to be taken into consideration in our model. Hence, we extend the model of the block volume as:

$$\boldsymbol{p} = \boldsymbol{x} + \boldsymbol{n}_1 = \alpha \boldsymbol{u} + \boldsymbol{n}_1, \tag{1}$$

where $\boldsymbol{n}_1$ is Gaussian noise. Intuitively, each block volume $\boldsymbol{p}$ undergoes spatiotemporal variations $\boldsymbol{v}$ over time, leading to a distorted version $\boldsymbol{q}$:

$$\boldsymbol{q} = \boldsymbol{y} + \boldsymbol{n}_2 = g\alpha \boldsymbol{u} + \boldsymbol{v} + \boldsymbol{n}_2, \tag{2}$$

where $\boldsymbol{y}$ represents deformation of $\boldsymbol{x}$ (*i.e.* $\boldsymbol{y} = g\alpha\boldsymbol{u} + \boldsymbol{v}$), g is a gain factor, and $\boldsymbol{n}_2$ denotes Gaussian noise. $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are independent with covariance matrices $\mathbf{C}_{\boldsymbol{n}_1} = \mathbf{C}_{\boldsymbol{n}_2} = \sigma_n^2 \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. Then, the covariance matrices of $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}$, and $\boldsymbol{q}$ are derived straightforwardly as:

$$\mathbf{C}_{\boldsymbol{x}} = \alpha^2 \mathbf{C}_{\boldsymbol{u}}, \qquad \mathbf{C}_{\boldsymbol{y}} = g^2\alpha^2 \mathbf{C}_{\boldsymbol{u}} + \sigma_v^2 \mathbf{I} \tag{3}$$

$$\mathbf{C}_{\boldsymbol{p}} = \alpha^2 \mathbf{C}_{\boldsymbol{u}} + \sigma_n^2 \mathbf{I}, \qquad \mathbf{C}_{\boldsymbol{q}} = g^2\alpha^2 \mathbf{C}_{\boldsymbol{u}} + \sigma_v^2 \mathbf{I} + \sigma_n^2 \mathbf{I} \tag{4}$$

where $\mathbf{C}_{\boldsymbol{u}}$ is the covariance matrix of $\boldsymbol{u}$.

At each point, the perceived visual information of the reference and deformed block volumes is obtained by the mutual information $\mathcal{I}(\boldsymbol{x}|\boldsymbol{p})$ and $\mathcal{I}(\boldsymbol{y}|\boldsymbol{q})$, respectively. We aim to approximate the perceptual information content from both blocks. To be specific, we subtract the common information shared between $\boldsymbol{p}$ and $\boldsymbol{q}$ ($\mathcal{I}(\boldsymbol{p}|\boldsymbol{q})$)

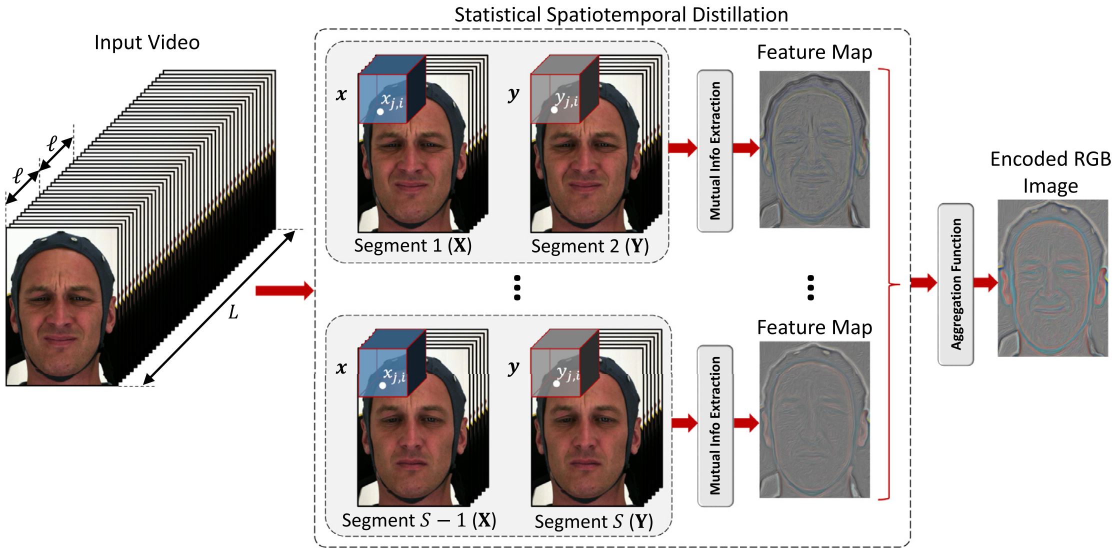

Fig. 1. The outline of our proposed SSD for video representation. Given a video of length L, we divide it into non-overlapping segments of smaller length ℓ. SSD generates feature maps by computing mutual information between block volumes (spatiotemporal neighborhood) of consecutive segments. Using an aggregation function, we obtain an encoded RGB image from the input video.

from $\mathcal{I}(\boldsymbol{x}|\boldsymbol{q})$ and $\mathcal{I}(\boldsymbol{y}|\boldsymbol{q})$. So, we define a weight based on the mutual information as:

$$w = \mathcal{I}(\boldsymbol{x}|\boldsymbol{p}) + \mathcal{I}(\boldsymbol{y}|\boldsymbol{q}) - \mathcal{I}(\boldsymbol{p}|\boldsymbol{q}), \tag{5}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ are all Gaussian for a given α. The mutual information approximation can be computed using the determinants of the covariances. Details for deriving the mutual information weight function in Eq. (5) will be given in the supplementary material.

Let x_i and y_i be the ith points of two frames in consecutive segments $\mathbf{X}$ and $\mathbf{Y}$, respectively. The Mean Square Error (MSE) between two frames is given by $\mathrm{MSE} = \frac{1}{P}\sum_{i=1}^{P}(x_i - y_i)^2$, where P is the number of pixels in the frame. For each of the three color channels, we compute a set of weights by moving a sliding voxel across two consecutive segments (see Fig. 1), where the spatial size of voxel is $H \times W$ pixels. This process results in a feature map for each two consecutive segments of the video. We define a weighted MSE for the corresponding location of the central point in the spatial neighborhood using w defined in Eq. (5). Suppose $x_{j,i}$ and $y_{j,i}$ are the ith points at the jth frame and $w_{j,\,i}$ is the weight computed at the corresponding location. The proposed Statistical Spatiotemporal Distillation (SSD) is defined as:

$$\mathrm{SSD}(\mathbf{X}, \mathbf{Y}) = \prod_{j=1}^{\ell}\left(\frac{\sum_{i=1}^{P} w_{j,i}\big(x_{j,i} - y_{j,i}\big)^2}{\sum_{i=1}^{P} w_{j,i}}\right), \tag{6}$$

where ℓ is the length of each segment of the video.

Repeating this process for all two consecutive segments, for a given video of length L, we can obtain $\lfloor L/\ell \rfloor - 1$ feature maps per color channel, which encode the appearance and dynamic variations within video segments. This distilled information can not be used as the input of pretrained CNN models due to multiple channels. To tackle this issue, we use a weighted aggregation approach (*i.e.* weighted sum, for simplicity) to generate a single RGB image from the obtained $\lfloor L/\ell \rfloor - 1$ channel feature maps. The weights are calculated as: $\beta_i = \frac{\exp(e_i)}{\sum_i \exp(e_i)}$, where e_i is the ith point of the image map. We compute the weighted sum of points for each channel separately to generate the RGB distilled representation of the video: $\mathcal{S}_j = \sum_{j=1}^{\lfloor L/\ell \rfloor - 1} \beta_j e_j$, where $\mathcal{S}_j$ denotes the jth point from one channel of the obtained representation.

Furthermore, we can simplify the mutual information weight function in Eq. (5) as:

$$w = \frac{1}{2}\log\left[\frac{|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}|}{\sigma_n^{4K}}\right], \tag{7}$$

where K is the total number of points in a block volume and

$$|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}| = |\big((\sigma_v^2 + \sigma_n^2)\alpha^2 + \sigma_n^2 g^2 \alpha^2\big)\mathbf{C}_{\boldsymbol{u}} + \sigma_n^2\big(\sigma_v^2 + \sigma_n^2\big)\mathbf{I}| \tag{8}$$

Applying an eigenvalue decomposition to the covariance matrix $\mathbf{C}_{\boldsymbol{u}} = \mathbf{O}\Lambda\mathbf{O}^T$, where $\mathbf{O}$ is an orthogonal matrix and Λ is a diagonal matrix with eigenvalues λ_k for $k = 1, \ldots, K$ along its diagonal entries, we can compute $|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}|$ as:

$$|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}| = |\mathbf{O}\big\{\big(\sigma_v^2 + (1+g^2)\sigma_n^2\big)\alpha^2\Lambda + \sigma_n^2\big(\sigma_v^2 + \sigma_n^2\big)\mathbf{I}\big\}\mathbf{O}^T|. \tag{9}$$

Due to the orthogonal property of $\mathbf{O}$, $|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}|$ has a closed-form:

$$|\mathbf{C}_{(\boldsymbol{p},\boldsymbol{q})}| = \prod_{k=1}^{K}\big\{\big(\sigma_v^2 + (1+g^2)\sigma_n^2\big)\alpha^2\lambda_k + \sigma_n^2(\sigma_v^2 + \sigma_n^2)\big\}. \tag{10}$$

Hence, again, the mutual information weight function in Eq. (5) can be expressed as:

$$w = \frac{1}{2}\sum_{k=1}^{K}\log\left(1 + \frac{\sigma_v^2}{\sigma_n^2} + \left(\frac{\sigma_v^2}{\sigma_n^4} + \frac{1+g^2}{\sigma_n^2}\right)\alpha^2\lambda_k\right). \tag{11}$$

The obtained weight function shows an interesting connection with the local deformation within frames of video. According to the deformation model in Eq. (2), the variations from $\boldsymbol{x}$ to $\boldsymbol{y}$ are characterized by the gain factor g and the random deformation σ_v^2. As g is a scale factor along the temporal frame evolution, it does not cause any changes in the image spatial structure. Thus, the changes in image spatial structure are captured by σ_v^2. Our weight function increases monotonically with σ_v^2. This demonstrates that more weights are cast to the areas that have larger variations.

In the derived weight function of Eq. (11), we still need to derive approximation for a set of parameters: $\mathbf{C}_{\boldsymbol{u}}$, α^2, g, and σ_v^2. $\mathbf{C}_{\boldsymbol{u}}$ is computed as:

$$\hat{\mathbf{C}}_{\boldsymbol{u}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T, \tag{12}$$

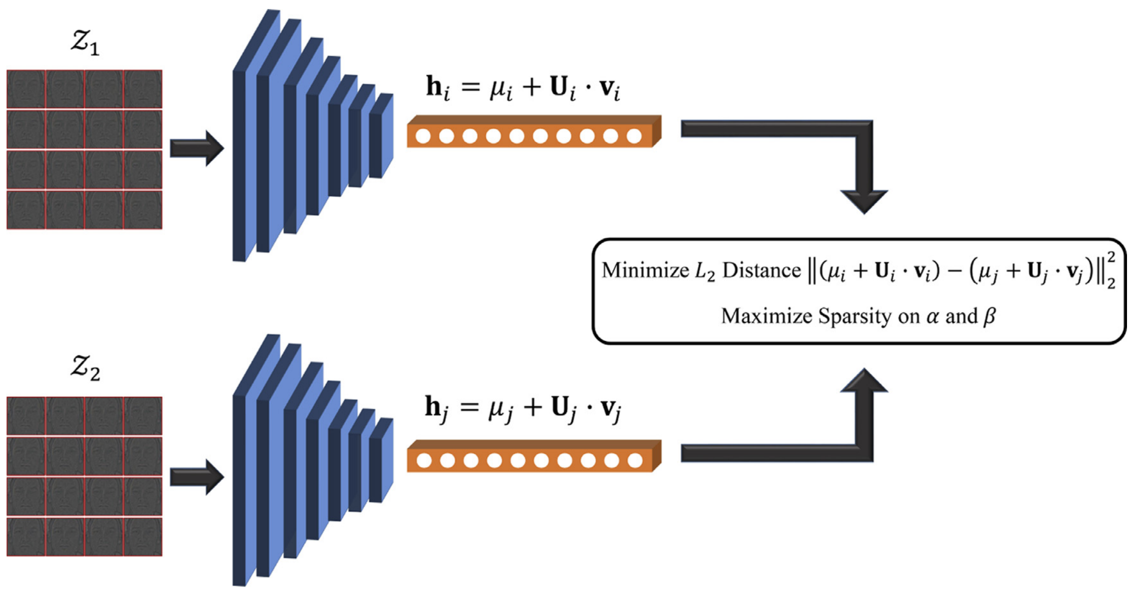

Fig. 2. An illustration of the Siamese network for learning the similarity between samples in an unsupervised manner.

where N is the number of block volumes and $\boldsymbol{x}_i$ is the ith neighboring vector. The multiplier α can be approximated using a maximum likelihood estimator:

$$\hat{\alpha}^2 = \frac{1}{K}\boldsymbol{x}^T \mathbf{C}_{\boldsymbol{u}}^{-1}\boldsymbol{x}. \tag{13}$$

The deformation parameters g and σ_v^2 can be computed via optimizing the following least square regression problem:

$$\hat{g} = \arg\min_{g} \|\boldsymbol{y} - g\boldsymbol{x}\|_2^2, \tag{14}$$

resulting $\hat{g} = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x}}$. Substituting this into Eq. (2), we can compute σ_v^2 using $\mathbf{v}^T\mathbf{v}/K$, which leads to:

$$\hat{\sigma}_v^2 = \frac{1}{K}\left(\boldsymbol{y}^T\boldsymbol{y} - \hat{g}\boldsymbol{x}^T\boldsymbol{y}\right). \tag{15}$$

2.2. Self-supervised representation learning

Self-supervised Learning is formulated as firstly learning a pretext task in an *unsupervised* manner, *i.e.* the supervisory labels for the pretext task are automatically generated based on the data itself rather than human annotations, and after the unsupervised training finished, it transfers the learned representations into a downstream supervised task to evaluate the quality of the learned representations. In our problem, we define the pretext task as learning a general similarity function for sample pairs. In the downstream task, we define the downstream task as supervised classification of different pain intensity levels. To encode such function, we propose a method based on Siamese neural networks (Fig. 2). We denote a batch of unlabeled data as $\mathcal{Z} = \{\mathbf{Z}_i\}_{i=1}^N$, where $\mathbf{Z}_i$ is a sample instance. Feeding these data into a CNN model $f(\cdot)$, we obtain their deep representations from the last fully connected layer. Then, we define a representation matrix $\mathbf{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N]$, where $\boldsymbol{h}_i = f(\mathbf{Z}_i)$ is the extracted deep representation of the ith sample in the batch.

We propose a joint representation for samples in each batch. The joint representation benefits from constructing a linear model to approximate the structure of each batch in a high-dimensional feature space. We can model a batch as an affine hull of its samples. The affine hull model is used to account for unseen samples in terms of affine combinations of existing samples. Hence, we model a batch as:

$$AH = \left\{ \sum_{i=1}^{N} \alpha_i \cdot \boldsymbol{h}_i \middle| \sum_{i=1}^{N} \alpha_i = 1 \right\}, \tag{16}$$

It is possible to represent the affine hull of Eq. (16) by another parametric form using batch mean $\boldsymbol{\mu}$ as a reference point to represent every data:

$$AH = \left\{ \boldsymbol{\mu} + \mathbf{U} \cdot \boldsymbol{v} | \boldsymbol{v} \in R^d \right\}, \tag{17}$$

where the d columns of $\mathbf{U}$ are the orthogonal bases obtained from singular value decomposition (SVD) of the centered deep representation matrix. We present a batch of samples as a triplet ($\boldsymbol{\mu}$**, U, H**) by including both the structure information and deep representations. The information of deep representation is used to reduce the ambiguity of the affine hull space.

Similar samples can be noisy or vulnerable to outliers. Hence, the direct search between nearest neighbors degrades the performance because it is possible to find a pair of similar samples with very small distance that represent different pain intensity levels. To overcome this limitation, we measure the dissimilarity between samples so that the Euclidean distance between them is small and samples of each class could be represented by a combination of its neighborhood points in the deep feature space.

Considering the above point, we propose a convex optimization formulation to define the dissimilarity metric. Given the deep representation matrix of each batch $\mathbf{H}$, we evenly split it into two matrices $\mathbf{H}_i$ and $\mathbf{H}_j$. Therefore, their corresponding affine hull representations are $(\mu_i, \mathbf{U}_i)$ and $(\mu_j, \mathbf{U}_j)$. To the define the objective function for the convex optimization, we define a series of functions as follows:

$$F_{\boldsymbol{v}_i, \boldsymbol{v}_j} = |(\mu_i + \mathbf{U}_i \cdot \boldsymbol{v}_i) - \left(\mu_j + \mathbf{U}_j \cdot \mathbf{v}_j\right)|_2^2, \tag{18}$$

$$G_{\boldsymbol{v}_i, \alpha} = |(\mu_i + \mathbf{U}_i \cdot \boldsymbol{v}_i) - \mathbf{H}_i \cdot \alpha|_2^2, \tag{19}$$

$$Q_{\boldsymbol{v}_j, \beta} = |\left(\mu_j + \mathbf{U}_j \cdot \boldsymbol{v}_j\right) - \mathbf{H}_j \cdot \beta|_2^2, \tag{20}$$

where the optimal model coefficients $\{\boldsymbol{v}_i^*, \boldsymbol{v}_j^*\}$ and sample coefficients $\{\alpha^*, \beta^*\}$ of our metric are achieved by optimizing the following unconstrained objective function:

$$\min_{\boldsymbol{v}_i, \boldsymbol{v}_j, \alpha, \beta} F_{\boldsymbol{v}_i, \boldsymbol{v}_j} + \lambda_1 \left(G_{\boldsymbol{v}_i, \alpha} + Q_{\boldsymbol{v}_j, \beta}\right) + \lambda_2 \|\alpha\|_1 + \lambda_3 \|\beta\|_1, \tag{21}$$

Table 1
Characteristics of the pain recognition datasets.

Dataset	Subjects	Videos	Stimuli	Classes
UNBC-McMaster	25	200	Shoulder Pain	16
BioVid (Part A)	87	8700	Induced Heat	5

where the first term is to keep the distance between similar samples $\boldsymbol{h}_i = \mu_i + \mathbf{U}_i \cdot \boldsymbol{v}_i$ and $\boldsymbol{h}_j = \mu_j + \mathbf{U}_j \cdot \boldsymbol{v}_j$ small. The second term is to preserve the individual fidelity between such samples and their neighbors. The last two terms enforces the coefficients to be sparse. λ_1, λ_2, and λ_3 are trade-off values to control the relative importance of different terms.

3. Experiments

We evaluated the performance of our proposed method by conducting experiments on two publicly available datasets, namely UNBC-McMaster Shoulder Pain Expression Archive [18] and the BioVid Heat Pain - Part A [24]. Table 1 summarizes the content of these datasets.

3.1. Experimental setup

In our experiments, we use four different network architectures, *i.e.* VGG16 [22], Inception-V1 [12], ResNet-18 [9], and ResNet-50 [9]. All of these models are trained using stochastic gradient descent with the momentum of 0.9 and an annealed learning rate, starting from 10^{-3} and multiplied by a factor of 0.2 per epoch. During the training, we randomly perform size jittering, cropping, flipping, and rescaling on the input samples. We also applied our SSD on optical flow data. For the computation of the optical flow, we use TLV1 optical flow algorithm [31], which is implemented in OpenCV with CUDA.

The value of λ_1 is fixed as 0.01 for all the experiments conducted in this paper. For λ_2 and λ_3, we design an automatic mechanism to control the relative sparsity of α and β. Note that if $\lambda_2 \geq \max(|2\lambda_1 \cdot (\mathbf{H}^T\mu)|)$, the zero vector is optimal for α at zero. The same statement applies for λ_3 and β. By performing a grid search and following the guidelines in [1], we adaptively set $\lambda_2 = 0.1\lambda_2^*$ and $\lambda_3 = 0.1\lambda_3^*$ for all experiments.

3.2. Analysis of SSD representations

In this section, we evaluate the performance of our method by analyzing different parameters in SSD representations. As described in Section 2.1, SSD divides the video into smaller fix-sized non-overlapping segments. The spatiotemporal patch-wise statistical analysis between two consecutive segments allows SSD to capture and encode the appearance and dynamic variations into one single RGB image map. Hence, the performance of SSD depends on two parameters, *i.e.* the length of segments and the spatial neighborhood around each point. We analyze the performance of our method by varying the length of video segments from 5 to 60 frames. Fig. 3 shows the results of experiments using the UNBC-McMaster dataset. As depicted, the Area Under the Curve (AUC) accuracy increases as the number of frames per segment increases. However, after a certain number of frames (20 frames per segment), the performance drops significantly. The deterioration in performance is likely due to capturing a wide range of temporal information from the face, which covers different pain intensity levels throughout the segment. Hence, SSD is not able to effectively encode the appearance and dynamic of the sequence.

Another important parameter, which influences the quality of obtained representations using SSD, is the spatial size of the block volumes. We change the spatial size of neighborhood around each point of the video segments to investigate its effect. Fig. 3 demonstrates the results of this experiment. Using the small spatial windows, the receptive field of SSD is also small. Hence, the statistical operation is performed in a tiny region of the video, which ignores most important relationships between neighboring points. However, choosing a large size for spatial patches leads to low accuracy. We argue that this drop in the performance is due capturing too much information and increasing the complexity of the obtained image map. Based on the results of this experiment, we will use video segments of 20 frames with the spatial size of 5×5 in the rest of our experiment, unless otherwise it is mentioned.

Table 2
The AUC accuracy (%) of our proposed method versus the accuracy of representations from optical flow and RGB data using ResNet-50.

Data Type	UNBC-McMaster	BioVid
RGB	85.9	69.4
OF	86.2	69.8
SSD (RGB)	88.5	71.0
SSD (OF)	86.6	70.3

In order to gain a better insight into the effect of segment length on SSD, we computed SSD representations using different segment lengths in Fig. 4. We varied the number of frames per segment from 5 to 60 frames. As can be seen, the smaller segment, lower spatiotemporal variations. In contrast, SSD representations obtained from longer segments (*e.g.* 60 frames) show noisy behavior, which interferes the distilled spatiotemporal information. Based on Eqs. (1) and (2), enlarging the temporal size of segments increases the spatiotemporal information to noise ratio, while decreasing the temporal size of segments limits SSD computation to a small portion of the video. Hence, it is important to select sufficiently enough number of frames per segments for SSD computation. This qualitative analysis of SSD representations further validates the quantitative results in Fig. 3.

We compared our proposed SSD representation against two frame-based baseline models. The first model operates on RGB images and the second model works on the optical flow. We also applied SSD on the optical flow sequence and the RGB video. The results are summarized in Table 2. The proposed SSD has 2.3% and 1.2% improvement over optical flow on the UNBC-McMaster and the BioVid datasets, respectively, indicating the superiority of SSD in capturing the appearance and dynamic of the video.

3.3. 2D Models vs. 3D models

The proposed SSD does not require any parameter learning process. It captures appearance and dynamic information of the video and encodes it into one RGB image. This enables its output representations to be processed by models devised for images. Consequently, the need for complex 3D models and large amount of annotated training videos can be avoided. Although extracting the distilled representations adds overhead to training of 2D models, this time is negligible when compared with the training time of 3D models. For instance, the training time of ResNet-50 on RGB frames of the UNBC-McMaster is 1257 min, which is raised to 1326 min by applying SSD. However, our method is more efficient in the test time. To demonstrate this efficiency, we randomly selected a set of 50 videos from the test set of the UNBC-McMaster dataset. The videos have lengths in range of 150 – 200 frames. We, then, compare the runtime of our method using 2D models against their 3D counterparts in Table 3, where we replace 2D filters of CNNs by 3D filters. From Table 3, we conclude that encoding information via SSD enables us to estimation pain intensity levels by using less

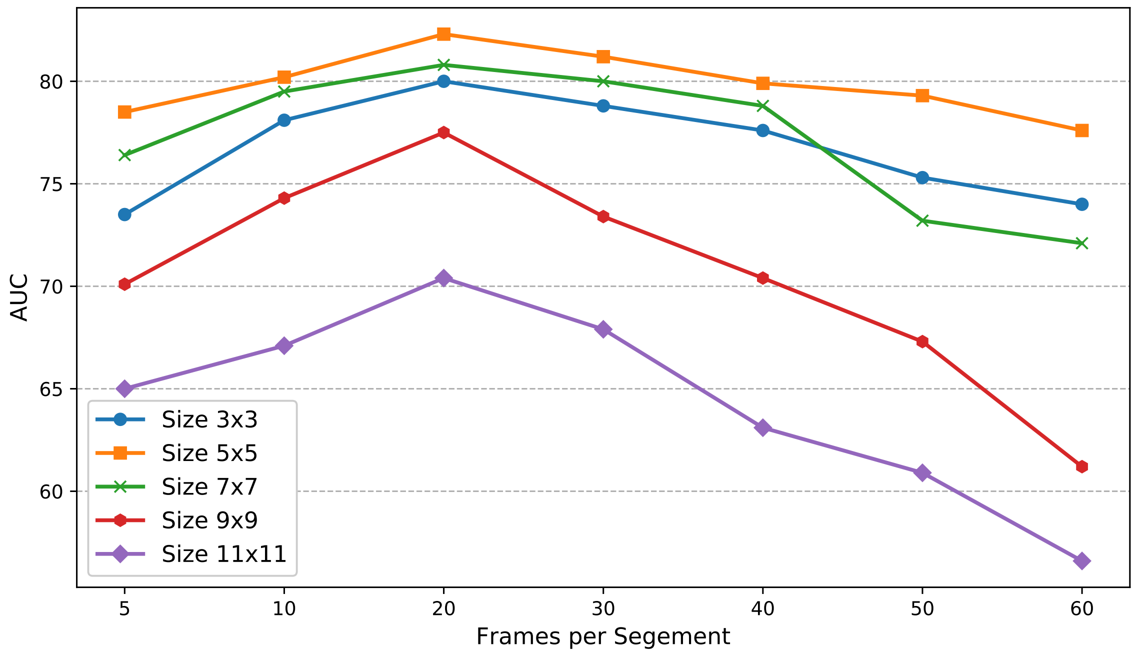

Fig. 3. Accuracy (%) of the proposed method by varying the segment length size and the spatial size of neighborhood for SSD representation computation using ResNet-50.

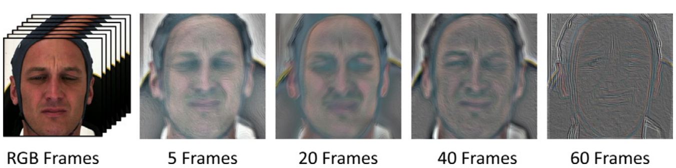

Fig. 4. Visualization of SSD representations versus different segment lengths. Small segments do not cover wide range of spatiotemporal variations, while too long segments have more noise.

Table 3
Average testing runtime (sec.) per frame and accuracy (%) on the UNBC-McMaster.

Architecture	2D		3D	
Method	Time	AUC	Time	AUC
VGG16	0.924	75.3	2.769	77.1
Inception-V1	0.985	78.5	2.854	78.8
ResNet-18	1.085	80.5	2.866	79.3
ResNet-50	1.299	83.2	2.892	81.5

complex models devised for images and still achieve high performance.

3.4. Cross database analysis

The performance analysis of automatic pain assessment methods in the cross-database setting is undermined. The existing approaches' performance degrades when the training and testing datasets are different [20]. This is mainly due to lack of generalization in the learned representations. In this section, we conduct cross-dataset experiments to show the generalization of the learned representations in our self-supervised framework. In these experiments, the model learns the pretext task using unlabeled data (SSD representations of videos) of either the UNBC-McMaster, BioVid, or their combination and is fine-tuned and tested using labeled SSD representations of the other dataset. We combined different portion of the target data with the training data for learning the pretext task to analyze how the performance is influenced using combined datasets. Table 4 summarizes the results. As can be seen, the model achieves reasonably good performance in self-supervised fashion and benefits from a larger dataset in the pretext task learning. However, including a portion of target data to the training data of the pretext task can slightly improve the performance.

Table 4
The accuracy (%) of the proposed self-supervised learning method in cross-dataset setting. ResNet-50 is used as the backbone of our model.

Pretext Data	Downstream Data	AUC
BioVid	UNBC-McMaster	69.2
UNB-McMaster	BioVid	65.5
BioVid + UNBC-McMaster (10%)	UNBC-McMaster	71.3
BioVid + UNBC-McMaster (30%)	UNBC-McMaster	73.5
BioVid + UNBC-McMaster (50%)	UNBC-McMaster	74.2
UNBC-McMaster + BioVid (10%)	BioVid	64.4
UNBC-McMaster + BioVid (30%)	BioVid	69.0
UNBC-McMaster + BioVid (50%)	BioVid	71.8

Table 5
Comparison of supervised pain intensity estimation using SSD representations in the cross-dataset experimental setting using ResNet-50.

Pretext Data	Downstream Data	AUC
BioVid	UNBC-McMaster	80.1
UNB-McMaster	BioVid	75.5
BioVid + UNBC-McMaster (10%)	UNBC-McMaster	81.2
BioVid + UNBC-McMaster (30%)	UNBC-McMaster	82.3
BioVid + UNBC-McMaster (50%)	UNBC-McMaster	83.5
UNBC-McMaster + BioVid (10%)	BioVid	76.3
UNBC-McMaster + BioVid (30%)	BioVid	78.2
UNBC-McMaster + BioVid (50%)	BioVid	79.0

To gain a better insight in effectiveness of our proposed self-supervised method, we conduct complementary cross-dataset experiments in a supervised manner. In these experiments, we train the CNN model using SSD representations of one dataset (with labels) and, then, use the SSD representations of other dataset for testing. Table 5 lists the results of this experiment. Although the accuracy drops in the cross-dataset setting, the SSD still shows a

Table 6
Comparative analysis in terms of the Mean Square Error (MSE) and the Pearson Correlation Coefficient (PCC) on the UNBC-McMaster dataset using RGB and SSD representations. ResNet-50 is used as the backbone of our method.

Method		RGB		SSD	
		MSE	PCC	MSE	PCC
Supervised	RVR [14]	1.39	0.59	1.15	0.63
	HoT [7]	1.21	0.53	0.98	0.66
	OSVR [32]	N/A	0.60	1.06	0.63
	RCNN [33]	1.54	0.64	1.27	0.73
	LSTM [21]	0.74	0.78	0.69	0.80
	SCN [23]	0.32	0.92	N/A	N/A
Self-supervised	Geometry [8]	1.81	0.51	1.65	0.57
	OPN [15]	1.76	0.57	1.60	0.61
	Jigsaw [19]	1.73	0.68	1.58	0.70
	[25]	1.59	0.65	1.33	0.70
	[4]	1.47	0.71	1.12	0.73
	Our Method	1.03	0.74	0.92	0.78

Table 7
Comparative analysis in terms of the AUC accuracy (%) on the BioVid dataset. ResNet-50 is used as the backbone of our method.

Method		RGB	SSD
Supervised	Head-movement [24]	67.00	70.36
	Time-windows [24]	71.00	72.15
	LBP [30]	63.72	65.20
	BSIF [30]	65.17	67.83
	FAD set [28]	72.40	74.59
	SCN [23]	86.02	N/A
Self-supervised	Geometry [8]	61.29	62.50
	OPN [15]	63.07	64.83
	Jigsaw [19]	63.44	65.19
	[25]	65.23	68.41
	[4]	66.01	69.19
	Our Method	69.35	71.02

good performance, demonstrating its high capability to represent video sequences discriminatively. From the results of Tables 4 and 5, we notice that there is still a gap in performance between supervised and self-supervised techniques. However, the cross-dataset experiments show promising results by using SSD representations.

3.5. Comparison against the state-of-the-art

We compare the performance of our proposed method with the-state-of-the-art methods for pain intensity estimation on the UNBC-McMaster [18] and the BioVid [24] datasets. In these experiments, VGGFace2 dataset [3] is used as unlabeled data for learning the pretext task. We conduct experiments using the original implementation of supervised and self-supervised approaches on both RGB and SSD representations. To make a direct and fair comparison, we report the Mean Square Error (MSE) and the Pearson Correlation Coefficient (PCC) for the UNBC-McMaster dataset. Table 6 summarizes the comparative results on the UNBC-McMaster dataset. We observe that the proposed method improves the performance of self-supervised pain intensity estimation using either RGB or SSD data following leave-one-subject-out cross-validation. Our method achieve 1.03 and 0.92 MSE using RBG and SSD data, respectively. It also improves PCC among self-supervised methods. We assert that this improvement is due to learning generalized representations in the pretext task thanks to the proposed similarity function, which push dissimilar samples away from each other. However, there is still a margin between the performance of the self-supervised and the supervised approaches in Table 6.

Table 7 draws comparison against the state-of-the-art methods on the BioVid dataset. We use part A of this dataset, which contains only unoccluded facial videos. As can be seen, our method achieves 69.35% and 71.02% AUC accuracy using RGB and SSD data, respectively, and improves the highest performance of self-supervised techniques by 3.34% and 1.83%, showing results comparable to some of the supervised techniques.

4. Conclusion

In this paper, we proposed a novel self-supervised representation learning framework for automatic pain intensity estimation. By learning a similarity function on a large portion of unlabeled samples using a Siamese network as the pretext task, we achieved generalized representations of data in an unsupervised manner. The learned representations were further transferred to the downstream supervised task, where the CNN model is fine-tuned using a small subset of labeled samples. In order to reduce the complexity of the learning process, we presented a statistical spatiotemporal distillation technique to capture and encode the appearance and dynamic of the video into one single RGB image. Hence, we could use 2D models to process video data. We conducted extensive experiments on two publicly available datasets to demonstrate the effectiveness of our proposed method. The experimental results of the cross-dataset pain intensity estimation validated our initial hypothesis of the generalization of the learned representations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to acknowledge the financial support of the Academy of Finland (No. 331883), Infotech Oulu, Tauno Tönning, Nokia, and KAUTE foundations.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2020.09.012.

References

[1] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 437–478.
[2] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a “siamese” time delay neural network, in: NIPS, 1994, pp. 737–744.
[3] Q. Cao, L. Shen, W. Xie, O.M. Parkhi, A. Zisserman, VGGFace2: a dataset for recognising faces across pose and age, FG, 2018.
[4] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: ECCV, 2018, pp. 132–149.
[5] M.N. Dailey, C. Joyce, M.J. Lyons, M. Kamachi, H. Ishi, J. Gyoba, G.W. Cottrell, Evidence and a computational explanation of cultural differences in facial expression recognition, Emotion 10 (6) (2010) 874–893.
[6] P. Ekman, W.V. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movements, Consulting Psychologist Press, 1978.
[7] C. Florea, L. Florea, C. Vertan, Learning pain from emotion: Transferred HoT data representation for pain intensity estimation, in: ECCV Workshops, 2014, pp. 778–790.
[8] C. Gan, B. Gong, K. Liu, H. Su, L.J. Guibas, Geometry guided convolutional neural networks for self-supervised video representation learning, in: CVPR, 2018, pp. 5589–5597.
[9] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[10] S. Hossein Khatoonabadi, N. Vasconcelos, I.V. Bajic, Y. Shan, How many bits does it take for a stimulus to be salient? in: CVPR, 2015, pp. 5501–5510.
[11] J. Hu, J. Lu, Y.-P. Tan, Discriminative deep metric learning for face verification in the wild, in: IEEE CVPR, 2014, pp. 1875–1882.
[12] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).
[13] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: a survey, IEEE PAMI (2020).

[14] S. Kaltwang, O. Rudovic, M. Pantic, Continuous pain intensity estimation from facial expressions, in: Advances in Visual Computing, 2012, pp. 368–377.
[15] H.-Y. Lee, J.-B. Huang, M. Singh, M.-H. Yang, Unsupervised representation learning by sorting sequences, in: ICCV, 2017, pp. 667–676.
[16] G.C. Littlewort, M.S. Bartlett, K. Lee, Automatic coding of facial expressions displayed during posed and genuine pain, Image Vis. Comput. 27 (12) (2009) 1797–1803.
[17] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, S. Chew, I. Matthews, Painful monitoring: automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database, Image Vis. Comput. 30 (3) (2012) 197–205.
[18] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, I. Matthews, Painful data: the UNBC-McMaster shoulder pain expression archive database, FG, 2011.
[19] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving Jigsaw puzzles, in: ECCV, 2016, pp. 69–84.
[20] E. Othman, P. Werner, F. Saxen, A. Al-Hamadi, S. Walter, Cross-database evaluation of pain recognition from facial video, in: ISPA, IEEE, 2019, pp. 181–186.
[21] P. Rodriguez, G. Cucurull, J. Gonalez, J.M. Gonfaus, K. Nasrollahi, T.B. Moeslund, F.X. Roca, Deep pain: exploiting long short-term memory networks for facial expression classification, IEEE Trans. Cybern. (2017) 1–11.
[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).
[23] M. Tavakolian, A. Hadid, A spatiotemporal convolutional neural network for automatic pain intensity estimation from facial dynamics, IJCV 127 (10) (2019) 1413–1425.
[24] S. Walter, S. Gruss, H. Ehleiter, J. Tan, H.C. Traue, S. Crawcour, P. Werner, A. Al-Hamadi, A.O. Andrade, The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system, in: Int. Conf. Cybernetics, 2013.
[25] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, W. Liu, Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics, in: CVPR, 2019, pp. 4006–4015.
[26] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Trans. PAMI (2018).
[27] X. Wang, X. Wang, Y. Ni, Unsupervised domain adaptation for facial expression recognition using generative adversarial networks, Comput. Intell. Neurosci. 2018 (2018) 1–10.
[28] P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, S. Gruss, H.C. Traue, Automatic pain assessment with facial activity descriptors, IEEE Trans. Affect. Comput. 8 (3) (2017) 286–299.
[29] H. Yan, Transfer subspace learning for cross-dataset facial expression recognition, Neurocomputing 208 (2016) 165–173.
[30] R. Yang, S. Tong, M. Bordallo, E. Boutellaa, J. Peng, X. Feng, A. Hadid, On pain assessment from facial videos using spatio-temporal local descriptors, IPTA, 2016.
[31] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime TV-L 1 optical flow, in: Joint Pattern Recognition Symposium, 2007, pp. 214–223.
[32] R. Zhao, Q. Gan, S. Wang, Q. Ji, Facial expression intensity estimation using ordinal information, in: CVPR, 2016, pp. 3466–3474.
[33] J. Zhou, X. Hong, F. Su, G. Zhao, Recurrent convolutional neural network regression for continuous pain intensity estimation in video, in: CVPR Workshops, 2016, pp. 84–92.