



A Spatiotemporal Convolutional Neural Network for Automatic Pain Intensity Estimation from Facial Dynamics

Mohammad Tavakolian¹ · Abdenour Hadid¹

Received: 12 February 2018 / Accepted: 16 June 2019 / Published online: 25 June 2019
© The Author(s) 2019

Abstract

Devising computational models for detecting abnormalities reflective of diseases from facial structures is a novel and emerging field of research in automatic face analysis. In this paper, we focus on automatic pain intensity estimation from faces. This has a paramount potential diagnosis values in healthcare applications. In this context, we present a novel 3D deep model for dynamic spatiotemporal representation of faces in videos. Using several convolutional layers with diverse temporal depths, our proposed model captures a wide range of spatiotemporal variations in the faces. Moreover, we introduce a cross-architecture knowledge transfer technique for training 3D convolutional neural networks using a pre-trained 2D architecture. This strategy is a practical approach for training 3D models, especially when the size of the database is relatively small. Our extensive experiments and analysis on two benchmarking and publicly available databases, namely the UNBC-McMaster shoulder pain and the BioVid, clearly show that our proposed method consistently outperforms many state-of-the-art methods in automatic pain intensity estimation.

Keywords Deep learning · Convolutional neural network · Facial dynamics · Pain intensity estimation · Cross-architecture knowledge transfer · Healthcare

1 Introduction

Pain is among vital indicators of our health condition. It can be defined as a highly unpleasant sensation which is caused by diseases, injuries, or mental distress. Pain is often considered as the fifth vital sign in disease diagnosis (Lynch 2001). Chronic pain can carry a wide array of pathophysiological risks. Pain is usually reported by patients themselves (self-report), either in clinical inspection or using Visual Analog Scale (VAS) (Lesage et al. 2012). Pain assessment based on the self-report is however highly subjective, and cannot be used for population that are incapable of articulating their pain experiences (Brahnam et al. 2006; Werner et al. 2013). Technologies that automatically recognize such a state from the facial patterns of a patient can be extremely powerful, both diagnostically and therapeutically. Automatic pain expression detection has indeed an important potential diag-

nostic value, especially for populations, such as neonates and post-surgery patients, that are incapable of articulating their pain experiences (Brahnam et al. 2006; Werner et al. 2013). At present, health professionals must infer pain in individuals by examining various physiological and behavioural indicators that are strongly associated with pain. Face analysis is particularly relevant in pain assessment, since research has shown that facial expressions of pain provide the most reliable and accurate source of information regarding a subject's health condition. However, people exhibit an increase in pain behavior in the presence of health practitioners (Flor et al. 1995); i.e. subjects who experience pain tend to exaggerate their pain expressions in order to attract more attention. In order to tackle such issues, developing an unbiased solution for pain assessment is crucial.

A potential approach to automatic pain assessment is through the use of facial expression analysis. The human face is indeed a rich source for non-verbal information regarding the health condition (Thevenot et al. 2017). Facial expression can be considered as a reflective and spontaneous reaction of painful experiences (Craig et al. 2011). Most studies on facial expression are based on the Facial Action Coding System (FACS) (Ekman and Friesen 1978). FACS is a system

Communicated by Xiaou Tang.

✉ Mohammad Tavakolian
mohammad.tavakolian@oulu.fi

¹ Center for Machine Vision and Signal Analysis (CMVS),
University of Oulu, Oulu, Finland

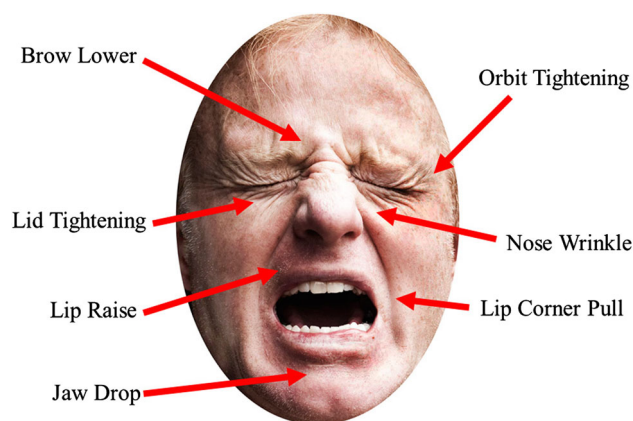


Fig. 1 An example of facial action coding

for objectively scoring facial expressions in terms of elementary facial movements, called Action Units (AUs). Each AU is coded with onset, offset, and an intensity on a five-point scale. Figure 1 shows an example of a facial pain expression that is coded in terms of seven component facial actions based on FACS.

Most of previous works on automatic pain assessment focused on extracting features from consecutive frames of face videos to detect and measure the intensity of pain—great progress has been achieved in these endeavors (Brahnam et al. 2006; Littlewort et al. 2009; Lucey et al. 2012). However, traditional image descriptors represent video frames based on static features, hence limiting their ability in encoding rich dynamic information required for pain intensity estimation. Some pioneering approaches have addressed the challenge of relevant spatiotemporal representation in the context of pain expression recognition (Rodriguez et al. 2017; Werner et al. 2017). These methods rely on exploiting fixed-range temporal information from videos. However, facial expression variations comprise short, mid, and long-range terms.

Based on the above observations, we strive to take an important step towards the goal of robust automatic pain assessment by introducing a novel 3D convolutional model. Our primary challenge is to develop a model that exploits facial dynamics using both the appearance and the motion information. The model should preferably be free of assumptions about the length of the videos and templates while learns embedded spatiotemporal information effectively in an end-to-end fashion. The second challenge is related to the need of a large amount of annotated training data to achieve good performance in video representation using deep models. Moreover, training on large databases is difficult and time-consuming. Our core insight is that we can leverage an efficacious transfer learning that bridges the knowledge transfer between different architectures so that there is no need to train the network from scratch. We propose to extend

ResNet architecture (He et al. 2016), which has 2D filters and pooling kernels, to incorporate 3D filters and pooling kernels. Unlike Res3D (Tran et al. 2017), we replace the standard 3D convolutional blocks with 3D convolutional filters of variable depths to capture the appearance and temporal information in short, mid, and long-range terms. We adopt ResNet because residual connections make it possible to train deeper network while minimizing overfitting problems (He et al. 2016). We call our proposed model *Spatiotemporal Convolutional Network (SCN)*.

Among our salient contributions in this paper, we can cite:

- (i) We propose a novel 3D deep convolutional neural network that captures both appearance and temporal information in different temporal ranges. The model learns the spatiotemporal representation of facial pain expression throughout the SCN (Spatiotemporal Convolutional Network) architecture and is trained end-to-end.
- (ii) We introduce and develop a cross-architecture knowledge transfer technique to train our 3D deep model, hence avoiding training from scratch. Our extensive analysis demonstrates that a 2D pre-trained model on a large database can be used in a transfer learning process for stable parameter initialization of a 3D model.
- (iii) We extensively validate the effectiveness of our proposed method on automatic pain intensity estimation using two benchmark and publicly available databases.

2 Related Work

In the recent years, there has been a considerable interest in automatic pain assessment from facial patterns. The existing works can be broadly divided into two categories: *determining the presence of pain* versus *measuring the intensity of pain*. Early studies tend to design models that automatically distinguish pain from no-pain (Ashraf et al. 2009; Lucey et al. 2011a; Hammal and Kunz 2012). For instance, Brahnam et al. (2007) exploited Discrete Cosine Transform (DCT) for image description followed by Sequential Forward Selection (SFS) for dimensionality reduction and nearest neighbor for pain classification. Gholami et al. (2010) relied on Relevance Vector Machine (RVM) that is applied on manually selected face images. Guo et al. (2012) used Local Binary Pattern (LBP) and its variants for improving both face description and pain detection accuracy. Ashraf et al. (2009) used Active Appearance Model (AAM) to detect the pain. By using AAM, Lucey et al. (2011a) tracked and aligned faces on manually labeled key-frames and fed them to a support Vector Machine (SVM) classifier for frame-level pain classification.

We note that all the abovementioned methods deal with pain analysis as a binary problem (i.e. pain vs. non-pain).

According to Prkachin and Solomon's pain intensity metric (Prkachin and Solomon 2008), pain can however be classified into several discrete levels. So, the most recent works on automatic pain assessment have focused on the challenging task of estimating the intensity of pain. For instance, Lucey et al. (2012) trained extended SVM classifiers for three-level pain intensity estimation. Kaltwang et al. (2012) extracted LBP and DCT features from facial images and used them and their combinations as appearance-based features. They fed these features to Relevance Vector Regression (RVR) for pain intensity detection. Hammal and Cohn (2012) used Log normal filters to identify four levels of pain. Florea et al. (2014) improved the performance of pain intensity recognition by using a histogram of topographical features and an SVM classifier. Recently, Zhao et al. (2016) proposed an alternating direction method of multipliers to solve Ordinal Support Vector Regression (OSVR).

It appears that the majority of these works (classifying pain into several levels) have mainly been focused on traditional hand-engineered representations which are obtained from individual frames. This yields in indisputable limitations in describing relevant dynamic information often required for accurate pain intensity estimation.

More recently, a few attempts have been made to model temporal information within video sequences by using deep neural networks. For instance, Zhou et al. (2016) proposed a Recurrent Convolutional Neural Network (RCNN) as a regressor model to estimate the pain intensity. They converted video frames into vectors and fed them to their model. However, this spatial conversion results in losing the structural information of the face. Rodriguez et al. (2017) extracted features in each frame from the fully connected layer of a CNN architecture. These features are fed to a Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) to exploit the temporal information. In this way, they consider a temporal relationship between video frames by integrating the extracted features from the CNN.

In order to have an efficient representation of facial videos for pain intensity estimation, it is crucial to simultaneously encode both appearance and temporal information. In recent years, several deep models have been proposed for spatiotemporal representations. These models mainly use 3D filters and pooling kernels with fixed temporal kernel depths. The most intuitive architecture is based on 3D convolutions (Ji et al. 2013) where the depth of the kernels corresponds to the number of frames used as input. Simonyan and Zisserman (2014) proposed a two-stream network, including RGB spatial and optical-flow CNNs. Tran et al. (2015) explored 3D CNNs with a kernel size of $3 \times 3 \times 3$ to learn both the spatial and temporal features with 3D convolution operation. In Tran et al. (2017), Tran *et al.* extended the ResNet architecture with 3D convolutions. Sun et al. (2015) decomposed 3D convolutions into 2D spatial and 1D temporal convolu-

tions. Carreira and Zisserman (2017) proposed to convert a pre-trained Inception-V1 (Ioffe and Szegedy 2015) model into 3D by inflating all the filters and pooling kernels with an additional temporal dimension. They achieved this by replicating the weights of the 2D filters. All these structures have fixed temporal 3D convolution kernel depths throughout the whole architecture. This makes them often incapable of capturing short, mid, and long temporal ranges. We address this problem by incorporating several temporal kernel depths in our proposed architecture for pain intensity estimation.

3 The Proposed Method: Spatiotemporal Convolutional Network

A subject experiencing pain often exhibits spontaneous spatiotemporal variations in his/her face. Our aim is to capture the dynamics of the face that embody most of the relevant information for automatic pain intensity estimation. We extend the residual block of ResNet architecture (He et al. 2016) to 3D convolution kernels with diverse temporal depths. Figure 2 illustrates an overview of our proposed model. By using an identity shortcut connection, the input of each 3D residual block is connected to its output feature maps. The obtained feature maps are fed to the subsequent block. We use bottleneck building block to make the training process of deep networks more efficient (He et al. 2016). Additionally, we adopt cross-architecture knowledge transfer (2D to 3D CNNs) to avoid cumbersome training of 3D CNNs from scratch.

3.1 Spatiotemporal Convolutional Network

ResNet architecture (He et al. 2016) uses 2D convolutional filters and pooling kernels. In our present work, we introduce an extended ResNet architecture that uses 3D convolutional filters and pooling kernels. The motivations behind adopting the ResNet architecture include the compact structure, the ability of training deep networks without overfitting thanks to residual connections, and the state-of-the-art performance on visual classification tasks. We develop a Spatiotemporal Convolutional Network (SCN) by introducing several 3D convolutional kernels of different temporal depths in the bottleneck building block instead of the residual building blocks in the 3D ResNet architecture (Res3D) (Tran et al. 2017). Figure 3 depicts our proposed bottleneck building block. As can be seen from the block diagram in Fig. 3, the bottleneck block comprises two 3D convolutional layers with a fixed temporal depth and several 3D convolutional layers with variable temporal depths. The depth of the 3D convolutional kernels ranges within $t \in \{t_1, \dots, t_K\}$. Rather than capturing fixed temporal range homogeneously, our proposed bottleneck is

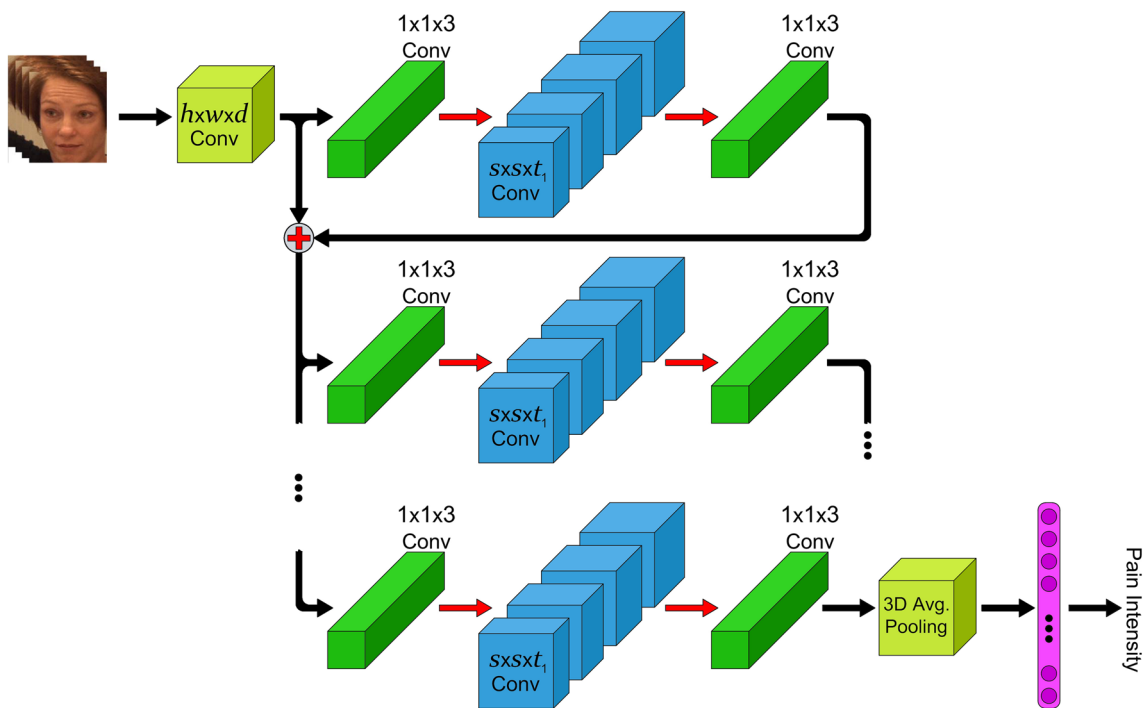


Fig. 2 An overview of our proposed spatiotemporal convolutional network (SCN). The input of each 3D bottleneck block is connected to its output by an identity shortcut connection

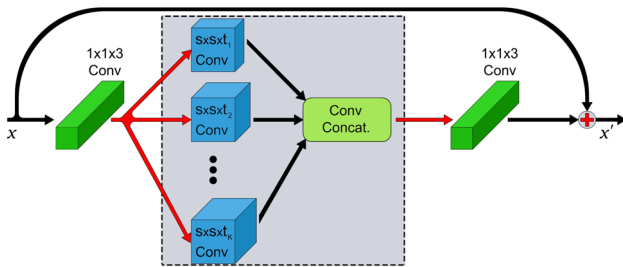


Fig. 3 The proposed bottleneck building block. The red arrows represent ReLu non-linearity (Color figure online)

able to capture a wide range of dynamics that encode complementary information for video representation.

The output feature maps of the 3D convolutions and pooling kernels at the l th layer extracted from an input video is a tensor $x \in R^{H \times W \times C}$, where H , W , and C are the height, width, and the number of channels of the feature maps, respectively. The 3D convolution and pooling kernels are of size $s \times s \times t$, where s is the spatial size of the kernels and t denotes the temporal depth.

Similar to the standard ResNet connectivity, we consider a bottleneck building block defined as:

$$x' = F(x, \{W_1, W_2, W_{t_k}\}) + x \quad (1)$$

where x and x' are the input and output of the layer, respectively. The function $F(x, \{W_1, W_2, W_{t_k}\})$ represents the

residual mapping to be learned. $\{W_1, W_2\}$ are the weight parameters of the first and last convolutional layers in the block and $\{W_{t_k}\}_{k=1}^K$ denote the weight parameters of middle convolutional layers. According to Fig. 3, the function F is defined as

$$F(x) = W_2 \sigma \left(\bigcup_{k=1}^K W_{t_k} \sigma (W_1 x) \right) \quad (2)$$

where \bigcup stands for concatenation operation and σ denotes ReLu non-linearity.

Within each bottleneck building block, after convolving the feature map of the preceding layer, x , with a $1 \times 1 \times 3$ convolution kernel, K parallel 3D convolutions with different temporal depths are applied on it, resulting K intermediate feature maps $\{S_1, S_2, \dots, S_K\}$, where $S_k \in R^{H \times W \times C_k}$. It should be noted that each intermediate feature map has different number of channels as they are obtained by convolution operations of different temporal depths, while the spatial size of all feature maps $H \times W$ is the same. These intermediate feature maps $\{S_k\}_{k=1}^K$ are simply concatenated into a single tensor and then fed into a $1 \times 1 \times 3$ convolutional layer. By using a shortcut connection and element-wise addition, the output feature map of the bottleneck block, $x' \in R^{H \times W \times C'}$, is computed. We employ $1 \times 1 \times 3$ convolutional kernels at the beginning and the end of bottleneck building blocks to perform a feature pooling operation and control the temporal

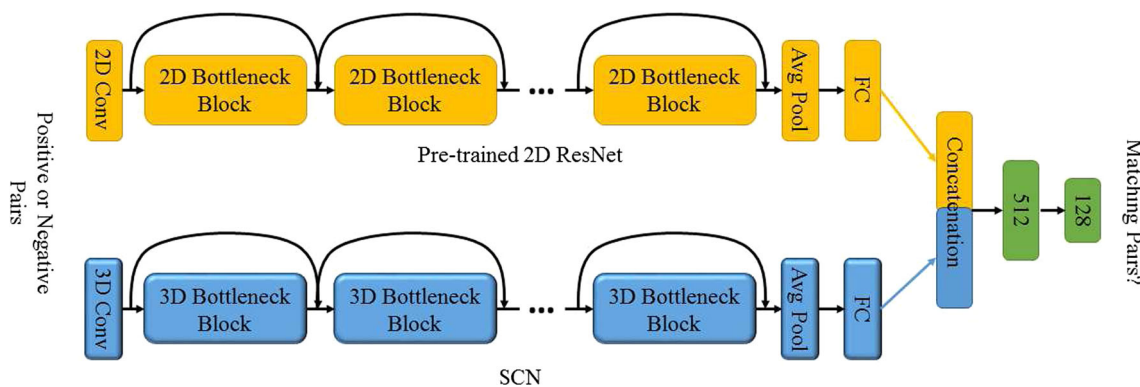


Fig. 4 The cross-architecture knowledge transfer architecture. The 2D pre-trained model deals with frames and the 3D model operates on video sequences of the same time stamp. The 3D model learns mid-level representations by image-video correspondence task

depth of the intermediate feature maps. 1×1 convolutions act as coordinate-dependent transformation in the filter space. So, $1 \times 1 \times 3$ convolution operations allow the network to go deeper and simultaneously reduce the dimensions inside the bottleneck building block. As shown in Fig. 2, the bottleneck building blocks are learned in an end-to-end network training.

Provided that the number of input and output channels are not equal, we can reformulate Eq. (1) by applying a linear projection to match the temporal dimensions.

$$x' = F(x, \{W_1, W_2, W_{t_k}\}) + W_s x \tag{3}$$

For pain intensity estimation, the model should be able to make a continuous-valued prediction. Therefore, instead of cross-entropy loss function which is widely used for classification in deep architectures, we use the mean squared error function to solve the regression problem. We calculate the Euclidean distance between the predicted output \hat{y} and the actual one y to determine the error. The training is carried out using Stochastic Gradient Descent (SGD) and backpropagation.

$$E = \frac{1}{N} \sum_{n=1}^N \|\hat{y}_n - y_n\|_2^2 \tag{4}$$

3.2 Cross-Architecture Knowledge Transfer

Training 3D deep models usually takes a lot of time due to the large number of parameters. In this section, we describe how to avoid training 3D CNNs from scratch by using cross-architecture knowledge transfer, i.e. pre-trained 2D CNN to 3D CNN. Suppose there is a pre-trained 2D model which has learned a rich representation from static images, while a 3D model is randomly initialized following the procedure in He et al. (2015). We aim to transfer substantial knowledge from 2D model to its 3D counterpart for an appropriate weight

initialization. The training of 3D CNN may fail provided that the model is initialized with inappropriate weights. It is worth noting that weight initialization plays an important role in the network convergence. Specifically talking, arbitrary weight initialization may compromise or slowdown the convergence, e.g. getting the network stuck in local minima.

Given N frames as an image sequence and a video showing pain for the same time stamp, the appearance information in both the frames and video sequence are similar. To build the image sequence, we randomly sample frames for the training videos in which they have the same level of pain intensity. The number of frames in the image sequence is proportional to the length of the input video to the 3D model. We leverage this for learning mid-level feature representation by image-video correspondence task between 2D and 3D model architectures (see Fig. 4). In this setup, we use a pre-trained 2D ResNet on a large database and our proposed SCN as the 3D model. The architecture of the both networks are similar. We concatenate the fully connected layers of both architectures to make a single fully connected layer that is further followed by two more layers with the dimension of 512 and 128 for binary classification. We use a binary classifier to decide whether the given N frames and the video belong to the same class or not.

During this knowledge transferring process, the parameters of the 2D model remain unchanged, while the task is to learn the model parameters for the 3D model. In backpropagation, only the parameters of the 3D model are updated. The pairs that belong to the same time stamp from the same video are considered as positive pairs, while the pairs drawn from two different videos by random sampling of N frames and video from two different videos are considered as negative pairs. It should be noted that the cross-architecture knowledge transfer technique is an unsupervised method and does not require labeled data.

In our experiments, we show that adequate weight initialization of SCN followed by fine-tuning on the target



Fig. 5 Face samples from the UNBC-McMaster shoulder pain expression archive database (Lucey et al. 2011b)

database significantly improve the performance. Additionally, we demonstrate that our proposed knowledge transfer technique can effectively be considered for training with small video database (note that all existing databases for pain intensity estimation are indeed small in size).

4 Experimental Analysis

For performance evaluation, we conducted extensive experiments on two benchmarking and publicly available databases namely UNBC-McMaster Shoulder Pain Expression Archive (Lucey et al. 2011b) and the BioVid Heat Pain (Walter et al. 2013). First, we adjusted the values for the hyper-parameters by performing a grid search and following the guidelines in Bengio (2012). Then, we made a comparison with the 3D CNN benchmarks to evaluate the effectiveness of our proposed cross-architecture knowledge transfer technique. Finally, we compared the performance of our method against state-of-the-art approaches. In all experiments, we followed the standard protocols corresponding to each database.

4.1 Experimental Data

UNBC-McMaster Database The UNBC-McMaster Shoulder Pain Expression Archive database (Lucey et al. 2011b) is widely used for pain expression recognition. This database contains facial videos of subjects performing a series of active and passive range-of-motion tests to their either affected or unaffected limbs on two sessions. Figure 5 shows some face samples from this database. Each video sequence was annotated in a frame-level fashion by FACS, resulting in 16 discrete pain intensity level (0-15) based on AUs. In our experiments, we considered the active test set that includes 200 face videos of 25 subjects with 48,398 frames of the size of 320×240 pixels.

BioVid Database The BioVid Heat Pain database (Walter et al. 2013) was collected from 90 participants from three age groups. Four distinct pain levels were induced in the right arm of each subject. Moreover, bio-psychological signals

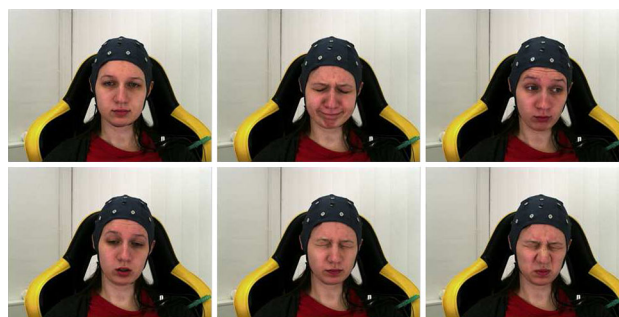


Fig. 6 Face samples from the BioVid Heat Pain database (Walter et al. 2013)

such as the Skin Conductance Level (SCL), the electrocardiogram (ECG), the electromyography (EMG), and the electroencephalogram (EEG) were recorded. However, in our experiment, we only use part A of this database (see Fig. 6). BioVid Part A includes 8,700 videos of 87 subjects which are labeled with respect to pain stimulus intensity. So, we distinguish five pain intensity levels, i.e. no pain (level 0), low pain (level 1), severe pain (level 4), and two intermediate pain intensities (levels 2 and 3).

4.2 Experimental Setup

To define the architecture of our SCN, we started with Res3D based on the standard 2D ResNet. Then, we explored SCN architecture based on Res3D. Due to the heavy computations when training deep models, we conducted the analysis (i.e. search for the optimal architecture) on the UNBC-McMaster database, as the BioVid database (Walter et al. 2013) is much larger than the UNBC-McMaster (Lucey et al. 2011b). Finally, we fine-tuned the optimal architecture for pain intensity estimation on each database separately.

We designed the Res3D architecture by replacing all the 2D kernels with 3D kernels and removing the first max pooling layer as in Tran et al. (2017). To achieve the optimal configuration for this architecture, we conducted a series of experiments on the model size and the temporal depths of the input data. Using bottleneck building block, we employed five versions of 2D ResNet with network sizes of 26, 39, 50, 101, and 152 for designing the Res3D. Similarly, we developed our SCN with these five network sizes. Table 1 summarizes the Area Under the Curve (AUC) as the accuracy measurement for these two models with different network sizes. As can be seen from Table 1, both models' accuracies increase with the increment of the network size. However, our SCN performs much better than the Res3D model and achieves 98.53% accuracy in pain intensity estimation. However, its accuracy drops when the network size becomes 152 layers. This implies that detailed spatiotemporal information for an effective representation of the video can be readily

Table 1 The area under the curve (AUC) accuracy (%) of the Res3D and SCN for different network sizes on the UNBC-McMaster database (Lucey et al. 2011b)

	Network size				
	26	39	50	101	152
Res3D	80.92	84.55	86.27	91.05	91.75
SCN	87.33	91.74	93.81	98.53	98.44

extracted using several 3D convolutional kernels with diverse temporal depths. Hence, enlarging the network depth does not necessarily improve the model performance, while it significantly increases the number of model’s parameters.

Facial expressions exhibit short, mid, and long-range spatiotemporal variations. So, the number and temporal depth, and the receptive field’s size of parallel 3D convolutional kernels in the bottleneck building blocks architecture play a crucial role in capturing those changes. In our experiments, we empirically changed the number of parallel 3D convolutional kernels from 1 to 4 to determine an optimal value. Simultaneously, we varied the temporal depth of each 3D convolutional kernel to find a good arrangement for the temporal coverage of kernels. Table 2 shows the accuracy of our proposed SCN for different number of parallel 3D convolutional kernels versus various temporal depths of kernels. We explored the performance of the SCN on capturing the facial dynamics by trying various gaps in temporal coverage of the 3D convolutional kernels. The results in Table 2 show that the accuracy of the SCN improves as the number of kernels increases. This increase allows the model to have more kernels with diverse temporal depths, which is important for obtaining a rich representation of the input data. The highest accuracy is achieved, i.e. 97.32%, when the number of parallel 3D convolutional kernels is set to 3 and the temporal depths of kernels are [3, 5, 9]. As can be seen, the accuracy drops provided that we do not consider a continuous temporal coverage. For instance, SCN achieves 91.33% using kernels with temporal depths of [3, 9, 13]. This ignores the mid-

range spatiotemporal variations. However, the performance improves significantly as the temporal depths are set to [3, 7, 9]. These results further validate our initial hypothesis of capturing short, mid, and long-range variations using kernels with diverse temporal depths. We conclude that the temporal depths of convolutional kernels should not be very dense nor loose to effectively capture the dynamics of video.

The receptive field of convolutional kernels is one of the basic concepts in CNNs. It is essential to carefully adjust the receptive field to ensure that it encompasses the relevant image regions. In order to effectively represent the input data, we analyzed the effect of kernels’ receptive field size. We conducted experiments by changing the spatial size of the 3D convolutional kernels. Figure 7 illustrates the performance of our SCN with different sizes of kernel’s receptive field versus a range of temporal depth combinations. Our experiments demonstrate that the smaller receptive fields can capture more detailed information. Hence, the overall accuracy of the proposed method is high when the spatial size of the convolutional kernels are 3 × 3. On the other hand, the performance declines as the receptive field’s size and the steps between temporal depths of 3D convolutional kernels become larger.

According to Tran et al. (2015), the temporal depth of input data affects the model performance in spatiotemporal representations. We evaluated the accuracy of our SCN with inputs of different temporal depths (see Fig. 8). Among all the considered temporal depths, input data with temporal depth of 32 frames seems to perform better. This result validates our initial hypotheses in the sense that larger input depth allows the model to capture short, mid, and long-range spatiotemporal terms in the video for more efficient representation. According to the results in Tables 1 and 2 and Fig. 7, different temporal lengths of the convolutional kernels in the bottleneck building block allow the model to capture more relevant spatiotemporal information from the videos. The details of our SCN architecture are illustrated in Table 3.

Table 2 The area under the curve (AUC) accuracy (%) of SCN for different number of parallel 3D convolutional kernels versus various temporal depths of convolutional kernels on the UNBC-McMaster database (Lucey et al. 2011b)

Number of parallel 3D convolutional kernels		3	5	7	9	11	13
1	Temporal Depths	3	5	7	9	11	13
	AUC	90.34	90.05	89.47	86.33	81.35	77.49
2	Temporal Depths	[3, 5]	[3, 7]	[3, 9]	[5, 7]	[5, 9]	[7, 9]
	AUC	93.54	93.24	92.81	90.75	89.79	88.46
3	Temporal Depths	[3, 5, 7]	[3, 5, 9]	[3, 5, 11]	[3, 7, 9]	[3, 9, 11]	[3, 9, 13]
	AUC	96.54	97.32	95.77	96.64	93.21	91.33
4	Temporal Depths	[3, 5, 7, 9]	[5, 7, 9, 11]	[3, 7, 9, 11]	[7, 9, 11, 13]	[3, 5, 9, 11]	[3, 5, 7, 13]
	AUC	96.43	97.04	96.79	94.37	95.46	95.57

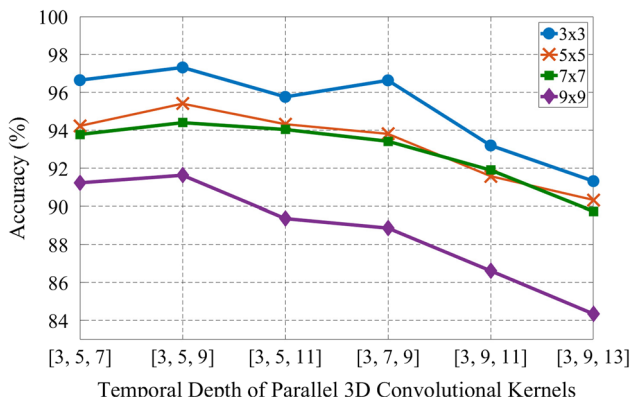


Fig. 7 The area under the curve (AUC) accuracy of the proposed SCN versus different spatial sizes of the convolutional kernels receptive field

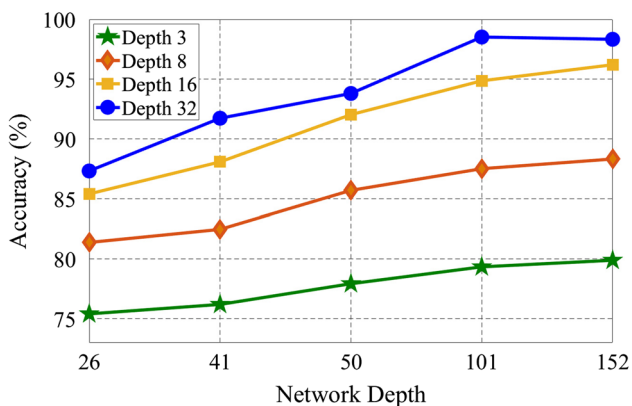


Fig. 8 Evaluation of SCN for different temporal depths of input data on the UNBC-McMaster database (Lucey et al. 2011b)

The input structure plays a crucial role in capturing detailed appearance and temporal information of the video (Tran et al. 2015). We conducted a series of experiments to determine the optimal input frame resolution and the frame sampling rate. We performed experiments on three different input frame resolutions, i.e. 224×224 , 112×112 , and 56×56 . Based on the results obtained in the previous section, we constrained our search to the network that has achieved the best result. Table 4 reports the accuracy of SCN on these three input resolutions. As can be seen, an input resolution of 112×112 pixels gains the best performance, while resolutions of 224×224 and 56×56 seem to be very large and very small, respectively. These results can partially be explained by the fact that small input resolution does not provide enough spatial information for representation. On the other hand, large input resolution does not necessarily introduce more information to the model.

Another model's hyper-parameter that has an influence on the output accuracy is the frame sampling rate. Following the previous network setup, we evaluated the performance of our model by changing the temporal stride of the input frames.

Table 5 presents the model accuracy trained on inputs with different frame sampling rates. The optimal performance is obtained when the temporal stride is set to 2 frames.

Based on the obtained results in our experimental setup and design, the model achieves its optimal performance with network depth of 101 layers and input temporal depth of 32 frames along with input frame resolution of 112×112 pixels and the temporal stride of 2. Hereafter, we use these settings in all the remaining experiments.

4.3 Training the Model

Our proposed SCN works on a chunk of 32 RGB frames. We cropped and resized the detected face images in the video sequence into 112×112 pixels. We followed the same weight initialization strategy as in He et al. (2015). For the training stage, we adopted Stochastic Gradient Descent (SGD) with a momentum of 0.9, weight decay 10^{-4} , and batch size of 64. The initial learning rate is set to 0.01 and decreased by a factor of 10 after every 10 epochs. The maximum number of epochs for the training was set to 200.

Moreover, we used a pre-trained 2D ResNet architecture on the CASIA WebFaces database (Yi et al. 2014) in the cross-architecture knowledge transfer scheme. In this framework, 32 RGB mean-subtracted frames were the inputs of the 2D network. To transfer knowledge to SCN, we substituted the classification layer of the 2D network with a two-way softmax classifier to determine positive and negative pairs. Our experiments showed that a proper weight initialization followed by a transfer learning improve the training of 3D CNNs on small databases like the UNBC-McMaster (Lucey et al. 2011b).

To transfer knowledge between architectures, we used positive and negative video sequence pairs. The videos are considered as positive pairs if they belong to the same class. A pair of 32 frames and a video sequence for the same time stamp will go through the 2D ResNet and SCN. An average pooling is done on the last layer of the 2D ResNet. The obtained representations of the 2D ResNet were concatenated with the video representations of SCN and passed into two fully connected layers afterward. The binary classifier distinguishes positive pairs from negative ones. The SCN network is trained using backpropagation to learn the 3D kernels' parameters, while the parameters of the 2D ResNet remained unchanged.

The proposed cross-architecture knowledge transfer technique is an elegant way to train 3D CNN architectures when large-scale databases are not available for a specific application. In addition, this learning technique is an unsupervised approach. Hence, the labeled data are not required for the

Table 3 The architecture of the proposed SCN

Layer	Spatial output	Structure	Number of layers				
			26	41	50	101	152
Conv1	112 × 112	[7 × 7 × 3], 64					
Conv2	56 × 56	[1 × 1 × 3], 64	× 2	× 3	× 3	× 3	× 3
		$\begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 5 \\ 3 \times 3 \times 9 \end{bmatrix}, 64$					
Conv3	28 × 28	[1 × 1 × 3], 256	× 2	× 3	× 4	× 4	× 8
		$\begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 5 \\ 3 \times 3 \times 9 \end{bmatrix}, 128$					
Conv4	14 × 14	[1 × 1 × 3], 512	× 2	× 4	× 6	× 23	× 36
		$\begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 5 \\ 3 \times 3 \times 7 \end{bmatrix}, 256$					
Conv5	7 × 7	[1 × 1 × 3], 1024	× 2	× 3	× 3	× 3	× 3
		$\begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 5 \\ 3 \times 3 \times 7 \end{bmatrix}, 512$					
		[1 × 1 × 3], 2048					
Pooling	1 × 1	Average pooling, 1000D fully connected					

Table 4 The accuracy (%) of SCN for different input frame resolutions on the UNBC-McMaster database (Lucey et al. 2011b)

Input resolution	224 × 224	112 × 112	56 × 56
AUC	92.07	98.53	90.22

Table 5 The accuracy (%) of SCN for different input sampling rates on the UNBC-McMaster database (Lucey et al. 2011b)

Temporal stride	1	2	4	8	16
AUC	93.50	98.53	97.61	90.33	87.09

knowledge transferring. After pre-training our SCN using the mentioned cross-architecture knowledge transfer technique, we can fine-tune the model on the target database. It is worth noting that the proposed method for transfer learning can also be adapted to different 3D CNN architec-

Table 6 The AUC accuracy (%) of the cross-architecture knowledge transfer on 3D CNN architectures

Model	With transfer learning	Without transfer learning
VGG3D	82.54	80.32
Inception3D	85.31	82.65
Res3D	91.05	87.33
SCN	98.53	95.27

tures. A direct comparison between the accuracy of different 3D CNN architectures (Res3D, Inception3D, VGG3D, and SCN) with and without cross-architecture knowledge transfer learning is given in Table 6. These experimental results in Table 6 clearly indicate that the proposed transfer learning technique significantly improves the accuracy of all 3D CNNs.

Table 7 Comparative analysis in terms of mean squared error and the Pearson correlation coefficient on the UNBC-McMaster database (Lucey et al. 2011b)

Method	MSE	PCC
RVR + (LBP and DCT) (Kaltwang et al. 2012)	1.39	0.59
HoT (Florea et al. 2014)	1.21	0.53
OSVR (Zhao et al. 2016)	N/A	0.60
RCNN (Zhou et al. 2016)	1.54	0.64
LSTM (Rodriguez et al. 2017)	0.74	0.78
WRN (Zagoruyko and Komodakis 2016)	0.75	0.78
BORMIR (Zhang et al. 2018)	1.38	0.61
DDM (Tavakolian and Hadid 2018) [†]	0.69	0.83
TSN (Wang et al. 2018) [†]	0.75	0.80
S3D-G (Xie et al. 2018) [†]	0.60	0.88
C3D (Tran et al. 2015) [†]	0.71	0.81
Res3D (Tran et al. 2017) [†]	0.65	0.86
SCN	0.32	0.92

[†]The method was originally proposed for action recognition and is adapted for pain intensity estimation

4.4 Experimental Results and Comparison with State-of-the-Art

We compared the performance of our proposed SCN with the recent state-of-the-art methods for automatic pain intensity estimation on the UNBC-McMaster (Lucey et al. 2011b) and the BioVid (Walter et al. 2013) databases.

In order to make a direct and fair comparison with the state-of-the-art methods, we report the Mean Squared Error (MSE) and the Pearson Correlation Coefficient (PCC) for the UNBC-McMaster database (Lucey et al. 2011b). Table 7 shows that the SCN improves the performance of pain intensity estimation using leave-one-subject-out cross-validation. As can be seen, our proposed method consistently outperforms the existing benchmark approaches by a large margin both in terms of MSE and PCC. It significantly reduces the MSE by 0.33 compared to Res3D architecture. In addition, its high PCC reveals that our proposed method is able to effectively extract detailed information from the face videos for accurate pain intensity estimation. The results in Table 7 further confirm that the deep spatiotemporal representation based methods (C3D, Res3D, and SCN) perform substantially better than Recurrent Neural Networks (RNNs) or LSTM for automatic pain intensity estimation, since these methods consider both the spatial and temporal information at the same time rather than treating the video sequences frame by frame.

To visualize the effectiveness of the SCN in determining different levels of pain, we compare in Fig. 9 the output of our model with the ground truth and Res3D on a video sequence of one subject from the UNBC-McMaster

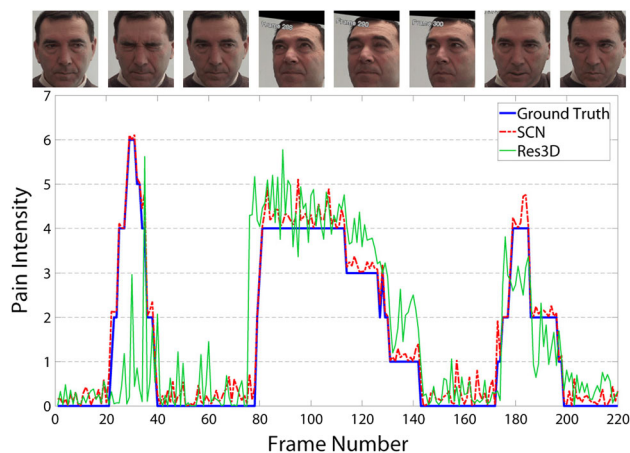


Fig. 9 An example of continuous pain intensity estimation using Res3D and SCN on a sample video from the UNBC-McMaster database (Lucey et al. 2011b)

database. It can be seen that SCN captures well different levels of pain and keeps track of pain intensity variations in each time instance thanks to its short, mid, and long-range spatiotemporal representation. Although Res3D can also determine pain intensity quite well, it is unable to correctly estimate the pain intensity when sudden changes happen in the face. Furthermore, Fig. 10 illustrates qualitative comparisons between Res3D and SCN with varied speed of facial expression changes. When changes in the facial structure happen smoothly (Fig. 10 left), both Res3D and SCN track changes in the pain intensity quite well. However, the performance of Res3D drops as sudden changes occur in the video sequence, while SCN firmly estimates the pain intensity level. As changes in the facial expression of pain happen rapidly, Res3D shows a noisy behaviour and cannot accurately determine the right level of pain intensity. However, our proposed SCN adapts itself to various speeds due to capturing a wide range of facial dynamics by the parallel 3D convolutional kernels.

Table 8 compares our results against the state-of-the-art on the BioVid database (Walter et al. 2013). As mentioned in Sect. 4.1, we considered BioVid Part A and use only the facial video data. The obtained results demonstrate that the 3D deep architecture has a higher accuracy in pain intensity estimation compared to conventional methods like LBP (Yang et al. 2016). Among the deep models, our SCN achieves the best accuracy which illustrates its robustness in the representation of a wide range of facial variations, i.e. subtle and large changes.

Although some methods in Tables 7 and 8 have achieved state-of-the-art performance in action recognition and video classification, not all of them are suitable for pain intensity estimation. To be specific, TSN (Wang et al. 2018) is a good alternative for 3D models such as C3D in action recognition.

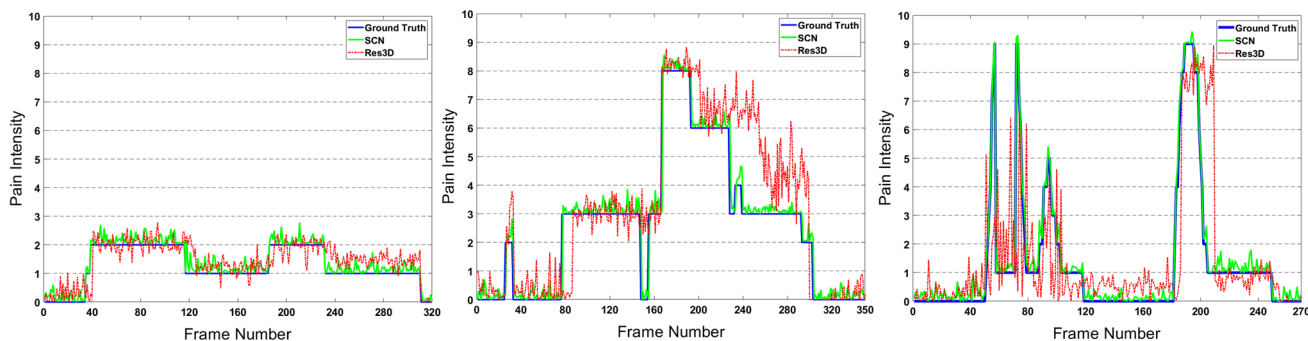


Fig. 10 Performance comparisons between Res3D and SCN when facial structures undergo changes with various speeds. Left: low speed, middle: medium speed, and right: high speed of facial expression vari-

ation. Res3D shows a noisy behaviour in response to sudden changes in the facial structure produced by pain

Table 8 Comparative analysis in terms of accuracy (%) on the BioVid database (Walter et al. 2013)

Method	AUC
Head-movement (Walter et al. 2013)	67.00
Time-windows (Walter et al. 2013)	71.00
LBP (Yang et al. 2016)	63.72
LPQ (Yang et al. 2016)	63.19
BSIF (Yang et al. 2016)	65.17
FAD set (Werner et al. 2017)	72.40
WRN (Zagoruyko and Komodakis 2016)	73.55
BORMIR (Zhang et al. 2018)	72.85
DDM (Tavakolian and Hadid 2018) [†]	81.05
TSN (Wang et al. 2018) [†]	75.39
S3D-G (Xie et al. 2018) [†]	83.26
C3D (Tran et al. 2015) [†]	80.31
Res3D (Tran et al. 2017) [†]	82.54
SCN	86.02

[†]The method was originally proposed for action recognition and is adopted for pain intensity estimation

However, its performance is deteriorated in pain intensity estimation due to random sampling of frames from the segments of videos. The random sampling discards the crucial information of sudden and subtle transitions of facial structure produced by pain. S3D-G (Xie et al. 2018) replaces 3D convolution kernels in I3D architecture with two consecutive convolution layers: one 2D convolution kernel to capture the spatial information followed by one 1D convolution kernel to incorporate the temporal information. The S3D-G’s performance is higher than the vanilla C3D and Res3D on both UNBC and BioVid databases. Nevertheless, it is still unable to encode different ranges of facial dynamics due to disjoint learning of spatial and temporal information. These results confirm our hypothesis that capturing different ranges of spatiotemporal information is important for encoding subtle and

sudden facial expression variations for pain intensity estimation from faces.

4.5 Computational Complexity Analysis

Although deep models extract more information from the inputs, the model’s complexity and computational time tend to increase when enlarging the depth of the network. Insights into the computational complexity of our proposed SCN versus Res3D on the UNBC-McMaster database are given in Table 9. The average training time (in seconds) and the time required to estimate the pain intensity from the query video are listed in Table 9. Although our SCN requires more time for training, it needs 3.263 s to estimate the pain intensity in the test phase. We argue that this short testing time is attributed to the parallel 3D convolution kernels in the bottleneck blocks that allow the model learn rich representation of the input video. All the feature maps are computed simultaneously within each layer by the virtue of parallelization of 3D convolutional kernels. So, widening the model to achieve short, mid, and long-range representations substantiate the short testing time of SCN in comparison with Res3D in Table 9, i.e. once the model learned efficient representations, the processing of test samples is quick and straightforward.

On the other hand, we assert that the training time significantly increases as we train the model without knowledge transferring. In this case, we pre-train the SCN using the CASIA WebFace database (Yi et al. 2014). Thanks to the substantial number of parameters, training 3D architecture from scratch demands heavy computational workload and long training time. By using the proposed cross-architecture knowledge transfer approach, the training time for both Res3D and SCN is reduced, significantly. This is due to appropriate parameter initialization of the 3D model by gaining knowledge from the 2D model, which results in decreasing the number of epochs required for the convergence.

Table 9 Training and testing times (seconds) of SCN and Res3D on the UNBC-McMaster database (Lucey et al. 2011b)

# of Layers	Train		Test	# of Parameters ($\times 10^6$)
	Without transfer learning	With transfer learning		
<i>SCN</i>				
26	104.520	27.433	2.065	126.0
41	130.029	31.182	2.103	199.6
50	145.814	36.729	2.558	226.0
101	197.816	41.537	3.263	423.2
152	353.270	55.809	4.316	586.8
<i>Res3D</i>				
26	30.284	9.347	4.821	33.4
41	58.990	13.501	5.740	52.0
50	77.995	17.294	6.233	58.6
101	129.595	24.360	6.861	108.8
152	177.352	30.631	7.592	150.0

The SCN has 3.8 times more model parameters compared to Res3D due to the parallel convolutional kernels in the bottleneck building blocks. However, its performance is higher than Res3D by a large margin. We note that Res3D and the other 3D CNN-based models only capture the spatiotemporal information within a fixed and homogeneous temporal range. Moreover, once our model is pre-initialized using the proposed cross-architecture knowledge transfer, it can be readily employed on different databases. In fact, the model is fine-tuned on the new target database. This strategy reduces the training time for the new database.

5 Conclusion

Automatic pain intensity assessment has a high value in diagnosis applications. This paper proposed a spatiotemporal convolutional network for pain intensity estimation from face videos. It leverages the detailed spatiotemporal information of spontaneous variations in the facial expression by deploying several 3D convolution operations with different temporal depths. Unlike 3D convolutional neural networks with fixed 3D kernel depths, our proposed architecture captures short, mid, and long-range spatiotemporal variations that are essential for representing spontaneous facial variations. Furthermore, we developed a cross-architecture knowledge transfer approach to avoid training 3D model from scratch. We used a pre-trained 2D deep model to train our 3D architecture with a relatively small number of training data. Using this technique, we further fine-tuned the 3D model on the target database. Our extensive experiments on the UNBC-McMaster Shoulder Pain and the BioVid databases showed the effectiveness of our proposed approach compared to the state of the art. As a future work, we plan to validate the generalization of the proposed model on other face analysis tasks.

Acknowledgements Open access funding provided by University of Oulu including Oulu University Hospital. The financial support of the Academy of Finland, Infotech Oulu, Nokia Foundation, and Tauno Tönnö Foundation is acknowledged.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Ashraf, A. B., Lucey, S., Cohn, J. F., Chen, T., Ambadar, Z., Prkachin, K. M., et al. (2009). The painful face—pain expression recognition using active appearance models. *Image and Vision Computing*, 27(12), 1788–1796.
- Bengio, Y. (2012). *Practical recommendations for gradient-based training of deep architectures* (pp. 437–478). Berlin: Springer.
- Brahnam, S., Chuang, C. F., Shih, F. Y., & Slack, M. R. (2006). Machine recognition and representation of neonatal facial displays of acute pain. *Artificial Intelligence in Medicine*, 36(3), 211–222.
- Brahnam, S., Nanni, L., & Sexton, R. (2007). Introduction to neonatal facial pain detection using common and advanced face classification techniques. In *Advanced computational intelligence paradigms in healthcare* (pp. 225–253).
- Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE CVPR* (pp. 4724–4733).
- Craig, K. D., Prkachin, K. M., & Grunau, R. V. E. (2011). *Handbook of pain assessment*. New York: Guilford Press. Chapter: The facial expression of pain.
- Ekman, P., & Friesen, W. (1978). *Facial action coding system: A technique for the measurement of facial movement*. Palo Alto: Consulting Psychologists Press.
- Flor, H., Breitenstein, C., Birbaumer, N., & Fürst, M. (1995). A psychophysiological analysis of spouse solicitousness towards pain behaviors, spouse interaction, and pain perception. *Behavior Therapy*, 26(2), 255–272.

- Florea, C., Florea, L., & Vertan, C. (2014). Learning pain from emotion: Transferred HoT data representation for pain intensity estimation. In *ECCV workshops* (pp. 778–790).
- Gholami, B., Haddad, W. M., & Tannenbaum, A. R. (2010). Relevance vector machine learning for neonate pain intensity assessment using digital imaging. *IEEE Transactions on Biomedical Engineering*, 57(6), 1457–1466.
- Guo, Y., Zhao, G., & Pietikäinen, M. (2012). Discriminative features for texture description. *Pattern Recognition*, 45(10), 3834–3843.
- Hammal, Z., & Cohn, J. F. (2012). Automatic detection of pain intensity. In *ACM international conference on multimodal interaction* (pp. 47–52).
- Hammal, Z., & Kunz, M. (2012). Pain monitoring: A dynamic and context-sensitive system. *Pattern Recognition*, 45(4), 1265–1280.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE ICCV* (pp. 1026–1034).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE CVPR* (pp. 770–778).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML* (pp. 448–456).
- Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on PAMI*, 35(1), 221–231.
- Kaltwang, S., Rudovic, O., & Pantic, M. (2012). Continuous pain intensity estimation from facial expressions. In *International symposium on advances in visual computing* (pp. 368–377).
- Lesage, F. X., Berjot, S., & Deschamps, F. (2012). Clinical stress assessment using a visual analogue scale. *Occupational Medicine*, 62(8), 600–605.
- Littlewort, G. C., Bartlett, M. S., & Lee, K. (2009). Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27(12), 1797–1803.
- Lucey, P., Cohn, J. F., Matthews, I., Lucey, S., Sridharan, S., Howlett, J., et al. (2011a). Automatically detecting pain in video through facial action units. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3), 664–674.
- Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., Chew, S., & Matthews, I. (2012). Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database. *Image and Vision Computing*, 30(3), 197–205.
- Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., & Matthews, I. (2011b). Painful data: The UNBC-McMaster shoulder pain expression archive database. In *IEEE international conference on face and gesture* (pp. 57–64).
- Lynch, M. (2001). Pain as the fifth vital sign. *Journal of Intravenous Nursing*, 24(2), 85–94.
- Prkachin, K. M., & Solomon, P. E. (2008). The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2), 267–274.
- Rodríguez, P., Cucurull, G., González, J., Gonfaus, J. M., Nasrollahi, K., Moeslund, T. B., et al. (2017). Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE Transactions on Cybernetics, PP(99)*, 1–11.
- Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS* (pp. 568–576).
- Sun, L., Jia, K., Yeung, D. Y., & Shi, B. E. (2015). Human action recognition using factorized spatio-temporal convolutional networks. In *IEEE ICCV* (pp. 4597–4605).
- Tavakolian, M., & Hadid, A. (2018). Deep discriminative model for video classification. In *The European conference on computer vision (ECCV)*.
- Thevenot, J., López, M. B., & Hadid, A. (2017). A survey on computer vision for assistive medical diagnosis from faces. *IEEE Journal of Biomedical and Health Informatics, PP(99)*, 1–1.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *IEEE ICCV* (pp. 4489–4497).
- Tran, D., Ray, J., Shou, Z., Chang, S. F., & Paluri, M. (2017). *Convnet architecture search for spatiotemporal feature learning*. [arXiv:1708.05038](https://arxiv.org/abs/1708.05038).
- Walter, S., Gruss, S., Ehleiter, H., Tan, J., Traue, H. C., Crawcour, S., et al. (2013). The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *IEEE international conference on cybernetics* (pp. 128–131).
- Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2018). Temporal segment networks for action recognition in videos. *IEEE Transaction on PAMI*. (in press).
- Werner, P., Al-Hamadi, A., Limbrecht-Ecklundt, K., Walter, S., Gruss, S., & Traue, H. C. (2017). Automatic pain assessment with facial activity descriptors. *IEEE Transactions on Affective Computing*, 8(3), 286–299.
- Werner, P., Al-Hamadi, A., Niese, R., Walter, S., Gruss, S., & Traue, H. C. (2013). Towards pain monitoring: Facial expression, head pose, a new database, an automatic system and remaining challenges. In *BMVC* (pp. 119.1–119.13).
- Xie, S., Sun, C., Huang, J., Tu, Z., & Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *The European conference on computer vision (ECCV)*.
- Yang, R., Tong, S., López, M. B., Boutellaa, E., Peng, J., Feng, X., et al. (2016). On pain assessment from facial videos using spatio-temporal local descriptors. In *IEEE IPTA* (pp. 1–6).
- Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). *Learning face representation from scratch*. [arXiv:1411.7923](https://arxiv.org/abs/1411.7923).
- Zagoruyko, S., & Komodakis, N. (2016). *Wide residual networks*. [arXiv:1605.07146](https://arxiv.org/abs/1605.07146).
- Zhang, Y., Zhao, R., Dong, W., Hu, B. G., & Ji, Q. (2018). Bilateral ordinal relevance multi-instance regression for facial action unit intensity estimation. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
- Zhao, R., Gan, Q., Wang, S., & Ji, Q. (2016). Facial expression intensity estimation using ordinal information. In *IEEE CVPR* (pp. 3466–3474).
- Zhou, J., Hong, X., Su, F., & Zhao, G. (2016). Recurrent convolutional neural network regression for continuous pain intensity estimation in video. In *IEEE CVPR workshops* (pp. 1535–1543).