Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification

Rodriguez, Pau; Cucurull, Guillem; Gonzàlez, Jordi; M. Gonfaus, Josep ; Nasrollahi, Kamal; Moeslund, Thomas B.

Published in:
IEEE TRANSACTIONS ON CYBERNETICS

DOI (link to publication from Publisher):
10.1109/TCYB.2017.2662199

Publication date:
2017

Document Version
Peer reviewed version

Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification

Pau Rodriguez, Guillem Cucurull, Jordi Gonzàlez, Josep M. Gonfaus,
Kamal Nasrollahi, Thomas B. Moeslund, F. Xavier Roca

***Abstract*—Pain is an unpleasant feeling that has been shown to be an important factor for the recovery of patients. Since this is costly in human resources and difficult to do objectively, there is the need for automatic systems to measure it. In this paper, contrary to current state-of-the-art techniques in pain assessment, which are based on facial features only, we suggest that the performance can be enhanced by feeding the raw frames to deep learning models, outperforming the latest state-of-the-art results while also directly facing the problem of imbalanced data. As a baseline, our approach first uses convolutional neural networks (CNN) to learned facial features from VGG_Faces, which are then linked to a Long Short-Term Memory (LSTM) to exploit the temporal relation between video frames. We further compare the performances of using the so popular schema based on the canonically normalized appearance versus taking into account the whole image: As a result, we outperform current state-of-the-art AUC performance in the UNBC-McMaster Shoulder Pain Expression Archive Database. In addition, to evaluate the generalization properties of our proposed methodology on facial motion recognition, we also report competitive results in the Cohn Kanade+ facial expression database.**

I. Introduction

The automatic detection of pain is a subject of high interest in the health domain since it is not only an important indicator for medical diagnosis, but has also been shown to be an obstacle for patient recuperation in Intensive Care Units [1] and after surgery [2]. In [3], it is shown how good pain assessment is crucial for a good pain control, which is usually verbally checked by professional nurses, known as self-report. However, this is not always possible due to the age of the patient, the particular illness or language impairments. Moreover, pain is a subjective feeling which can be described differently across cultures [4]. Thus, pain assessment could be highly benefited from automatic tools.

Indeed this goal has been already addressed several times in the past, for example in 2011 the authors of [5] tackle the problem using brain activity imaging. So pain detection is also an important task from the point of view of computer vision, since it is a clear step towards an automatic detector of spontaneous face expressions [6], [7], [8], [9] and [10]. In particular, it was of high importance for the computer vision community the release of a database published by Lucey *et al.* in [11], in order to alleviate the lack of representative data of the other existing databases. Their UNBC-McMaster database consists of 200 video sequences taken from 25 patients who were suffering from shoulder pain. The frames were labeled using the validated Prkachin and Solomon metric [12] (PSPI) based on the Facial Action Coding System (FACS) [13], which codes different movements of the face muscles with different intensity levels. It is a very challenging dataset, and as it can be seen in Fig. 1, in some cases it can be very hard to determine whether a subject is in pain or not, even for clinical professionals.

So the UNBC-McMaster Painful dataset has been used to propose new models for facial pain detection. In the first place, Lucey *et al.* in [11] already released a baseline along with the dataset, using Support Vector Machines (SVMs) on top of the pixel and landmark features extracted using Active Appearance Models (AAM) [14] in order to predict painful Action Units (AUs) and the PSPI for the presence of pain. [10] proposed a late fusion of shape and appearance features in order to predict the continuous PSPI scores of the Painful data.

In fact, facial Action Units have been typically used to encode facial motion corresponding to different facial expressions such as pain or anger. As stated by Rudovic *et al.* [15], the task of AU intensity estimation is very challenging, due to the high variability in facial expressions depending on the context, such as illumination, head movements or subject-specific expressions. Being a complex task, Action Unit intensity estimation has received a lot of attention over two decades for generic facial motion analysis. It has been approached by Kim *et al.* in [16], where they use a dynamic ranking model to overcome the difficulty of the emotion intensities differing substantially across subjects. Valstar *et al.* [17] also tackle the task of facial AUs recognition by using a facial point detector to localize 20 facial fiducial points. Then these points are tracked through a sequence of images and then a combination of GentleBoost, SVMs and hidden Markov models (HMM) is used for AU recognition. According to [18] most of the temporal graphical models such as HMM or conditional random fields (CRF) used for AU recognition fail to jointly model different emotions. To overcome this issue, they propose the use of a Hidden Conditional Ordinal Random Field (H-CORF) to achieve both intensity estimation of facial expressions and dynamic recognition of multiple emotions at the same time. Ming *et al.* [19] proposed a method based

P. Rodriguez, G. Cucurull, J. Gonzàlez, and F.X. Roca are with the Computer Vision Center - Universitat Autònoma de Barcelona (Catalonia Spain); J. Gonzàlez, J.M. Gonfaus, and F.X. Roca are with Visual Tagging Services, Campus UAB Barcelona (Catalonia Spain); and K. Nasrollahi and T.B. Moeslund are with Aalborg University, Denmark. Authors acknowledge the support of the Spanish project TIN2015-65464-R (MINECO/FEDER), the 2016FI_B 01163 grant of Generalitat de Catalunya, and the COST Action IC1307 iV&L Net (European Network on Integrating Vision and Language) supported by COST (European Cooperation in Science and Technology). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU and a GTX TITAN GPU, used for this research.

Figure 1: **Examples of pain and no pain frames** This figure shows how hard it can be to distinguish between pain and no pain frames. The subject was not in pain in the frames of the first row (a), whereas it was suffering pain in all frames of row (b). At first glance it is very hard to determine which row contains pain frames and which one contains frames labeled as zero pain level, demonstrating that the task of pain detection is not trivial and that the proposed model faces a lot of difficult cases like the ones being shown.

on multi-kernel SVM and feature fusion to approach AUs intensity estimation.

Focused on facial landmark estimation for pain detection, Rudovic *et al.* [20] proposed to use a heteroscedastic Conditional Ordinal Random Field (CORF) model in order to deal with the inter-subject variability of the pain expressions. Authors in [21] and [22] used weakly supervised learning and multiple instance learning to predict pain only using sequence-level annotations. Khan *et al.* [23] also used the referenced dataset for pain/no-pain recognition using shape information extracted with a pyramid histogram of orientation gradients (PROG) and appearance information using a pyramid local binary patterns (PLBP). Subsequently, Zafar and Khan's [24] used a K-NN classifier to classify AUs using 22 facial characteristic points. In 2015, Irani *et al.* [25] use Spatiotemporal Feature Extraction in order to model the exploits the released energy of the facial muscles in the spatial and temporal domains. They applied their system to both RGB [25] and RGB-Thermal-Depth [26] facial images. Presti and Cascia [27] use Hankel Matrices to represent the temporal dynamics of a sequence of Face Image Descriptors. Pedersen [28] addressed the identity bias of the dataset using autoencoders, ensuring the presence of discriminative features by training with a combined loss function that balances the reconstruction error and the classification error. Later in the same year, Neshov and Manolova [29] used SVMs on top of Scale Invariant Feature Transform (SIFT) features for continuous and discrete PSPI prediction. Rathee and Ganotra [30] proposed the use of Thin Plate Spline (TPS) mapping [31] for modeling the deformation of facial features and a Distance Metric Learning (DML) method to ensure the distance between features belonging to different levels of pain. The recent work of Zhao *et al.* [32] proposes the novel Alternating Direction Method of Multipliers (ADMM) to solve Ordinal Support Vector Regression (OSVR) achieving competitive performances in supervised, semi-supervised and unsupervised prediction of PSPI scores.

In this paper we continue current trends on deep learning [33][34][35][36][37] applied to pain estimation [38]. Similarly to [38], we also perform regression with Deep Convolutional Neural Networks (DCNNs) in order to predict the PSPI score for each frame. Subsequently, we adapt the resulting CNN model for pain classification inspired by [11]. In order to alleviate the problem of data scarcity, we use VGG_Faces, i.e. a VGG-16 CNN [37] pre-trained with millions of faces [39], which already obtains state-of-the-art scores compared with other leave-one-subject-out methods.

Differently to [38], we follow the ideas exposed in [25], by directly exploiting the temporal axis information using Long Short-Term Memory (LSTM) [40][41] on top of the previously-learned deep features, boosting our scores even more. So the main difference of our Deep Learning methodology as described above and the Recurrent Convolutional Neural Networks used in [38] is that we leverage the temporal information without renouncing to the representational power of generic pre-trained CNN features like the ones learned from VGG_Faces, i.e. we link the VGG_Faces features to the LSTM Recurrent Network. In other words, the approach of [38] either discards the temporal information of the data when considering pre-trained features from VGG_Faces or considers temporal information but using less-discriminative features, since the RCNN is learned from scratch.

In addition, differently to [38], we consider the raw image as the input of the CNN, rather than using facial landmarks. By doing so, the proposed method is able to outperform current state-of-the-art in pain intensity estimation.

As pain detection is a form of facial expression recognition, similar methods can be applied to the more general task of emotion recognition. For example Lucey *et al.* [42] used an SVM on top of features extracted using AAM to build a facial emotion classifier. Based on the observation that only a few facial patches are important for expression recognition, Zhong *et al.* [43] use a two-stage approach. First LBP features are

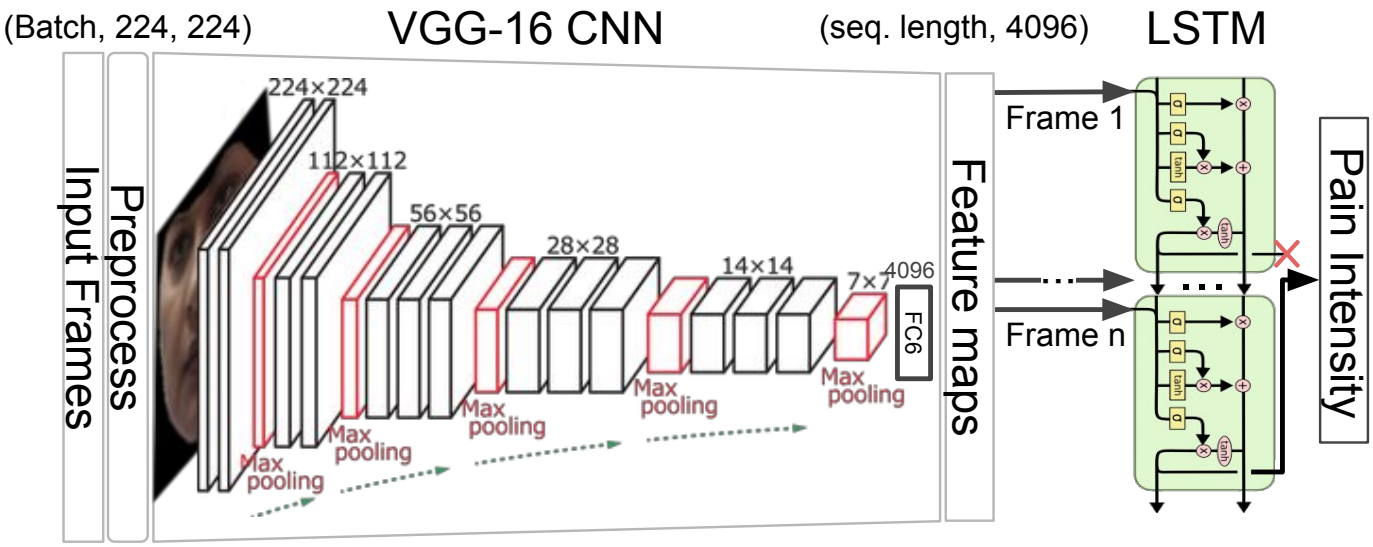

Figure 2: **Proposed framework.** Schematic depicting the different stages of our proposed pain detection model.

used to describe every patch on a grid of 8×8 over the images of 96×96 pixels. Then Multi-task sparse learning (MTSL) is used to learn common patches across expressions. Similar to this idea, Liu *et al.* [44] propose a method which adapts 3D Convolutional Neural Networks (3D CNN) to detect facial action parts under spatial constraints. In the work by Liu *et al.* [45] they propose to use a Boosted Deep Belief Network to jointly learn the best set of features to describe expression related facial appearance and a classifier on top of these features to perform emotion recognition. Jung *et al.* [46] approached the task by using deep learning techniques. Specifically, their method combines two deep networks: the deep temporal appearance network (DTAN) and the deep temporal geometry network (DTGN). The DTAN receives as input raw images, whereas the DTGN receives the position of the facial landmarks points. Thus, the DTAN learns to extract appearance features and the DTGN extracts geometrical features. Mollahosseini *et al.* also used a deep learning approach, but in this case, they use only one CNN, with the difference that it has several Inception modules. In the work by Zhao *et al.* [32] they propose the Peak-Piloted Deep Network (PPDN) to use the peak samples (frames with maximum expression) to supervise the feature responses for the non-peak frames of the same emotion and the same subject. Their approach is to minimize both the classification error and the difference in the representations of both frames, and at the same time, they propose the usage of Peak Gradient Suppression (PGS) to prevent the representations of peak-frames driving towards the representations of non-peak frames.

II. The Proposed System

The block-diagram of the proposed system is shown in Fig. 2. We use the same data registration as the one used by Lucey *et al.* [11] for fair comparison: images are cropped using the provided landmarks and then frontalized. Then, we apply global contrast normalization before feeding the images to a deep convolutional neural network pre-trained with faces [39]. Contrary to most of the approaches and in the same line as Kaltwang *et al.* [10], we try to solve the regression task because it fits best to this problem. However, we finally threshold the predictions in order to get performance metrics so that we can compare to [47] or [11] as previously seen in the introduction. The following sub-sections go through the steps of the system.

- **Data Pre-processing.** As it can be seen in Figures 3 and 4, we use the provided landmarks in order to crop and frontalize the faces. Following the procedure in [11], we use Generalized Procrustes Analysis (GPA) to align the landmarks [48]. This method is no more than an extension of the Procrustes Analysis for comparing more than two ordered sets of landmarks. For the simple case, in order to align two sets $X = \{x_1, x_2, ..., x_n\}$, $Y = \{y_1, y_2, ..., y_n\}$ of N landmarks, one has to (i) move their centroids $\bar{x}, \bar{y}$ to the origin (ii) find their scaling factor s:

$$s = \frac{\sqrt{\sum (x_i - \bar{x})^2 + (y_i - \bar{y})^2}}{N} \forall x_i, y_i \in X, Y, \quad (1)$$

so that we can remove it from the landmarks by dividing them by s. Then, one can find the rotation θ between two sets of landmarks by optimizing the rotation angle needed to minimize the mean squared distance between the two sets. This leads to the following equation:

$$\theta = \tan^{-1}\left(\frac{\sum_{i=1}^{N}(w_i y_i - z_i x_i)}{\sum_{i=1}^{N}(w_i y_i + z_i x_i)}\right). \quad (2)$$

Then, for K sets of points, the GPA consists in choosing one of the sets as a reference in order to align the rest, use the mean of the alignment as a new reference and repeat the process until the Procrustes distance $d = \sqrt{\sum (x_i - y_i)^2}$ between the new reference and the previous one are below a threshold. Once the final reference is obtained, the images are aligned so that their respective landmarks are aligned to it. Then, Delaunay

triangulation is used to create a mesh corresponding to the dual graph of the Voronoi diagram of the points so that piecewise-affine warping can be used to get the so called *canonical normalized appearance*. As it can be seen in Fig. 4, we did not use all the provided landmarks since it forces too much the facial expression, i.e. eliminates mouth gestures and closed eyes, and we did not want to lose any pain-related information.

Contrary to the procedure described in [11] and followed by others, e.g. [28], we do not grayscale the image and we warp it to 224×244 because it is the common input size for most deep neural network models after cropping. We do not crop patches during training due to the fact that faces are already aligned so there is no need for translation invariance.

Finally, per-pixel mean subtraction is performed in order to pass real zeros for the black areas to the neural network. Global contrast normalization is then applied to ease the training of the model.

- **Facing imbalanced data** Since there are about 8K pain frames and about 40K labeled as no pain in PSPI score, it is probable that any model gets biased towards the prediction of no-pain at the cost of missing pain frames. There are two common approaches to overcome this problem: (i) balancing data, (ii) using weighted loss functions. In this work we balance the training data (i) and validate the original validation data, but we also complement the results by giving normalized scores, as proposed by [49] (i.e. balancing the validation dataset). To balance the data, we randomly under-sample the majority class, i.e the no-pain class, so that both pain and no-pain categories have the same probability to be randomly picked by the training algorithm. To create the training sequences for the RNN we also need to balance the data, but instead of balancing at the frame level, we balance at the sequence level so that there are no frame skips. This means that we sort the frames in time, split them in sequences, and discard entire sequences with no pain in all their frames until they match the number of sequences with pain inside.
- **Target pre-processing** Because MSE is very sensitive and most suited for the cases where Gaussian noise is present, it is good practice to standardize the labels, i.e. the pain levels, before training.

After data is pre-processed, it is used to train a CNN to perform the pain level recognition task. This is achieved by fine-tunning a VGG-16 CNN pre-trained with Faces [39]. Instead of using the log-likelihood objective function, we used the $L2$ between the predicted label $\hat{Y}$ and the actual label Y in an attempt to make the model get a better insight on pain detection since it is not binary and it actually proved to perform slightly better:

$$E = \frac{1}{2N} \sum_{n=1}^{N} ||\hat{y}_n - y_n||_2^2. \tag{3}$$

In order to improve the model generalization, data augmentation is used. This is done by (i) flipping images with 50% probability, and (ii), adding random noise to the reference landmarks before performing piece-wise affine warping in order to introduce small deformations to faces, see Fig. 3.

The masking and the frontalization performed during the pre-process alter the original face, resulting in an image considerably different from a non-processed face like the ones that the CNN used has been pre-trained with. These differences between the pre-training data and the fine-tunning data could affect the results obtained, because the network has learned to extract specific features from raw face images, and it may not be able to extract them from the processed faces. Thus, we also provide results with a network trained with raw faces, similar to the ones used during pre-training, and each frame is processed only to extract a crop around the face, see Fig. 3, and then the mean pixel value is subtracted to each image.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are an architecture of neural networks proposed by LeCun *et al.* that localize local features in images to extract information of the visual content [50]. CNNs are made of different types of layers, stacked on top of each other. The basic layer of a CNN is the convolution layer, which convolves a given tensor of size

$$W \times H \times D,$$

with K different filters of size

$$F \times F \times D,$$

with a stride of S between convolutions and padding the input with P zeros. This convolution of the input by K filters outputs a tensor with dimensions:

$$W' \times H' \times D',$$

where

$$W' = (W - F + 2P)/S + 1,$$
$$H' = (H - F + 2P)/S + 1,$$
$$D' = K.$$

The values of the convolution filters are learned by initializing them randomly and updating them by performing gradient descent using the backpropagation algorithm [51]. To compute the error for a given input to the network, the last layer of the network is a loss layer which computes the error between the ground truth label of an input image and the predicted output for that image. This error at the output is backpropagated to previous layers in order to compute the gradients for the weights of previous layers.

This architecture is specially designed to capture 2D information, so it performs very well on images, where pixels intensities are related to their neighbors. The recent increase in computational power provided by GPUs and the availability of large datasets like Imagenet [52] have made the initial CNN implementations evolve to very deep networks [36], [37]. These deep networks have been proven to perform very well in a variety of computer vision tasks such as human action recognition [53], handwritten digit recognition [54] or automatic face detection [55].

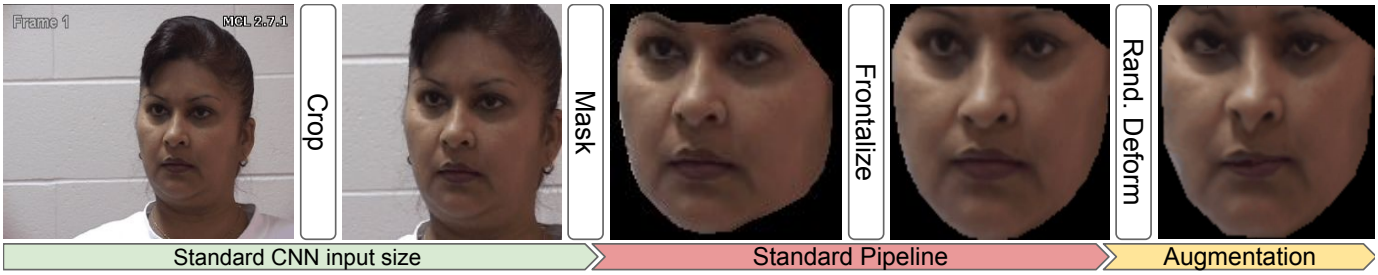

Figure 3: **Pre-processing pipeline**. The different stages of data pre-processing that were applied to the image. First, the image is aligned and cropped so as to fit the standard CNN input size. Then, it is masked and frontalized using piece-wise affine warping to match the standard pipeline proposed in [47]. Finally we perform data augmentation by applying landmark-based random deformations.

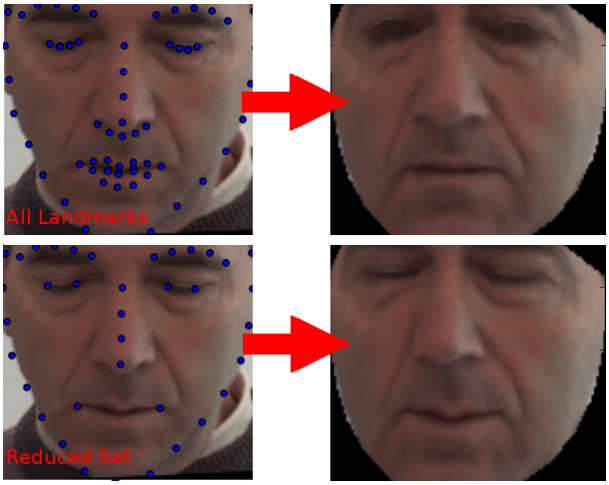

Figure 4: **Frontalized Images.** This figure shows the difference between frontalizing using all the provided landmarks or a coarser subset. It can be seen that using a smaller subset, the eyes preserve their state and the line of the mouth is more similar to the original frame.

B. Using temporal information

Although we are using video data, the previous sections only deal with the problem of labeling isolated frames. Thus, temporal information can still be used in order to improve the model. In order to take it into account, similarly to the work of [38], the features from the `fc6` layer are extracted and used to feed a recurrent neural network (RNN). This kind of neural nets is especially suited for sequential data since their neurons do not only have connections (weights) between the next layers but to themselves, which are used to keep information from previous inputs. Since they have to be unrolled, the training of this kind of networks is done with an extension of the back-propagation algorithm [51], called back-propagation through time BPTT [56].

In this work we use LSTM, a type of RNN which is capable of learning long-term dependencies present on sequential data. Standard RNNs are theoretically capable of learning long-term dependencies, but in practice, it is difficult to train them because the gradients tend to either explode or vanish [57]. LSTM differs from standard RNN because it has a cell state controlled by 3 gates, which decide how much information should be let through. These gates are known as forget, input and output gates, see 2. The amount of information that is let through each gate is controlled by a point-wise multiplication and sigmoid function, as the sigmoid function output is between 0 and 1, indicating how much of the information should let through the gate.

At each time-step, the input gate is computed depending on the input to the LSTM for that time-step and the previously hidden state. The cell state candidate is also computed by:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{4}$$

$$\hat{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c). \tag{5}$$

Then output of the forget get is computed as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f). \tag{6}$$

And when the forget and input gates have determined how much information of the previous cell state C_{t-1} and the new cell state candidate $\hat{C}_t$ should be let through, the cell state for the current time-step is computed:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t. \tag{7}$$

Then, the state can be used in order to predict the output of the cell:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{8}$$

$$h_t = o_t * \tanh C_t. \tag{9}$$

In order to train the RNN for pain detection, we used the MSE loss since it better suits the nature of the problem, where pain levels have distances in the output space. In case we need to compare in terms of binary accuracy, we can just use a binary threshold. In fact, we empirically found that using the cross-entropy error for binary classification yielded worse performance than just using a threshold after regression. Concretely, we could only reach 81% of accuracy on the test set with the initial settings shown in Table II, which presents a 83.1% for the same model after regression and thresholding.

To train the LSTM, first a feature vector has to be extracted for each image, being this vector the input to the LSTM. We can think of this feature vector as a low-dimensional representation of the image in the feature space. To create this vector for each frame, the frame is processed through the VGG-16

	Feature descriptors	Classifier	Performance Measure	Metric	Score	Use all images
Lucey *et al.* [58]	PTS, APP,	SVM	Leave one subject out	AUC	78%	Yes
Lucey *et al.* [59]	PTS, APP,	SVM	Leave one subject out	AUC	78.4%	Yes
Lucey *et al.* [11]	SPTS, CAPP	SVM	Leave one subject out	AUC	83.9%	Yes
Lucey *et al.* [47]	SPTS, SAPP, CAPP	SVM	Leave one subject out	AUC	84.7	Yes
Kaltwang *et al.* [10]	PTS, DCT, LBP	RVR	Leave one subject out	MSE, PCC, ICC	1.39, 0.59, 0.50	Yes
Florea *et al.* [60]	HoT	SVR	Leave one subject out	MSE, PCC	1.21, 0.53	Yes
Zhou *et al.* [38]	learnt	RCNN	Leave one subject out	MSE, PCC	1.54, 0.65	Yes
Zhao *et al.* [32]	LBP,Gabor	OSVR-L1, OSVR-L2	Leave one subject out	MAE,PCC,ICC	0.81, 0.60, 0.56	Yes
Ashraf *et al.* [9]	S-PTS, S-APP, C-APP	SVM	Leave one subject out	Hit rate	82%	No
Hammal *et al.* [61]	CAPP	SVM	Leave one subject out	Recall, F1 Precision	61%, 57% 65%	No (Only 15%)
Rudovic *et al.* [20]	LBP	KCORF	Custom split	F1	40.2%	No
Khan *et al.* [23]	PHOG, PLBP	SVM, DT RF, 2NN	10 fold CV	Accuracy	96.4%	Yes
Rathee *et al.* [30]	TPS	SVM	Leave one frame out	Accuracy	96.0%	Yes
Pedersen *et al.* [28]	Custom features	SVM	Leave one subject out	AUC, Accuracy	96.5, 86.1%	No

Table I: **Summary of previous approaches.** This table compares the experimental setup of previous approaches to solve the task of automatic pain detection. We compare our method against the previous approaches that have used a subject-exclusive leave-one-subject-out performance measure and do not discard any painful image.

CNN fine-tuned to perform pain level detection and the outputs of the a fully-connected layer are used as the encoding for that frame. As it can be seen in Table III, we found that the outputs in the `fc6` layer had less temporal invariability than the ones from the `fc7` and thus, the former yielded better performance when fed to the LSTM. Hence, the `fc6` is always used for comparison with the state-of-the-art. This process results in M feature vectors v where M is the number of frames and $v \in \mathbb{R}^{4096}$ since the `fc6` layer of the VGG-16 network has 4096 units. Then, the M feature vectors have to be grouped together in sequences of length ρ. The sequences are created so that each frame is the last of a sequence once, e.g if the first sequence is $s_0 = \{v_0, v_1, ..., v_{n-1}, v_n\}$, the next sequence is $s_1 = \{v_1, v_2, ..., v_n, v_{n+1}\}$. Each sequence s is labeled with one label t, corresponding to the label of the last frame of the sequence. In the classification task, t is a binary one-hot vector $t \in \{0, 1\}^2$, and for the regression task t is a real number $t \in \mathbb{R}$. As each sequence has only one label, only the hidden state of the last time-step h_{t_n} is used to compute the output of the network.

Hence, the label of a frame is predicted taking into account the past ρ frames. For this problem, we found that $\rho = 16$ worked well, and an LSTM [40] RNN is used in order to avoid the problem of gradient vanishing for long sequences. The network is optimized with ADAM since it has proved to be more stable than SGD with momentum [62].

III. Experiments and Results

As said in the previous sections, we center our experimentation on The UNBC-McMaster Shoulder Pain Expression Archive Database [11]. In addition, we prove the generality of our model by testing it on the Cohn Kanade+ face emotion detection dataset [42] and obtaining competitive results.

A. Results on Pain Recognition

A quick skim through the pain detection literature concerning the database will show the reader that there are multiple benchmark procedures. While the original paper [11] and some posterior ones [28] use leave-one-subject-out cross-validation, others like [20], [23], and [30] use k-fold cross-validation or even leave-one-frame-out cross-validation. In addition, Jeni *et al.* face the problem of data imbalance in [49], proposing normalized metrics that take the skew into account.

In Table I there is a summary of previous approaches to performing pain detection on the same dataset, indicating the method used to extract features and the classifier or regressor trained with those features. It also shows the metric used to evaluate their approach, along with the score obtained and the performance measure. The main difference between most of the listed previous approaches and our approach is that they manually extract a set of features, and then train a model with them, whereas we use an end-to-end deep learning model which learns to extract features from the data and how to combine them to give the correct output. Our approach is also based on Convolution Neural Networks as in [38], but in contrast, we apply temporal modeling using LSTM onto the features learned from the VGG_faces network. This is different from the method proposed in [38], which discards the temporal information of the data when considering pre-trained features from VGG_Faces, and considers temporal information on low-discriminative features, since the RCNN is learnt from scratch in an unbalanced, quite small dataset (even smaller in [38], since no data augmentation pre-processing is applied).

In this work, we compare within the dataset authors' scheme: AUC score on leave-one-subject-out cross-validation, since subject-exclusiveness increases the confidence that the model will behave similarly with new data. In addition to comparing our model in a binary setting by using the AUC score, we also test it against other state-of-the-art continuous prediction models with the Intraclass Correlation Coefficient (ICC), Pearson Correlation Coefficient (PCC), the Mean Square Error (MSE), and the Mean Absolute Error (MAE). For the continuous setting, we aggregated the pain levels as indicated in [32] so that the levels 4 and 5 are merged, as well as 6+, that become the 5th level.

	Normalized [49]		Unbalanced	
Metric	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Align	77.1	83.2	83.1	83.1
Align + Fron.	83.2	82.4	86.4	82.1
Align + Front. + Data aug.	85.9	89.9	88.8	89.9
Aligned Crop	80.8	90.0	87.5	89.6
Aligned Crop + LSTM	83.8	90.1	90.3	91.3

Table II: **Unbalanced and normalized scores.** This table reports the accuracy and area under the ROC curve obtained by different versions of our method.

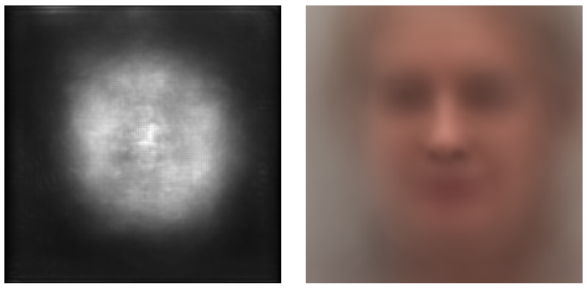

Figure 5: **Average saliency map and average face.** The first picture (a) corresponds to the average saliency map computed for each image as described by Simonyan *et al.* [63]. the second picture is the average of all the training images. The saliency map shows where the CNN is looking to decide the level of pain of a frame.

In our case, and only for comparison purposes, we also trained on aligned and canonical normalized faces but including data augmentation to add robustness to the model predictions. In Table II, we show the effect of the different stages of pre-processing shown in Fig. 3 on the performance of the model. Specifically, it can be seen that the aligned frontalized facial landmarks proposed in [11] already provides a good performance, but the VGG_faces model is not pre-trained with similar kind of images [39]. In fact, it is interesting that with canonically normalized appearance, the position and translation invariances of the faces are not enough to compensate their difference with the pre-trained model. We also found very important the mean subtraction step since the pre-trained model was trained with faces with some background and the canonically normalized appearance contains a black background. Hence, subtracting the global pixel mean was making all those zeros to be non-zero and thus lower the performance. The solution was to subtract the per-pixel mean. The best score for the AUC metric, 89.9 is achieved by considering the so popular pre-processing step, as used in [11].

The last 2 rows in Table II show the results obtained by our model when it is trained with centrally cropped Procrustes aligned faces. With this different setting, the performance of the model is enhanced, only matched by the canonically normalized setting when heavy data augmentation was used (face deformations). The main reason to this gain is due to the fact that VGG_Faces is pretrained with millions of raw images.

A possible drawback of keeping the image background could be that the CNN is helped by non-facial information (such as the arms) to improve its performance. In order to verify that the model is ignoring the background and that it is using only face information, we performed a class saliency visualization as described by Simonyan *et al.* [63]. In Fig. 5 it can be seen the average saliency map compared to the mean face, and by comparing both pictures it can be seen that the network bases its decision looking at the face region, without using background information. The average saliency map has been obtained by computing the saliency map of all the images and averaging them. According to [63] the saliency map of an image can be thought as the magnitude of the derivative of the output S_c with respect to the input image I, because the magnitude of the derivative indicates which pixels need to be changed the least to affect the output the most, and therefore those pixels correspond to the region of the image that the network is using to give its output. The derivative is computed as following:

$$w = \frac{\partial S_c}{\partial I}, \tag{10}$$

and the saliency map $M \in \mathbb{R}^{m \times n}$ for an image $I \in \mathbb{R}^{m \times n}$ is computed as:

$$M_{ij} = max_c |w_{h(i,j,c)}|, \tag{11}$$

where $h(i, j, c)$ is the index of the element in w that corresponds to the i-th row, j-th column and c-th colour channel value of the image I. As the saliency map does not have a color dimension, the maximum magnitude of w across all colour channels is selected to create the map.

The UNBC-McMaster Shoulder Pain Expression Archive Database is unbalanced, meaning that there are a lot more frames labeled as zero pain than frames labeled with some level of pain. There is a total of 48398 frames coded with a pain intensity, 40029 of them being labeled as zero pain-intensity. This means that the 83.6% of the examples of the dataset belong to the same class, whereas only the other 16.4% examples have some level of pain [11]. As stated by the authors in [49] the results of the accuracy metric is influenced by the skew in the testing data, whereas the AUC metric is not affected that much. Therefore, to avoid providing a score which is influenced by the skew in the data set, in Table II the first two columns correspond to the accuracy and AUC obtained when the score is skew normalized to mitigate the effect of imbalanced data. The last two columns correspond to the scores obtained testing the models with an unbalanced distribution. In the same way as the authors in [49], to calculate the skew normalized scores shown in Table II, we undersample the majority class at test time. This means that we randomly choose a set of no-pain samples (the majority class) that has as many images as the pain class (the minority class). Then, the normalized scores provided are calculated based on those samples. As stated by [49] the results of the accuracy metric are influenced by the skew in the testing data, whereas the AUC metric is not affected that much. That is why the

	AUC
Lucey *et al.* [11]	83.9
Lucey *et al.* [47]	84.7
Aligned crop (Ours)	89.6
Frontalization (Ours)	89.9
Aligned crop + LSTM on fc7 (Ours)	**91.3**
Aligned crop + LSTM on fc6 (Ours)	**93.3**

Table III: **Comparison against binary leave-one-subject-out methods with AUC scores.**

	MAE	MSE	PCC	ICC
Kaltwang *et al.* [10]	-	1.39	0.59	0.50
Florea *et al.* [60]	-	1.21	0.53	-
Zhou *et al.* [38]	-	1.54	0.64	-
Zhao *et al.* [32]	0.81	-	0.60	**0.56**
Aligned crop + LSTM	**0.5**	**0.74**	**0.78**	0.45

Table IV: **Comparison against continuous leave-one-subject-out methods with MAE, MSE, PCC, and Intraclass Correlation (ICC).**

accuracy scores change significantly when score normalization is applied and the AUC scores don't differ much. Accuracies are reported with a threshold interval of $[0, 1)$ for no-pain and $[1, \infty)$ for pain. It is important to remark that just a square crop centered on the nose of the subjects already performed very good in terms of AUC. However, for a fair comparison with previous work, scores for cut faces are also provided. Fig. 2 shows a fragment of the ground-truth data compared to the predictions of our model. It can be seen the model is highly correlated with the data and most of the mistakes are due to frontier effects. E.g. when a subject just stopped to feel pain, muscles relax with some lag. A similar effect happens when a subject reported pain before the facial expression completely changed.

Tables III and IV show the achieved model is competitive enough to achieve state-of-the-art results using the thorough leave-one-subject-out setting. A more detailed analysis of the binary performance of our model has been conducted, evaluating the results on each subject. Table V shows the number of pain frames and no-pain frames per subject, indicating how many of them have been correctly classified by our model. As it can be seen in Table III, using the same preprocessing as [47], our model already outperforms the previous state-of-the-art AUC scores. Namely, Lucey *et al.* [47] train a model to detect the presence of facial action units (AUs) from a set of facial features, while our model tries to directly find the best hierarchy of features to infer pain from the pixel level. Then, [47] use these features to train an SVM to detect each AU while the neural network is end-to-end, i.e. it learns to extract the features and also learns to use them to predict the level of pain. Furthermore, when frames are just aligned using Procrustes analysis, we leverage all the potential of the pre-trained model, not only outperforming previous AUC scores by a large margin, but achieving state-of-the-art results in terms of MAE, MSE, and PCC; when compared with the most recent literature (as it can be seen in Table IV).

Summarizing, we have demonstrated that considering the raw image and temporal information at the pixel level allows our model to outperform the results obtained by previous canonical normalized appearance [11] approaches.

	Not pain		Pain	
Subject	**Correct**	**Total**	**Correct**	**Total**
0	1807	1827	122	221
1	354	408	15	40
2	547	571	60	133
3	1461	1472	57	64
4	1867	2059	158	181
5	2148	2171	463	517
6	876	1000	339	408
7	2344	2403	45	93
8	2486	2699	539	821
9	1060	1116	55	100
10	2277	2361	350	455
11	1371	1396	42	76
12	1564	1863	468	505
13	913	944	59	80
14	3034	3116	45	148
15	2026	2164	267	524
16	428	641	784	959
17	713	734	183	354
18	1376	1376	71	160
20	806	844	494	1076
21	1421	1478	218	442
22	1603	1613	103	179
23	634	684	37	84
24	300	311	376	393

Table V: **Number of correctly classified pain and no-pain frames for each subject.** This table shows the number of pain and no-pain frames per subject, and how many of them are correctly classified. It can be seen that the main source of classification error is subject 20.

B. Results on Emotion Recognition

Pain recognition from facial gestures is a specific task within the broader task of facial expression recognition. In order to evaluate the effectiveness and robustness of our proposed method, we apply it to the task of emotion recognition from facial pictures. Facial expressions can show different human emotions such as anger, disgust or happiness [64] so the task of emotion recognition from pictures of faces can be approached as a facial expression recognition task. Our method for pain recognition can be adapted to perform facial expression recognition very easily. For pain detection we perform a regression task, i.e. predicting the pain intensity of a face picture. To switch to emotion detection, we must now perform a classification task. To do so, we changed the number of output units in the output layer of the CNN from 1 output unit to N, where N is the number of emotions we want to recognize in one-hot encoding. The loss function was also be changed to the cross-entropy error between the correct output y and the predicted output $\hat{y}$ as defined by the equation 12:

$$E(y, \hat{y}) = \sum_{n=1}^{N} y_n log(\hat{y}_n) \tag{12}$$

The output of the network $\hat{y}$ is the result of applying the softmax function to the outputs of the last layer, and the true label y, which is the one-hot representation of the emotion label assigned to a sample. To test our method on emotion

	Accuracy (%)
Zhong *et al.* [43]	89.9
Liu *et al.* [44]	92.4
Mollahosseini *et al.* [68]	93.2
Liu *et al.* [69]	94.2
Sikka *et al.* [66]	95.1
Liu *et al.* [45]	96.7
Jung *et al.* [46]	96.9
Zhao *et al.* [65]	**97.3**
Aligned crop (Ours)	94.5
Aligned crop + LSTM (Ours)	97.2

Table VI: **Results on the CK+ dataset**

recognition we used the Extended Cohn-Kanade Emotion Dataset (CK+) [42].

1) CK+ Dataset: The emotion recognition CK+ dataset [42] has 593 sequences of 123 subjects which are FACS coded at the peak frame. In each sequence, the subject face evolves from a neutral face to a peak facial expression. Only 327 of the sequences are labeled with one of the following seven emotions: anger, contempt, disgust, fear, happy, sadness, surprise. In Fig. 6 there is an example of a peak frame for each of the seven emotions present in the dataset. Following the trend in other works [43, 65, 66], we split the sequences into 10 subject-exclusive folds in order to perform a leave-one-fold-out cross-validation to test our method on this dataset. To make sure that the classes are evenly distributed among folds, the subjects are randomly separated into 10 groups. In the same way as in other works [45, 65] we select the last three frames of each sequence to train the CNN. To train the LSTM we must provide fixed-length sequential inputs, and as the videos vary in duration, from 10 to 60 frames approximately, we have chosen the length of the sequences to be 10. For each video, we generate three different sequences of length 10, each sequence ending in one of the last three frames. If there aren't enough frames in the video to build a sequence of length 10, the first frame is repeated at the beginning of the sequence. The results provided for the CK+ dataset are obtained by training on 9 of the 10 folds and leaving one out for testing, and repeating the process until each fold has been used for testing at least one. The accuracy provided is the average within the 10 folds.

2) Results on CK+: We provide two results for the CK+ dataset, the baseline accuracy obtained by the emotion classifier built on top of the CNN and the accuracy obtained by the LSTM model. In Table VI a comparison of our method scores against other state-of-the-art procedures reported in the literature can be seen. The results shown in the table are from seven emotion classes: anger, contempt, disgust, fear, happy, sadness, and surprise. The confusion matrix of the predictions on the test folds can be seen at Fig. 7. Other works [67] provide scores for the eight class problem where the neutral emotion is added. We can not construct sequences ending in a neutral frame because the neutral frame is always the first one, so we do not provide results for this task.

IV. Conclusions

Pain recognition has been proved to be an important task for health-care. In this work we have faced the task of binary pain recognition on facial images from the deep learning perspective achieving state-of-the-art results when compared to leave-one-subject-out setups. This, however, has also exposed the problem of stating which is the correct comparison methodology since results from other works have been provided in terms of accuracy, AUC, subject exclusive and non-exclusive settings. We believe subject-exclusiveness is crucial and thus, provided all the results computed this way. Our approach of training a deep CNN for pain-level estimation already provided good results, and we have proved that using an RNN to exploit the temporal relation between frames improves the results even more. By training a CNN end-to-end to perform pain-level estimation our approach obtained an AUC of 89.6, increasing up to 93.3 when that same CNN is used to the extract features to train the RNN. Moreover, we prove the generality of our method by obtaining an accuracy of 97.2% on the CK+ facial emotion recognition dataset, a competitive score when compared to the state-of-the-art (97.3% in [65]).

References

[1] A. Gawande. *The Checklist Manifesto: How to Get Things Right*. Macmillan, Apr. 1, 2010. 225 pp. ISBN: 978-1-4299-5338-2.

[2] G. P. Joshi and B. O. Ogunnaike. "Consequences of Inadequate Postoperative Pain Relief and Chronic Persistent Postoperative Pain". In: *Anesthesiology Clinics of North America* 23.1 (Mar. 2005), pp. 21–36.

[3] K. O. Anderson et al. "Minority cancer patients and their providers: pain management attitudes and practice". In: *Cancer* 88.8 (Apr. 15, 2000), pp. 1929–1938.

[4] J. A. Encandela. "Social science and the study of pain since Zborowski: a need for a new agenda". In: *Social Science & Medicine (1982)* 36.6 (Mar. 1993), pp. 783–791.

[5] J. E. Brown et al. "Towards a physiology-based measure of pain: patterns of human brain activity distinguish painful from non-painful thermal stimulation". In: *PloS One* 6.9 (), e24124.

[6] K. D. Craig, K. M. Prkachin, and V. E. "The facial expression of pain". In: *Handbook of pain assessment*. Ed. by D. C. Turk and R. Melzack. New York, NY, US: Guilford Press, 1992, pp. 257–276.

[7] B. Gholami, W. M. Haddad, and A. R. Tannenbaum. "Agitation and pain assessment using digital imaging". In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 2009 (2009), pp. 2176–2179.

[8] G. C. Littlewort, M. S. Bartlett, and K. Lee. "Automatic coding of facial expressions displayed during posed and genuine pain". In: *Image and Vision Computing*. Visual and multimodal analysis of human spontaneous behaviour: 27.12 (Nov. 2009), pp. 1797–1803.

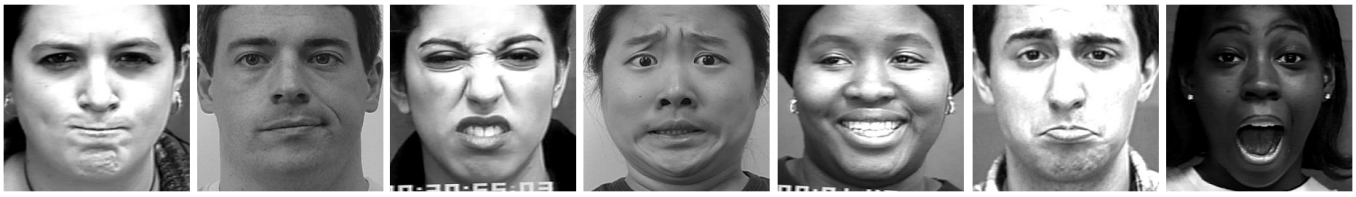

Figure 6: **Examples of emotion frames** This figure shows one frame of each of the seven emotions. From left to right: anger, contempt, disgust, fear, happiness, sadness, surprise.

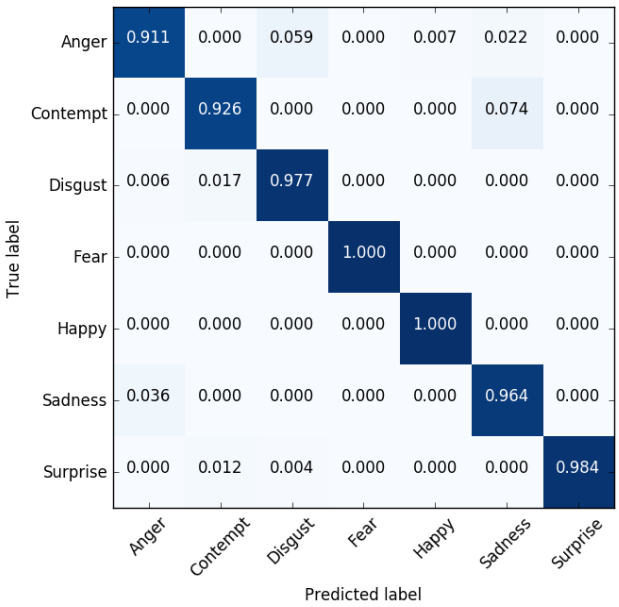

Figure 7: **Emotion detection confusion matrix.** Confusion matrix for the task of emotion detection in the CK+ dataset for the seven classes.

[9] A. B. Ashraf et al. "The painful face – Pain expression recognition using active appearance models". In: *Image and Vision Computing*. Visual and multimodal analysis of human spontaneous behaviour: 27.12 (Nov. 2009), pp. 1788–1796.

[10] S. Kaltwang, O. Rudovic, and M. Pantic. "Continuous Pain Intensity Estimation from Facial Expressions". In: *Advances in Visual Computing*. Lecture Notes in Computer Science 7432. Springer Berlin Heidelberg, July 16, 2012, pp. 368–377.

[11] P. Lucey et al. "Painful data: The UNBC-McMaster shoulder pain expression archive database". In: *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*. Mar. 2011, pp. 57–64.

[12] K. M. Prkachin and P. E. Solomon. "The structure, reliability and validity of pain expression: evidence from patients with shoulder pain". In: *Pain* 139.2 (Oct. 15, 2008), pp. 267–274.

[13] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System - The Manual on CD-ROM*. 2nd edition. A Human Face, 2002.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active Appearance Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (2001), pp. 681–685.

[15] O. Rudovic, V. Pavlovic, and M. Pantic. "Context-sensitive dynamic ordinal regression for intensity estimation of facial action units". In: *IEEE transactions on pattern analysis and machine intelligence* 37.5 (2015), pp. 944–958.

[16] M. Kim and V. Pavlovic. "Structured output ordinal regression for dynamic facial emotion intensity prediction". In: *European Conference on Computer Vision*. Springer. 2010, pp. 649–662.

[17] M. F. Valstar and M. Pantic. "Fully automatic recognition of the temporal phases of facial actions". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.1 (2012), pp. 28–43.

[18] O. Rudovic, V. Pavlovic, and M. Pantic. "Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2634–2641.

[19] Z. Ming et al. "Facial Action Units intensity estimation by the fusion of features with multi-kernel Support Vector Machine". In: *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. Vol. 6. IEEE. 2015, pp. 1–6.

[20] O. Rudovic, V. Pavlovic, and M. Pantic. "Automatic Pain Intensity Estimation with Heteroscedastic Conditional Ordinal Random Fields". In: *Advances in Visual Computing: 9th International Symposium, ISVC 2013, Rethymnon, Crete, Greece, July 29-31, 2013. Proceedings, Part II*. Ed. by G. Bebis et al. Berlin, Heidelberg: Springer, 2013, pp. 234–243. ISBN: 978-3-642-41939-3.

[21] K. Sikka, A. Dhall, and M. Bartlett. "Weakly supervised pain localization using multiple instance learning". In: *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). Apr. 2013, pp. 1–8.

[22] K. Sikka, A. Dhall, and M. S. Bartlett. "Classification and weakly supervised pain localization using multiple segment representation". In: *Image and Vision Computing*. Best of Automatic Face and Gesture Recognition 2013 32.10 (Oct. 2014), pp. 659–670.

[23] R. Khan et al. “Pain detection through shape and appearance features”. In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*. 2013 IEEE International Conference on Multimedia and Expo (ICME). July 2013, pp. 1–6.

[24] Z. Zafar and N. Khan. “Pain Intensity Evaluation through Facial Action Units”. In: *2014 22nd International Conference on Pattern Recognition (ICPR)*. 2014 22nd International Conference on Pattern Recognition (ICPR). Aug. 2014, pp. 4696–4701.

[25] R. Irani, K. Nasrollahi, and T. Moeslund. “Pain recognition using spatiotemporal oriented energy of facial muscles”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). June 2015, pp. 80–87.

[26] R. Irani et al. “Spatiotemporal Analysis of RGB-D-T Facial Images for Multimodal Pain Level Recognition”. In: (2015).

[27] L. Presti and M. Cascia. “Using Hankel Matrices for Dynamics-Based Facial Emotion Recognition and Pain Detection”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015, pp. 26–33.

[28] H. Pedersen. “Learning Appearance Features for Pain Detection Using the UNBC-McMaster Shoulder Pain Expression Archive Database”. In: *Computer Vision Systems*. Ed. by L. Nalpantidis et al. Lecture Notes in Computer Science 9163. Springer International Publishing, July 6, 2015, pp. 128–136.

[29] N. Neshov and A. Manolova. “Pain detection from facial characteristics using supervised descent method”. In: *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Vol. 1. Sept. 2015, pp. 251–256.

[30] N. Rathee and D. Ganotra. “A novel approach for pain intensity detection based on facial feature deformations”. In: *Journal of Visual Communication and Image Representation* 33 (Nov. 2015), pp. 247–254.

[31] F. L. Bookstein. “Principal warps: thin-plate splines and the decomposition of deformations”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 11.6 (June 1989), pp. 567–585.

[32] R. Zhao et al. “Facial Expression Intensity Estimation Using Ordinal Information”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3466–3474.

[33] Y. Bengio. “Learning Deep Architectures for AI”. In: *Found. Trends Mach. Learn.* 2.1 (Jan. 2009), pp. 1–127. ISSN: 1935-8237.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks.” In: *NIPS*. 2012, pp. 1106–1114.

[35] Y. Bengio. “Deep Learning of Representations: Looking Forward”. In: *arXiv:1305.0445 [cs]* (May 2, 2013). arXiv: 1305.0445.

[36] C. Szegedy et al. “Going Deeper with Convolutions”. In: *arXiv:1409.4842 [cs]* (Sept. 16, 2014). arXiv: 1409.4842.

[37] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014).

[38] J. Zhou et al. “Recurrent Convolutional Neural Network Regression for Continuous Pain Intensity Estimation in Video”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26 - July 1, 2016*. 2016, pp. 1535–1543.

[39] O. M. Parkhi, A. Vedaldi, and A. Zisserman. “Deep Face Recognition”. In: *British Machine Vision Conference*. 2015.

[40] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780.

[41] F. A. Gers, J. Schmidhuber, and F. Cummins. “Learning to forget: Continual prediction with LSTM”. In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[42] P. Lucey et al. “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE. 2010, pp. 94–101.

[43] L. Zhong et al. “Learning active facial patches for expression analysis”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2562–2569.

[44] M. Liu et al. “Deeply learning deformable facial action parts model for dynamic expression analysis”. In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 143–157.

[45] P. Liu et al. “Facial expression recognition via a boosted deep belief network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1805–1812.

[46] H. Jung et al. “Deep Temporal Appearance-Geometry Network for Facial Expression Recognition”. In: *CoRR* abs/1503.01532 (2015).

[47] P. Lucey et al. “Automatically Detecting Pain in Video Through Facial Action Units”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 41.3 (June 2011), pp. 664–674.

[48] J. C. Gower. “Generalized procrustes analysis”. In: *Psychometrika* 40.1 (Mar. 1975), pp. 33–51.

[49] L. Jeni, J. Cohn, and F. De la Torre. “Facing Imbalanced Data–Recommendations for the Use of Performance Metrics”. In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII). Sept. 2013, pp. 245–251.

[50] Y. LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (Oct. 9, 1986), pp. 533–536.

[52] J. Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255.

[53] S. Ji et al. “3D convolutional neural networks for human action recognition”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.1 (2013), pp. 221–231.

[54] D. Ciresan, U. Meier, and J. Schmidhuber. “Multi-column deep neural networks for image classification”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 3642–3649.

[55] S. S. Farfade, M. J. Saberian, and L.-J. Li. “Multi-view face detection using deep convolutional neural networks”. In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM. 2015, pp. 643–650.

[56] P. J. Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (Oct. 1990), pp. 1550–1560.

[57] S. Hochreiter, Y. Bengio, and P. Frasconi. “Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies”. In: *Field Guide to Dynamical Recurrent Networks*. Ed. by J. Kolen and S. Kremer. IEEE Press, 2001.

[58] P. Lucey et al. “Automatically detecting action units from faces of pain: Comparing shape and appearance features”. In: *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE. 2009, pp. 12–18.

[59] P. Lucey et al. “Automatically detecting pain using facial actions”. In: *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*. IEEE. 2009, pp. 1–8.

[60] C. Florea, L. Florea, and C. Vertan. “Learning pain from emotion: transferred hot data representation for pain intensity estimation”. In: *Computer Vision-ECCV 2014 Workshops*. Springer. 2014, pp. 778–790.

[61] Z. Hammal and J. F. Cohn. “Automatic detection of pain intensity”. In: *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM. 2012, pp. 47–52.

[62] D. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015.

[63] K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*. 2014.

[64] T. Kanade, J. F. Cohn, and Y. Tian. “Comprehensive database for facial expression analysis”. In: *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE. 2000, pp. 46–53.

[65] X. Zhao et al. “Peak-Piloted Deep Network for Facial Expression Recognition”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 425–442.

[66] K. Sikka, G. Sharma, and M. Bartlett. “Lomo: Latent ordinal model for facial analysis in videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5580–5589.

[67] M. Liu et al. “Au-aware deep networks for facial expression recognition”. In: *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE. 2013, pp. 1–6.

[68] A. Mollahosseini, D. Chan, and M. H. Mahoor. “Going deeper in facial expression recognition using deep neural networks”. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2016, pp. 1–10.

[69] M. Liu et al. “Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1749–1756.

Pau Rodriguez received the MSc degree in Artificial Intelligence from KU Leuven in 2015. He is currently a Ph.D. student at the Image Social Evaluation (ISE) Lab of the Computer Vision Center and the Universitat Autònoma de Barcelona, Catalonia Spain. His research interests focus on machine learning, pattern recognition, and computer vision.

Guillem Cucurull received the BSc degree in Computer Science from Universitat Autònoma de Barcelona (UAB) in 2016. He is currently a MSc student in Computer Vision and a research assistant at the the Image Sequence Evaluation (ISE Lab) research group of the Computer Vision Center and UAB. His research interests include computer vision and machine learning techniques for human motion and behaviour analysis.

Jordi Gonzàlez received the PhD degree in Computer Engineering from Universitat Autònoma de Barcelona (UAB) in 2004. He is an associate professor in computer science at the Computer Science Department, UAB. He is also a research fellow at the Computer Vision Center, where he has co-founded three spin-offs (Cloud Size Services, Visual Tagging, Care Respite) and the Image Sequence Evaluation (ISE Lab) research group. His research interests include machine learning techniques for the computational interpretation of social images, or Visual Hermeneutics.

Josep M. Gonfaus received the PhD degree in Computer Engineering from Universitat Autònoma de Barcelona (UAB) in 2012. He participated in various Pascal challenges. He co-founded a spin-off based on their research by applying computer vision and deep learning techniques for analyzing social media data (Visual Tagging). His main research interests are deep learning techniques to mimic human cognitive capabilities.

Kamal Nasrollahi received the M.Sc. degree in computer engineering from Tehran Polytechnic, in 2007, and the Ph.D. degree in electrical engineering from Aalborg University, in 2010, with a focus on computer vision. He is currently an Associate Professor with the Visual Analysis of People Laboratory, Aalborg University, Denmark. He has been involved in five international research projects. His research interests include facial analysis systems, biometrics recognition, soft biometrics, and inverse problems. He has received an IEEE Conference Best Paper Award.

Thomas B. Moeslund (M'12) received the M.Sc.E.E. and Ph.D. degrees from Aalborg University, Aalborg, Denmark, in 2003 and 1996, respectively. He is head of a computer vision group at Aalborg University. His research is focused on all aspects of computer vision and image analysis. He has been involved in 12 national and international research projects, both as coordinator, WP leader, and researcher. He is reviewer for all major journals within the field. He serves as associate editor and editorial board member for four international journals. He has coedited one special journal issue, cochaired four international workshops/tutorials, and acted as PC member/reviewer for a number of conferences. He has published 90 peer-reviewed journal and conference papers.

F. Xavier Roca received the Ph.D. degree in computer science from Universitat Autònoma de Barcelona (UAB), Cerdanyola del Vallès, Spain, in 1990. He is an Associate Professor and the Director of the Department of Computer Science, UAB. He is also a Research Fellow with the Computer Vision Center. He has been a Principal Researcher in several projects (public and private funds). He is working in technological transfer computer vision. The topics of his research are active vision, biometrics and tracking.