Pain detection from facial images using unsupervised feature learning approach

Conference Paper *in* Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference · August 2016

DOI: 10.1109/EMBC.2016.7590729

3 authors:

Reza Kharghanian
Ferdowsi University Of Mashhad

Ali Peiravi
Ferdowsi University Of Mashhad

Farshad Moradi
Aarhus University

Pain Detection from Facial Images using Unsupervised Feature Learning Approach

Reza Kharghanian, Ali Peiravi, and Farshad Moradi, *Member, IEEE*

Abstract - **In this paper a new method for continuous pain detection is proposed. One approach to detect the presence of pain is by processing images taken from the face. It has been reported that expression of pain from the face can be detected utilizing Action Units (AUs). In this manner, each action units must be detected separately and then combined together through a linear expression. Also, pain detection can be directly done from a painful face. There are different methods to extract features of both shape and appearance. Shape and appearance features must be extracted separately, and then used to train a classifier. Here, a hierarchical unsupervised feature learning approach is proposed in order to extract the features needed for pain detection from facial images. In this work, features are extracted using convolutional deep belief network (CDBN). The extracted features include different properties of painful images such as head movements, shape and appearance information. The proposed model was tested on the publicly available UNBC MacMaster Shoulder Pain Archive Database and we achieved near 95% for the area under ROC curve metric that is prominent with respect to the other reported results.**

I. INTRODUCTION

Pain is usually reported by patient's self-report [1], either through a clinical interview or using a visual analog scale (VAS) [2]. These methods naturally have some drawbacks such as being unreliable and highly subjective. Yet another difficulty arises when patients cannot communicate. In some populations like neonates, children, unconscious patients, patients who have no ability to communicate, those who have critical conditions and are kept in intensive care unit (ICU) and patients who suffer from cerebral palsy (CP), self-report cannot be used. Therefore, in such cases, a care provider should be available to monitor the patient's physical conditions at least once every 4 hours and complete the clinical checklist with pain being one of the items in the checklist [3] and make an evaluation of pain intensity. Also, this can be done by analyzing tissue pathology, neurological signals, testing muscle strength [4] and so on all of which are invasive and inconvenient approaches.

Given the advances in machine learning, now there are computer vision systems for monitoring patients in a fully non-invasive manner that can operate around the clock and without any fatigue. Pain and its intensity are noticeable in one's face. Any movements in facial muscles can depict one's current emotional state. For example, in case of pain, we close our eyes and depending on the intensity we push them hard. Another indication of pain is lowering the eyebrows. Pain also affects the mouth.

Recently many works have been carried out in this field. In one sense, different methods can be categorized as AU-based and non-AU-based. In the case of AU-based methods, facial action units (AUs) are detected and pain can be detected using their combination. Some works have been done trying to extract features that can represent pain directly without using AUs [5, 18]. The authors in [5] have shown that direct pain estimation can be more accurate than its prediction from the AUs.

In this work, a non-AU-based method is proposed that can predict the pain with a high accuracy. Features are extracted from filters that are learned in a fully unsupervised manner. These filters are learned through a CDBN architecture. CDBN is a hierarchical generative model constructed by stacking max-pooling Convolutional Restricted Boltzmann Machine (CRBMs) on top of each other. Learned detectors (filters) can detect features of both shape and appearance defining emotional facial states. The extracted features are used to train a classifier so that it can classify a test sample into one of two predefined classes namely pain and no-pain.

The rest of the paper is organized as follows. First, a review of previous works on pain detection and its recognition is presented in section II. Then, the CDBN and its constructor CRBM are introduced in section III. The classifier used in this paper is described in section IV. Finally, the experimental results are presented in section V and the work is finally concluded in section VI.

II. RELATED WORKS

The most popular system on facial expression definition and coding, Facial Action Coding System (FACS), that was a measurement of facial movement was introduced in 1978 [6]. FACS described facial expression by coding facial muscle movements into 46 action units (AUs). Numerous good works which are capable of making a distinction between different human emotional states like being neutral, sadness, fear, surprise, anger, happiness, and disgust have been done based on this system. Some of the related algorithms are found in [7-14], which may use facial AUs or not. At first glance, it seems that it is able to detect pain or recognize its intensity. However, one must notice that there are relations between some emotional states like sadness and surprise, which makes it more difficult to use combinations of AUs to detect pain. Yet, it might be possible to use fuzzy classification by assigning some membership values to them. Anyway, this area of research is open for further

Reza Kharghanian and Ali Peiravi are with the Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. (e-mail: r_kharghanian@stu.um.ac.ir , peiravi@um.ac.ir).

Farshad Moradi is with the Integrated Circuits and Electronics Laboratory, Department of Engineering, Aarhus University, Denmark. (e-mail: moradi@eng.au.dk).

*Corresponding author: Ali Peiravi (peiravi@um.ac.ir)

978-1-4577-0220-4/16/$31.00 ©2016 IEEE

investigation and several research works have been done to achieve a more reliable and robust system.

Works on pain monitoring using AUs have been also reported [15-17]. A formulation defined to quantify pain intensity into 16 discrete levels from zero as no pain to 15 as extremely high pain is reported in [15]. The PAIN equation is based on a combination of AU intensities among 46 different AUs describing pain, (i.e. AUs 4, 6, 7, 9, 10, 12, 20, 25, 26, 27 and 43):

Pain = AU4 + max (AU6, AU7) + max (AU9, AU10) + AU43 (2)

A three step frame-based approach to continuous pain intensity estimation has been proposed in [5]. The first step in this work is shape and appearance feature extraction. The location of characteristic facial points as shape-based features and local binary patterns (LBP) are extracted from facial images with discrete cosine transform (DCT). For each set of features, a Relevance Vector Regression (RVR) is used for prediction of pain intensity. As the last step, the outputs of the three regressors are fused by computing their mean estimate or by using the forth RVR using the outputs of regressors as inputs.

An Active Appearance Model (AAM) and frame based system, which can automatically detect pain through AUs has been described in [16]. In this paper, pain detection is based on facial expressions coded by FACS. AAM can detect both shape and appearance information. They have also employed some other information to consider head motion. Moreover, they showed that fusing all AAM features can improve the performance. By the use of AAM parameters, three types of feature were extracted: the similarity normalized shape (SPTS), the similarity normalized appearance (SAPP) and the canonical normalized appearance (CAPP). Then, a Support Vector Machine (SVM) is trained using the extracted features. This SVM classifies each new input sample as either pain or no-pain and provides a score that can be used to estimate the pain intensity. This has been done by fusion of scores using Linear Logistic Regression (LLR) method [16]. The same approach is used in [17] in which detection and tracking of face and facial features is done by AAM. They tried to overcome two problems: 1) integrate the rigid and non-rigid head motion into a single feature representation, and 2) incorporate the salient temporal patterns into the classification stage. The former were tackled by using a high level representation which incorporates various modes of pain achieved using a *histogram of AUs*. The second problem was tackled utilizing a hidden conditional random field (HCRF).

So far, we argued about works that they were based on detecting AUs. In [18] a non-AU based method was proposed. Shape information using pyramid histogram of orientation gradients (PHOG) and appearance information using pyramid local binary pattern (PLBP) was used for pain detection in [18]. Then, features were used to learn multiple classifiers like SVM with χ^2 kernel, decision tree, random forest and 2 nearest neighbors. They achieved their best recognition rate using 2NN.

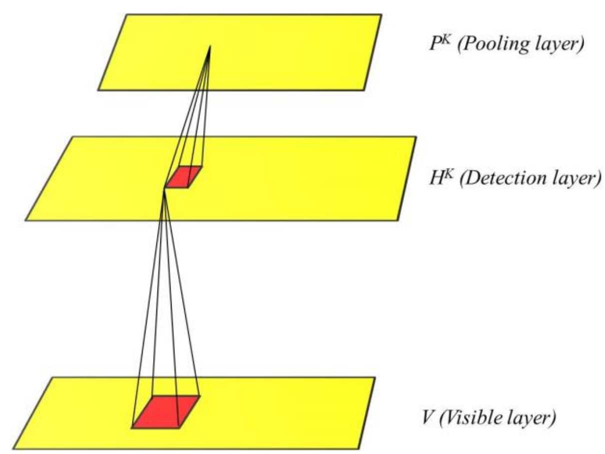
In this work, we use fully unsupervised learning of hierarchical representations in which the learned feature extractors (filters) are able to extract both shape and appearance features, simultaneously. We trained a convolutional deep belief network (CDBN), which extracts features from the lowest level i.e. edges to more complex features like object parts. Another key advantage of CDBN is its translation invariant property which is attributed to weight sharing. In the following section we describe the theory of CDBN based on a work that was done by [19].

III. CONVOLUTIONAL DEEP BELIEF NETWORK, CDBN

In this section, the basics of convolutional deep belief networks are reviewed. First, we need to describe a CRBM which is the main constructor block of CDBNs.

A. Convolutional Restricted Boltzmann Machine (CRBM)

CRBM is an extension of the standard RBM [20] to the convolutional settings. It comprises of three layers: visible, detection (hidden) and pooling layer as shown in Fig. 1. The input layer consists of real-valued units arranged in 2D form. The hidden layer consists of K groups, where each group consists of binary hidden units in 2D form. Each hidden group is obtained by using a filter, which is convolved with the input data. Filter weights are shared among all the hidden units within the group. Afterward, each group of hidden layer is reduced in both dimensions by the same factor such that a small region of the hidden group is mapped into only one unit in the pooling layer. This can be done using probabilistic max-pooling [19]. This shrinking step makes the extracted features robust against small translations in the input [19, 23].



Despite the weights of the filter, each group has a bias b_k and all visible units have a single bias c, that must be optimized.

The joint distribution of the probabilistic max-pooling CRBM with real-valued visible units is defined through its energy function as follows [19]:

$$P(v,h) = \frac{1}{Z}\exp(-E(v,h)) \tag{2}$$

$$\begin{aligned} E(v,h) = {} & \frac{1}{2}\sum_{i,j} v_{i,j} - \sum_{k}\sum_{i,j}\sum_{r,s} h_{ij}^{k} W_{rs}^{k} v_{i+r-1,j+s-1} \\ & -\sum_{k} b_k \sum_{i,j} h_{ij}^{k} - c\sum_{i,j} v_{i,j} \end{aligned} \tag{3}$$

where Z is known as the *partition function*, b_k are hidden unit biases and c is visible unit bias and *i, j* iterate over visible and hidden units, respectively.

As computing the exact gradient of the log-likelihood is intractable, model parameters can be optimized via minimization of a so called *Contrastive Divergence* (CD) [20] objective function as an approximation to maximizing the log-likelihood [22]. CD utilizes Gibbs sampling method to approximate the model distribution using the following [19]:

$$P(h_{ij}^{k} = 1 | v) = \sigma\left(\left(\tilde{W}^{k} *_{v} v\right)_{ij} + b_{k}\right) \tag{4}$$

$$P\left(v_{ij} | h\right) = \mathrm{N}\left(\sum_{k}\left(W^{k} *_{f} h^{k}\right)_{ij} + c, 1\right) \tag{5}$$

where $\sigma(.)$ is the sigmoid function and *N(.)* is normal distribution. In these equations, $*_v$ and $*_f$ stands for valid and full convolution, respectively and $\tilde{W}$ denote flipping the array horizontally and vertically.

By the use of probabilistic max-pooling [19], a simple constraint will be added to the energy function. As expressed, we divide each group of hidden layers into blocks and each block is mapped into only one unit of pooling layer. Considering probabilistic max-pooling, at most one of the units in the block may be on. Therefore, the corresponding unit in the pooling layer is on if and only if a detection unit in the block is on.

Due to the convolutional connections, overcompleteness must occur that results in trivial solutions. One way to overcome this problem is to add a penalty term to the objective function so that each hidden groups has a mean activation close to a small constant. This could be achieved by applying the following simple update rule on hidden biases vector [19]:

$$\Delta b_{k}^{sparsity} \propto p - \frac{1}{N_{H}} \sum_{i,j} P\left(h_{i,j}^{k} = 1 | v\right) \tag{6}$$

where p is a target sparsity.

B. Convolutional deep belief network

Now we can simply define a CDBN as stacking of several probabilistic-max-pooling CRBM on top of each other based on [19]. The energy function for this type of network is defined by summing the energy functions of all individual pairs of layers. The training of such a network is performed using greedy layer-wise strategy in which each CRBM is trained individually [20]. That is once a given layer is trained, its parameters are frozen, and its unit's expected values, also known as *activation,* are used as inputs for the next layer.

IV. CLASSIFICATION

Each learned expert (filter) extracts features from the input image. These extracted features are used to train a SVM as a classifier. SVM is a binary classifier [24] that can be used to classify each input image on one of the two predefined classes i.e. pain and no-pain. SVM tries to find the best hyperplane by maximizing the possible margin between positive and negative instances. Consider a two-class problem. Linear SVM classifies test sample to either classes by the following decision:

$$\begin{cases} w^{T} x > w_{0} & x \in pain \\ w^{T} x < w_{0} & x \notin pain \end{cases} \tag{7}$$

Where w is normal vector to the hyperplane and w_0 determines the offset of the hyperplane from the origin along the normal vector.

V. EXPERIMENTAL RESULTS

The publicly available UNBC-MacMaster Shoulder Pain Expression Archive dataset [25] is used in this study to evaluate the proposed algorithm. It contains a recorded video of the face of patients suffering from shoulder pain (only one arm is affected by pain). There are 200 sequences across 25 subjects (a total of 48,398 frames). All the frames are FACS coded and the PSPI (Prkachin and Solomon Pain Intensity) [15] score are provided by the creator of the database. We use this score as ground truth. We also used leave-one-subject-out strategy and training data were divided into positive and negative samples, where positive samples are all images labeled 1 or more and all other images are labeled zero (pain intensity of 0) form the negative set. In order to generate the training data we followed this strategy described here: after excluding out one subject, all the positive data in the remaining data were kept and only a small part of the negative data were used. In addition, we used a Viola-Jones detector [26] to locate the face in each image. We used these samples to train a system that comprises of a CDBN followed by an SVM. Training of the system was performed offline.

As mentioned earlier, we used CDBN that is a multilayer generative model for fully unsupervised feature learning. We formed a two-layer CDBN constructed from probabilistic max-pooling CRBM. For the first layer bases, we used natural images as done by Lee [19]. For the second layer, we used images from the database. The length of filters was set to 10 for both layers and the number of filters are set as 24 and 48, respectively. Furthermore, we used target sparsity values of 0.002 and 0.005, respectively. Also we set max-pooling ratio as 2 for each layer. Inputs to the higher layers were the max-pooled activation of the lower level layer. Fig. 2 shows these learned bases. The second layer bases visualization is performed by a linear combination of the lower layer bases [19].

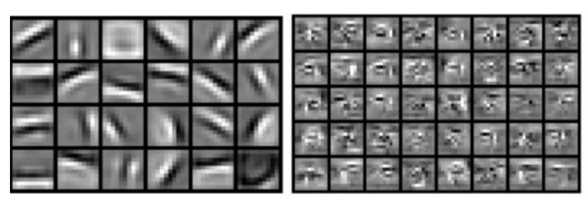


Figure 2. Learned filters of the two layers. Left, first layer filters laerned some edge-like fgeatures. Right, second layer filters learned face features.

Then we used the features extracted using learned filters for training an SVM with linear kernel duo to its ability to generalize well to unseen data. We used embedded SVM toolbox in Matlab for training and testing of the SVM. Also we used the same approach proposed in [19] to define CDBN. The results of the proposed method are shown in Table 1, which are calculated using 10-fold cross validation. The three metrics that were used to evaluate the performance of the proposed method are accuracy, F1-score and area underneath the ROC curve. This curve is obtained by plotting the true positives against the false positives as the decision threshold varies.

TABLE I. RESULTS AND COMPARSION

	Performance measurements		
	Accuracy%	*F1-Score%*	*Area under ROC %*
Proposed method	***87.2***	***86.44***	***94.48***
[16] SVM + SPTS +SAPP + CAPP	-	-	***84.7***
[18] SVM+Level2	***96.4***	-	-

Noted, the authors in [18] did not use leave-one-subject-out strategy. However, they evaluated their work on the complete database, which has a poor generalization on new unseen input data. The program was developed by Matlab 8.4.0 under a laptop configuration of core i7 2.4 GHz CPU and 8 GB memory. The proposed algorithm achieved an average time of 30 milliseconds for feature extraction and classification in the test step and it needs 80 seconds for training of the SVM and 75 seconds per epoch for training of the CDBN for the offline training step.

VI. CONCLUSION

In this paper, a method was proposed to emulate a care provider for patient monitoring and pain detection. This was done using features extracted from facial images. These features were extracted using filters learned through a CDBN. Then, these features were fed to a support vector machine to be classified into one of the two predefined classes namely pain and no-pain. Our experiments shows near 95% for area under ROC curve metric that is prominent with respect to the other reported results. Our future work will be designing a classifier that is able to determine the intensity of the pain for the instances classified as pain.

REFERENCES

[1] T. Hadjistavropoulos and K. Craig, "Social influences and the communication of pain", *Pain: Psychological Perspectives*. New York:Erlbaum, 2004.

[2] F.-X. Lesage, S. Berjot and F. Deschamps, "Clinical stress assessment using a visual analogue scale", Occupational Medicine, Sep. 2012.

[3] A. Gawande, "The checklist manifesto: How to get things right", New York: Metropolitan Books, 2010.

[4] D. Turk and R. Melzack, "The measurement of pain and the assessment of people experiencing pain," *Handbook of Pain Assessment*, Eds. New York: Guildford Press, pp 1–11, 2001.

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

[6] P. Ekman and W. Friesen, "Facial action coding system: A technique for the measurement of facial movement" Consulting Psychologists Press, Palo Alto, 1978.

[7] C. Shan, S. Gong and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study", Image and Vision Computing, Volume 27, Issue 6, pp 803–816, 4 May 2009.

[8] L. Zhang and D. Tjondronegoro, "Facial expression recognition using facial movement features", *IEEE Transactions on Affective Computing,* Volume 2, Issue 4, pp 219 – 229, June 2011.

[9] Y. Li, S. Wang, Y. Zhao and Q. Ji "Simultaneous facial feature tracking and facial expression recognition", *IEEE Transactions on Image Processing* , Volume 22, Issue 7, pp 2559 – 2573, July 2013.

[10] M. Lyons, A. Akamatsu, M. Kamachi, and J. Gyoba."Coding facial expressions with Gabor wavelets" Third IEEE International Conference on Automatic Face and Gesture Recognition, Proceedings, pp 200-205, 1998.

[11] E. Sariyanidi, H. Gunes and A. Cavallaro. "Automatic analysis of facial affect: A survey of registration, representation, and recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 37, Issue 6, pp.1113 – 1133, June 1 2015.

[12] Y. Rahulamathavan, R.-W. Phan, J. A. Chambers and D. J. Parish, "Facial expression recognition in the encrypted domain based on Local Fisher Discriminant Analysis", *IEEE Transactions on Affective Computing,* Volume 4, Issue 1, pp 83 – 92, Jan-March 2013.

[13] P. Siritanawan, K. Kotani and F. Chen "Independent subspace of dynamic Gabor features for facial expression classification", *IEEE International Symposium on Multimedia (ISM) ,pp* 47 – 54, 2014.

[14] Y. Piparsaniyan, V.K. Sharma and K.K. Mahapatra, "Robust facial expression recognition using Gabor feature and Bayesian discriminating classifier", *International Conference on Communications and Signal Processing (ICCSP),* pp 538 – 541, 2014.

[15] K. Prkachin and P. Solomon."The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain", Pain 139, pp 267–274, 2008.

[16] P. Lucey, J. F. Cohn, I. Matthews, S. Lucey S. Sridharan, J. Howlett and K. M. Prkachin ,"Automatically detecting pain in video through facial action units", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, pp 664 – 674, Volume 41 , Issue 3, May 2011.

[17] A. Ghasemi, W. Xinyu, P. Lucey, S. Sridharan and C. Fookes,"Social signal processing for pain monitoring using hidden condition random field.", IEEE Workshop on Statistical Signal Processing (SSP), pp 61-64, June 29-July 2, 2014.

[18] R. A. Khan, A. Meyer,H. Konik and S. Bouakaz," Pain detection through shape and appearance features", IEEE International Conference on Multimedia and Expo (ICME), pp 1-6, 15-19 July 2013.

[19] H. Lee, R. Grosse, R. Ranganath and A. YK Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks", Communications of the ACM, Vol 54, No 10, pp 95-103, Oct 2011.

[20] G. E. Hinton, S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets", Neural Computation, vol 18 no 7pp 1527-1554, July 2006.

[21] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks", *Advances in Neural Information Processing Systems*, 2007.

[22] G. E. Hinton, "Training products of experts by minimizing contrastive divergence", Neural Comput., vol 18, no 7, pp 1771- 1800, 2002.

[23] M. Norouzi, M. Ranjbar, G. Mori,"Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[24] VN. Vapnik,"The nature of statistical learning theory", New York: Springer-Verlag, 1995.

[25] P. Lucey, J. Cohn, K. Prkachin, P. Solomon and I. Matthews,"Painful data: The UNBCMcMaster shoulder pain expression archive database", Int'l Conf. on Automatic Face & Gesture Recognition and Workshops, IEEE, pp 57–64, 2011.

[26] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.