



## Saliency Supervision for Pain Intensity Regression

---

Conghui Li, Zhaocheng Zhu and Yuming Zhao

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 18, 2018

# Saliency Supervision: An intuitive and effective approach for pain intensity regression

Conghui Li<sup>1</sup>, Zhaocheng Zhu<sup>2</sup>, Yuming Zhao<sup>1\*</sup>

School of Electronic Information and Electrical Engineering  
Shanghai Jiao Tong University<sup>1</sup>

School of Electronics Engineering and Computer Science  
Peking University<sup>2</sup>

**Abstract.** Getting pain intensity from face images is an important problem in autonomous nursing systems. However, due to the limitation in data sources and the subjectiveness in pain intensity values, it is hard to adopt modern deep neural networks for this problem without domain-specific auxiliary design. Inspired by human vision priori, we propose a novel approach called saliency supervision, where we directly regularize deep networks to focus on facial area that is discriminative for pain regression. Through alternative training between saliency supervision and global loss, our method can learn sparse and robust features, which is proved helpful for pain intensity regression. We verified saliency supervision with face-verification network backbone [1] on the widely-used *UNBC-McMaster Shoulder-Pain* [2] dataset, and achieved state-of-art performance without bells and whistles. Our saliency supervision is intuitive in spirit, yet effective in performance. We believe such saliency supervision is essential in dealing with ill-posed datasets, and has potential in a wide range of vision tasks.

**Keywords:** regression, saliency supervision, regularization, triplet loss, multi-task training

## 1 Introduction

Excessive usage of anesthetic will cause bad effects on patients because the pain intensity of the patient is not well measured by tradition methods like skin conductance algometer or heart rate variability, which calls for a reliable approach that reports pain intensity in time. From the perspective of computer vision, the pain intensity task can be viewed as a semantic problem from single images or videos. Our job is aimed at pushing the benchmark of pain intensity to state-of-art level with a new design.

There are a few datasets that have pain intensity labels. Particularly, *UNBC-McMaster Shoulder-Pain* [2] dataset is the only dataset available for per-frame visual analysis. It contains only 200 videos of 25 patients who suffer from shoulder pain and repeatedly raise their arms and then put them down (onset-apex-offset).

---

\* Corresponding author (e-mail: arola\_zym@sjtu.edu.cn)

While all frames are labeled with pain intensities, the labels are reported by the patient, which is very subjective. Moreover, in most frames patients got zero pain intensity, making it hard to observe the pattern behind frames. Many methods [3] [4] have exploited deep neural networks trained in a data-extensive domain to alleviate the limited training data problem. For example, they finetune a well-trained face verification network with a regularized regression loss for pain intensity regression or valence-arousal estimation. In this case, their initial value is expected to be closer to some optimum for pain intensity regression, due to the similarity between domains. However, we doubt the finetuning procedure in those methods, as pain intensity values is not always a good supervision signal, especially for small dataset like *UNBC-McMaster Shoulder-Pain*.

Many people would give their reasons based on face attributes like eyes or lips when asked to determine individual’s pain intensity from his face. Inspired by this fact, we exploit Action Units (AUs) [5], a semantic representation of face attributes, to regularize the training of pain intensity regression. AUs have been utilized for facial expression recognition. For example, FATAUVA-Net [4] train a mid-level network for AU detection using the AU label, and then finetune mid-level network for valence-arousal estimation and facial expression recognition. They use the labeled AU values as additional information and get better result. Pain is also a kind of expression, and here we explicitly supervise the network to focus on the areas of AUs so that the network will pay attention to the areas related to pain which will give a closer initial value to the global optimum than the previous work [3] when finetuning the network for pain intensity regression.

Since deep neural networks work like a black box, we cannot exactly know where the network pay attention. Zeiler proposed a kind of visualizing network method [6] using deconvolution [7] to get the contribution of input to the network’s output. As we want to directly supervise the saliency map, which means the network for getting saliency should be able to merge with the origin network and support forward and backward calculation. Deconvolution is a good option for generating saliency map for our task. We add a deconvolution group after the bottleneck layer, making the network an encoder-decoder, note that in this encoder-decoder framework, the framework is different from the task of image generation [8] or image completion [9], the deconvolution group shares the architecture and weights with convolution group, and we hope the decoded saliency map is similar with attention map which means the saliency map observed from the bottleneck layer should follow certain distribution where some areas such as mouth, eyes should be more activated than others.

According to Facial Action Coding System, six common emotions: happiness, sadness, surprise, fear, anger, disgust, contempt are coded by the AU. Since not all AUs are related to a specific expression, we can not directly use the AU value as supervised signal. To avoid the intra-variance AUs that may case wrong supervision, we organize the inputs in the form of triplets [10] to use the relative value of pain intensity. In a triplet, the anchor and positive have the same pain intensity, and the negative has different pain intensity. If an AU is almost the same between anchor and positive and different between anchor and negative,

we think such AU is related to pain, otherwise it belongs to irrelevant AUs. We select possible related AUs by comparing the pain intensity and AU labels in every triplet and then build the attention map for the triplet. At the same time, using triplets as input for saliency supervision, we alleviate the imbalance problem of training data through utilizing most image instances whose pain intensity is zero and avoiding overfitting.

As for the encoder-decoder framework, the deconvolution group is used for saliency map generation while not for feature learning, so we freeze the parameter updating in this group. As we need to get the saliency map in time after the convolution group has been updated, we copy the parameter in convolution group to deconvolution group before we calculate the saliency map.

The saliency map is used to supervise the AU local features, and we also take the whole image feature into consideration. The global feature is extracted from the bottleneck layer and supervised by triplet loss, and the margin is determined by the difference of pain intensity between the anchor and negative example. We combine local and global feature through a multi-task learning framework, using alternate training strategy.

Finally we finetune the network from the bottleneck layer supervised by pain intensity with regression loss, and get better result than the previous work [3].

In summary, the contributions of this work include:

1. Use triplet to build the attention map for pain regression task without sampling and datasets balancing, which can make full use of the ill-posed datasets.
2. Supervise the saliency map with attention map so the bottleneck feature is embedded with more information of the interest area related with the pain regression task.
3. Propose a method to train the network as a multi-task problem combining local feature and global feature.

## 2 Related works

### 2.1 Pain Intensity Regression

The work on pain intensity based on computer vision has significant improvement due to the deep learning technology and the release of the *UNBC-McMaster Shoulder-Pain*. There are two main streams, video based and image based methods.

Personalized method [11] uses the facial point of face image as input and uses Bi-LSTM [12] to estimate the observed pain intensity (OPI) value. They build individual facial expressiveness score (I-FES) for each person and use Hidden Conditional Random Fields (HCRFs) to merge the sequence result to get personalized visual analog scales (VAS) estimation. Recurrent Convolutional Neural Network Regression (RCR) [13] uses recurrent convolution network leveraging sequence information, and is trained end-to-end yet achieving sub-optimal performance. The other direction is image based method [3] that changes the face verification task to be a regression task, using smooth  $L1$  [14] loss and adding

center loss [1] to make the result more discrete. FATAUVA-Net presents a deep learning framework for Valence-Arousal (V-A) estimation that employs AUs as mid-level representation where they map the AU labels to the feature maps as a surprised signal, training different branch for different AUs and using the AU detection results for Valence-Arousal (V-A) estimation.

## 2.2 Regulation Loss

Regulation restricts the size of the parameter space, so the deeper neural network can also have good generalization ability learning from small datasets.  $L1$  regulation makes parameters sparsely while  $L2$  weight decay restricts the norm of parameters. Recently proposed methods [15] [16] focus on small datasets training or semi-supervised learning, which are similar with ours, however, we add the regulation on saliency map. To our best knowledge, no previous work was carried in such direction.

## 3 Our method

### 3.1 Attention Map Generation

Since only some of the AUs are relevant with pain intensity, we organize the inputs images in the form of triplets to use the relative values of pain intensity instead of the absolute values. Building attention map in the form of triplets, we can exclude the irrelevant AUs that wave obviously among same pain intensity or barely change among different pain intensity. Supervised by such attention map, the saliency map areas corresponding to irrelevant AUs will have almost the same brightness, which means the areas corresponding to irrelevant AUs in input image are embedded as little as possible into the bottleneck features, and the areas corresponded with relevant AUs in input images are embedded as much as possible into the final bottleneck features.

In details, we choose the triplet: the anchor and positive have the same pain intensity, and the negative has different. We divide all AUs into set  $\mathcal{A}$  and set  $\mathcal{B}$ . AUs in set  $\mathcal{A}$  are relevant AUs whose corresponded areas in saliency map should discriminate in a triplet, and AUs in set  $\mathcal{B}$  are irrelevant AUs whose corresponded areas in saliency map should almost be the same in a triplet. AU  $\mathbf{k}$  is selected to set  $\mathcal{A}$  if it satisfy the following equation, otherwise it was added to set  $\mathcal{B}$ . Set  $\mathcal{A}$  contains relevant AUs that barely change between anchor and positive and wave obviously between anchor and negative.

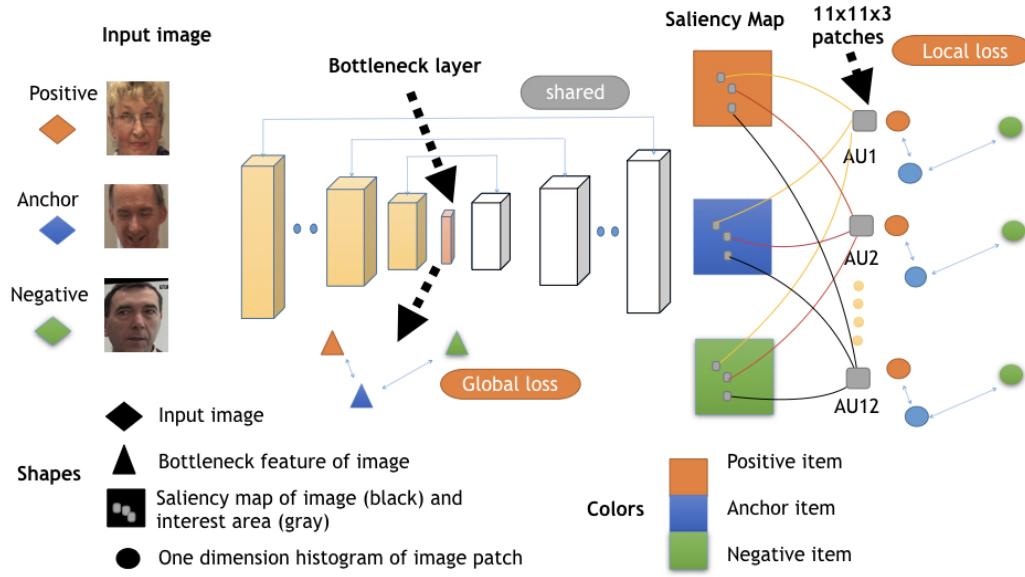
$$|V_a^k - V_p^k| < \alpha \quad \text{and} \quad |V_a^k - V_n^k| \geq \alpha \quad (1)$$

$V^{\mathbf{k}}$  is the value of facial AU  $\mathbf{k}$ . In a triplet, we denote the value of facial AU  $\mathbf{k}$  for anchor, positive, negative as  $V_{\mathbf{a}}^{\mathbf{k}}$ ,  $V_{\mathbf{p}}^{\mathbf{k}}$ ,  $V_{\mathbf{n}}^{\mathbf{k}}$  respectively.  $\alpha$  is the threshold depending on the facial AU value.

Since AUs are corresponding to the semantic pattern of facial attributes, we use areas around each AU to generate the attention map, which is similar to the

approach in EAC (enhancing and cropping) Net [17]. Note that EAC Net uses the absolute value of AU to generate the attention map for AU detection task. Different from their approach, we use the relative value of AU in a triplet for pain intensity regression as for not all AUs are related with pain intensity. We use attention map as direct signals for saliency supervision while EAC Net adds attention map to feature map.

### 3.2 Saliency Map Supervision



**Fig. 1.** Our proposed network structure.

Figure 1 shows our proposed network structure. The saliency map is observed from the bottleneck feature using deconvolution network (white blocks) that has the same architecture and parameters as the convolution network (yellow blocks). For each AU, we find the corresponding patch in anchor, positive and negative item and every patch is  $11 \times 11$  pixels around the corresponding facial landmarks. The local loss is the sum of triplet loss generated by each AU.

We divide AUs into two set  $\mathcal{A}$  and  $\mathcal{B}$  in Section 3.1. The loss function for two sets is different.

The loss for set  $\mathcal{A}$  is:

$$loss_{\mathcal{A}} = \sum_{k \subseteq \mathcal{A}} ([g(P_a^k, P_n^k) - g(P_a^k, P_p^k) + O * W^k]_+) \quad (2)$$

$$W^k = \frac{|V_{\mathbf{a}}^k - V_{\mathbf{n}}^k|}{\sum_{i \subseteq \mathcal{A}} (|V_{\mathbf{a}}^i - V_{\mathbf{n}}^i|)} \quad (3)$$

The loss for set  $\mathcal{B}$  is:

$$loss_{\mathcal{B}} = \sum_{k \subseteq \mathcal{B}} ([g(P_{\mathbf{a}}^k, P_{\mathbf{n}}^k) - g(P_{\mathbf{a}}^k, P_{\mathbf{p}}^k) + O * \frac{1}{N}]_{-}) \quad (4)$$

We denote saliency map patch of AU  $\mathbf{k}$  in anchor, positive, negative as  $\mathbf{P}_{\mathbf{a}}^k$ ,  $\mathbf{P}_{\mathbf{p}}^k$ ,  $\mathbf{P}_{\mathbf{n}}^k$  respectively.  $\mathbf{g}(\mathbf{m}, \mathbf{n})$  is the distance metric which we will discuss in details.  $\mathbf{O}$  is the absolute value between the pain intensity of anchor and negative.  $\mathbf{W}^k$  represents the contribution of AU  $\mathbf{k}$  to pain intensity in set  $\mathcal{A}$ .  $\mathbf{N}$  is the number of AUs in set  $\mathcal{B}$ , and we average the influence of each AU to pain intensity in set  $\mathcal{B}$ .

We denote the distance metric by  $\mathbf{g}(\mathbf{m}, \mathbf{n})$ , in practical, we use normalized Earth Mover’s Distance [18] between the one dimensional histogram of image patch  $\mathbf{m}$  and  $\mathbf{n}$ , as it is robust to face alignment error and performs better when evaluating the saliency map difference according to the image retrieval method [18].

The total local loss is:

$$loss_L = loss_{\mathcal{A}} + loss_{\mathcal{B}} \quad (5)$$

The supervision signal is saliency map, making the bottleneck feature embedded with similar information of the saliency map. To the best of our knowledge, we first supervise the saliency map for the origin task training.

### 3.3 Multi-task Training

The local loss we get in Section 3.2 focus on the facial AU areas which acts as the regulation loss. We find that if we only use the local loss, the network can not converge, so we combine it with the global feature together through multi-task training strategy and the local loss acts as the regulation loss. The normalized bottleneck feature is taken as the global feature presentation for the whole image, and we train it with triplets loss using Euclidean distance. The single triplet loss is as the following equation:

$$loss_G = [f(P_{\mathbf{a}}, P_{\mathbf{n}}) - f(P_{\mathbf{a}}, P_{\mathbf{p}}) + O * \beta]_{+} \quad (6)$$

We denote the bottleneck feature of anchor, positive, negative as  $\mathbf{P}_{\mathbf{a}}$ ,  $\mathbf{P}_{\mathbf{p}}$ ,  $\mathbf{P}_{\mathbf{n}}$  respectively.  $\mathbf{f}(\mathbf{m}, \mathbf{n})$  is the Euclidean distance.  $\mathbf{O}$  is the absolute value between the pain intensity of anchor and negative and we multiply  $\mathbf{O}$  by factor  $\beta$  as the triplet loss margin. During training, we carry hard negative sampling [19] in all triplets in a batch, and we use alternate training strategy to combine the global loss and local loss.

### 3.4 Pain Regression

After training the network using local feature and global feature, we take the pain regression task. We finetune the network from the bottleneck layer, and as for the pain intensity is set to  $[0, 5]$ , we use the activation function  $\sigma(x) = \frac{5}{1+e^{-x}}$ ,  $\mathbf{x}$  is the output of bottleneck layer. we choose the smooth  $L1$  [14] +  $L1$  center loss [1] as our regression loss.

## 4 Experiments

In this section we will explain our implement details and experiments result.

We choose the related face recognition task and use the state of art pertained model as pretrained model. We use MTCNN [20] to detect the face area, and do face alignment according to the facial landmarks provided by *UNBC-McMaster Shoulder-Pain* dataset. The AU labels are available from *UNBC-McMaster Shoulder-Pain* dataset.

We train the attention network alternatively using basic learning rate 0.001 for global loss, and 0.01 for local loss. When we train the local loss, the deconvolution group is frozen, and share the parameters with the convolution group. During training we first select hard negative examples [19] in a batch, and train the network using  $loss_G$ , and then we use the same hard negative examples to get the  $loss_L$  to train the network again.

When training the pain intensity regression, we set pain intensity value from  $[0, 15]$  to  $[0, 5]$  as proposed by the previous work[21] [13]. We evaluate the result using 25-cross-valuation. We try to train the network only using the local loss, however the network is not converging, and when the global loss is added, the network performance well which means the local loss acts like regulation loss here.

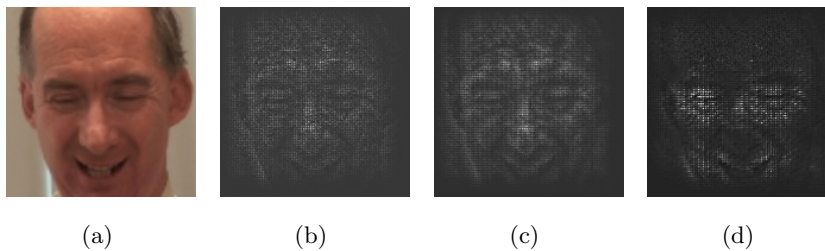
We compare the difference of directly taking regression for the task, using global feature then finetune the network for regression, combining global feature and local feature then finetune the network for regression. The comparison of saliency map can be found in Figure 2. The experiment results can be found in Table 1 and Table 2.

Brighter areas in the saliency map means that the corresponded areas in input image are more embedded into the bottleneck feature. As we can see from Figure 2, using global feature and then finetuned by the pain intensity, the model focuses some area of face while not accurate enough, when using local feature and global feature together, the model focuses on the eye, mouth areas that related to the pain expression.

## 5 Summary

In this paper, we propose a novel method for attention based network. We use deconvolution network to get the saliency map of the bottleneck layer, and design the attention map for pain intensity regression task. Through direct regulation





**Fig. 2.** Saliency maps of different network training approaches. (a) is the input picture, (b) is the saliency map from bottleneck embeddings trained directly by the pain intensity, (c) is the saliency map from bottleneck embeddings trained by the global feature and finetuned by the pain intensity, (d) is the saliency map from bottleneck embedding trained by the combination of global feature and local feature and finetuned by pain intensity.

on saliency map and multi-task training with global loss, we successfully push forward single image based method by a considerable margin without bells and whistles. We believe our job is great for single image based prediction, and the performance of joint prediction over image sequences should benefit from features extracted by our method, which we leave as a future direction.

## References

1. Y. Wen, K. Zhang, Z. Li, and Y. Qiao, *A Discriminative Feature Learning Approach for Deep Face Recognition*, 2016.
2. P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, “Painful data: The unbc-mcmaster shoulder pain expression archive database,” in *Face and Gesture 2011*, 2011, pp. 57–64.
3. F. Wang, X. Xiang, C. Liu, T. D. Tran, A. Reiter, G. D. Hager, H. Quon, J. Cheng, and A. L. Yuille, “Transferring face verification nets to pain and expression regression,” vol. abs/1702.06925, 2017.
4. W. Chang, S. Hsu, and J. Chien, “Fatauva-net: An integrated deep learning framework for facial attribute recognition, action unit detection, and valence-arousal estimation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 00, 2017, pp. 1963–1971.
5. P. Ekman and E. Rosenberg, “What the face reveals : basic and applied studies of spontaneous expression using the facial action coding system (facs),” vol. 68, no. 1, pp. 83–96, 1997.
6. M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” vol. 8689, pp. 818–833, 2013.
7. M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.
8. A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2015.

Methods	MAE↓	MSE↓	PCC↑
our proposed method 1	0.401	0.742	0.643
our proposed method 2	<b>0.334</b>	<b>0.626</b>	<b>0.804</b>
smooth $L1 + L1$ center loss[3]	0.456	0.804	0.651
OSVR- $L1$ [21]	1.025	N/A	0.600
OSVR- $L2$ [21]	0.810	N/A	0.601
RCR[13]	N/A	1.54	0.65

**Table 1.** Performance of our proposed methods and related works on the *UNBC-McMaster Shoulder-Pain* dataset for the estimation of pain intensity. Our proposed method 1: Use global feature to train the network and smooth  $L1 + L1$  center loss for regression. Our proposed method 2: Use the combination of global feature and local feature to train the network and smooth  $L1 + L1$  center loss for regression. MAE is short for mean absolute error deviated from the ground-truth labels over all frames per video. MSE is mean squared error which measures the curve fitting degree. PCC is Pearson correlation coefficient which measures the curve trend similarity ( $\uparrow$  indicates the larger, the better). The best is highlighted in bold.

Methods	wMAE↓	wMSE↓
our proposed method 1	0.883	1.697
our proposed method 2	<b>0.727</b>	<b>1.566</b>
smooth $L1 + L1$ center loss + sampling[3]	0.991	1.720
OSVR- $L1$ [21]	1.309	2.758
OSVR- $L2$ [21]	1.299	2.719

**Table 2.** Performance of our network when evaluated using the weighted MAE and weighted MSE proposed by [3]. Methods proposed by us are applied with uniform class sampling technique.

9. R. A. Yeh, C. Chen, T. Lim, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with perceptual and contextual losses,” vol. abs/1607.07539, 2016.
10. F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *CoRR*, vol. abs/1503.03832, 2015.
11. D. L. Martinez, O. Rudovic, and R. Picard, “Personalized automatic estimation of self-reported pain intensity from facial expressions,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 2318–2327.
12. X. Ma and E. H. Hovy, “End-to-end sequence labeling via bi-directional lstm-cnns-crf,” 2016.
13. J. Zhou, X. Hong, F. Su, and G. Zhao, “Recurrent convolutional neural network regression for continuous pain intensity estimation in video,” pp. 1535–1543, 2016.
14. R. B. Girshick, “Fast R-CNN,” vol. abs/1504.08083, 2015.
15. T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii, “Distributional smoothing with virtual adversarial training,” *Computer Science*, 2015.
16. T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” 2017.
17. W. Li, F. Abtahi, Z. Zhu, and L. Yin, “Eac-net: A region-based deep enhancing and cropping approach for facial action unit detection,” 2017.

18. Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," vol. 40, no. 2, pp. 99–121, 2000.
19. A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," vol. abs/1703.07737, 2017.
20. K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," vol. abs/1604.02878, 2016.
21. R. Zhao, Q. Gan, S. Wang, and Q. Ji, "Facial expression intensity estimation using ordinal information," in *Computer Vision and Pattern Recognition*, 2016, pp. 3466–3474.