UNIVERSITY OF CALIFORNIA SAN DIEGO

Automated Pain Detection in Facial Videos using Transfer Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Xiaojing Xu

Committee in charge:

Professor Virginia R. de Sa, Chair
Professor Truong Q. Nguyen, Co-Chair
Professor Pamela C. Cosman
Professor Gary W. Cottrell
Professor Nuno Vasconcelos

2021

The Dissertation of Xiaojing Xu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Professor Virginia R. de Sa for her support as the chair of my committee. Through multiple drafts and many long nights, her guidance has proved to be invaluable.

Chapter 1, in full, is a reprint of the material as it appears in 15th IEEE International Conference on Automatic Face and Gesture Recognition 2020. Xu, Xiaojing, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

Chapter 2, in full, is a reprint of the material as it appears in Artificial Intelligence in Health 2018. Xu, Xiaojing, Kenneth D. Craig, Damaris Diaz, Matthew S. Goodwin, Murat Akcakaya, Büşra Tuğçe Susam, Jeannie S. Huang, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part is currently being prepared for submission for publication of the material. Xu, Xiaojing, Jeannie S. Huang, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this material.

Chapter 4, in part, has been submitted for publication of the material as it may appear in International Conference of the IEEE Engineering in Medicine and Biology Society, 2021. Xu, Xiaojing, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

VITA

2014	B. S. in Information Engineering, Shanghai Jiao Tong University, Shanghai
2015	M. S. in Electrical and Computer Engineering, Ohio State University, Columbus
2017	Ph. D. Data Scientist Intern, Visa, Austin
2020	Software Engineer Intern, Facebook, Menlo Park
2021	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California San Diego

PUBLICATIONS

Xu, Xiaojing, Srinjoy Das, and Ken Kreutz-Delgado. “ApproxDBN: Approximate computing for discriminative deep belief networks”. *arXiv*, 2017

Xu, Xiaojing, Kenneth D. Craig, Damaris Diaz, Matthew S. Goodwin, Murat Akcakaya, Büşra Tuğçe Susam, Jeannie S. Huang, and Virginia R. de Sa. “Automated Pain Detection in Facial Videos of Children using Human-Assisted Transfer Learning”. *International Workshop on Artificial Intelligence in Health*, pp 162-180. Springer, Cham, 2018.

Xu, Xiaojing, Büşra Tuğçe Susam, Hooman Nezamfar, Damaris Diaz, Kenneth D. Craig, Matthew S. Goodwin, Murat Akcakaya, Jeannie S. Huang, and Virginia R. de Sa. “Towards Automated Pain Detection in Children using Facial and Electrodermal Activity”. *International Workshop on Artificial Intelligence in Health*, pp. 181-189. Springer, Cham, 2018.

Susam, Busra T., Murat Akcakaya, Hooman Nezamfar, Damaris Diaz, Xiaojing Xu, Virginia R. de Sa, Kenneth D. Craig, Jeannie S. Huang, and Matthew S. Goodwin. “Automated pain assessment using electrodermal activity data and machine learning”. *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 372-375. IEEE, 2018.

Xu, Xiaojing, Jeannie S. Huang, and Virginia R. De Sa. “Pain evaluation in video using extended multitask learning from multidimensional measurements”. *Machine Learning for Health Workshop, ML4H@NeurIPS*, pp. 141-154. PMLR, 2020.

Xu, Xiaojing, and Virginia R. de Sa. “Exploring multidimensional measurements for pain evaluation using facial action units”. *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG2020)*. 2020.

ABSTRACT OF THE DISSERTATION

Automated Pain Detection in Facial Videos using Transfer Learning

by

Xiaojing Xu

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2021

Professor Virginia R. de Sa, Chair
Professor Truong Q. Nguyen, Co-Chair

Accurately determining pain levels is difficult, even for trained professionals. Facial activity provides sensitive and specific information about pain, and computer vision algorithms have been developed to automatically detect facial activities such as Facial Action Units (AUs) defined by the Facial Action Coding System (FACS). Previous work on automated pain detection from facial expressions has primarily focused on frame-level objective pain metrics, such as the Prkachin and Solomon Pain Intensity (PSPI). However, the current gold standard pain metric is the visual analog scale (VAS), which is self-reported at the video level. In this thesis, we propose machine learning models to directly evaluate VAS in video.

First, we study the relationship between sequence-level metrics and frame-level metrics. Specifically, we explore an extended multitask learning model to predict VAS from human-labeled AUs with the help of other sequence-level pain measurements during training. This model consists of two parts: a multitask learning neural network model to predict multidimensional pain scores, and an ensemble learning model to linearly combine the multidimensional pain scores to best approximate VAS. Starting from human-labeled AUs, the model outperforms provided human sequence-level estimates.

Secondly, we explore ways to learn sequence-level metrics based on frame-level automatically predicted AUs. We start with an AU prediction software called iMotions. We apply transfer learning by training another machine learning model to map iMotions AU codings to a subspace of manual AU codings to enable more robust pain recognition performance when only automatically coded AUs are available for the test data. We then learn our own AU prediction system which is a VGGFace neural network multitask learning model to predict AUs.

Thirdly, we propose to improve our model using individual models and uncertainty estimation. For a new test video, we jointly consider which individual models generalize well generally, and which individual models are more similar/accurate to this test video, in order to choose the optimal combination of individual models and get the best performance on new test videos. Our structure achieves state-of-the-art performance on two datasets.

Introduction

Accurate measurement of pain severity is difficult even for trained professionals. This is a critical problem as over-medication can result in adverse side-effects, including opioid addiction, and under-medication can lead to unnecessary suffering, tumor growth and can compromise immune function and healing after surgery [LCP11b, QSH15].

The current clinical gold standard and most widely employed method of assessing clinical pain is patient self-report [ZPG$^+$16]. However, this subjective method is vulnerable to self-presentation bias. Consequently, clinicians often distrust pain self-reports, and find them more useful for comparisons over time within individuals, rather than comparisons between individuals [VB09]. Further, infants, young children, and others with communication/neurological disabilities do not have the ability or capacity to self-report pain levels [ZPG$^+$16, SAD$^+$15, AKRP$^+$16]. As a result, to evaluate pain in populations with communication limitations, observational tools based on nonverbal indicators associated with pain have been developed [SFV$^+$17], including physiological, speech, body movements and facial expressions. A systematic review of pain-recognition systems that are based on deep-learning models is provided by [MAEAKAS20].

Of the various modalities of nonverbal expression, it has been suggested that facial activity provides the most sensitive, specific, and accessible information about the presence, nature, and severity of pain across the life span, from infancy [GC87] to advanced age [HHP$^+$14]. Moreover, observers largely consider facial activity during painful events to be a relatively spontaneous reaction which is less subject to voluntary control than verbal expression [SFV$^+$17, Cra92, CPG11].

In this thesis, we focus on self-rated pain level prediction using facial videos.

0.1 Background

0.1.1 Pain

Pain is defined as “an unpleasant sensory and emotional experience associated with, or resembling that associated with, actual or potential tissue damage,” [RCC$^+$20] by the International Association for the Study of Pain (IASP). It is always a personal experience that is influenced to varying degrees by biological, psychological, and social factors. Pain is different from nociception and cannot be inferred solely from activity in sensory neurons. Individuals learn the concept of pain through their life experiences, and inability to communicate does not negate the possibility that a human or a nonhuman animal experiences pain [RCC$^+$20].

Pain is multidimensional. Major dimensions of pain include physiological, sensory, affective, cognitive, behavioral, and sociocultural [McG92] aspects. Self-report measures the subjective nature of pain, which has been shown to be not as stable and accurate as a multidimensional assessment [ABR83, CYT$^+$02, RRM$^+$07, vBVvdS$^+$17]. Such properties of self-reported pain level have made it hard to automatically evaluate across subjects.

0.1.2 Facial Action Units (AU)

Facial Action Units (AUs) is defined by the Facial Action Coding System (FACS) [EF76, MVJP17], which is a comprehensive, anatomically based system for describing all visually discernible facial movement. During AU coding, each present action unit is coded with onset, offset and with an intensity on a 5-point scale. The FACS manual was first published in 1978 by Ekman and Friesen, and was most recently revised in 2002. (Visualizations of facial activation units can be found at https://imotions.com/blog/facial-action-coding-system/). Identifying AUs in video traditionally requires time intensive offline coding by trained human coders, limiting application in real-time clinical settings. Recently, algorithms to automatically detect AUs [VP06, JVP11, MVJP17, LAZ17, TZY$^+$17, SLCM18, SLC$^+$19, ECJ$^+$19] have been developed and implemented in software such as iMotions (imotions.com) allowing automatic output of AU probabilities in real-time based on direct recording of face video.

FACS provides an objective, comprehensive and descriptive approach to decode facial expressions, and studies have found sets of AUs associated with pain. All studies of adults have identified lowering of the brows (AU4) and narrowing of the eyes, as a result of tightening the eyelids (AU7), as basic to the expression. Majority of studies also found raising the cheeks (AU6), eyes closed or blinking (AU43), raising the upper lip (AU10), and parting of lips (AU25) or dropping of the jaw (AU26) to be pain-related actions [CPG11]. People also found nose wrinkling (AU9) to be related to pain [PS08].

0.1.3 PSPI

The Prkachin and Solomon Pain Intensity (PSPI) [PS08] is proposed to evaluate pain levels in an image. It is defined as a combination of a set of AUs:

$$\mathrm{PSPI} = \mathrm{AU4} + \max(\mathrm{AU6{,}AU7}) + \max(\mathrm{AU9{,}AU10}) + \mathrm{AU43}$$

It has been widely used as a pain level indicator, and most research on automatic pain detection from facial expression has focused on predicting the frame-/image-level PSPI scores [ALC+09, LCP+11a, MR06, RPP13, WXL+17, RCG+17, TH18, ZGKS18]. However, it has some limitations. First, PSPI only consider 6 AUs, but there is evidence showing that pain is reflected by other AUs too. The PSPI may also be non-zero when the subject doesn't feel any pain, for instance, AU43 (eyes closed) occurs during sleep and is obviously not specific to pain expression. Secondly, it is defined on frame level, but the ground truth of pain level should be measured on sequence level. People usually talk about pain levels during a period of time, not at a point of time. PSPI may go up and down through a video while the feeling of pain stays unchanged. Thirdly, PSPI only measures the facial expression of pain, and shouldn't be confounded with the feeling of pain. PSPI is influenced by many factors, e.g. facial expressiveness.

With all these shortcomings with PSPI, although it is a good measure of pain, it doesn't reflect pain directly and successful estimation of PSPI doesn't mean successful estimation of pain.

0.1.4 VAS

Two types of pain metrics are usually considered in pain studies: frame-by-frame metrics and sequence-level metrics. AU and PSPI mentioned above are frame-by-frame, and VAS (Visual Analog Scale) is sequence level, in the sense that there is one VAS score labeled for each video (sequence of frames).

As stated above, self-report VAS is the gold standard of pain assessment and can be easily acquired as ground truth. Participants rate their pain level during a period of time using a 0-10 Numerical Rating Scale, where 0 = no-pain and 10 = worst pain ever.

VAS also has drawbacks as a measure of pain. As illustrated above, is it subjective, because the pain feeling is subjective, and VAS can also be affected by reporting bias and variances in memory and culture, and interpretation of the scales or descriptive words [CPG11, WAHLE+16].

Nevertheless, VAS is considered to be a metric that is the closest to the ground truth of pain. Test–retest reliability has been shown to be good, and validity has been proved with a high correlation with other pain scales [DLR+78, FQA+90]. VAS has also demonstrated sensitivity to changes in pain [JZHM75].

0.1.5 Challenges in automated pain recognition

1. Datasets are small, and hard to obtain.

   The most famous UNBC McMaster Shoulder Pain database [LCP+11a] only contains 25 subjects and 200 videos. The post-surgery child pain dataset [XCD+18] contains 134 videos of 70 subjects. These are the only two facial video datasets that have both clinically obtained VAS labels and AU labels. Not only the number of samples in each dataset is small, the number of videos from each subject is also small, because it is rare that a patient will experience all 11 pain levels during genuine pain data collection. Another famous pain dataset is the Biovid Heat Pain database [WGE$^+$13]. It includes 90 participants, but the videos are not AU-coded, and pain levels are determined by stimulation temperatures, not self-ratings. Other datasets such as BP4D [ZYC$^+$14], SenseEmotion [VGL$^+$16], EmoPain [AKRP$^+$15], X-ITE [HBN$^+$18] all have similar problems. In such a scenario, transfer learning, in particular, using CNN models trained on larger datasets, would benefit the learning of pain.

2. Noisy labels.

   Different people have different understanding of pain. Two people may experience the same level of pain, but give different pain ratings. This makes it hard for machine learning models to generalize to new subjects.

3. Noisy inputs.

   Not only the pain feeling itself is subjective, the facial expressions of pain also differs from person to person. There are studies suggesting that there are "different faces of pain" [KL14]. Two people may have the same level of pain, but show completely different facial expressions.

0.2 Dissertation overview

0.2.1 Chapter 1: Exploring Multidimensional Measurements for Pain Evaluation using Facial Action Units

Pain is multidimensional, and studies have suggested to evaluate pain in patients using multidimensional measures. In this chapter, we analyze the relationship between multidimensional pain measurements: VAS (Visual Analog Scale), OPR (Observer Pain Rating), AFF (Affective-motivational scale), SEN (Sensory scale), and their predictions from a machine learning model.

We also study the relationship between sequence-level and frame-level pain metrics, and explore ways of utilizing human-coded AUs and multidimensional pain ratings to improve VAS prediction, and study the contribution of each component of the multitask-ensemble multidimensional-pain model. People have developed methods to evaluate the frame-level PSPI and AUs in facial images. Our model serves as a baseline of how well one can predict VAS using human-coded AUs. Our model can be combined with automated AU/PSPI detection systems to achieve end-to-end VAS prediction and provides an upper-bound on expected performance.

0.2.2 Chapter 2: Automated Pain Detection in Facial Videos of Children using Human-Assisted Transfer Learning

In this work we combine our VAS prediction model from true AUs with an AU prediction system, the iMotions software, to achieve end-to-end VAS prediction from videos. The iMotions takes live/recorded facial videos as input, locates the face and provides 20 AU estimations. We find that AUs coded automatically are different from those coded by a human trained in the FACS system, and that the human coder is less sensitive to environmental changes. We improve the robustness of automatic AU codings by applying a transfer learning model to transfer automatically coded AUs to manually coded AUs.

0.2.3 Chapter 3: Pain Evaluation in Video using Extended Multitask Learning from Multidimensional Measurements

In Chapter 3, we use the iMotions software to obtain AU values. In this chapter, we develop our own AU prediction model using the VGG16 structure. We replace the last layer of a pre-trained VGGFace neural network and fine-tune it using our pain data. We build a three-stage multitask learning model that exploits multiple dimensions of pain to evaluate the current gold standard pain metric VAS in video from video frames directly.

0.2.4 Chapter 4: Personalized Pain Detection in Facial Video with Uncertainty Estimation

Building on our 3-stage model, we propose to improve pain detection in facial videos using individual models and uncertainty estimation. For a new test video, the system jointly considers which individual models generalize well generally, and which individual models are more similar/accurate to this test video, in order to choose the optimal combination of individual models and get the best performance on new test videos.

0.3 Contributions

1. We analyzed the relationship between multidimensional pain measurements and their predictions from a machine learning model

2. We studied the relationship between sequence-level and frame-level pain metrics, and built an extended multitask learning model to estimate sequence-level pain scores using human-coded frame-level features

3. We explored ways of utilizing human-coded AUs and multidimensional pain ratings to improve VAS prediction, and studied the contribution of each component of the multitask-ensemble multidimensional-pain model

4. We provided a baseline of how well one can predict VAS using human-coded AUs

5. We showed that transferring automated features to the manual feature space improves automatic recognition of clinical pain across different environmental domains.

6. We proposed a three-stage multitask learning model that exploits multiple dimensions of pain to evaluate the current gold standard pain metric VAS in video from video frames directly

7. We showed that multitask learning of pain-related ratings improves the learning of target pain ratings

8. We learned personalized individual models to evaluate the current gold standard pain metric VAS in video from video frames directly.

9. We learned PSPI and VAS as a combination of the output of individual models to improve the generalizability of the pain prediction model.

10. We learned the uncertainty of VAS prediction of each individual model, and improved the VAS prediction on new test subjects by adjusting ensemble weights based on the uncertainty of individual predictions

11. Our model beat the current state-of-the-art performance.

Chapter 1

Exploring Multidimensional Measurements for Pain Evaluation using Facial Action Units

1.1 Introduction

The current gold standard of estimating clinical pain is patient self-report given by visual analog scale (VAS), despite its known limitations [Cra92, ZPG+16]. One of these limitations is that it is difficult to obtain in populations with verbal or neurological disabilities [ZPG+16]. Automated pain recognition models have been developed to solve this problem using various nonverbal signals such as facial expressions, head/body movement and physiological signals [WGE+13, OBBMW15, JPP+15, CPS19]. Research has shown that facial expressions can provide sensitive and reliable information about pain across the life span [PBB+01, Wil02], from infants [GC87] to elderly patients [MBML03, HHP+14].

Two types of pain metrics are usually considered in pain studies: frame-by-frame metrics and sequence-level metrics. One prominent example of frame-level metrics, are the muscle-based facial action units (AUs) defined by the Facial Action Coding System (FACS) [EF76]; they have been widely used as a consistent and reliable way to represent facial expressions including pain [CAW18] expression. The names of some of the pain-related AUs can be found in Table 1.1. Another frame-level metric, built on top of the AUs, is the Prkachin and Solomon Pain Intensity

(PSPI) [PS08]. It defines a single number that measures pain as a combination of AU intensities:

$$\text{PSPI} = \text{AU4} + \max(\text{AU6,AU7}) + \max(\text{AU9,AU10}) + \text{AU43}$$

Most research on automatic pain detection from facial expression has focused on predicting frame-level PSPI scores. A widely used 2-step framework is to first extract low-dimensional relevant non-rigid geometric or appearance features from raw pixels and then learn a classification or regression model [ALC$^+$09, LCP$^+$11a, MR06, RPP13]. Otherwise, deep learning can be used to learn from raw pixels directly [WXL$^+$17, ZGKS18]. In addition to these "static approaches" that extract features from single frames, it is also useful to learn dynamic features when data is available in the form of video sequences [RCG$^+$17, TH18]. Multiple-instance learning has been used to learn frame-level scores using sequence-level labels in a weakly supervised manner [SDB13, RRBP16].

Automated detection of facial AUs has also been well studied, and PSPI ratings can be calculated directly from AU estimates. Many approaches of AU detection focused on finding regions of interest [ZCZ16, G$^+$17, LAZ17, LAZY18]. Jaiswal et al. and Chu et al. combined CNN and LSTM, and Kumawat proposed a 3D convolutional layer called Local Binary Volume layer, to learn temporal information [JV16, CDlTC17, KVR19]. Baltrušaitis et al. studied the benefit of person-specific neutral expression normalisation and multiple datasets for generic model training, and presented a pipeline that detects AUs in real-time [BMR15]. Tang et al. and Romero et al. fine-tuned VGG models pretrained on face datasets to detect AUs under different facial views [TZY$^+$17, RLA18]

In contrast, to the automated work above, sequence-level pain metrics are more often used in clinics, and the understanding and interpretation of pain in the literature is mostly based on sequence level assessments, rated by observers or by self-report. The sequence-level self-rated VAS is still the most commonly used pain score in clinical settings. Only a few papers have addressed the problem of estimating VAS score in facial videos. Sikka et al. [SAD$^+$15] and Xu et al. [XCD$^+$19] detected pain in children after surgery using AUs extracted by iMotions (imotions.com). Liu et al., Martinez et al., and Xu et al. used a two-stage method to first train a model to predict pain scores at the frame level, and then predicted video VAS score using these frame-level predictions [LPS+17, MRP+17, XHdS19] although only [XHdS19] started from raw pixels.

Although sequence-level metrics are considered to have more clinical relevance, frame-level pain recognition has been studied more thoroughly and there exist software packages and toolkits such as iMotions (imotions.com) and OpenFace [ALS16] to automatically detect AUs. There are many reasons for this. First, it is difficult to obtain a large number of sequence-level samples. A pain dataset with each video lasting less than 1 minute can have three orders of magnitude more frames than videos. Second, machine learning models on videos require significantly more space and time to train. This problem is not unique to pain; there are many deep neural networks trained on facial images, but there is no publically available model trained on facial videos, so it is hard to leverage prior work when working with videos. Currently most sequence-level models use frame-level models as building blocks [SAD+15, LPS+17, MRP+17, XCD+19, XHdS19], and the problem of learning sequence-level metrics is usually broken down into two parts: learning frame-level metrics and learning sequence-level metrics based on the frame-level metrics. Since there has been a lot of research addressing the first part (learning frame-level metrics), in this work, we focus on whether and how well we can solve the second part. In order to not be dependent on the quality of model solving the first part, we study the second problem for human coded frame-level AUs and PSPIs (which are usually used as ground truth in AU and PSPI estimation models). We do this through a two-stage model similar to the last two stages in the extended multitask learning model which is the current state-of-the-art for estimating VAS [XHdS19]. In the first stage, we send statistics of AUs and PSPI over frames of each video as inputs to a neural network to get a sequence-level VAS prediction, and use multitask learning to improve the VAS prediction while obtaining multidimensional pain scales. Then, as in [XHdS19], we extend the multitask learning framework by finding an optimal linear combination of these pain scales to further improve VAS prediction. We show on the UNBC-McMaster Shoulder Pain dataset [LCP+11a] that this method outperforms human video-level labels, and can be further improved when combined with those human ratings.

The contributions of this paper are as follows:

- We analyze the relationship between multidimensional pain measurements and their predictions from a machine learning model
- We study the relationship between sequence-level and frame-level pain metrics, and build an extended multitask learning model to estimate sequence-level pain scores using human-coded frame-level features
- We explore ways of utilizing human-coded AUs and multidimensional pain ratings to improve VAS prediction, and study the contribution of each component of the multitask-ensemble multidimensional-pain model
- Our model serves as a baseline of how well one can predict VAS using human-coded AUs
- Our model can be combined with automated AU/PSPI detection systems to achieve end-to-end VAS prediction and provides an upper-bound on expected performance.

1.2 Method

This paper studies the widely used UNBC-McMaster Shoulder Pain dataset [LCP+11a]. It contains videos of patient faces (who were suffering from shoulder pain) while they were performing a series of active and passive range-of-motion tests to their affected and unaffected limbs on two separate occasions. The dataset includes 25 subjects, 200 videos and 48,398 frames.

Table 1.1. AU Description

AU4	brow lowering	AU12	oblique lip raising
AU6	cheek raising	AU20	horizontal lip stretch
AU7	eyelid tightening	AU25	lips parting
AU9	Nose wrinkling	AU26	jaw dropping
AU10	upper lip raising	AU43	eye closure

The dataset provides 11 facial action unit (AU) intensities coded each frame by certified FACS coders, and 1 PSPI score calculated from the AUs. AUs are defined by FACS (Facial Action Coding System) [EF76] to code movements of individual facial muscles. In this work, we work with the 9 AUs (AU4, 6, 7, 10, 12, 20, 25, 26 and 43) present in more than 500 frames.

In addition to the frame-level features, the dataset also provides 4 sequence-level labels: VAS (Visual Analog Scale) 0-10, OPR (Observers Pain Rating) 0-5, AFF (Affective-motivational scale) 0-15 and SEN (Sensory Scale) 0-15. OPR is the human observers' rating of pain level of the video. The other three measures are provided by the patients themselves. The sensory scale consists of a numeric scaling associated with the following words of increasing SEN scale: extremely weak, faint, very weak, weak, very mild, mild, slightly moderate, moderate, barely strong, clear cut, slightly intense, strong, intense, very intense, extremely intense. The affective-motivational (AFF) scale uses the following affect-based words: slightly unpleasant, slightly annoying, annoying, unpleasant, slightly distressing, slightly miserable, very annoying, distressing, very unpleasant, miserable, very distressing, slightly intolerable, very miserable, intolerable, very intolerable [GMD78, HGDM80].

With the features and labels described above, our goal is to train a model that predicts VAS using AU and PSPI intensities. Our model structure and hyper-parameters follow that of stage 2 and 3 of the model proposed in [XHdS19].

1.2.1 VAS Estimation in Facial Videos using AU Sequences

For each video, we form a 10-D feature vector by taking the maximum rating over all frames for each of the 9 AUs and 1 PSPI to form a 10 dimensional feature vector of the video that is input to a fully connected neural network with one 20 unit hidden layer to predict VAS in a linear output layer using batch-weighted MSE loss [SH19]. We used the Adam optimizer, initial learning rate of 1e-2, batch size of 32, max number of epochs of 200, and used early stopping when the validation loss hadn't decreased for 20 epochs.

1.2.2 Multitask Learning

As mentioned in [XHdS19], the three other sequence level pain ratings are very related to the VAS pain score which motivates a multitask learning (MTL) approach [Car97] that leverages "the domain-specific information contained in the training signals of related tasks" [Car97]. OPR may be especially useful as it should be fully constrained by information in the video (unlike VAS that may reflect strong pain but masked facial expression). The multitask architecture is straight forward. We use 4 scores instead of a single VAS as outputs of the neural network. The labels are normalized into the same range so that all elements contribute equally to the loss during training. The losses are weighted based on the distribution of VAS scores, and the validation loss is the mean MSE of the 4 outputs.

1.2.3 Ensemble Learning of Multidimensional Pain Scores

Each of the four sequence-level scores (VAS, OPR, AFF, and SEN) reports on different aspects of pain. VAS reflects the patient's overall rating of their perceived pain. AFF and SEN are designed to try to separate out affective vs sensory aspects of pain and are also reported by the patient. OPR, on the other hand, is scored by an external observer and is only based on the facial video so may be a more predictable function of the video for training a machine learning system. If humans are considered the gold standard at facial pain recognition, then OPR could be considered an approximate upper bound for a machine-learning facial video system.

OPR, AFF, and SEN are all highly correlated with VAS (see Figure 1.1 LEFT) and can be considered as predictions of VAS. After scaling their outputs to the same range as VAS, they all do a reasonable job at estimating VAS and can be considered as four different "experts" (Fig. 1.1 RIGHT). Ensemble averaging can be used to compute the optimal linear combination of experts to reduce variance of the estimator [Has97].

As in [XHdS19], the final prediction of VAS is learned as a weighted sum of the four experts. If each expert outputs f_i, then the overall model $\tilde{f}$ is defined as:

$$\tilde{f} = \sum_{i=1}^{4} \alpha_i f_i$$

We solve the optimization problem minimizing MSE of the final prediction $\tilde{f}$ subject to $\sum_{i=1}^{4} \alpha_i = 1$ [Cle86, TL86, Has97, XHdS19]. The optimal $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$ is:

$$\alpha = \frac{\Omega^{-1} 1}{1^T \Omega^{-1} 1}$$

where $\Omega = [\omega_{ij}] = [E[(f_i - VAS)(f_j - VAS)]]$ and VAS gives the true VAS labels. The ensemble weights an expert more if it is more accurate in estimating VAS.

1.3 Experimental Analysis

On the UNBC-McMaster dataset, we performed 5-fold cross validation with each fold consisting of 5 subjects. To prevent overfitting, we used the same training/test splits for the two stages in each iteration. One of the 4 training folds is randomly selected as the validation set for neural network training. After 5 iterations we evaluate the models using Mean Absolute Error (MAE), Mean Squared Error (MSE), Intraclass Correlation Coefficient (ICC) and Pearson Correlation Coefficient (PCC) on all test data. ICC is useful when MAE scores are deceptively low. For example, for the current dataset, if the model outputs the average VAS for all samples, the MAE will be 2.44, but the ICC will be approximately zero. So we want low MAE with high ICC.

For all models in this paper, we performed the above 5-fold cross validation 5 times, and report mean and standard deviation over 5 experiments. All experiments were run on a single NVIDIA Titan V GPU.

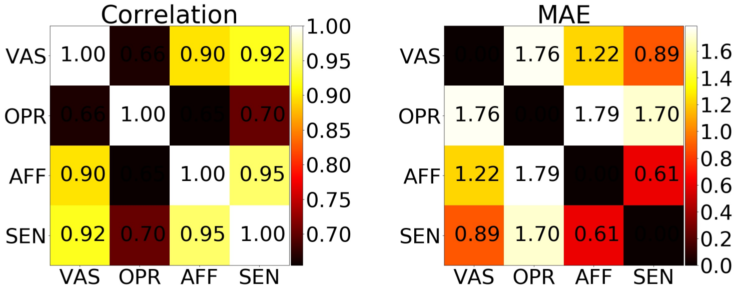


Figure 1.1. Correlation (left) and MAE (right) between each pair of the 4 sequence-level true scores. The scores have been scaled to the same range 0-10.

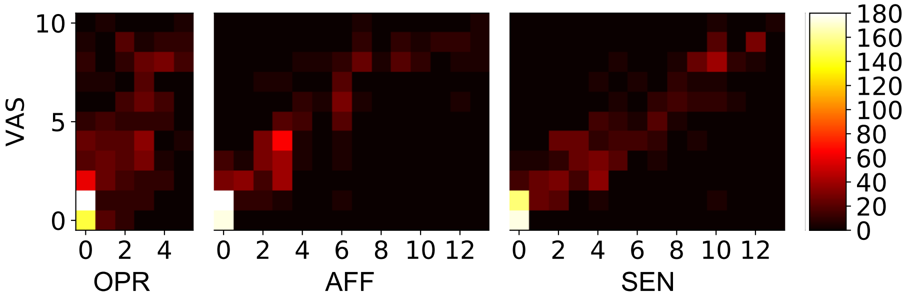


Figure 1.2. 2D histogram of sequence-level score pairs.

1.3.1 Relationship between Sequence-level Metrics in the Data

The relationship between the 4 sequence-level scores in the UNBC-McMaster dataset is shown in Fig. 1.1. We can see from the heatmap on the left that VAS, AFF and SEN are highly correlated, and OPR is also correlated with these 3 self-rated scores but not as much. The right side of the figure shows how well (in terms of MAE) each of the multimodal pain measures predicts the others (after appropriate rescaling). For example OPR (human ratings) predicts VAS with an MAE of 1.76.

Figure 1.2 shows the joint distributions of VAS with OPR, AFF and SEN plotted as 2D histograms. It can be seen that although VAS is linearly correlated with the three other scores, they are not strictly proportional.

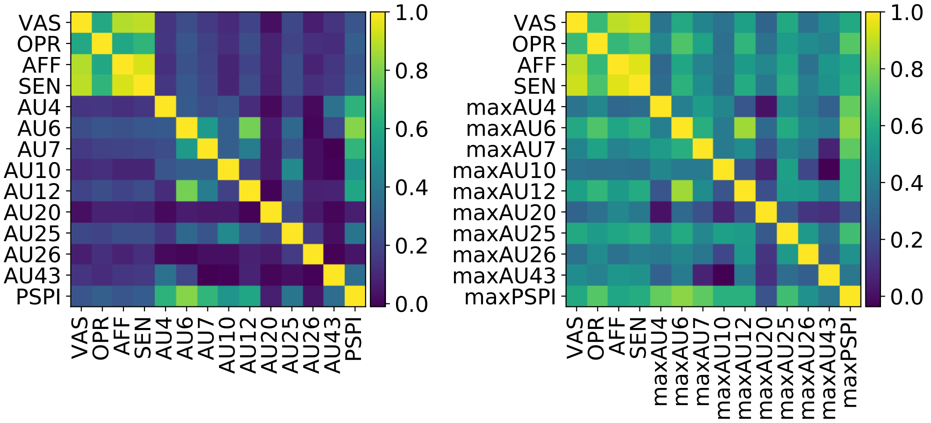


Figure 1.3. The correlation between 4 sequence-level scores (VAS, OPR, AFF, SEN) and 10 frame-level scores (9 AUs and PSPI) in the data. On the left is the correlation at the frame level, where the VAS for a frame is the VAS of the video it belongs to. On the right is the correlation at the sequence level, where the maximum AU/PSPI for a video is taken.

1.3.2 Relating Sequence- and Frame-level Metrics in the Data

Fig. 1.3, shows the correlation between the frame-level and sequence-level pain scores. We see again the high correlations between the sequence-level measures and some correlation between the frame-level measures. Of the sequence measures, OPR generally has a higher correlation with the AUs and PSPI. This shows the potential of predicting sequence-level pain ratings from frame-level measurements.

1.3.3 Multidimensional Pain Prediction using Neural Networks

While the previous subsections discussed properties of the UNBC-McMaster Pain dataset, in this subsection we discuss relations between predictions from our neural networks.

Fig. 1.4, presents, as heatmaps, the MAEs of the multitask neural network with 4 outputs (prior to ensembling) corresponding to the 4 sequence-level pain ratings. For example, diagonal elements show the MAEs of each output predicting the corresponding metric, and the second element in the first row shows the MAE of using the OPR output to predict VAS. Interestingly, the best MAE in predicting a metric is not always given by its corresponding output, e.g. the OPR output predicts VAS better than the VAS output, and the OPR output works better in SEN

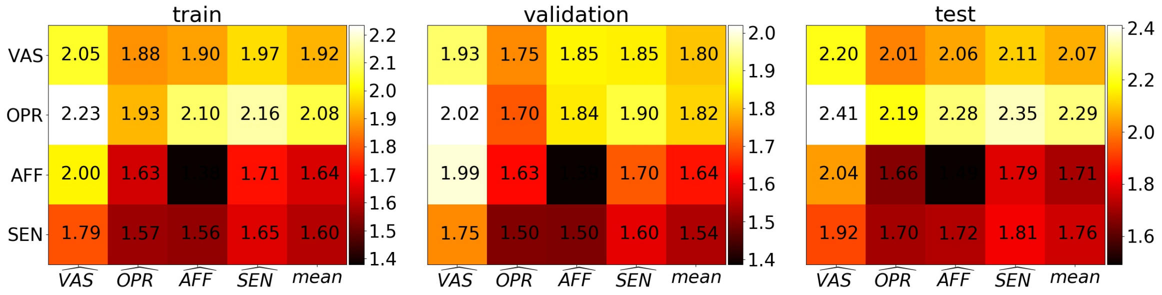


Figure 1.4. Average MAE on training and test data. The y axis gives the true label, and x axis the predictions. Each entry is the mean absolute difference between the two variables. All the labels and predictions have been mapped to the range 0-10 before calculation, but MAEs in different rows are not strictly comparable because OPR only takes 6 values while AFF and SEN can take 16.

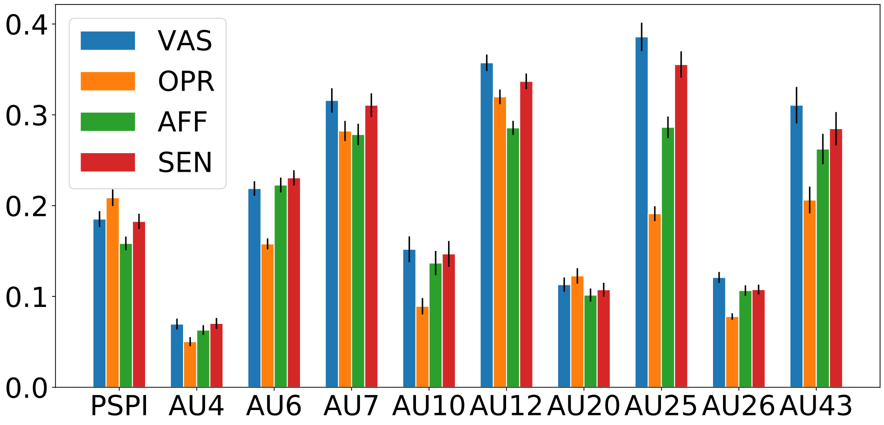


Figure 1.5. Contributions of each of the AUs to the neural network outputs that use max of AUs and PSPI as input. The heights of the bars represent feature importance measured as the mean absolute shap values. Error bars show the standard deviation of the mean absolute shap values.

prediction than the SEN output. Actually, the OPR output works well when used to estimate all the metrics despite being trained to only estimate OPR, the metric with the lowest correlation with the other pain scores. This may be because OPR is more consistent across subjects and is based purely on video features. As a result, OPR may be learned more easily from facial features such as AUs and PSPI, and serve as a better pain metric when tested across subjects.

AU Importance. We use the shap framework [LL17] to calculate the contribution of each of the AUs to the four output scores, and plot the importance values in Fig. 1.5. The bar graph shows, for example, that AU7, 12, 25 and 43 are very useful in pain prediction, except that

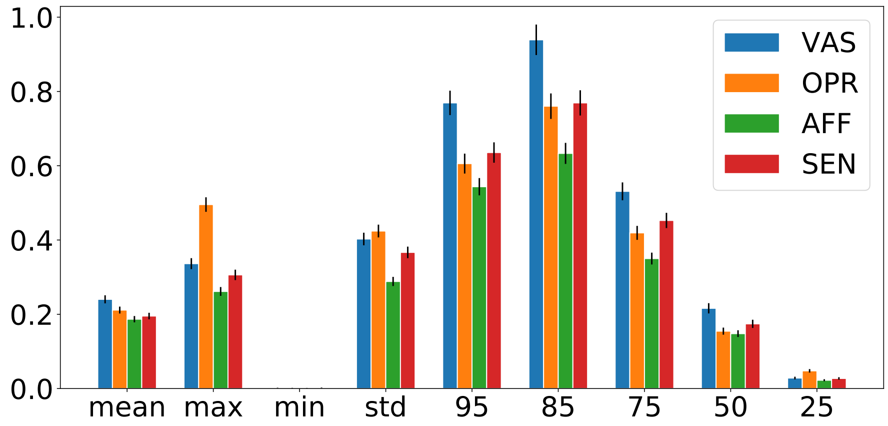


Figure 1.6. Contributions of each of the 9 PSPI statistics to the neural network outputs that use 9 PSPI stats as input. The heights of the bars represent feature importance measured as the mean absolute shap values. Error bars show the standard deviation of the mean absolute shap values.

AU25 is much less important when predicting OPR than predicting the self-ratings. OPR uses PSPI more while not using as much the individual AUs compared to the self-report measures of VAS, AFF and SEN. Interestingly, while AU4 is considered to be among the “core expressions of pain” and contributes to PSPI score [Prk92, PS08, Prk09], it is not a very important feature in this model on this dataset.

There is a fair amount of consistency between Fig. 1.5 and Fig. 1.3. For example, PSPI has higher correlation with OPR than the 3 self-rated scores, and also higher importance for predicting OPR. AU25 and 43 are less important for OPR and also less correlated with OPR than the other 3 pain scores.

Benefit of Multitask Learning. We explore the benefit of multitask learning in the neural network in Table 4.2 row 1-2. The first row shows the VAS prediction performance without multitask learning, i.e. when the neural network only has one output predicting VAS. The second row corresponds to the multitask learning model, where the performance is evaluated only with the output trained to predict VAS. Learning the three other scores from a shared hidden layer, together with VAS helps the model’s VAS output to better predict VAS.

Different Input Features. When extracting sequence-level features for a video from a sequence of frame-level features, we simply take the maximum of the AU/PSPI sequence as in [ALC+09], but it is also common to use other statistics such as standard deviation, minimum, mean, etc. [SAD+15, LPS+17, XCD+19, XHdS19]. To explore how different choices of input features work, we extracted 9 statistics (mean, max, min, standard deviation, 95th, 85th, 75th, 50th, 25th percentiles) from the PSPI and AU sequences to form a length-90 (9 stats × (9 AU + 1 PSPI)) feature vector. The performance using 90 features is not as good as using 10 maxima (row 4-6 compared to row 1-3 in Table 4.2). The reason may be that 90 dimensional inputs is too large for our model. To address this, we also tried using 9 statistics of PSPI only following [LPS+17, XHdS19] since PSPI is defined to represent pain and contains the most comprehensive information about pain expressions. The results are shown in the last three rows in Table 4.2. Using 9 statistics of PSPI works fine, but still not as good as using 10 maxima of PSPI and AUs. The shap importance values for this model are plotted in Fig. 1.6. Min and 25 percentile are two inputs that are not very useful for this model.

Table 1.2. Sequence-level VAS Prediction using Frame-level Labels

NN Input	NN Output	Ensemble Learning	MAE	MSE	ICC	PCC
PSPI+AU max	VAS	-	1.94 ± 0.05	5.25 ± 0.18	0.57 ± 0.02	0.64 ± 0.02
PSPI+AU max	4 scores MTL	-	1.90 ± 0.04	4.98 ± 0.10	0.59 ± 0.01	0.67 ± 0.01
PSPI+AU max	4 scores MTL	Ensemble	1.73 ± 0.03	4.61 ± 0.19	0.61 ± 0.02	0.67 ± 0.02
PSPI+AU stats	VAS	-	2.02 ± 0.05	5.83 ± 0.14	0.51 ± 0.04	0.58 ± 0.02
PSPI+AU stats	4 scores MTL	-	1.94 ± 0.05	5.39 ± 0.22	0.56 ± 0.02	0.61 ± 0.02
PSPI+AU stats	4 scores MTL	Ensemble	1.81 ± 0.04	5.04 ± 0.17	0.58 ± 0.01	0.63 ± 0.01
PSPI stats	VAS	-	2.07 ± 0.05	5.81 ± 0.23	0.52 ± 0.04	0.63 ± 0.03
PSPI stats	4 scores MTL	-	2.03 ± 0.05	5.58 ± 0.23	0.53 ± 0.03	0.65 ± 0.02
PSPI stats	4 scores MTL	Ensemble	1.76 ± 0.03	4.81 ± 0.18	0.59 ± 0.02	0.65 ± 0.02

Table 1.3. Comparison with Humans and Other Work

	MAE	MSE	ICC	PCC
EMTL with true AU (this paper)	1.73 ± 0.03	4.61 ± 0.19	0.61 ± 0.02	0.67 ± 0.02
EMTL from pixels [XHdS19]	1.95 ± 0.06	5.90 ± 0.23	0.43 ± 0.03	0.55 ± 0.03
Human (OPR)	1.76	6.26	0.66	0.66
Average of EMTL (with true AU) and Human	1.48 ± 0.02	4.22 ± 0.10	0.70 ± 0.01	0.71 ± 0.01

1.3.4 Optimal Linear Combination of Multidimensional Pain

While multitask learning results in improved training of VAS prediction through joint learning of all 4 measures, ensembling the 4 predicted outputs discussed in Section 1.2.3 results in significantly ($p < 0.0001$) better performance as shown in row 3 in Table 1.2.

1.3.5 Contributions of Different Components: Multitask Learning, Ensemble Learning and Multidimensional Pain

In this section, we perform ablation studies to explore the relative contributions of different components of the extended multitask learning model.

In order to see whether multitask learning helps, we trained models with separate hidden layer for each of the four sequence-level outputs i.e. with the same inputs and outputs but without multitask learning/hidden layer sharing. The performance ("4 scores") is not as good as using multitask learning ("4 scores MTL") (see Table 1.2 and Fig. 1.7).

In order to compare the importance of ensemble learning to that of multi-task learning, we trained a model with the same structure as our best model, i.e. with 4 neural network outputs and ensemble learning on top of them, but instead of using 4 different pain scores as labels for NN outputs, we trained each of the 4 outputs with identical VAS labels (but different initial conditions). This allows the model to start from 4 different initial states and explore different areas of the weight space with different final predictions. The ensemble model will then find the best way to linearly combine these predictions to obtain a new random variable as the prediction of VAS. The results show that this simple ensemble model also performs better than a single network predicting only VAS but slightly worse than the best model predicting 4 different pain scores, as plotted in Fig. 1.7 "VAS $\times$4 MTL". From these results we conclude that ensembling is most helpful for the excellent performance, but that using multidimensional pain scores is also helpful.

We also trained a version of the network with 4 VAS outputs where each output had its own (unshared) hidden layer. ("VAS $\times$4" in Fig. 1.7). This model performed slightly worse.

This is likely because the multitask learning model has less parameters and so learns faster with less overfitting.

Lastly, since ensemble learning contributes significantly to the performance, we considered a model with extra copies of outputs to provide more “expert” predictions to ensemble. We considered 4 copies of the 4 different sequence-level scores, and separately, 16 copies of VAS, to make 16 output NNs. This didn’t further improve the performance.

To summarize, with the same inputs, the model with ensemble learning on multidimensional pain predictions yields the best performance. This corresponds to the third row in Table 1.2 for each input type, as well as the first (blue) bar in Fig. 1.7 in each group.

1.3.6 Comparison with Humans and Other Work

We compare our model with humans in Table 1.3. The human ratings are given by the OPR scores in the dataset. Our extended multitask learning model using AU features and multidimensional pain outputs beats the MAE of those humans. Moreover, when averaging our prediction with the human predictions, the performance can be further improved. This implies that learning pain as a function of individual AUs may be a more accurate and systematic way than learning pain from the whole face.

We also compare our model using true AUs with [XHdS19] that has a model with similar structure but uses AUs predicted automatically from the output of a deep convolutional network. Our results significantly outperform [XHdS19] demonstrating the potential of an end-to-end VAS prediction model if the AU prediction stage is improved.

1.4 Discussion and Conclusion

We explored a model that predicts VAS using facial actions units, and beats human observers on the UNBC-McMaster Shoulder Pain dataset. When a human observer is available, the performance can be largely improved simply by averaging our prediction and the human prediction. While the human observer in the UMBC-McMaster dataset is not necessarily the same human that labeled the AUs, it would be interesting to explore whether this method of using human-labeled AUs can beat the same observer at VAS prediction.

We studied ablations of the Extended Multitask Learning Model. The approaches using multitask learning, multidimensional pain measurement and ensemble learning can be used in similar healthcare datasets and tasks. Our model can be combined with existing frame-level pain estimation models such as AU or PSPI extractors to easily form a video-level metric prediction model. In this case, the performance shown in this paper provides an upper bound on the accuracy that can be achieved when using automatically estimated AUs instead of manually labeled AUs. It also provides a baseline for estimating sequence-level pain ratings such as VAS using widely-used frame-level pain related measurements such as AUs and PSPI.

Acknowledgments

Chapter 1, in full, is a reprint of the material as it appears in 15th IEEE International Conference on Automatic Face and Gesture Recognition 2020. Xu, Xiaojing, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

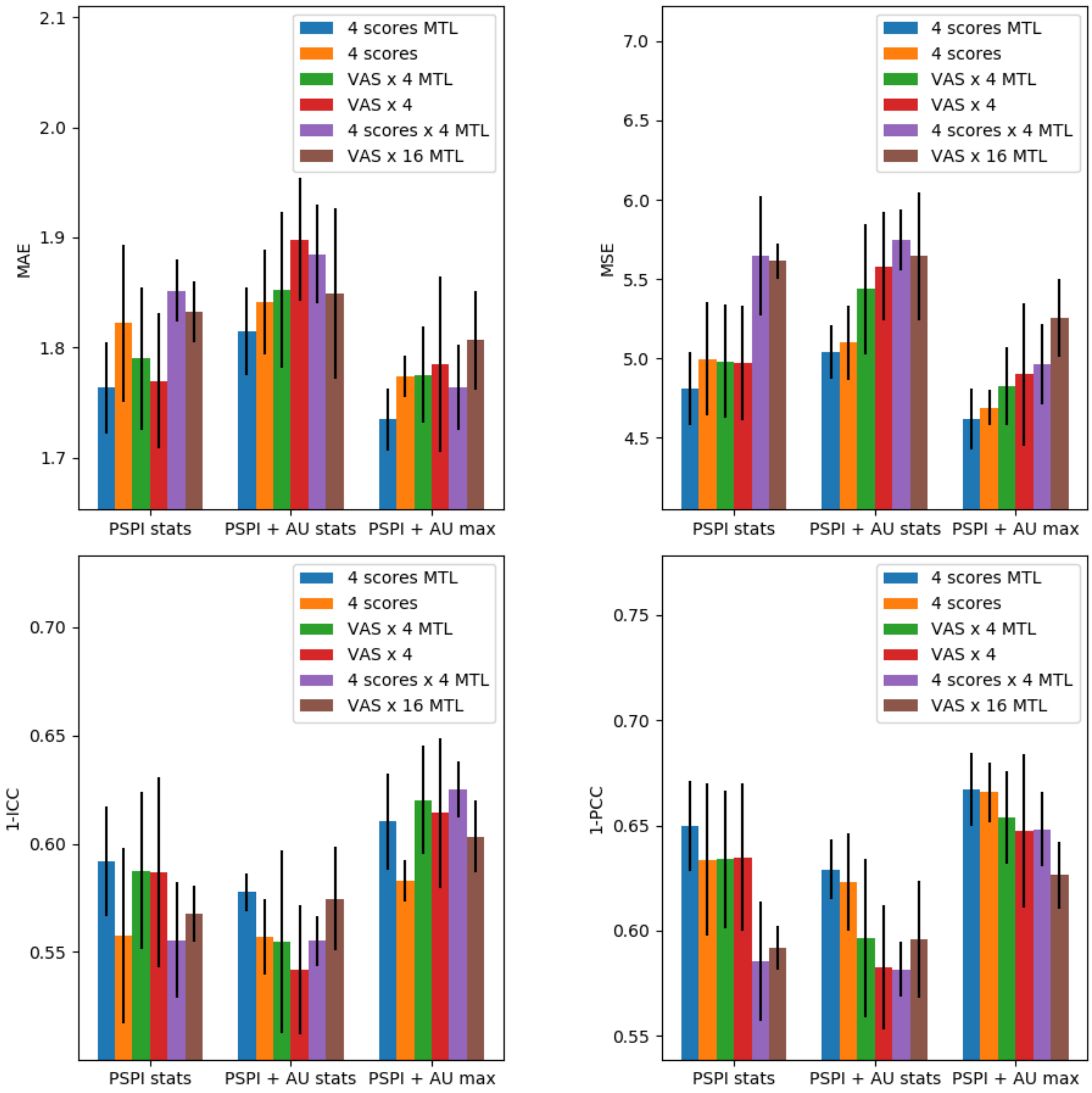


Figure 1.7. Bar graphs showing the MAE, MSE, 1-ICC, 1-PCC (we plot 1-ICC and 1-PCC instead of ICC and PCC so that for all sub-figures shorter bars mean better performance) of the following models predicting VAS using 3 different combinations (PSPI, PSPI+AU, PSPI+AU max) of frame-level labels: (1) 4 scores MTL. Predicting 4 scores using multitask learning. (2) 4 scores. Predicting 4 scores using 4 separate models. (3) VAS × 4 MTL. Predicting 4 VAS using multitask learning. (4) VAS × 4 predicting 4 VAS using 4 separate models. (5) 4 scores × 4 MTL. Predicting 4 copies of 4 scores using multitask learning. (6) VAS × 16 MTL. Predicting 16 copies of VAS using multitask learning.

Chapter 2

Automated Pain Detection in Facial Videos of Children using Human-Assisted Transfer Learning

2.1 Introduction

In the classic model of machine learning, scientists train models on a collected dataset to accurately predict a desired outcome and then apply learned models to new data measured under identical circumstances to validate performance. Given the notable variation in real world data, it is tempting to apply learned models to data collected under similar but non-identical circumstances. However, performance in such circumstances often deteriorates due to unmeasured factors not accounted for between the original and new datasets. Nevertheless, knowledge can be extracted in these scenarios. Transfer learning, or inductive transfer in machine learning parlance, focuses on using knowledge gained from solving one problem to improve performance on a different but related problem [WVW07]. The present paper describes application of transfer learning to the important clinical problem of automated pain detection in children.

Accurate measurement of pain severity in children is difficult, even for trained professionals and parents. This is a critical problem as over-medication can result in adverse side-effects, including opioid addiction, and under-medication can lead to unnecessary suffering [QSH15].

The current clinical gold standard and most widely employed method of assessing clinical pain is patient self-report [ZPG$^+$16]. However, this subjective method is vulnerable to self-presentation bias. Consequently, clinicians often distrust pain self-reports, and find them more useful for comparisons over time within individuals, rather than comparisons between individuals [VB09]. Further, infants, young children, and others with communication/neurological disabilities do not have the ability or capacity to self-report pain levels [ZPG$^+$16, SAD$^+$15, AKRP$^+$16]. As a result, to evaluate pain in populations with communication limitations, observational tools based on nonverbal indicators associated with pain have been developed [SFV$^+$17].

Of the various modalities of nonverbal expression (e.g., bodily movement, vocal qualities of speech), it has been suggested that facial activity provides the most sensitive, specific, and accessible information about the presence, nature, and severity of pain across the life span, from infancy [GC87] to advanced age [HHP$^+$14]. Moreover, observers largely consider facial activity during painful events to be a relatively spontaneous reaction [SFV$^+$17].

Evaluation of pain based on facial indicators requires two steps: (1) Extraction of facial pain features and (2) pain recognition based on these features. For step (1), researchers have searched for reliable facial indicators of pain, such as anatomically-based, objectively coded Facial Action Units (AUs) defined by the Facial Action Coding System (FACS) [EF76, MVJP17]. (Visualizations of facial activation units can also be found at https://imotions.com/blog/facial-action-coding-system/). However, identifying AUs traditionally requires time intensive offline coding by trained human coders, limiting application in real-time clinical settings. Recently, algorithms to automatically detect AUs [MVJP17] have been developed and implemented in software such as iMotions (imotions.com) allowing automatic output of AU probabilities in real-time based on direct recording of face video. In step (2), machine learning algorithms such as linear models [SAD$^+$15], SVM [ALC$^+$09], and Neural Networks [MR06] have been used to automatically recognize pain based on facial features.

Although a simple machine learning model based on features extracted by a well-designed algorithm can perform well when training and test data have similar statistical properties, problems arise when the data follow different distributions, as happens, for example, when videos are recorded in two different environments. We discovered this issue when training videos were recorded in an outpatient setting and test videos in the hospital. One way to deal with this problem is to use transfer learning, which discovers "common knowledge" across domains and uses this knowledge to complete tasks in a new domain with a model learned in the old domain [PY10]. In this work, we show that features extracted from human-coded (manual) AUs are less sensitive to domain changes than features extracted from iMotions (automated) AU codings, and thus develop a simple method that learns a projection from automated features onto a subspace of manual features. Once this mapping is learned, future automatically coded data can be transformed to a representation that is more robust between domains. In this work, we use a neural network model to learn a mapping from automated features to manual features, and another neural network model to recognize pain using the mapped facial features.

To summarize, our contributions of this work include demonstrating that:

- Manually/automatically coded AUs can be used to successfully recognize clinical pain in videos with machine learning.
- Environmental factors modulate the ability of automatically coded AUs to recognize clinical pain in videos.
- Manually coded AUs (especially previously established "pain-related" ones) can be used to successfully recognize pain in videos with machine learning across different environmental domains.
- Automatically coded AUs from iMotions do not directly represent or correlate with AUs defined in FACS.
- Transfering automated features to the manual feature space improves automatic recognition of clinical pain across different environmental domains.

This work was presented at the Joint Workshop on Artificial Intelligence in Health and a shorter version of this paper appeared in the proceedings [XCD+18].

2.2 Methods

2.2.1 Participants

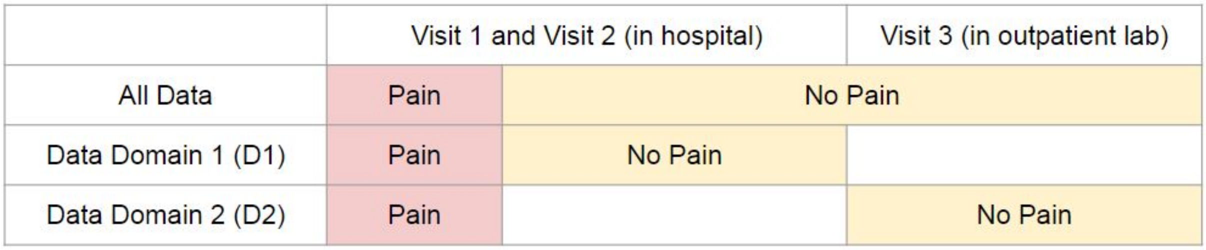
One hundred and forty-three pediatric research participants (94 males, 49 females) aged 12 [10, 15] (median [25%, 75%]) years old and primarily Hispanic (78%) who had undergone medically necessary laparoscopic appendectomy were videotaped for facial expressions during surgical recovery. Videos were subsequently categorized into two conditions: pain and no-pain. Participating children had been hospitalized following surgery for post-surgical recovery and were recruited for participation within 24 hours of surgery at a pediatric tertiary care center. Exclusion criteria included regular opioid use within the past six months, documented mental or neurological deficits preventing study protocol compliance, and any facial anomaly that might alter computer vision facial expression analysis. Parents provided written informed consent and youth gave written assent [HHG+18]. The local institutional review board approved the research protocol.

2.2.2 Experimental Design and Data Collection

Data were collected over three visits (V): V1 within 24 hours after appendectomy; V2 within the calendar day after the first visit; and V3 at a follow-up visit 25 [19, 28] (median [25%, 75%]) days postoperatively when pain was expected to have fully subsided. Data were collected in two environmental conditions: V1 and V2 in hospital and V3 in the outpatient setting. At every visit, two 10-second videos (60 frames per second at 853×480 pixel resolution) of the face were recorded while manual pressure was exerted at the surgical site for 10 seconds (equivalent of a clinical examination). During hospital visits (V1, V2), participants were lying in the hospital bed with the head of the bed raised. In the outpatient lab in V3, they were seated in a reclined

Table 2.1. Numbers of Samples at Different Pain Levels in Each Visit.

Pain Level	0	1	2	3	4	5	6	7	8	9	10
V1	16	12	18	28	31	26	26	19	24	15	11
V2	4	18	24	40	21	23	16	13	14	8	4
V3	166	17	3	1	0	0	0	0	0	0	0



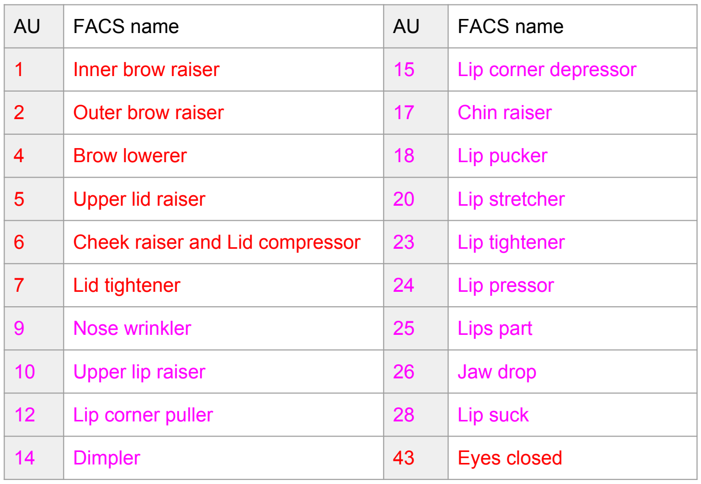
	Visit 1 and Visit 2 (in hospital)		Visit 3 (in outpatient lab)
All Data	Pain	No Pain	
Data Domain 1 (D1)	Pain	No Pain	
Data Domain 2 (D2)	Pain		No Pain

Figure 2.1. Data Domain Illustration. The area of category is not proportional to the number of samples.

chair. Participants rated their pain level during manual pressure using a 0-10 Numerical Rating Scale, where 0 = no-pain and 10 = worst pain ever. For classification purposes, and following convention used by clinicians for rating clinically significant pain [HSDA10], videos with pain ratings of 0-3 were labeled as no-pain, and videos with pain ratings of 4-10 were labeled as pain. Two hundred and fifty-one pain videos were collected from V1/2, 160 no-pain videos were collected from V1/2, and 187 no-pain videos were collected from V3. The numbers of samples collected for different pain levels and visits are shown in Table 2.1. Note that all V3 data are labeled as no-pain and there are only 4 pain ratings over 1 in V3. In contrast, the majority of no-pain data in V1 and V2 are ratings of 2 and 3. Figure 2.1 “All Data” demonstrates the distribution of pain and no-pain videos across environmental conditions.

2.2.3 Feature Extraction

For each 10-second video sample we extracted AU codings per frame to obtain a sequence of AUs. This was done both automatically by iMotions software (www.imotions.com) and manually by a FACS trained human in a limited subset. A second trained human independently coded a subset of the videos coded by the first human. We then extracted features from the



AU	FACS name	AU	FACS name
1	Inner brow raiser	15	Lip corner depressor
2	Outer brow raiser	17	Chin raiser
4	Brow lowerer	18	Lip pucker
5	Upper lid raiser	20	Lip stretcher
6	Cheek raiser and Lid compressor	23	Lip tightener
7	Lid tightener	24	Lip pressor
9	Nose wrinkler	25	Lips part
10	Upper lip raiser	26	Jaw drop
12	Lip corner puller	28	Lip suck
14	Dimpler	43	Eyes closed

Figure 2.2. FACS names (descriptions) of 20 AUs coded by iMotions. AUs 1-7 and 43 are upper face AUs, and the others are lower face AUs.

sequence of AUs.

Automated Facial Action Unit Detection:

The iMotions software integrates Emotient's FACET technology (www.imotions.com), formally known as CERT [LWW$^+$11]. In the described work, iMotions software was used to process videos to automatically extract 20 AUs as listed in Figure 2.2 and three head pose indicators (yaw, pitch and roll) from each frame. The values of these codings represent estimated log probabilities of AUs, ranging from -4 to 4.

Manual Facial Action Unit Detection:

A trained human FACS AU coder manually coded 64 AUs (AU1-64) for each frame of a subset (54%) of videos and labeled AU intensities (0-5, 0 = absence). In order to evaluate the reliability of the manual codings, we had another trained human coder code a subset (15%) of videos coded by the first human.

Feature Dimension Reduction:

The number of frames in our videos was too large to use full sequences of frame-coded AUs. To reduce dimensionality, we applied 11 statistics (mean, max, min, standard deviation, 95th, 85th, 75th, 50th, 25th percentiles, half-rectified mean, and max-min) to each AU over all frames as in [SAD$^+$15] to obtain 11×23 features for automatically coded AUs, and 11×64 features for manually coded AUs. We call these automated features and manual features, respectively. The range of each feature was rescaled to $[0,1]$ to normalize features over the training data.

2.2.4 Machine Learning Models

Neural Network Model to Recognize Pain with Extracted Features:

A neural network with one hidden layer was used to recognize pain with extracted automated or manual features. The number of neurons in the hidden layer was twice the number of neurons in the input layer, and the Sigmoid activation function $\sigma(x) = 1/(1+\exp(-x))$ was used with batch normalization for the hidden layer. The output layer used Softmax activation and cross-entropy error.

Neural Network Model to Predict Manual Features with Automated Features:

A neural network with the same structure was used to predict manual features from automated features, except that the output layer was linear and mean squared error was used as the loss function.

Model Training and Testing:

Experiments were conducted in a participant-based (each participant restricted to one fold) 10-fold cross-validation fashion. Participants were divided into 10 folds, and each time 1 fold was used as the test set, and the other 9 folds together were used as the training set. We balanced classes for each participant in each training set by randomly duplicating samples from the under-represented class. One out of nine participants in the training sets were picked randomly as a nested-validation set for early stopping in the neural network training. A batch size of $1/8$ the size of training set was used.

Table 2.2. AUC for Classification with SEM (Standard Error of The Mean).

Train on	Test on	Automated	Manual	Automated "Pain" Features	Manual "Pain" Features
All	D1	0.61 ± 0.006	0.66 ± 0.006	0.63 ± 0.007	$\mathbf{0.69 \pm 0.006}$
D1	D1	0.58 ± 0.014	0.62 ± 0.008	0.61 ± 0.008	$\mathbf{0.65 \pm 0.008}$
D2	D1	0.57 ± 0.005	0.67 ± 0.007	0.62 ± 0.004	$\mathbf{0.7 \pm 0.006}$
All	D2	0.9 ± 0.005	0.79 ± 0.007	0.88 ± 0.005	0.8 ± 0.003
D1	D2	0.69 ± 0.011	0.68 ± 0.008	0.73 ± 0.012	0.73 ± 0.01
D2	D2	0.92 ± 0.01	0.79 ± 0.009	0.9 ± 0.007	0.8 ± 0.005

We then examined the receiver operating characteristic curve (ROC curve) which plots True Positive Rate against False Positive Rate as the discrimination threshold varies. We used Area under the Curve (AUC) to evaluate classification performance. We considered data from three domains (D) as shown in Figure 2.1: (1) D1 with pain and no-pain both from V1/2 in hospital; (2) D2 with pain from V1/2 in hospital and no-pain from V3 from outpatient lab; and (3) All data, i.e., pain from V1/2 and no-pain from V1/2/3. The clinical goal was to be able to discriminate pain levels in the hospital; thus evaluation on D1 (where all samples were from the hospital bed) was the most clinically relevant evaluation.

2.3 Analysis and Discussion

Data from 73 participants labeled by both human and iMotions were used through section 2.3.1 to 2.3.5, and data from the remaining 70 participants using only automated (iMotions) AU codings were included for independent test set evaluation in the results section.

2.3.1 Automated Classifier Performance Varies by Environment

Using automated features, we first combined all visit data and trained a classifier to distinguish pain from no-pain. This classifier performed well in general (AUC$= 0.77 \pm 0.011$ on All data), but when we looked at different domains, the performance of D1 (the most clinically relevant in-hospital environment) was inferior to that on D2, as shown in data rows 1 and 4 under the "Automated" column in Table 2.2.

There were two main differences between D1 and D2, i.e., between V1/2 and V3 no-pain samples. The first was that in V1/2, participants still had some pain and their self-ratings were greater than 0, while in V3, no-pain ratings were usually 0 reflecting a “purer” no-pain signal. The second difference was that V1/2 occurred in the hospital with patients in beds and V3 videos were recorded in an outpatient setting with the participant sitting in a reclined chair. Lighting was also inherently different between hospital and outpatient environments. Since automated recognition of AUs is known to be sensitive to facial pose and lighting differences, we hypothesized that added discrepancy in classification performance between D1 and D2 was mainly due to the model classifying on environmental differences between V1/2 and V3. In other words, when trained and tested on D2, the classifier might distinguish “lying in hospital bed” vs “more upright in outpatient chair” as much as pain vs no-pain (this is similar to a computer vision algorithm doing well at recognizing cows by recognizing a green background).

In order to investigate this hypothesis and attempt to improve classification on the clinically relevant D1, we trained a classifier using only videos from D1. Within the “Automated” column, row 2 in Table 2.2 shows that performance on automated D1 classification does not drop much when D2 samples are removed from the training set. At the same time, training using only D2 data results in the worst classification on D1 (row 3), but the best classification on D2 (last row) as the network is able to exploit environmental differences (no-pain+more upright from V3, pain+lying-down from V1/2).

Figure 2.3 (b) (LEFT) shows ROC curves of within and across domain tests for models trained on automated features in D2. The dotted (red) curve corresponds to testing on D2 (within domain) and the solid (blue) curve corresponds to testing on D1 (across domain). The model performed well on within domain classification, but failed on across domain tasks.

2.3.2 Classification Based on Manual AUs Are Less Sensitive to Environmental Changes

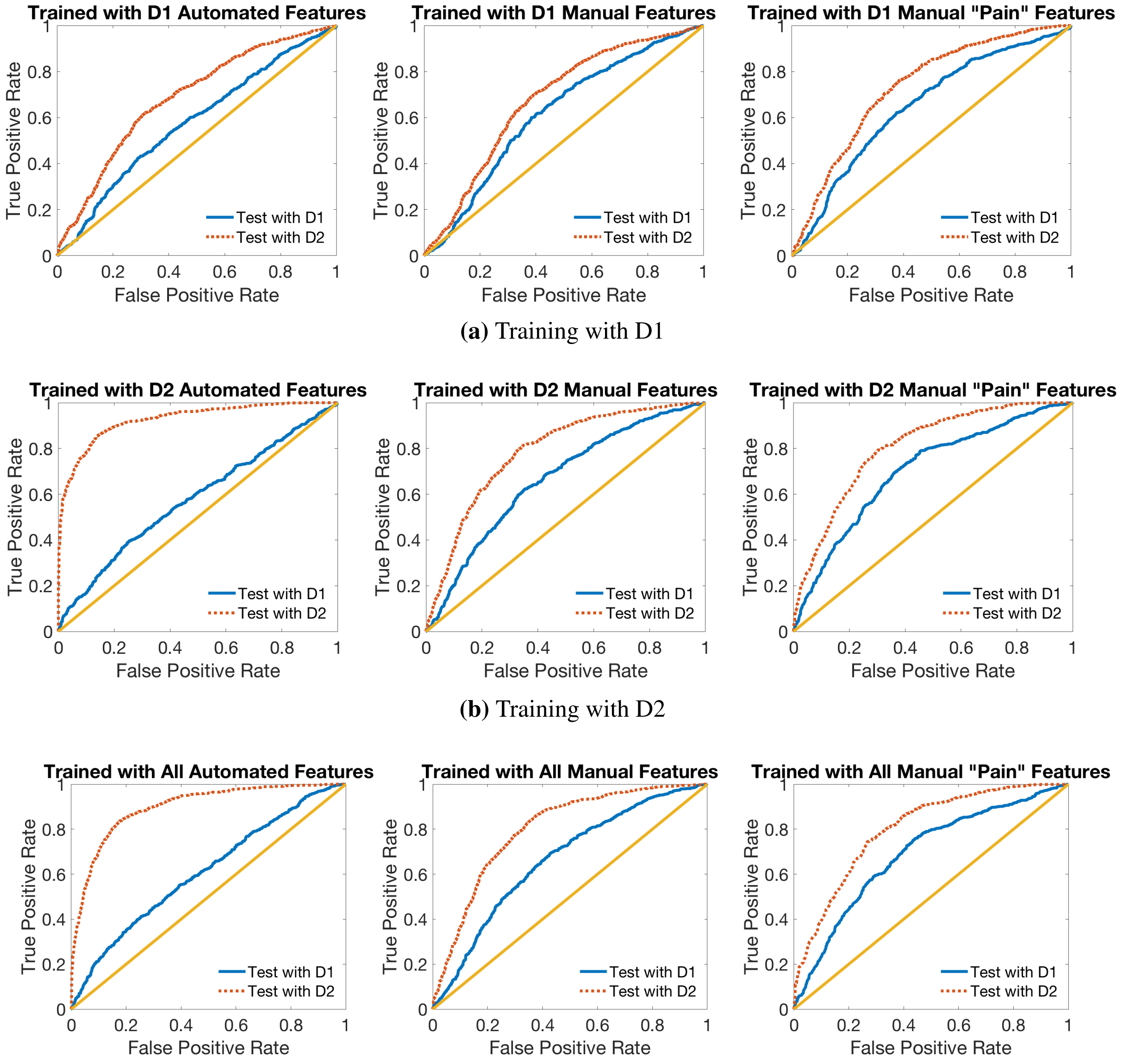
We also trained a classifier on manual AUs labeled by a human coder. Interestingly, results from the classifier trained on manual AUs showed less of a difference in AUCs between domains, with a higher AUC for D1 and a lower AUC for D2 relative to those with automated AUs (see Table 2.2 “Manual” and “Automated” columns). Overall, manual AUs appeared to be less sensitive to changes in the environment, reflecting the ability of human labelers to consistently code AUs without being affected by lighting and pose variations.

When we restricted training data from All to only D1 or only D2 data, classification performance using manual AUs went down, likely due to the reduction in training data, and training with D2 always gave better performance than training with D1 on both D1 and D2 test data, which should be the case since pain and no-pain samples in D2 are more discrepant in average pain rating. These results appear consistent with our hypothesis that human coding of AUs is not as sensitive as machine coding of AUs to environmental differences between V1/2 and V3.

Figure 2.3 (b) (MIDDLE) displays ROC curves for manual features. As discussed above, in contrast to the plot on the left for automated features, manual coding performance outperformed automated coding performance in the clinically relevant test in D1. The dotted (red) curve representing within-domain performance is only slightly higher than the solid (blue) curve, likely due in part to the quality difference in no-pain samples in V1/2 and V3, and also possibly any small amount of environmental information that the human labeler was affected by. Note that ignoring the correlated environmental information in D2 (i.e., pain faces were more reclined and no-pain faces were more upright) resulted in a lower numerical performance on D2 but does not likely reflect worse classification of pain but instead the failure to “cheat” by using features affected by pose angle to classify all upright faces as “no-pain.”

2.3.3 Restricting Manual AUs to Those Associated with Pain Improves Classification

In an attempt to reduce the influence of environmental conditions to further improve performance on D1, we restricted the classifier to the eight AUs consistently associated with pain: 4 (Brow Lowerer), 6 (Cheek Raiser), 7 (Lid Tightener), 9 (Nose Wrinkler), 10 (Upper



(**a**) Training with D1

(**b**) Training with D2

(**c**) Training with All

Figure 2.3. ROC Curves for classification on D1 and D2 using automated features (left), manual features (middle) and pain-related manual features (right), when the model is trained on (a) D1, (b) D2 and (c) All data. The dotted (red) lines are ROCs when the machine is able to use environment information to differentiate pain and no-pain conditions, and the solid (blue) lines show the machine's ability to discriminate between pain and no-pain based on AU information alone. The straight (yellow) line graphs the performance of random chance.

Lip Raiser), 12 (Lip Corner Puller), 20 (Lip Stretcher), and 43 (Eyes Closed) [Prk92, Prk09] as illustrated in Figure 2.4 to obtain 11 (statistics) ×8 (AUs) features. Pain prediction results using these "pain" features are shown in the last two columns in Table 2.2. Results show that using

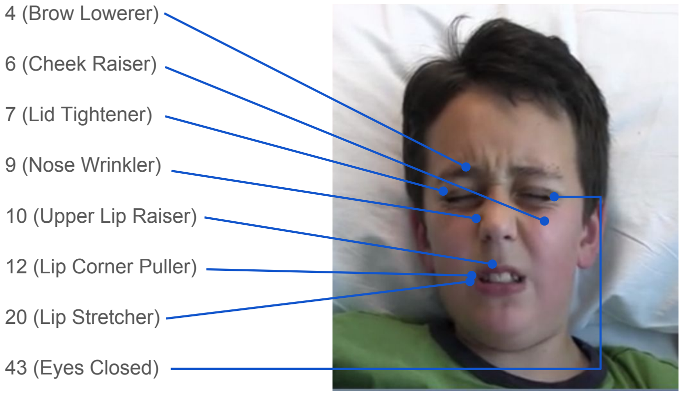


Figure 2.4. Illustration of eight "pain-related" facial AUs.

only pain-related AUs improved classification performance of manual features. However, it did not seem to help as much for automated features.

Similarly, Figure 2.3 (b) (RIGHT) shows that limiting manual features to use only pain-related AUs further improved D1 performance when training with D2. We also employed PCA on pain-related features and found that performance in the hospital domain was similar if using four or more principal components.

In Figure 2.3 (a) and (c) we show ROC curves similar to Figure 2.3 (b) except with different training data. These curves correspond to row 2 and 5 (a), or 1 and 4 (c), under "Automated," "Manual," and "Manual 'Pain' Features" in Table 2.2.

2.3.4 iMotions AUs Are Different Than Manual FACS AUs

Computer Vision AU automatic detection algorithms have been programmed/ trained on manual FACS data. However, we demonstrate differential performance of AUs encoded automatically versus manually. To understand the relationship between automatically encoded v. manually coded AUs, we computed correlations between binarized automatically coded AUs and manually coded AUs at the frame level as depicted in Figure 2.5. The FACS names corresponding to AU numbers are listed in Figure 2.2, in which AUs 1, 2, 4, 5, 6, 7, 43 are upper face AUs and all others are lower face AUs. If two sets of AUs were identical, the diagonal of the matrix (marked with small centered dots) should yield the highest correlations, which was

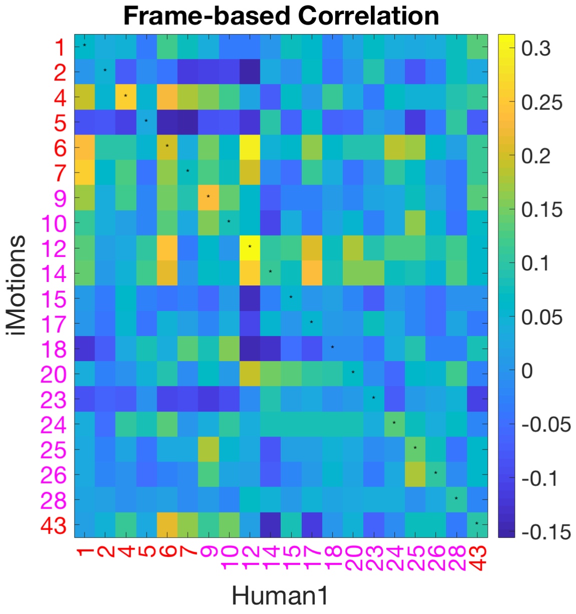


Figure 2.5. Correlation matrix of AU pairs from automated and manual codings using All data.

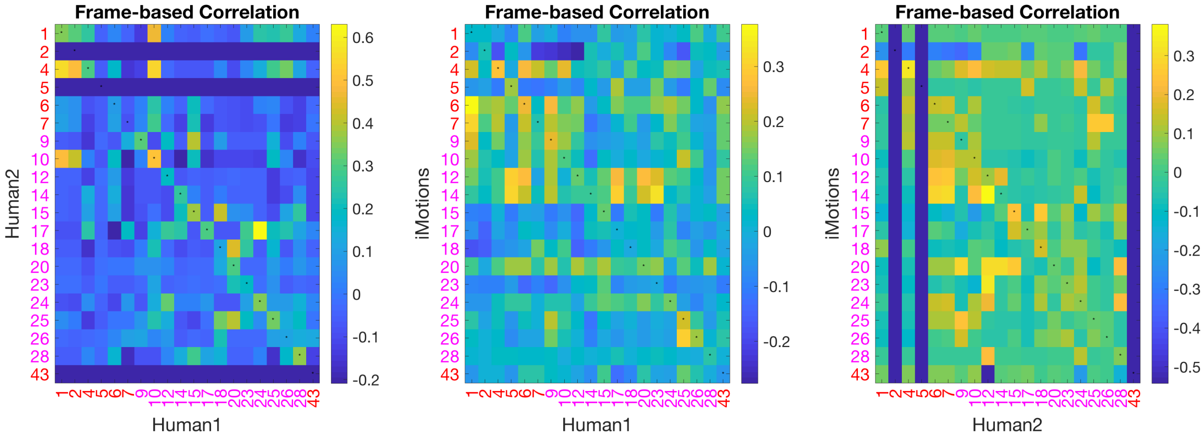


Figure 2.6. Correlations of AU pairs from two of (1)iMotions; (2)human 1; and (3) human 2 on a subset of the data.

not the case. For example, manual AU 6 was highly correlated with automated AU 12 and 14, but had relatively low correlation with automated AU 6.

The correlation matrix shows that not only is our first human coder less affected by environmental changes, the AUs she coded are not in agreement with the automated AUs. Our second trained human coder (human 2) shows a better correlation with the coding of human 1 than between each human and iMotions, shown in Figure 2.6 (LEFT). The correlation between

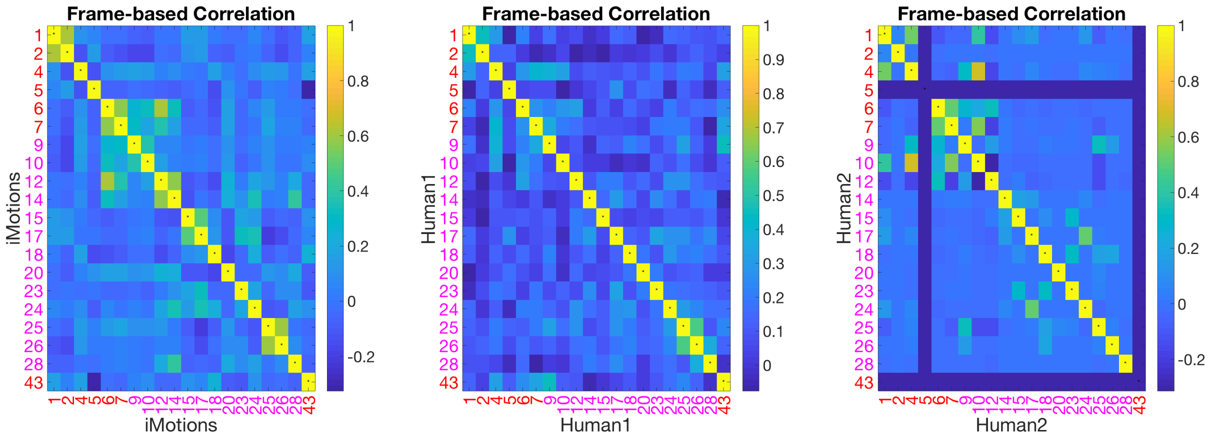


Figure 2.7. Self-correlation Matrices of AU pairs from iMotions or humans.

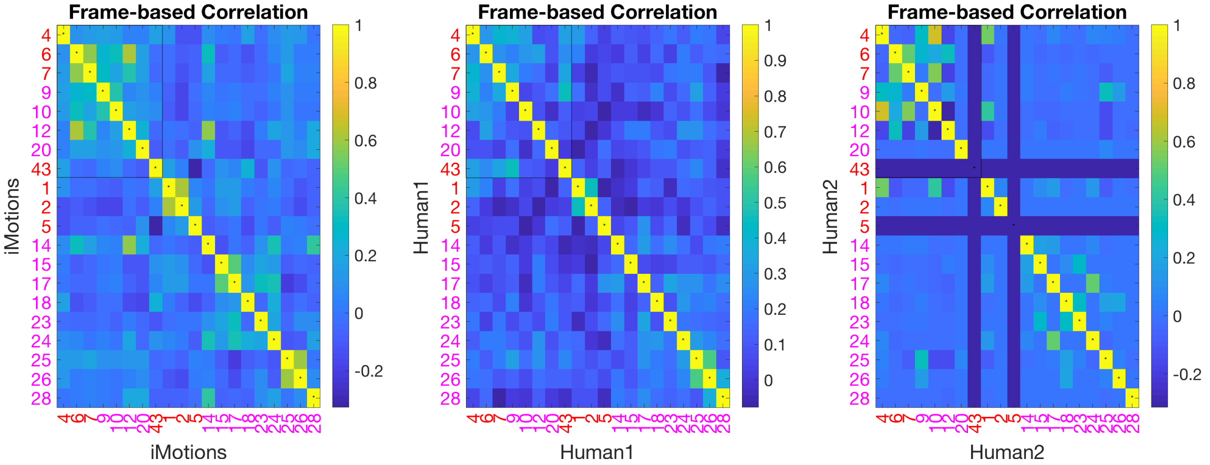


Figure 2.8. Self-correlation Matrices of AU pairs from iMotions or humans with "pain" AUs arranged together at the top left corner.

each of the humans and the software on the same subset is shown in Figure 2.6 (MIDDLE, RIGHT). This likely explains the reduced improvement by restricting the automated features model to "pain-related AUs" as these have been determined based on human FACS coded AUs.

The self-correlation matrices between AUs in iMotions and the human coder are shown in Figure 2.7. AUs coded by iMotions show higher correlations (between different iMotions coded AUs) than AUs coded by humans. Some human AU codings were also correlated, which is expected since specific AUs often occur together (e.g., AU 1 and 2 for inner and outer brow raiser and AU 25 and 26 for lips part and jaw drop) and other AUs tend to occur together in pain.

Table 2.3. AUC (and SEM) with Transferred Automated Features.

Train on	Test on	All Features	"Pain" Features	7 PCs	4 PCs	1 PC
All	D1	0.61 ± 0.009	0.63 ± 0.009	0.68 ± 0.006	$\mathbf{0.69 \pm 0.008}$	0.65 ± 0.009
D1	D1	0.62 ± 0.009	0.64 ± 0.014	0.66 ± 0.012	$\mathbf{0.67 \pm 0.011}$	0.65 ± 0.009
D2	D1	0.58 ± 0.011	0.59 ± 0.01	0.66 ± 0.008	$\mathbf{0.68 \pm 0.006}$	0.66 ± 0.009
All	D2	0.82 ± 0.009	0.82 ± 0.009	0.76 ± 0.009	0.75 ± 0.012	0.7 ± 0.01
D1	D2	0.69 ± 0.009	0.71 ± 0.013	0.7 ± 0.015	0.71 ± 0.015	0.69 ± 0.011
D2	D2	0.88 ± 0.011	0.86 ± 0.006	0.76 ± 0.013	0.74 ± 0.01	0.7 ± 0.009

This latter correlation of pain AUs is more evident in Figure 2.4 which shows the same content as Figure 2.7 except that in Figure 2.4 the eight pain-related AUs are put together at the upper left corner to highlight their higher correlations. Interestingly, higher correlations within the pain AUs for iMotions coding was observed but the pattern is different.

2.3.5 Transfer Learning via Mapping to Manual Features Improves Performance

We have shown that manual codings are not as sensitive to domain change. However, manual coding of AUs is very time-consuming and not amenable to an automated real-time system. In an attempt to leverage manual coding to achieve similar robustness with automatic AUs, we utilized transfer learning and mapped automated features to the space of manual features. Specifically, we trained a neural network model to estimate manual features from automated features using data coded by both iMotions and a human. Separate models were trained to predict: manual features of 64 AUs, manual features of the eight pain-related AUs, and principal components (PCs) of the manual features of the eight pain-related AUs. PCA dimensionality reduction was used due to insufficient data for learning an accurate mapping from all automated AUs to all manual AUs.

Once the mapping network was trained, we used it to transform the automated features and trained a new network on these transformed data for pain/no-pain classification. The 10-fold cross-validation was done consistently so that the same training data was used to train the mapping network and the pain-classification network.

In Table 2.3, we show classification AUCs when the classification model was trained

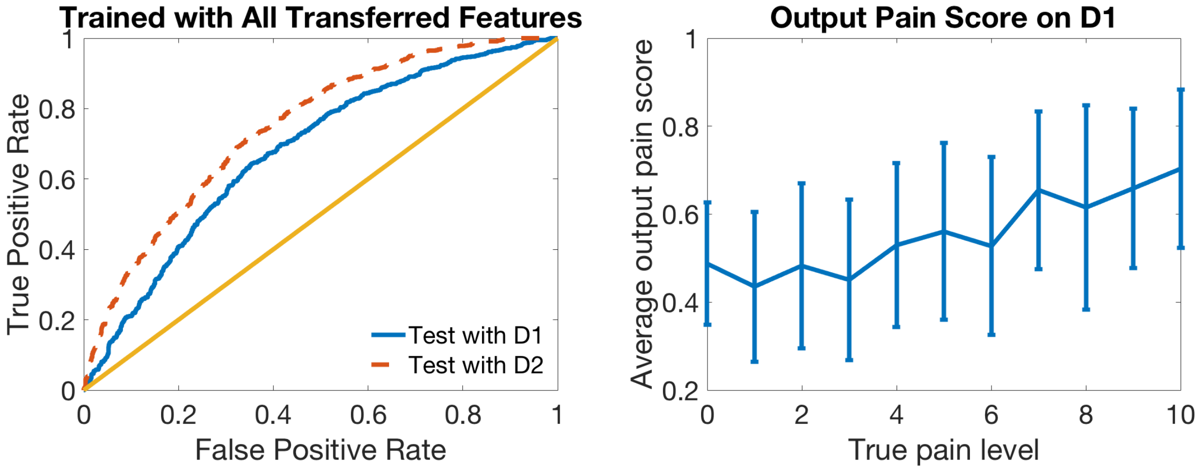


Figure 2.9. ROC Curves for classification on two domains using our transfer learning model (left) and plot of average model output pain score (with error bars indicating standard deviation) over true pain level (right).

and tested with outputs from the prediction network. We observed that when using All data to train (which performed best), with the transfer learning prediction network, automated features performed much better in classification on D1 ($0.68-0.69$ compared to $0.61-0.63$ in Table 2.2). Predicting four principal components of manual pain-related features yielded the best performance in our data. Overall, the prediction network helped in domain adaptation of a pain recognition model using automatically extracted AUs.

Figure 2.9 (LEFT) plots the ROC curves on two domains using the transfer learning classifier trained and tested using four predicted features. The model performed well in across-domain classification. Compared to Figure 2.3 (c) (LEFT), the transferred automated features showed properties more similar to manual features (Figure 2.3 (c) (RIGHT)), with smaller differences between performance on the two domains and higher AUC on the clinically relevant D1. Table 2.3 shows numerically how transfer learning helped automated features ignore environmental information in D2 like humans, and learn pure pain information that can be used in classification on D1.

Within-domain classification performance for D1 was also improved with the prediction network. These results show that by mapping to the manual feature space, automated features

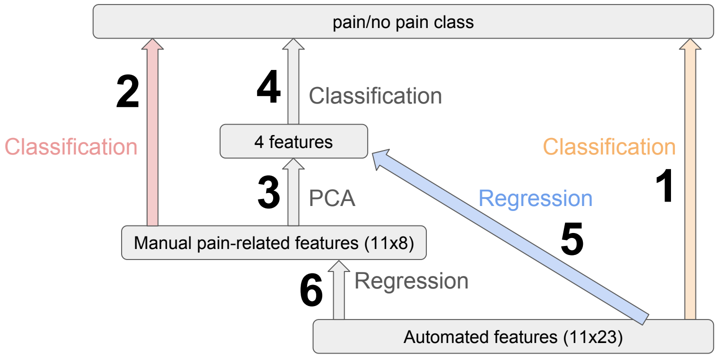


Figure 2.10. Illustration of Machine Learning Models. 1/2 are classifications using automated/manual pain features, in which 2 does better than 1. 3-4 can be done to reduce feature dimensions while maintaining performance. 6-2 and 5-4 are our transfer learning models, training a regression network to map automated features to a subspace of manual pain features before classification.

can be promoted to perform better in pain classification.

Figure 2.9 (RIGHT) plots output pain scores of our model tested on D1 versus 0-10 self-reported pain levels. The model output pain score increases with true pain level, indicating that our model indeed reflects pain levels.

2.4 Results

In the previous section we showed that in Figure 2.10 classification with pain-related pain features (2) performed better than automated features (1) on D1, which was the clinically relevant classification. We also found that applying PCA to manual features (3-4) does not change performance on D1 much. Thus, we introduced a transfer learning model to map automated features first to manual pain-related features (or the top few principal components of them), and then used the transferred features for classification (6-2 or 5-4). We obtained similar results to manual features on D1 with the transfer learning model (5-4) mapping to four principal components of manual features.

Table 2.2 shows that without our transfer learning method, training on all data and restricting to pain-related AUs results in the best performance using automated features for

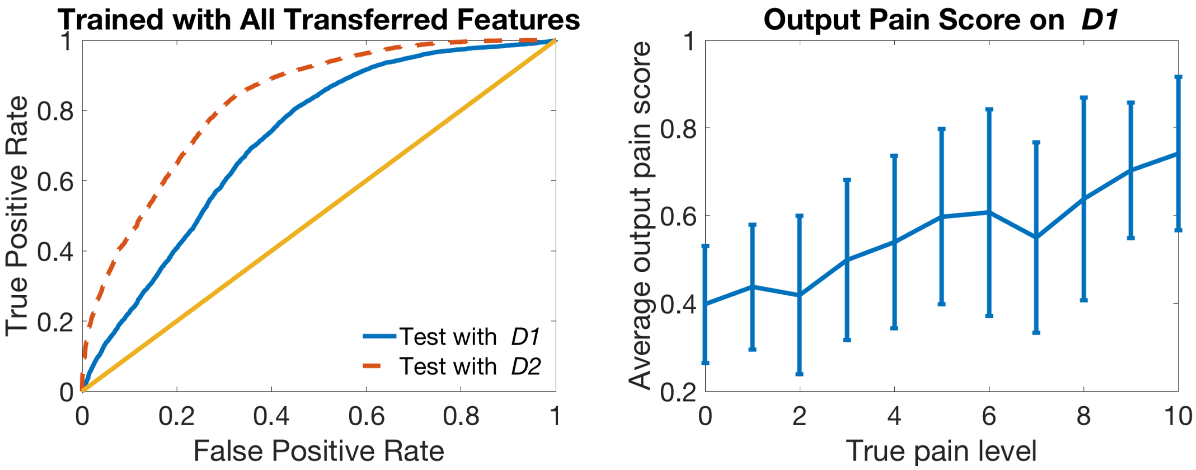


Figure 2.11. ROC Curves for classification on NEW test domains *D1* and *D2* using our transfer learning model (left) and plot of average model output pain score (with error bars indicating standard deviation) over true pain level (right).

D1. And cross-validation results in Table 2.3 shows that with our method, using all data and predicting four PCs yielded the best performance for D1. With these optimal choices of model structure and training domain before and after transfer learning, we show the benefits of transfer learning in two experiments.

2.4.1 Test on New Subjects with Only iMotions AU Codings

In this section we report on the results from testing our transfer learning method on a new separate dataset (new participants), which contained only automated features. We trained two models, with and without transfer learning, using all the data in section 2.3 labeled by both iMotions and humans, and tested the model on this new dataset only labeled by iMotions *D1, D2*. (We use italicized domain names to indicate that this is independent test data *D1, D2*.) Our model with transfer learning (AUC$= 0.72 \pm 0.002$) performed better than the model without it (AUC$= 0.67 \pm 0.002$) on *D1* with a p-value$= 1.33e-45$ in a one-tailed two-sample t-test.

Similar to Figure 2.9, in Figure 2.11 we plot ROC curves for classification on the NEW test dataset (LEFT) and output pain scores at 0-10 pain levels (RIGHT) using our transfer learning model.

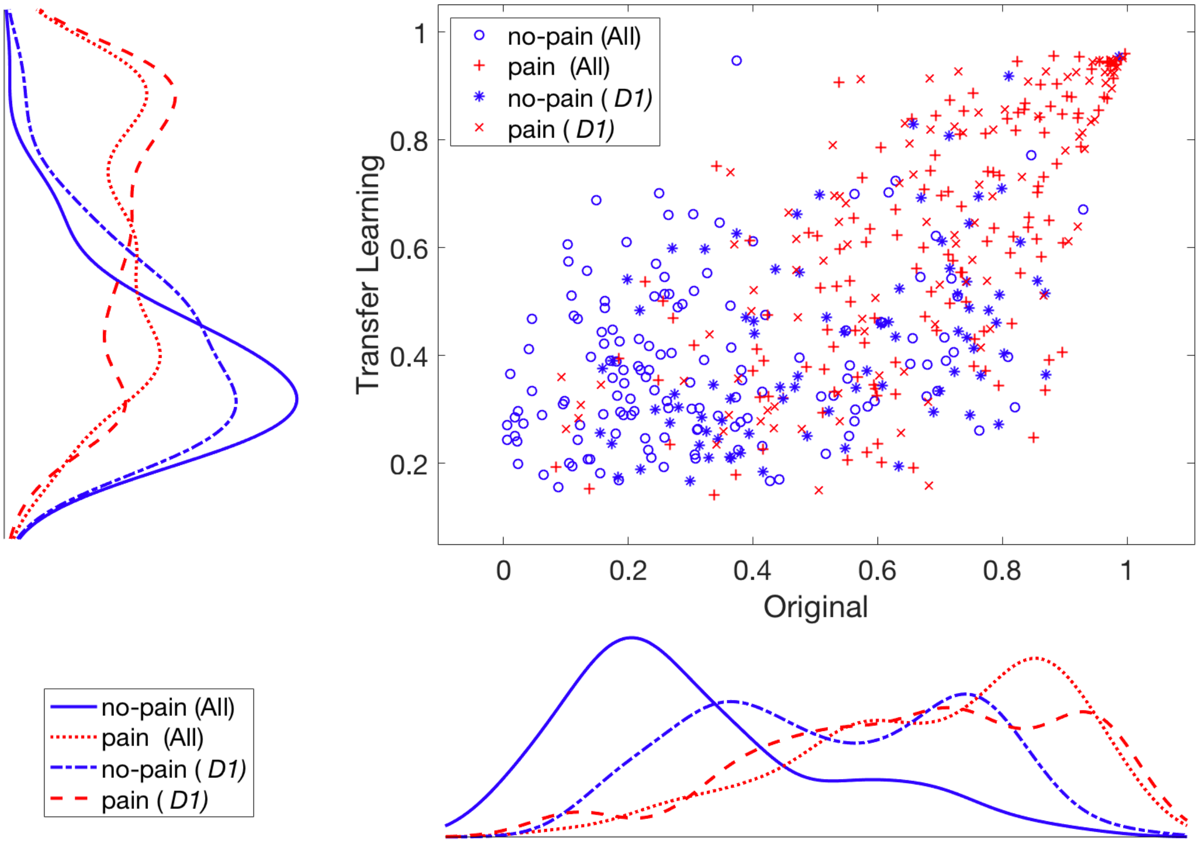


Figure 2.12. Scatter plot and distributions of pain scores (transfer learning vs original) using original iMotions features (on the x-axis) and transferred iMotions features (on the y-axis).

In Figure 2.12, we show a scatter plot of neural network output pain scores using transferred automated features versus those using original automated features, as well as pain score distributions, separately for training (All Data from section 2.3) and test (*D1* from NEW test data in the current section), pain and no-pain. We can see for original automated features scores, no-pain samples from *D1* are distributed very differently from no-pain in All data domain used for training and fall mostly in the range of the pain class. Results using transfer learning do not appear to have this problem.

2.4.2 Test with Masked Pain and Faked Pain

As another test of the effect of our transfer learning model, we looked at results of classifying whether participants are in pain or not from videos where children were asked to fake pain when they were not really in pain as well as when they were asked to suppress visual expressions of pain when they were in pain.

Although facial expressions convey rich and objective information about pain, they can be deceptive because people can inhibit or exaggerate their pain displays when under observation [HC04]. It has been shown that human observers discriminate real expressions of pain from faked expressions only marginally better than chance [HC04, BLFL14]. Children can also be very good at suppressing pain, but not fully successful in faking expressions of pain [LCC06]. In this section we discuss performance of masked and faked pain in machine learning models trained to distinguish genuine pain and no-pain.

In addition to the data described in section 2.2.2, we recorded videos of "masked pain" in V1 and V2 by asking participants to suppress pain during the 10-second manual pressure, and videos of "faked pain" during V3 by asking participants to fake the worst pain ever during manual pressure. As in section 2.2.2, we asked participants to rate their true pain level during manual pressure with a number from 0 to 10. We then labeled masked-pain videos with pain ratings of 4-10 as masked-pain and faked-pain videos with pain ratings of 0-3 as faked-pain, and discarded other samples. This ensured that in masked-pain videos participants actually experienced pain and in faked-pain videos participants in fact felt no pain. One hundred and seventeen masked-pain samples and 116 faked-pain samples were collected. The distribution of the four classes within the three visits is shown in Figure 2.13.

Using the best models before and after transfer learning trained to distinguish between genuine pain and no-pain described above, the masked and faked pain samples were processed to obtain pain labels. The results are shown in Figure 2.14. We can see that without transfer learning (LEFT), most masked-pain data were classified as real-pain and most faked-pain as no-pain. This appeared to be the case because the AU features coded automatically were sensitive to environmental factors, and during training the machine learned to discriminate between genuine pain and no-pain by recognizing environmental differences between them. At test time, since masked-pain is in the same environmental domain as real-pain and faked-pain is in the

<table>
<tr><th></th><th colspan="2">Visit 1 and Visit 2 (in hospital)</th><th>Visit 3 (in outpatient lab)</th></tr>
<tr><td>Genuine Expression (All Data)</td><td>Real Pain</td><td colspan="2">No Pain</td></tr>
<tr><td>Non-genuine Expression</td><td colspan="2">Masked Pain</td><td>Faked Pain</td></tr>
</table>

Figure 2.13. Distribution of four classes in three visits. The area of category is not proportional to the number of samples.

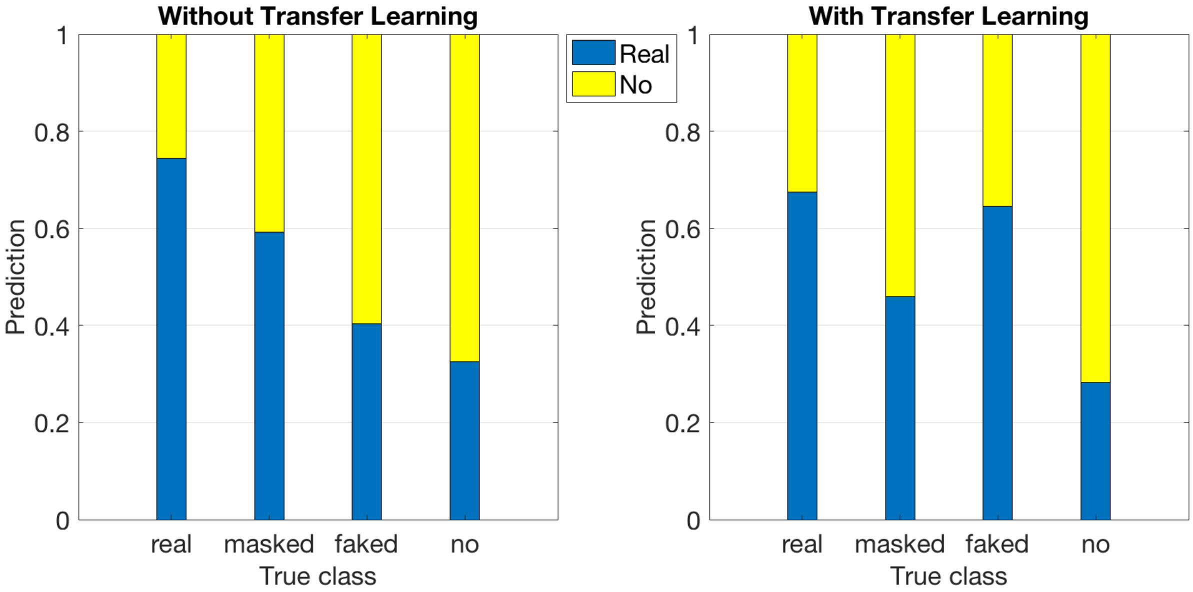


Figure 2.14. Bar graph showing classification of real-pain, masked-pain, faked-pain and no-pain. The area of bars shows the distribution of predicting pain and no-pain.

similar environment as no-pain, they are assigned to the corresponding classes. In contrast, with transfer learning (Figure 2.14 (RIGHT)), masked-pain was mostly classified as no-pain and faked-pain as real-pain. This might be because automated features were transferred to ignore the difference between the two classes caused by environmental change, and the machine can only use differences in facial actions to complete the classification task. Humans' attempts to mask pain are to mimic no-pain faces and, similarly, humans' attempts to fake pain are to mimic pain faces. The machine in this way classifies pain and no-pain according to expressed facial actions.

2.5 Conclusion

In the present work we recognized differences in classifier model performance (pain vs no-pain) across domains that reflect environmental differences as well as differences reflecting how the data were encoded (automatically v. manually). We demonstrate that manually coded facial features are more robust than automatically coded facial features to environmental changes which allow us to obtain the best performance on our target data domain. We then introduced a transfer learning model to map automated features first to manual pain-related features (or principal components of them), and then used the transferred features for classification (6-2 or 5-4 in Figure 2.10). This allowed us to leverage data from another domain to improve classifier performance on the clinically relevant task of automatically distinguishing pain levels in the hospital. Further, we were able to demonstrate improved classifier performance on a separate, new data set.

2.6 Future Work

Planned future work:

1. Classification of real-pain, masked-pain, faked-pain, and no-pain using machine learning, and comparison to human judgments.

2. Classification of genuine expression and non-genuine expression using machine learning, and comparison to human judgments.

3. Using transfer learning to improve fusion analysis of video features and peripheral physiological features in [XSN$^+$18].

4. Multidimensional pain assessment such as pain catastrophizing and anxiety based on facial activities.

Acknowledgments

This work was supported by National Institutes of Health National Institute of Nursing Research grant R01 NR013500, NSF IIS 1528214, and by IBM Research AI through the AI Horizons Network. Many thanks to Ryley Unrau for manual FACS coding and Karan Sikka for sharing his code and ideas used in [SAD^{+}15].

Chapter 2, in full, is a reprint of the material as it appears in Artificial Intelligence in Healt 2018. Xu, Xiaojing, Kenneth D. Craig, Damaris Diaz, Matthew S. Goodwin, Murat Akcakaya, Büşra Tuğçe Susam, Jeannie S. Huang, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Pain Evaluation in Video using Extended Multitask Learning from Multidimensional Measurements

3.1 Introduction

Reading facial expressions is one of the most useful ways that humans perceive pain in others [For76]. Accurate measurement of the pain severity, however, is difficult even for trained professionals. The current clinical gold standard and most widely employed method of assessing clinical pain is patient self-report [ZPG$^+$16]. However, this method is subjective and vulnerable to social and self-presentation biases and requires substantial cognitive, linguistic, and social competencies [ZPG$^+$16, SAD$^+$15, AKRP$^+$16]. The goal of an automated facial pain recognition model is to generate a pain level based on facial videos that predicts the patient's self-reported visual analog scale (VAS) pain level. The model should be able to generalize to new patients, for example those with communication disabilities.

Pain is multidimensional. Major dimensions of pain include physiological, sensory, affective, cognitive, behavioral, and sociocultural [McG92] aspects. Self-reported VAS values the subjective nature of pain, which has been shown to be not as stable and accurate as a multidimensional assessment [RRM$^+$07, ABR83, CYT$^+$02]. In this paper, we analyzed the relationship between several pain measurements and their predictions from a machine learning model, and proposed a novel method to learn a pain score as a combination of several dimensions of pain to better approximate the patient's VAS level.

A natural way to predict a pain score using video is to use a 3D CNN. However, this is difficult in clinical pain detection because (1) clinical pain datasets are usually too small to train a deep model and (2) the length of the video is not fixed. By contrast, there are many models designed and trained for image face analysis, and we can fine-tune such a model to apply to pain data frames. We propose an efficient three-stage model to estimate pain in video. In the first stage, we use deep neural networks pre-trained on other face datasets to predict frame-level pain features such as Prkachin and Solomon Pain Intensity (PSPI) [PS08] scores directly from raw images. We then extract statistics from the output of the first stage, and send them into a neural network to get the sequence-level multidimensional pain scales. Further, we find an optimal linear combination of these pain scales to estimate VAS. We also propose to use multitask learning in each of the first two stages, and here show that both help improve the final VAS estimation. We show on the UNBC-McMaster Shoulder Pain dataset [LCP$^+$11a] that the proposed extended multitask-learning multidimensional-pain approach outperforms current state-of-the-art methods on pain intensity estimation in video.

3.1.1 Contributions

- We propose a three-stage multitask learning model that exploits multiple dimensions of pain to evaluate the current gold standard pain metric VAS in video from video frames directly
- We explore different frame-level features and discuss the relationship between frame-level and sequence-level pain ratings
- We analyze the contribution of different aspects of our model
- Our model beats the current state-of-the-art performance on the UNBC-McMaster dataset

3.1.2 Related Work

Two types of pain metrics are considered in pain studies [ALC$^+$09]. In facial video pain recognition, frame-level pain metrics are calculated from the intensity of objective facial action units (AUs), such as PSPI. Sequence-level pain metrics are rated by observers or subjects themselves.

Most research on automatic pain detection using facial expression has focused on objective frame-level pain metrics. Early studies have primarily involved two steps: extracting features from facial images, and then using machine learning models to predict pain levels. Ashraf et al. and Lucy et al. used Active Appearance Model (AAM)-based features and Support Vector Machine (SVM) to detect pain [ALC$^+$09, LCP$^+$11a]. Monwar et al. extracted location and shape features of the face and used a neural network to recognize pain expressions [MR06]. Rudovic proposed the heteroscedastic Conditional Ordinal Random Field to change the variance in the ordinal probit model to adapt to the pain expressiveness level specific to each subject [RPP13]. Recently, deep learning has been increasingly used to assess pain directly from raw pixels. Wang et al. fine-tuned a face verification network [WXL$^+$17]. Zamzmi et al. combine deep features from pre-trained VGGFace with traditional features for neonates' pain facial expression detection [ZGKS18]. Tavakolian et al. encoded CNN extracted features into a compact binary code using a deep network so that videos with same level of pain have smaller Hamming distance. [TH18].

There is also work considering spatiotemporal information when estimating pain in a single frame. Zhou et al. implemented a Recurrent Convolutional Neural Network (RCNN) model that took temporal information into feature extraction by adding recurrent connections within each convolutional layer [ZHSZ16]. Rodriguez et al. linked CNNs to a Long Short-Term Memory Networks (LSTM) model [RCG$^+$17]. Tavakolian and Hadid used 3D CNNs to capture a wide range of spatiotemporal variations of the faces [TH18]. Other work has attempted to detect peak pain intensity of the entire video using multiple-instance learning [SDB13, RRBP16].

None of the above methods estimate a sequence-level self-reported pain, but pain is a subjective experience and self-rating such as VAS is still the most commonly used pain score in clinical settings. Only a few works addressed the problem of estimating VAS score in facial videos. Sikka et al. [SAD$^+$15] and Xu et al. [XCD$^+$18] detected postoperative pain in children using AUs extracted by iMotions (imotions.com). Liu et al. proposed a two-stage method to first train a neural network model at the frame level using sequence-level VAS as labels and AAM landmarks as inputs, and then obtained video VAS score from frame-level predictions using a Gaussian process regression model [LPS$^+$17]. Martinez et al. used a bidirectional LSTM model to predict PSPI of each video frame using AAM landmarks and then applied personalized HCRFs (Hidden Conditional Random Fields) to predict VAS using the PSPI sequences [MRP$^+$17].

Our model can be decomposed to frame-level and sequence-level predictions in a similar way to the two stages in [LPS$^+$17, MRP$^+$17, XCD$^+$18] but our model takes raw images as inputs in stage 1, which involves the use of deep learning and transfer learning, and doesn't require AAM landmarks or AUs on test data which are obtained from expensive human annotation of key frames and automated landmark/AU detector and tracking algorithms. In [XdS20], we explored sequence-level models similar to our sequence-level stage 2 and 3 independently, using true AU and PSPI labels In this paper, we use a similar stage 2 and 3 but use AU and PSPI predictions from stage 1 as inputs.

3.2 Method

We developed our model based on the widely used UNBC-McMaster Shoulder Pain dataset [LCP$^+$11a]. It includes facial videos of participants suffering from shoulder pain while performing a series of active and passive range-of-motion tests to their affected and unaffected limbs on two separate occasions. The dataset has 25 subjects, 200 videos and 48,398 frames of size 320 x 240 pixels in total.

The dataset has two types of labels: frame-level labels and sequence-level labels. Frame-level labels include 66 AAM landmarks, 11 facial action unit (AU) intensities and 1 PSPI score. Both of the previous works predicting VAS using this dataset [MRP$^+$17, LPS$^+$17] used AAM landmarks as features but in this work we only used images as inputs. We also used AUs and PSPIs as outputs during training.

AUs are defined by FACS (Facial Action Coding System) [EF76] to code nearly all anatomically possible facial expressions. Figure 3.5(a) shows names of some AUs. In this work, we work with the 9 AUs (AU4, 6, 7, 10, 12, 20, 25, 26 and 43) present in more than 500 frames in the dataset. PSPI [PS08] is a pain evaluation metric computed from a specific set of AU intensities: PSPI = AU4 + max(AU6,AU7) + max(AU9,AU10) + AU43

AU intensities are integers ranging from 0-5 (weakest trace to maximum intensity possible), except for AU43 which can only take values from 0 and 1, so PSPI rating is also an integer and ranges from 0-16 (with larger values reflecting more pain) .

Sequence-level labels include the gold standard self-rating VAS pain score ranging from 0-10, as well as three other pain ratings: OPR (Observers Pain Rating - A value given by a human observer of the video) 0-5, AFF (Affective-motivational scale) 0-15 and SEN (Sensory Scale) 0-15. The properties of AFF and SEN are discussed in [GMD78, HGDM80]. The description for SEN/AFF scales is shown in Fig 3.1.

SENSORY WORDS

EXTREMELY WEAK
FAINT
VERY WEAK
WEAK
VERY MILD
MILD
SLIGHTLY MODERATE
MODERATE

BARELY STRONG
CLEAR-CUT
SLIGHTLY INTENSE
STRONG
INTENSE
VERY INTENSE
EXTREMELY INTENSE

(a) SEN Sensory intensity descriptors

UNPLEASANTNESS WORDS

SLIGHTLY UNPLEASANT
SLIGHTLY ANNOYING
ANNOYING
UNPLEASANT
SLIGHTLY DISTRESSING
SLIGHTLY MISERABLE
VERY ANNOYING
DISTRESSING

VERY UNPLEASANT
MISERABLE
VERY DISTRESSING
SLIGHTLY INTOLERABLE
VERY MISERABLE
INTOLERABLE
VERY INTOLERABLE

(b) AFF Affective-motivational descriptors

Figure 3.1. Word Descriptors for (a) SEN and (b) AFF

With the help of the labels described above, our goal is to train a model that predicts

VAS from image sequences directly. We chose the hyper-parameters of the neural networks based on training/validation learning curves and validation performance. The learning rates were selected using grid search at logarithmic intervals. The number of epochs and early stopping criterion were decided by observing the learning curves. The choice of optimizer doesn't affect the validation performance much so we chose it based on previous work [PVZ15, XCD+18] and experience.

3.2.1 Stage 1: PSPI Estimation in Facial Images

Our first stage predicts the frame-level PSPI score using RGB images. We built our model based on the VGGFace model [PVZ15]. The architecture was designed and pre-trained to classify 2622 individuals, and we simply replaced the last layer with our own linear fully-connected regression layer. During training, we updated all parameters in the neural network, but we used different initial learning rates (1e-4 for the last layer and 1e-5 for other layers). We used the Adam optimizer and a weight decay of 5e-4. We applied batch-weighted [SH19] Mean Squared Error (MSE) loss, where the weight of a sample in the loss is inversely proportional to the proportion of its label (which is PSPI score here) in the current batch, to overcome the class imbalance problem. The batch-weighted loss has been shown to help reduce overfitting compared to weighted loss [CJL+19] where weights are calculated for all data. We used a batch size of 32 and max epochs of 50 and early stopping when the validation loss hadn't decreased for 20 epochs.

For image preprocessing, we used the cascade DPM Face Detector [WHM11, MBPVG14] to detect the face and then extended the bounding box by a factor of 0.1 when cropping the face. We then resized the image to 224 × 224 and normalized each channel with the mean and standard deviation of the data the model was pre-trained on.

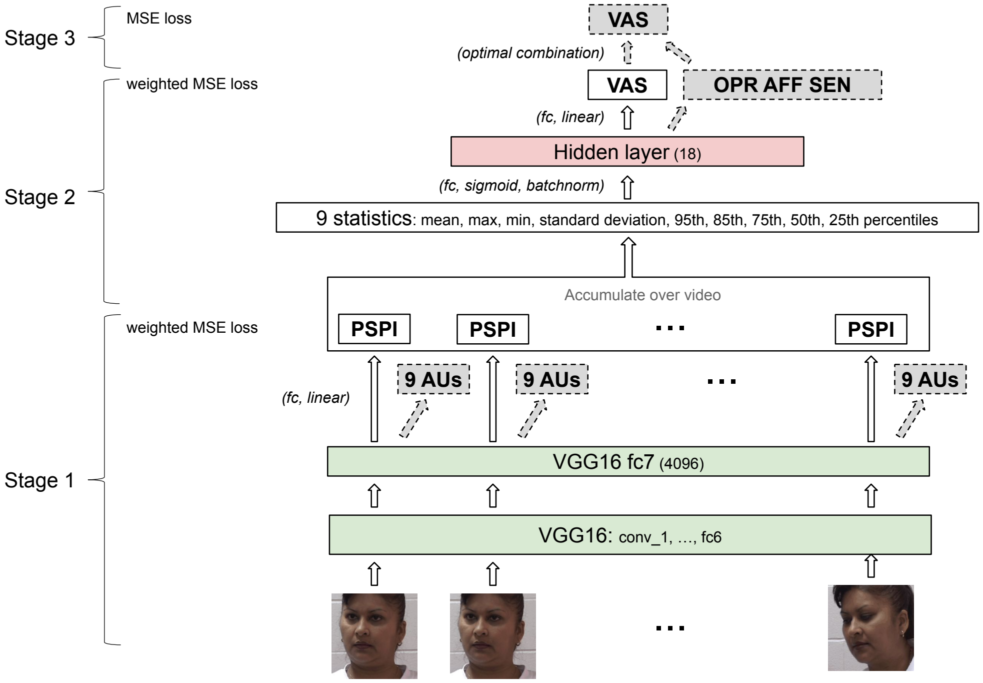


Figure 3.2. The proposed three-stage structure. The baseline model is represented by solid blocks, and shaded blocks with dashed outlines show added parts in multitask learning and ensemble learning with multidimensional pain scales.

3.2.2 Stage 2: VAS Estimation in Facial Videos using Sequence of Predictions

After we obtained PSPI predictions of all frames in stage 1, we extracted 9 statistics (mean, max, min, standard deviation, 95th, 85th, 75th, 50th, 25th percentiles) over all frames of a video to form a video feature vector. We then sent it to a fully connected neural network with 1 hidden layer with twice the number of units as the input layer to predict VAS in a linear output layer using batch-weighted MSE loss similar to stage 1. We used Adam and started with a learning rate of 1e-2. We set the batch size to 32, max number of epochs to 200, and used early stopping when the validation loss hadn't decreased for 20 epochs.

Combining stage 1 and 2 we obtained our baseline model which predicts the VAS score from video. This is illustrated in Fig 3.2. Stage 1 and 2 were trained separately because the memory capacity of our GPU doesn't allow end-to-end training using video data.

3.2.3 Multitask Learning

The UNBC-McMaster Shoulder Pain dataset contains other pain metrics besides VAS and PSPI at both the frame and sequence level. At the frame level, it provides several manually coded FACS AUs. At the sequence level, three other pain ratings are available. We reasoned that a multitask network [Car97] learning these metrics with the same hidden layer/representation learning to predict PSPI and VAS may better learn PSPI and VAS.

For example, in stage 1, PSPI is a non-linear combination (due to the max operation) of 6 AUs. The same PSPI could be due to many different combinations of AUs and underlying facial expressions. Thus there is a noisy many to one mapping. Learning individual AU activations is a simpler mapping, and a network that performs well on the underlying AU representations should be able to compute PSPI.

Similarly, in the second stage, OPR, AFF, and SEN are very related to the VAS pain score. OPR in particular is more simply related to the video than VAS is. In particular, OPR should be a possibly noisy function of the video features whereas VAS may be not fully constrained by the video; if the person is particularly expressive or stoic, their VAS score may be more or less related to the video features.

Our proposed multitask architecture is illustrated in Fig 3.2. In stage 1, instead of having only one output estimating PSPI, we concatenated several AU values and the PSPI score to form a multitask vector output. During training, we scaled the labels into the same range to make sure all elements contribute equally to the loss. AU labels are even more sparse than PSPI labels, so we only used 9 AUs (AU4, 6, 7, 10, 12, 20, 23, 26, 43) labeled in more than 500 frames out of the 48,398 frames in the dataset. For a similar reason, we weighted the loss function using PSPI score distribution and only looked at PSPI for validation loss for early stopping.

In stage 2, similarly, we used a 4-dimensional vector representing the four pain ratings instead of a single value representing VAS as output. The losses are weighted based on the distribution of VAS scores, and the validation loss is the mean MSE of the 4 outputs.

3.2.4 Stage 3: Ensemble Learning of Multidimensional Pain Measurement

On the UNBC-McMaster dataset, each of the 4 sequence-level scores can be seen as an evaluation of pain level, but focusing on different aspects of pain. For example, VAS reflects how much pain the patient perceives and relies on the patient's personal understanding of pain, whilst OPR is based on third-party observation of facial expressions, and will be influenced by how much "pain expression" the patient shows on his/her face and how good the observer is at reading facial expressions of pain. They also have different properties. For example, OPR may be more consistent across subjects when scored by the same observer. As OPR entirely depends on facial video it should be more easily learned from facial video than VAS in the same way that AUs should be more learnable from video than any non-linear function of them. At the same time, OPR may be limited as a measure of actual pain as it is only able to reflect pain revealed by facial expressions and will be biased if the subject hides it. But any computer vision system will face the same limitations unless it incorporate features from other sensors [XSN$^+$18].

OPR, AFF, and SEN are all highly correlated with VAS and can be considered as predictions of VAS. In fact, after scaling the outputs to the same range as VAS, i.e. multiplying the outputs corresponding to VAS, OPR, AFF, and SEN by 1, 10/5, 10/15, 10/15 respectively, all 4 outputs do a reasonable job at estimating VAS. In other words, we now have 4 "experts" each with its own prediction of pain level.

Ensemble averaging can be used in this case in the hope of reducing variance at no cost to bias [Has97]. This corresponds to the last layer in Fig 3.2. The optimal linear combination of experts to form a least mean squared error estimation of the target score was discussed in [Has97]. Below we briefly discuss the derivation of our ensemble model weights.

Consider each data point (x,y) as an observation of random variables (X,Y) from an unknown multivariate distribution over $\mathbb{R}^9 \times \mathbb{R}$. And $f_i : \mathbb{R}^9 \to \mathbb{R}$ $(i = 1,2,3,4)$ maps stage 2 inputs to a real number, each corresponding to one of the 4 scores.

We learn the final prediction of VAS as a weighted sum of the four experts f_i. The overall model $\tilde{f}$ can be defined as:

$$\tilde{f}(x) = \sum_{i=1}^{4} \alpha_i f_i(x) \tag{3.1}$$

where we apply the constraint $\sum_{i=1}^{4} \alpha_i = 1$ (and $\alpha_0 = 0$) as suggested by [Cle86, TL86, Has97].

The MSE loss of the final model is:

$$\begin{aligned} \text{MSE}(\tilde{f}(X)) &= E[(\tilde{f}(X) - Y)^2] && (3.2)\\ &= E[(\sum_{i=1}^{4} \alpha_i f_i(X) - Y)^2] && (3.3)\\ &= E[(\sum_{i=1}^{4} \alpha_i (f_i(X) - Y))^2] && (3.4) \end{aligned}$$

Our goal is to minimize the MSE subject to $\sum_{i=1}^{4} \alpha_i = 1$. The Lagrangian expression of this problem is:

$$L(\mathbf{X}, \lambda) = \text{MSE}(\tilde{f}(\mathbf{X})) - \lambda(\sum_{i=1}^{4} \alpha_i - 1) \tag{3.5}$$

where λ is the Lagrange multiplier.

First, we compute the partial derivative of Eq (3.5) with respect to α_k for $k = 1, 2, 3, 4$:

$$\frac{\partial L(\mathbf{X}, \lambda)}{\partial \alpha_k} = E[2\sum_{i=1}^{4} \alpha_i (f_i(\mathbf{X}) - Y)(f_k(\mathbf{X}) - Y)] - \lambda \tag{3.6}$$

Then set the gradients to 0:

$$\sum_{i=1}^{4} \alpha_i E[(f_i(\mathbf{X}) - Y)(f_k(\mathbf{X}) - Y)] = \frac{\lambda}{2} \text{ for } k = 1, 2, 3, 4 \tag{3.7}$$

Let $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$, $\Omega = [\omega_{ij}] = [E[(f_i(X) - Y)(f_j(X) - Y)]]$, the equation above becomes:

$$\Omega\alpha = \frac{\lambda}{2}1 \tag{3.8}$$

This together with the constraint $\sum_{i=1}^{4} \alpha_i = 1$ gives us the optimal weight vector α as:

$$\alpha = \frac{\Omega^{-1}1}{1^T\Omega^{-1}1} \tag{3.9}$$

3.3 Experimental Analysis

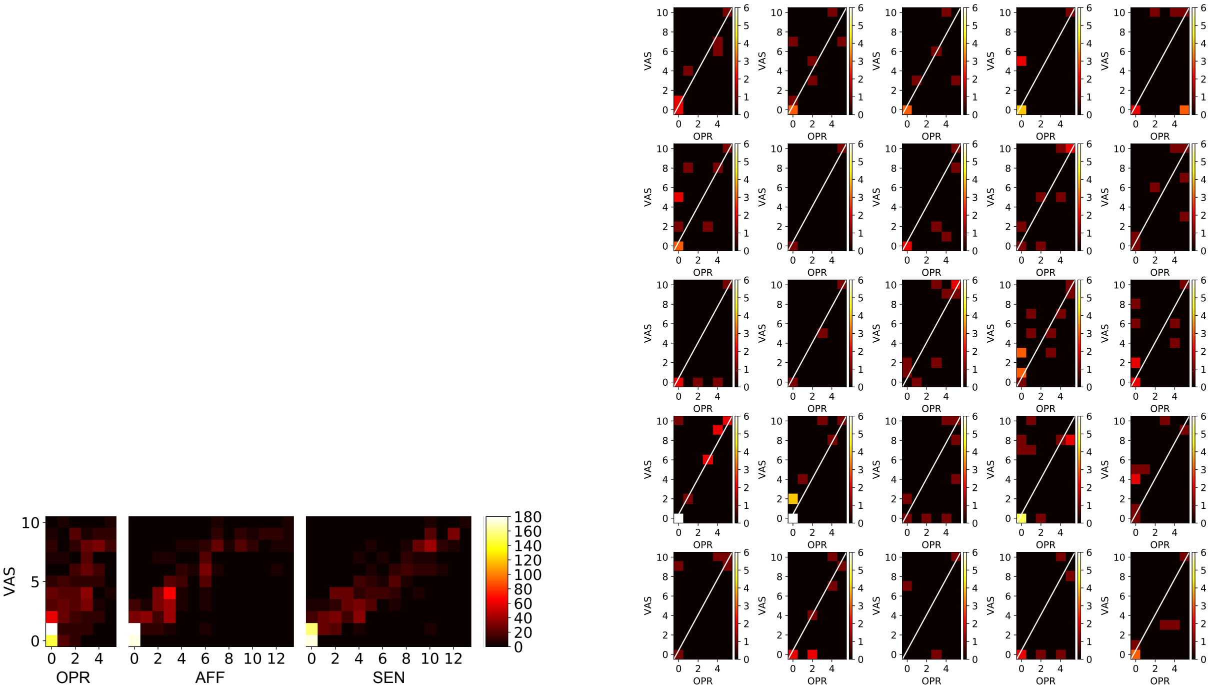
On the UNBC-McMaster dataset, we performed 5-fold cross validation with each fold consisting of 5 subjects. We used the same training/test splits for the three stages in each iteration. One of the 4 training folds is randomly selected as the validation set during neural network training. After 5 iterations, we concatenated all the test samples and calculated the Mean Absolute Error (MAE), Mean Squared Error (MSE), Intraclass Correlation Coefficient (ICC) and Pearson Correlation Coefficient (PCC). ICC is useful when MAE scores are deceptively low. For example, for the current dataset, if the model outputs the average VAS for all samples, the MAE will be 2.44, but the ICC will be approximately zero. So we want a model with low MAE and high ICC.

For all models in this paper, we performed the 5-fold cross validation 5 times, and report mean and standard deviation of MAE, MSE, ICC and PCC over 5 runs of the 5-fold cross validation. To ensure reproducibility, we used the same set of random seeds to make sure all models are trained and tested on the same data and have the same initial states. We run all our experiments on a single GPU (NVIDIA GeForce RTX 2080); it takes about 4 hours to train a three-stage model using 4 folds of the UNBC-Mcmaster data.

3.3.1 Relationship between Frame- and Sequence-level Metrics in the Data

Relationship between Sequence-level Metrics

The correlation between the 4 sequence-level scores in the UNBC-McMaster dataset is shown in Fig. 3.4 top left block. We can see from the heatmap that VAS, AFF and SEN are highly correlated, and OPR is also correlated with these 3 self-rated scores but not as much.



(a) VAS vs OPR/AFF/SEN **(b)** VAS vs OPR distribution for each subject

Figure 3.3. 2D histogram of distribution of sequence-level score pairs. Mass above the diagonal represents videos where the observer (OPR) underestimated the patient's VAS.

Figure 3.3(a) shows the joint distributions of VAS with OPR, AFF and SEN plotted as 2D histograms. It can be seen that although VAS is linearly correlated with the three other scores, they are not strictly proportional.

Relationship between Sequence-level and Frame-level Metrics

Fig. 3.4 shows the correlation between the frame-level and sequence-level pain scores. We see again the high correlations between the sequence-level measures and some correlation

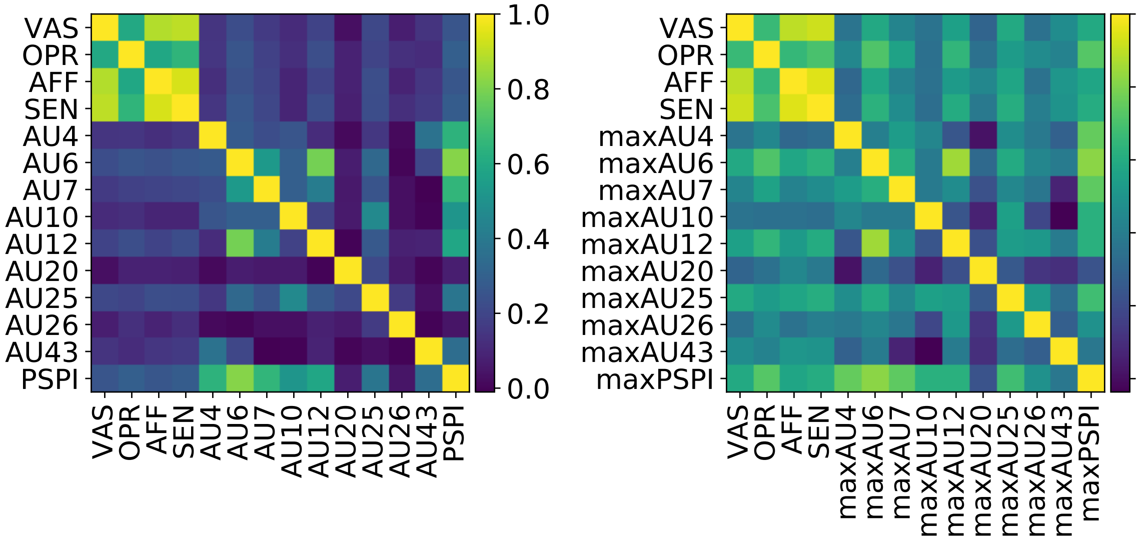


Figure 3.4. The correlation between 4 sequence-level scores (VAS, OPR, AFF, SEN) and 10 frame-level scores (9 AUs and PSPI) in the data. On the left is the correlation at the frame level, where the VAS for a frame is the VAS of the video it belongs to. On the right is the correlation at the sequence level, where the maximum AU/PSPI for a video is taken.

between the frame-level measures. Of the sequence measures, OPR generally has a higher correlation with the AUs and PSPI. This shows the potential of predicting sequence-level pain ratings from frame-level measurements.

Sequence-level Metric Estimation using Frame-level Metrics

In [XdS20], we analyzed the last two stages of our model using true PSPI and AUs to predict VAS and showed that such a model can achieve an MAE of 1.73 and ICC of 0.61, outperforming the human labeler (MAE=1.76, calculated using OPR provided). In this paper the last two stages use predicted PSPI and AUs output by our stage 1, thus will likely result in worse performance, but this current approach does not need manual labeled AUs on test data and can be used on videos directly making it much more practical.

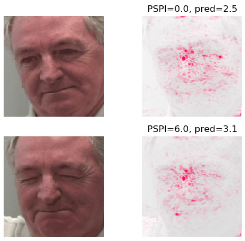
3.3.2 Stage 1: PSPI Estimation using Multitask Learning

The performance of our stage 1 PSPI prediction model is shown in Table 3.1. In order to better understand the importance of different components, we did ablation analyses to explore the importance of components. We found that improving PSPI estimation in stage 1 doesn't necessarily improve VAS estimation in stage 2 or 3. So for the well-performing models in stage 1, we continued to look at the final performance on VAS directly in Table 3.2 to decide which one is better.

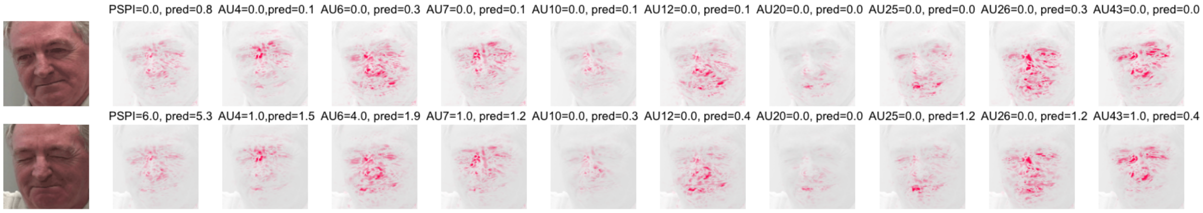
Multitask Learning in S1. Comparing rows 1 and 2 in Table 3.1 shows that multitask learning of AUs helps the model to better predict PSPI. In Table 3.2, comparing "PSPI+AU MTL" in the "Stage 1" column to "PSPI" rows shows that multitask learning in Stage 1 also helps the model better predict VAS.

AU4	brow lowering	AU12	oblique lip raising
AU6	cheek raising	AU20	horizontal lip stretch
AU7	eyelid tightening	AU25	lips parting
AU9	Nose wrinkling	AU26	jaw dropping
AU10	upper lip raising	AU43	eye closure

(a) AU Description. PSPI=AU4+max(AU6,AU7)+max(AU9,AU10)+AU43)



(b) Output PSPI



(c) Output PSPI and 9 AUs

Figure 3.5. Contributions of pixels for two frames are explained in the figures above. (b) explains the baseline stage 1 VGG model predicting only PSPI, and (c) explains the multitask learning VGG model predicting PSPI and 9 AUs. The first frame has a PSPI score of 0 and corresponds to the first row in each of the two figures. The second frame has a PSPI score of 6 and corresponds to the second row. The first column in both (b) and (c) is the input image, and other columns correspond to model outputs. Larger absolute SHAP value (corresponding to darker pixels) means larger contribution of a pixel to the corresponding output. For example, in (c), when predicting AU4, the model focuses on the area around eyes and eyebrows, especially the inner portion of the eyebrows and the area between them, which is consistent with the description of AU4.

To better understand this aspect, we plot the contributions of image pixels to the outputs using SHAP introduced by [LL17] in Fig 3.5. SHAP is a framework that interprets complex models by assigning each feature an importance value for a particular prediction. Pixels with larger absolute values of SHAP (darker red on the image) reflect a greater influence on the output. In Fig 3.5, the 1st column in (c) shows the PSPI output of the MTL model has captured more meaningful pixels on the face compared to the baseline model (Fig 3.5 (b) 2nd column). E.g. we can see clearer outlines of the eyebrows, the eyes, the nose and the mouth. Fig 3.5 (c) shows that many of these areas are relevant for the prediction of several AUs, such as eyebrows in AU4, eye area in AU7 and AU43, corners of the lip and nasolabial furrows in AU12, mouth in AU25, etc. Note this is true even though some of the AUs are not well learned because of a lack of training data (AU10 and AU20 both are present in less than 1000 frames).

We also tried ensemble learning in stage 1, where we viewed the PSPI prediction as one expert and a PSPI score calculated from AU predictions as another expert, and used the same method in section 3.2.4 to obtain an ensembled PSPI prediction. The performance was not improved by doing this, possibly because PSPI and AUs are not as “complementary” in stage 1 as the 4 scores in stage 2.

The ICC of the outputs of the multitask learning S1 model is shown in Fig 3.6. The model learns PSPI more accurately than most AUs, possibly because it has more samples with positive PSPI scores. Poor performance has been observer in AU4 and AU20 due to the lack of positive samples.

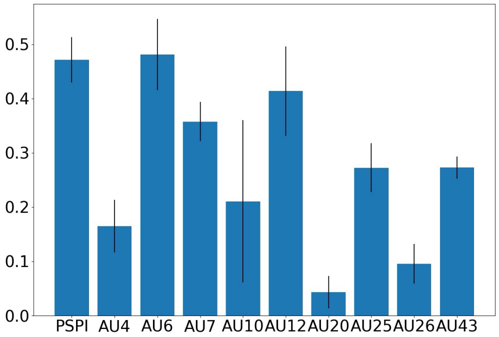


Figure 3.6. ICC of PSPI and AU predictions of the MTL S1 model

Benefit of Transfer Learning. We observe the benefit of transfer learning in stage 1 in row 2-5 in Table 3.1. Pretraining on face recognition (VGGFace) outperforms pretraining on general object recognition (Imagenet), and training from scratch simply fails. Using VGG16 pretrained on ImageNet instead of VGGFace in the final model is also shown to increase the MAE in Table 3.2 “ImageNet PSPI+AU MTL” under “Stage 1 Model”.

In our stage 1, we take a pretrained VGGFace model and replace the last layer with a regression layer, and learn this new last layer while fine-tuning the other layers, but the trained VGG16 model can also work as a feature extractor without fine-tuning, e.g. in [ZGKS18]. This will save a lot of training time, but it doesn’t work as well for our model, as shown in “No finetuning” under “Stage 1 Model” in Table 3.2

We also compared our method fine-tuning the VGGFace model to extract AU/PSPI scores with using the commercial software iMotions (imotions.com) to detect the face and automatically estimate AU intensities for each raw frame. AUs and computed PSPI from iMotions are not on the same scale as AUs we use here, so they are not directly comparable, but iMotions AUs can be used as input to train our stage 2 and 3 to predict VAS. The result is shown in row “iMotions PSPI+AU” under “Stage 1 Model” in Table 3.2. AUs from iMotions are not as good as AUs from our transferred S1 model, but perform better than our transferred S1 model pretrained on ImageNet (“ImageNet PSPI+AU MTL” under “Stage 1 Model” in Table 3.2).

Face Pre-processing. For comparison purposes, we also trained our model on aligned and warped faces. We used the provided 66 AAM facial landmarks to pre-process the face images following procedures in [ALC+09, LCP+11a]. As described in [ALC+09, LCP+11a], the shape s of an AAM is described by a 2D triangulated mesh defined by coordinates $s = [x_0, y_0, x_1, y_1, ..., x_n, y_n]$ where n is the number of vertices. The shape s can be expressed as a base shape s_0 plus a linear combination of m shape vectors s_i:

$$s = s_0 + \sum_{i=1}^{m} p_i s_i \tag{3.10}$$

where the coefficients $p = (p_1, \ldots, p_m)^T$ are the shape parameters. These shape parameters can typically be divided into rigid similarity parameters p_s and non-rigid object deformation parameters p_o, such that $p^T = [p_s^T, p_0^T]$. Following [ALC$^+$09, LCP$^+$11a, RCG$^+$17], we use Generalized Procustes Analysis to align the faces. This removes all rigid geometric variation p_s in Eq (3.10) by translation, scale, and rotation. We also applied a piece-wise affine warping to each triangle in the mesh to warp/frontalize the faces and then masked them following [RCG$^+$17]. This step removes all variation p in Eq (3.10).

The results are shown in the last two rows in Table 3.1. Alignment and warping reduce the MAE in PSPI prediction. However, final VAS prediction was not improved significantly (Table 3.2 last two rows). One explanation can be that stage 1 model has been pre-trained with faces without alignment or warping, so it can deal with varying scales, rotations and positions. Another reason is that shape information is very important in pain detection and face warping removes these cues and keeps only texture information. Similar conclusions have also been made in [ALC$^+$09] where S-APP (similarity normalized appearance representation) of AAM features which applies the same transformation as our warping yields the worst performance. The fact that alignment and warping is more helpful for the frame-level model than the sequence-level model also indicates that the movement of facial points may provide useful temporal information in identifying pain expression, so alignment/warping across images may help pain detection in still images, but not in videos.

We have also compared warping using all 66 facial landmarks to results using a reduced set with 37 landmarks in order to keep useful information as in [RCG$^+$17] and didn't see much difference.

The process of alignment and warping involves AAM landmarks provided by the dataset, which were obtained by hand-labeling and pre-processing of the data. Requiring AAM landmarks on new test datasets would make this method very expensive.

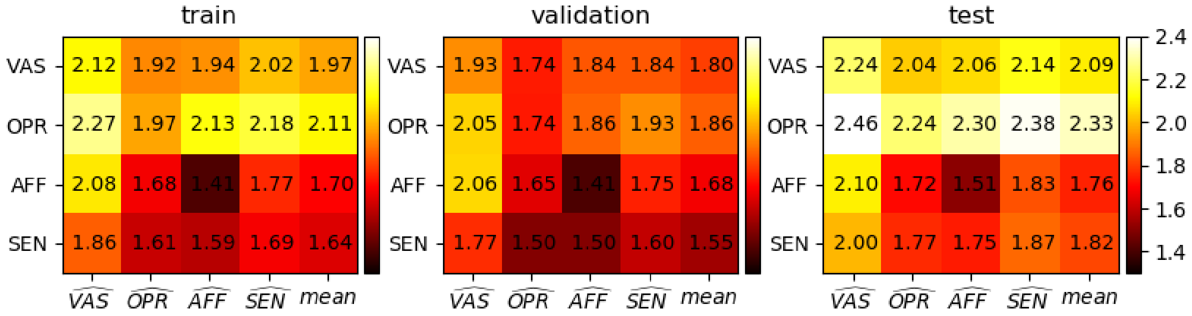


Figure 3.7. Average MAE matrices on training, validation and test data. y axis is the true label, and x axis is the prediction (or the mean of the 4 predictions). Each entry is the mean absolute difference between the two variables.

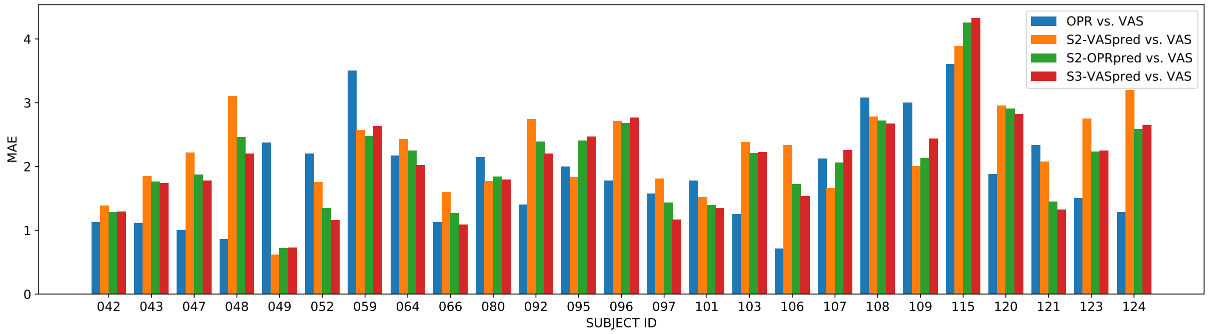


Figure 3.8. Bar graphs (ordered by subject id) showing the per-subject MAE of different nodes predicting VAS

3.3.3 Stage 2 and 3: VAS Estimation using Extended Multitask Learning of Multidimensional Pain Scales

Using the PSPI (and AU) estimations from stage 1, we trained a neural network to predict VAS. Ablation analyses in this section follow and are compared to [XdS20].

S2 Performance. We first observed the performance of each of the 4 outputs from stage 2, shown in Fig 3.7, after re-scaling each variable to 0-10. Interestingly, the best approximation of a metric is not always given by its corresponding output. OPR output does a better job in estimating OPR than other outputs. The same is true for AFF. However, OPR output gives a better estimate of VAS than the VAS output. This is possibly because OPR is also based on facial videos, whereas VAS involves other factors that may not be determinable from video frames, and OPR is also more consistent across subjects. So as a result, OPR is an easier measure to be

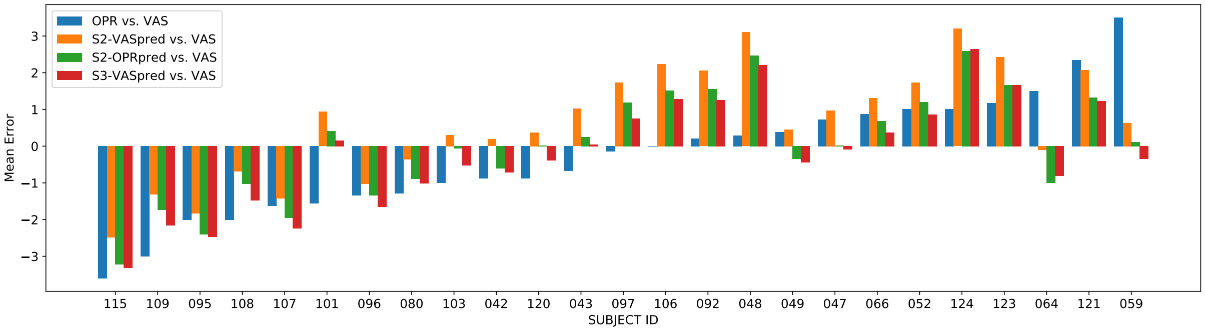
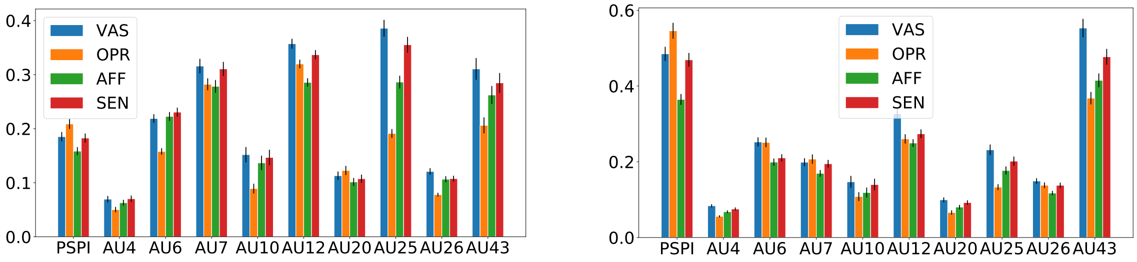


Figure 3.9. Bar graphs (ordered by OPR-VAS) showing the per-subject Mean Error of different nodes predicting VAS

estimated by a computer vision model across subjects. At this point, in order to achieve the best performance on VAS estimation, the OPR output should be used instead of the VAS output.

Following [XdS20], other than using 9 statistics of PSPI predictions as input, we also tried using a length-10 vector concatenating the maximum of PSPI and 9 AU predictions as input to S2, corresponding to “PSPI+AU max” rows under “Stage 2 Input” in Table 3.2. Unlike in [XdS20] where the model was trained with hand-labeled AUs, statistics of PSPI perform better than maximums of PSPI and AUs. This is possibly because our S1 model doesn’t predict AUs perfectly. As a result, while true AUs work better as features for S2, predicted AUs are worse than predicted PSPI statistics.

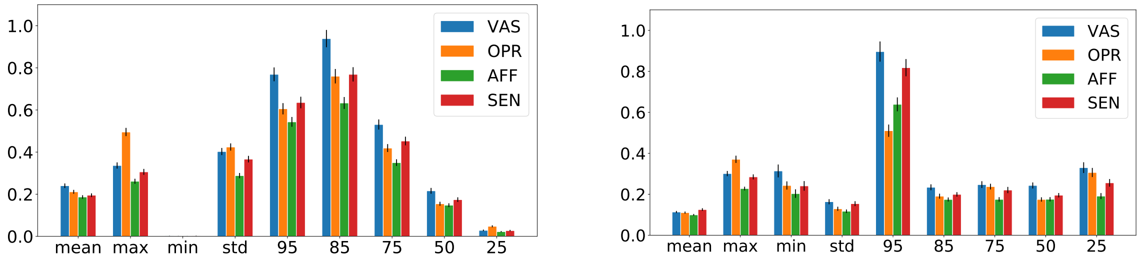


(a) Max of true PSPI and AUs as input

(b) Max PSPI and AU predictions as input

Figure 3.10. Contributions of each of the maximum measurements to the S2 model outputs. The heights of the bars represent feature importance measured as the mean absolute shap values. Error bars show the standard deviation of the mean absolute shap values.

Input Importance in S2. We plot the contribution of input features for the S2 model predicting sequence-level scores using frame-level predictions from S1 in Fig 3.10(b) and 3.11(b),



(a) Statistics of true PSPI as input

(b) Statistics of PSPI predictions as input

Figure 3.11. Contributions of each of the statistics to the S2 model outputs. The heights of the bars represent feature importance measured as the mean absolute shap values. Error bars show the standard deviation of the mean absolute shap values.

and compare them to versions using true AU and PSPI as inputs (as studied in [XdS20]. We plot them in (a) in both figures). When using the max of PSPI and AU predictions as input, unlike Fig 3.10(a), PSPI is the most important. This is because our S1 model learns to predict PSPI much better than AUs as there are more positive samples. So although AUs are useful to multi-task train PSPI, the AU predictions are not good enough to be used in subsequent stages. This explains why in Table 3.2, different from [XdS20] where using the max of PSPI and AU yields the best performance, when PSPI and AUs are learned from S1, using PSPI statistics as input leads to better performance than using AU maximums.

For both Fig 3.10(a) and (b), PSPI contributes more to OPR than self-rated pain scores, which makes sense because PSPI is designed based on human observer's understanding of pain. In Fig 3.4 OPR is also the most correlated sequence-level score with PSPI.

When using statistics of PSPI as input, Fig 3.11 shows that when the model uses the predicted PSPI values instead of the true ones, more statistics are considered important, with the 95th percentile feature (a form of trimmed or robust max) the most important.

Benefit of S3. Since the four S2 outputs are all performing well in VAS prediction, we can regard them as outcomes of different experts trying to predict VAS, and learn an ensemble model on top of them. This third stage of our model has been discussed in Section 3.2.4 and experimental results are shown in row 7-9 in Table 3.2. The optimal weights were found on training and validation data, and the ensemble outperforms each of the 4 outputs on the test data.

Fig 3.8 plots the MAE for each of the 25 subjects. For a single subject, S3 VAS prediction may not beat S2 VAS output or S2 OPR output when used to estimate VAS, but in general the combined S3 score outperforms one of the S2 outputs, and for some of the subjects outperforms both. The ensemble learning model in stage 3 works well in deciding how to best weight the S2 outputs.

Fig 3.9 plots the Mean Error for each subject. In Fig 3.3 (b) we can see for most of the subjects, the human observer tends to either overestimate or underestimate pain for the same patient, so the heights of the OPR vs. VAS bars are meaningful. However, the third bar is always lower than the second bar, and this is because our model is trained on all subjects and the human overall underestimates pain, so that S2 learns to output smaller estimations for OPR than VAS.

It's also interesting that in Fig 3.9 the last bar is almost always lower (closer to negative infinity) than both the second and the third bar. This is because S1 and S2 of our method have been optimized for weighted MSE loss. This is a common approach to handle imbalanced data. [WXL$^+$17] has suggested to use weighted MSE/MAE as evaluation metrics for this imbalanced dataset, but most work still report unweighted metrics which reflect the errors on the true distribution of the data. However, the weighted MSE trained model is thus not optimal for the evaluation metrics (unweighted MSE/MAE) on the current data distribution. In other words, the model is not unbiased because it is trained with weighted MSE loss, not MSE loss. We found that simply adding a constant to the output of S2 to make the VAS prediction unbiased will reduce the MAE of S2 VAS prediction from 2.20 to 2.01 and MSE from 6.53 to 6.08. The same trick doesn't work as well for the S3 output, which will only reduce the MAE of S3 by at most 0.01. This indicates that S3, optimized for MSE, can "calibrate" the bias in S2 outputs, possibly through assigning higher weights to smaller S2 predictions. This explains why in Fig 3.9 S3 VAS prediction is almost always lower than both OPR and VAS output by S2.

Multitask Learning in S2. Our S2 model again uses multitask learning and the 4 pain scales share the same hidden layer. In Table 3.2, we show that multitask learning using multidimensional pain (row 1-3 vs 4-6) in stage 2 improves VAS prediction.

We analyze variations of the extended multitask learning model similar to [XdS20] in Fig 3.12. None are significantly better. “4 scores MTL” corresponds to body row 8 in Table 3.2, which is the final model.

In order to show that multitask learning in stage 2 is also helpful for stage 3, we trained 4 separate networks for the 4 scores with no shared parameters. We combined these scores using the same method for learning an ensemble model and obtained a final prediction of VAS. The performance (body row 10 in Table 3.2 and “4 scores” bar in Fig 3.12) is slightly worse than using multitask learning at this stage.

In order to compare the importance of ensemble learning to that of multitask learning, we trained a model with the same structure as our best model, i.e. with 4 neural network outputs and ensemble learning on top of them, but instead of using 4 different pain scores as labels for S2 outputs, we trained each of the 4 outputs with identical VAS labels (but different initial conditions). The corresponding bars “VAS$\times$4 MTL” shows that an ensemble of 4 VAS is not as good as our final model (ensemble of multidimensional pain), but is much better than learning only 1 VAS, so the ensemble learning method contributes a lot to the final performance.

This observation lead us to train our S2 model to learn several copies of 4 scores (or VAS) and learn stage 3 on top of them so that S3 has more variant input “experts” to ensemble. We tried 4 copies of 4 scores and 16 copies of VAS and show results in “4 scores $\times$ 4 MTL” and “VAS $\times$ 16 MTL” in Fig 3.12, and “4 scores $\times$ 4 MTL” under “Stage 2 Output” in Table 3.2). While increasing the risk of overfitting, increasing the number of outputs of S2 by learning multiple versions for the same variables doesn’t result in significant improvement in final VAS estimation.

Lastly, it should also be noted that learning the weights in stage 2 and 3 together through back propagation didn’t give as much improvement as in the “extended multitask learning” where we learned to predict multiple pain dimensions first and combine them afterward.

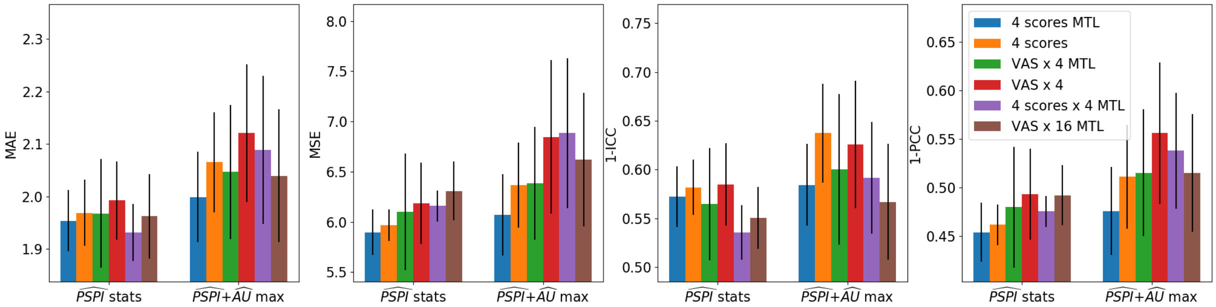


Figure 3.12. Bar graphs showing the S3 VAS output MAE, MSE, 1-ICC, 1-PCC of the following different S2 models using 2 different combinations of input (stats of PSPI prediction, max of PSPI and AU prediction) of frame-level predictions: (1) 4 scores MTL: (Our final model) Predicting 4 scores using multitask learning. (2) 4 scores: Predicting 4 scores using 4 separate models. (3) VAS × 4 MTL: Predicting 4 VAS using multitask learning. (4) VAS × 4: Predicting 4 VAS using 4 separate models. (5) 4 scores × 4 MTL: Predicting 4 copies of 4 scores using multitask learning. (6) VAS × 16 MTL: Predicting 16 copies of VAS using multitask learning .

3.3.4 Comparison with Other Work

We compare to results from previous work estimating VAS using the UNBC-McMaster dataset or a child pain dataset in Table 3.3. The child pain dataset contains facial video from children aged 10 to 15 who had undergone medically necessary laparoscopic appendectomy. Details of this dataset can be found in [XCD$^+$18]. Without retraining, we tested the model trained on the UNBC-McMaster dataset with 134 videos of 70 subjects from the child pain dataset. Our 95% Confidence Interval of MAE on the shoulder pain dataset is 1.95 ± 0.0526 and that of ICC is 0.43 ± 0.0175. If we assume equal variability for the previous state-of-the-art [LPS$^+$17] (not provided in the paper), our MAE is significantly lower ($p = 0.0002$). In [XHdS19] we also applied our model with no extra fine-tuning to a dataset of children post surgery and outperformed our earlier performance with a model based on iMotions [XCD$^+$18].

3.3.5 Comparison with Human

Since OPR is obtained from human observers estimating the subject's pain level, we can see OPR as human's estimation of VAS, and compare it with our model. The results are shown in Table 3.4 in row 1-2. The human predicts VAS better than our model.

From the per-subject MAE plot Fig 3.8, we can see that human is not always better than the model. For almost half of the subjects, the model performs better than the human.

The model and human also don't seem to make the same mistakes, i.e. the model may perform pretty well on a subject the human fails at. The correlation between the error (difference between VAS and the estimation) of human estimation of VAS and the error of model estimation of VAS over videos is 0.5, which also shows that our model is not very highly correlated with OPR.

This indicates that our model learns additional information than human observers. Based on this observation, we take an average of OPR and the model output, and show the results in row 3 Table 1.3. The average outperforms the human. This shows the potential of our machine learning system for a clinical settings. When a human's estimation is available, a more accurate estimation of the VAS score of a patient can be obtained by simply averaging our model's prediction and the human's rating. When a human observer is not present, the model can serve as a cheap, consistent monitoring system that provides live feedback that is almost as accurate as human observers.

3.3.6 Comparison with Model using True AU/PSPI labels

The last two rows in Table 3.4 shows the performance of a model with the same structure as our last two stages but using true AU and PSPI as input. Details of this model can be found in [XdS20]. Because our S1 is not making perfect predictions, our final results (and its average with human) are not as good as [XdS20]. But the large difference between row 1 and row 4 (as well as row 3 and row 5) in Table 1.3 indicates that there is a lot of space for improvement of our S1.

3.4 Conclusions and Future Directions

We propose a three-stage model to predict VAS in facial videos directly, and propose a method using multitask learning, multidimensional pain measurement and ensemble learning to effectively improve the performance of the model. Our approach achieves state-of-the-art performance on the UNBC-McMaster Shoulder Pain dataset. Our MAE (1.95) is not as good as human observers (1.76), but our model is cheaper and more consistent than humans. In the UNBC-McMaster dataset, simply averaging our prediction and the human prediction reduces the MAE to 1.58.

Our model can be broken into two parts similar to other work on video analysis, where the first part (stage 1 of our model) focuses on frame-level feature extraction and the second part (stage 2 and 3 of our model) uses the previous stage outputs to learn sequence-level targets. We can improve the model at each of the parts. Our model is not as good as [XdS20] (MAE=1.73) which uses true labels instead of predictions of AUs, indicating that if our stage 1 can achieve better prediction of AUs, our final VAS prediction will also be improved. Our future work will include exploring different deep models, e.g. a model that is designed and trained to recognize AUs, to improve AU and PSPI estimation and further improve VAS prediction.

For the second part, the sequence-level model, we have studied the difference and relationship between multidimensional pain scores, and designed an extended multitask learning framework to take best advantage of them. However, the difference and relationship between patients has not yet been explored. Since VAS is subjective and facial expressions of pain may also be different for different people, we will learn different models to predict pain levels for different patients, and an optimal ensemble of these models with a goal of generalizing better to new subjects. Moreover, we can also try to learn similarities between subjects in terms of their facial expression to pain level mapping. Then given a new video of a new subject, we can use the model(s) of the most similar subject(s) to determine the pain score.

Acknowledgments

Chapter 3, in part is currently being prepared for submission for publication of the material. Xu, Xiaojing, Jeannie S. Huang, and Virginia R. de Sa. The dissertation author was the

primary investigator and author of this material.

Table 3.1. Frame-level PSPI Prediction

Stage 1 Input	Stage 1 Model	Stage 1 Output	MAE	MSE	ICC	PCC
Cropped	**VGGFace**	**PSPI+AU MTL**	0.80 ± 0.07	1.53 ± 0.14	0.47 ± 0.04	0.49 ± 0.04
Cropped	VGGFace	PSPI	0.91 ± 0.19	1.82 ± 0.37	0.51 ± 0.04	0.52 ± 0.04
Cropped	Imagenet	PSPI+AU MTL	0.88 ± 0.04	2.25 ± 0.27	0.40 ± 0.02	0.41 ± 0.03
Cropped	No pretraining	PSPI+AU MTL	2.39 ± 0.08	6.30 ± 0.37	-0.01 ± 0.01	-0.02 ± 0.03
Cropped	No finetuning	PSPI+AU MTL	1.78 ± 0.07	4.87 ± 0.42	0.16 ± 0.04	0.16 ± 0.04
Aligned	VGGFace	PSPI+AU MTL	0.74 ± 0.08	1.51 ± 0.19	0.44 ± 0.04	0.46 ± 0.05
Warped	VGGFace	PSPI+AU MTL	0.73 ± 0.05	1.44 ± 0.12	0.48 ± 0.04	0.50 ± 0.04

Table 3.2. Sequence-level VAS Prediction using Frame-level Predictions from Stage 1

Stage 1 Input	Stage 1 Model	Stage 2 Input	Stage 2 Output	Stage 3	MAE	MSE	ICC	PCC
Cropped	PSPI	PSPI	VAS	-	2.34 ± 0.09	7.27 ± 0.51	0.34 ± 0.04	0.50 ± 0.04
Cropped	PSPI+AU MTL	PSPI	VAS	-	2.23 ± 0.08	6.76 ± 0.37	0.37 ± 0.02	0.52 ± 0.03
Cropped	PSPI+AU MTL	PSPI+AU max	VAS	-	2.27 ± 0.08	7.08 ± 0.56	0.34 ± 0.07	0.47 ± 0.09
Cropped	PSPI	PSPI	4 scores MTL	-	2.30 ± 0.06	7.06 ± 0.31	0.37 ± 0.04	0.52 ± 0.02
Cropped	PSPI+AU MTL	PSPI	4 scores MTL	-	2.20 ± 0.06	6.53 ± 0.30	0.37 ± 0.03	0.54 ± 0.02
Cropped	PSPI+AU MTL	PSPI+AU max	4 scores MTL	-	2.20 ± 0.07	6.73 ± 0.45	0.37 ± 0.05	0.51 ± 0.06
Cropped	PSPI	PSPI	4 scores MTL	Ensemble	1.94 ± 0.05	5.76 ± 0.23	0.45 ± 0.03	0.56 ± 0.03
Cropped	PSPI+AU MTL	PSPI	4 scores MTL	Ensemble	1.95 ± 0.06	5.90 ± 0.22	0.43 ± 0.02	0.55 ± 0.03
Cropped	PSPI+AU MTL	PSPI+AU max	4 scores MTL	Ensemble	2.00 ± 0.09	6.07 ± 0.41	0.42 ± 0.04	0.52 ± 0.05
Cropped	PSPI+AU MTL	PSPI	4 scores	Ensemble	1.97 ± 0.04	6.10 ± 0.26	0.40 ± 0.03	0.52 ± 0.03
Cropped	PSPI	PSPI	4 scores ×4 MTL	Ensemble	1.98 ± 0.09	6.48 ± 0.84	0.46 ± 0.03	0.51 ± 0.05
Cropped	PSPI+AU MTL	PSPI	4 scores ×4 MTL	Ensemble	1.93 ± 0.05	6.16 ± 0.16	0.46 ± 0.03	0.52 ± 0.02
Cropped	iMotions PSPI+AU	PSPI	4 scores MTL	Ensemble	2.05 ± 0.05	6.22 ± 0.24	0.43 ± 0.02	0.51 ± 0.02
Cropped	iMotions PSPI+AU	PSPI+AU max	4 scores MTL	Ensemble	2.06 ± 0.07	6.29 ± 0.39	0.43 ± 0.04	0.50 ± 0.04
Cropped	ImageNet PSPI+AU MTL	PSPI	4 scores MTL	Ensemble	2.12 ± 0.04	6.66 ± 0.12	0.38 ± 0.02	0.46 ± 0.01
Aligned	PSPI+AU MTL	PSPI	4 scores MTL	Ensemble	1.94 ± 0.01	5.88 ± 0.27	0.43 ± 0.02	0.55 ± 0.03
Warped	PSPI+AU MTL	PSPI	4 scores MTL	Ensemble	1.97 ± 0.03	6.04 ± 0.07	0.42 ± 0.02	0.53 ± 0.01

Table 3.3. Comparison with Other Work

Model	Dataset	MAE	ICC	AUC
pRNN-HCRF (p=1) [MRP$^+$17]	UNBC	2.47 ± 0.18	0.36 ± 0.08	
pRNN-HCRF (p=2) [MRP$^+$17]	UNBC	2.46 ± 0.23	0.34 ± 0.04	
DeepFaceLIFT [LPS$^+$17]	UNBC	2.18	0.35	
EMTL (this paper)	UNBC	1.95 ± 0.06	0.43 ± 0.02	
TransferLearning [XCD$^+$18]	Child	-	-	0.72 ± 0.02
Extended MTL (Our Model)	Child	2.22 ± 0.10	0.33 ± 0.05	0.76 ± 0.01

Table 3.4. Comparison with Human

	MAE	MSE	ICC	PCC
EMTL (our model)	1.95 ± 0.06	5.90 ± 0.23	0.43 ± 0.03	0.55 ± 0.03
Human	1.76	6.26	0.66	0.66
Average of EMTL (our model) and Human	1.58 ± 0.03	4.58 ± 0.05	0.64 ± 0.01	0.68 ± 0.01
EMTL-TrueAU [XdS20]	1.73 ± 0.03	4.61 ± 0.19	0.61 ± 0.02	0.67 ± 0.02
Average of EMTL-TrueAU [XdS20] and Human	1.48 ± 0.02	4.22 ± 0.10	0.70 ± 0.01	0.71 ± 0.01

Chapter 4

Personalized Pain Detection in Facial Video with Uncertainty Estimation

4.1 Introduction

Two types of pain metrics are considered in pain studies [ALC$^+$09]. In facial video pain recognition, frame-level pain metrics are calculated from the intensity of objective facial muscle movements called facial action units (AUs) defined by the Facial Action Coding System (FACS). A commonly used combination of pain-related action units developed by [PS08] is called the Prkachin and Solomon Pain Intensity(PSPI) measure PSPI: PSPI=AU4+max(AU6,AU7)+ max(AU9,AU10)+AU43). The AU descriptions are: AU4 brow lowering, AU6 cheek raising, AU7 eyelid tightening, AU9 nose wrinkling, AU10 upper lip raising, AU12 oblique lip raising, AU20 horizontal lip stretch, AU25 lips parting, AU26 draw dropping, AU43 eye closure. Sequence-level pain metrics are overall pain levels rated by observers or the subjects themselves.

The current gold standard to evaluate pain is the sequence-level self-rated Visual Analog Scale (VAS). Automated pain evaluation systems developed to help detect pain [SAD$^+$15, XHdS19, XCD$^+$18, LPS$^+$17, MVJP17] can usually be broken down into two stages: Stage 1 predicts the PSPI score in each frame, and Stage 2 learns VAS using predicted PSPI scores in a video. This work follows the same two stage approach to predict VAS.

Pain is a personal, subjective experience, and VAS is a noisy label that differs in its relationship to facial expression across subjects. This makes automated pain estimation difficult when generalizing to subjects not in the training dataset. To address this issue, Martinez et al. introduced a facial expressiveness score, unique for each person, but their method requires labeled data for new subjects [MRP$^+$17]. Liu et al. personalized the estimation of self-reported pain via a set of hand-crafted personal features including age, gender and complexion [LPS$^+$17]. The labeling of these personal features is easier, but still the model can't automatically generalize to unseen subjects. There are also works tackling pain personalization in images instead of videos [RMZ$^+$20, RTK$^+$21].

In this work, we propose a systematic way to model the noise and bias in VAS in different subjects, and design a pain estimation model that can be optimized for new subjects using uncertainty estimation.

4.1.1 Uncertainty in Machine Learning Models

Uncertainty can be generally categorized into two types: epistemic or aleatory [DKD09, KG17]. Epistemic uncertainty can be reduced given enough data, while aleatoric uncertainty captures noise that is inherent in the observations.

In a supervised learning problem, suppose data points (x_i, y_i) are related via a model $y_i = f(x_i) + \varepsilon_i$, where f is the true function that maps data input to output, and ε_i is the noise inherent in the observations with zero mean and variance σ_i^2. A machine learning model seeks to find a function $\hat{f}(x;D)$ that approximates the true $f(x)$ as well as possible, using training data $D = \{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$. Using mean squared error to evaluate the approximation, then given a new observation (x,y), the expected squared error between $\hat{f}(x;D)$ and y is:

$$E[(y-\hat{f}(x;D))^2] \tag{4.1}$$

$$=E[(f(x)+\varepsilon-\hat{f}(x;D))^2] \tag{4.2}$$

$$=E[\varepsilon^2+(f(x)-\hat{f}(x;D))^2+2\varepsilon(f(x)-\hat{f}(x;D))] \tag{4.3}$$

$$=\sigma^2+E[(f(x)-\hat{f}(x;D))^2] \tag{4.4}$$

Equation (4.4) follows from (4.3) because ε is independent of $\hat{f}$.

σ^2 is often called the irreducible error. It is a property of the data, not the model, so it captures the aleatoric uncertainty. The second term doesn't exactly capture epistemic uncertainty because $\hat{f}$ is only one deterministic model, but it is correlated to epistemic uncertainty, evaluating how much the solution $\hat{f}$ over D varies from the true solution f assuming infinite number of data points.

Many authors have proposed to use neural networks to estimate the input dependent $f(x)$ as well as the variance $\sigma^2(x)$ of the prediction $\hat{f}(x)$ [NW94, LPB17]. In this work, we make the same assumption that the noise ε is input/subject dependent and can be predicted using a machine learning model.

4.1.2 Contributions

- We learn personalized individual models to evaluate the current gold standard pain metric VAS in video from video frames directly.
- We learn PSPI and VAS as a combination of the output of individual models to improve the generalizability of the pain prediction model.
- We learn the uncertainty of VAS prediction of each individual model, and improve the VAS prediction on new test subjects by adjusting ensemble weights based on the uncertainty of individual predictions
- Our model beats the current state-of-the-art performance on the UNBC-McMaster dataset.

4.2 Methods

Our model uses the Extended Multi-Task Learning (EMTL) model described in [XHdS19] as the baseline structure. The original EMTL model is trained using all training subjects together; in this work we train individual models on data from individual training subjects, and explore ways to combine these individual models so that the ensemble prediction is optimal for samples from test subjects.

4.2.1 Optimal Linear Combination of Individual Models

[XHdS19] proposed an optimal linear combination of multidimensional pain estimations (VAS, OPR, SEN, and AFF) to obtain an improved prediction of VAS. This method works very well in aggregating different aspects of pain to produce a better estimation. However, it didn't consider the subject-dependent aspect of pain, i.e. different patients having different understanding of pain and expressing pain in different ways through facial expression.

We address this problem by training personalized models: instead of training one model using all training subjects, we train several models each using video samples from one subject.

Consider each data point (x,y) as an observation of random variables (X,Y), and the model for subject s is denoted as $\hat{f}_s$. We learn the final prediction of VAS as a weighted sum of the predictions $\hat{f}_s(x)$. The overall model $\tilde{f}$ can be represented as:

$$\tilde{f}(x) = \sum_s \alpha_s \hat{f}_s(x) = \alpha^T \hat{f}(x) \tag{4.5}$$

The solution to minimizing the MSE of the the final model $E[(\tilde{f}(X) - Y)^2]$ subject to $\sum_s \alpha_s = 1$ can be obtained using the Lagrangian function:

$$L(X,\lambda) \tag{4.6}$$

$$=E[(Y - \alpha^T \hat{f}(X))^2] - \lambda \alpha^T 1 \tag{4.7}$$

$$=E\left[\left(\alpha^T (Y - \hat{f}(X))\right)^2\right] - \lambda \alpha^T 1 \tag{4.8}$$

$$=\alpha^T E\left[(Y - \hat{f}(X))(Y - \hat{f}(X))^T\right] \alpha - \lambda \alpha^T 1 \tag{4.9}$$

Setting the derivative of $L(X,\lambda)$ with respect to α to zero:

$$E\left[(Y-\hat{f}(X))(Y-\hat{f}(X))^T\right]\alpha-\lambda 1=0 \tag{4.10}$$

The solution contains the error matrix:

$$\hat{\alpha}=\frac{\Omega^{-1}1}{1^T\Omega^{-1}1} \tag{4.11}$$

where $\Omega=E\left[(Y-\hat{f}(X))(Y-\hat{f}(X))^T\right]$.

What this means is that, if a subject generalizes to others better than another subject, then the weight of the first subject should be larger than the weight of the second subject in the ensemble model $\tilde{f}$. The optimal linear combination takes into account the covariance between the different estimators and is optimal for the whole data distribution in the sense of mean squared error.

4.2.2 Ensemble using Predicted Variance

The optimal linear combination(OLC) model in section 4.2.1 only aims to reduce epistemic uncertainty, and helps the model generalize to data in the same distribution.

However, the data distribution is different for different subjects, and this is captured in the first term σ^2 in equation (4.4). In this section we propose to learn the variances of individual model predictions to account for both aleatoric and epistemic uncertainties. In practice, we learn $\hat{\sigma}_s^2(x)$ to approximate $(y-\hat{f}_s(x))^2$. This is not the variance exactly, but equals to the variance of label noise if $\hat{f}_s=f_s$.

The original MSE loss is only dependent on predicted means $\hat{f}_s(x)$, and assumes the same σ^2 for all data points. This is not true especially across subjects because both x, facial expression of pain, and y, the self-rated pain level VAS, are quite different across subjects. In other words, for different subjects, (x,y) data are in different domains. Our variance prediction model is able to predict such uncertainty due to domain shift, which can be used to determine parameters for the ensemble model. For example, if a video is quite similar to training subject 1, and completely different from training subject 2, then the pain score prediction from the model trained on subject 1 should have smaller σ^2 for this sample than the model trained on subject 2, meaning this sample is out-of-distribution for subject 2 model and prediction from subject 1 model is more trustworthy, so that the ensemble model should assign higher weights to scores output by subject 1 model.

The OLC model in section 4.2.1 can't do this because the optimal weight in equation (4.11) is only dependent on training samples. We bring in $\hat{\sigma}_s^2(x)$ which also depends on the input x to predict the best weighting in the ensemble model for specific test samples.

Input-dependent Regularization Using Learned Variance

We propose a new loss function which applies Tikhonov regularization to integrate predicted variance in personalized models:

$$Loss(\tilde{f}) = (y - \tilde{f})^2 + \beta\tilde{\sigma}^2 \tag{4.12}$$

where

$$\tilde{f} = \sum_s a_s \hat{f}_s = a^T \hat{f} \tag{4.13}$$

$$\tilde{\sigma}^2 = var(\tilde{\varepsilon}) = a^T \Sigma a \tag{4.14}$$

$a = [a_s]$ is the weight vector, $\hat{f} = [\hat{f}_s(x)]$ is the input vector, and $\Sigma = diag(\hat{\sigma}_s^2(x))$ is a diagonal matrix where the learned variances are on the diagonal.

So the loss (4.12) can be expressed as

$$Loss(\tilde{f}) = a^T(\Omega + \beta\Sigma)a \tag{4.15}$$

The first term is the MSE of the final prediction. It finds individual models that generalize well on the whole data distribution, and the MSE matrix Ω is the same as the Ω in equation (4.11), and is learned on the training data. The second term on the other hand looks for models performing better especially for the current video, and is different for each sample. At test time, the ensemble model will calculate the optimal weights a for the loss above using the same method as in section 4.2.1, using Ω learned from training data and Σ arising from the variance prediction model $\hat{\sigma}_s^2(x)$.

The optimal weight vector is determined by the following equation:

$$\hat{a} = \frac{(\Omega + \beta\Sigma)^{-1}1}{1^T(\Omega + \beta\Sigma)^{-1}1} \tag{4.16}$$

Here because Σ is dependent on input, the optimal $\hat{a}$ is dependent on the input as well.

Maximum Likelihood Using Learned Variance

Another way we propose to integrate the input-dependent $\sigma(x)^2$ is to use a maximum likelihood estimation framework. When we apply the ensemble model on a sample (x,y), we can represent the probability distribution of its output as a weighted sum of distributions of candidates:

$$P(y|x) = \sum_s \pi_s P_s(y|x) \tag{4.17}$$

$\pi = [\pi_s]$ is the weighting coefficient, and should meet the condition $\sum_s \pi_s = 1$. The mean and variance of $P_s(y|x)$ are approximated by $\hat{f}_s$ and $\hat{\sigma}_s^2$.

Note that $y = f(x) + \varepsilon(x)$ and $E[y] = f(x)$, so the ensemble prediction is:

$$\tilde{f} = \sum_s \pi_s \hat{f}_s \tag{4.18}$$

If we assume a multivariate Gaussian distribution for $y|x$ on the subjects, we have the probability of the training data:

$$P(D) \propto \prod_{(x,y)\in D} \exp\left((y-\pi^T \hat{f})^T \Sigma^{-1} (y-\pi^T \hat{f})\right) \tag{4.19}$$

Maximizing the likelihood is equivalent to minimizing:

$$\sum_{(x,y)} (y-\pi^T \hat{f})^T \Sigma^{-1} (y-\pi^T \hat{f}) \tag{4.20}$$

This has a similar form as the MSE in section 4.2.1, and the solution is similar to equation (4.11):

$$\hat{\pi} = \frac{W^{-1}1}{1^T W^{-1} 1} \tag{4.21}$$

where $W = E\left[(Y-\hat{f}(X))\Sigma^{-1}(Y-\hat{f}(X))^T\right]$.

Loss under Gaussian Assumption. In this section we make the strong assumption of Gaussian noise in labels. We can actually do more with this assumption. For example, if $\varepsilon \sim \mathcal{N}(0,\sigma^2)$, then the distribution of y given x is:

$$P(y|x;\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(y-f(x))^2}{2\sigma^2}\right) \tag{4.22}$$

The natural logarithm of the probability density function is:

$$\ln P(y|x;\sigma^2) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2) - \frac{(y-f(x))^2}{2\sigma^2} \tag{4.23}$$

The first term is a constant, so in a feed-forward neural network, the maximum log-likelihood estimation can be formulated as minimizing:

$$-L(f;D) = \sum_i \left(\frac{(y_i - f(x_i))^2}{2\sigma_i^2} + \frac{1}{2}\ln(\sigma_i^2)\right) \tag{4.24}$$

This negative log-likelihood can be used as a loss function in a maximum likelihood learning framework, and can be used in a neural network with split heads to learn f and σ^2 at the same time [NW94]. Note that MSE loss is a special case of this loss function when σ is constant across all data points.

The reason we didn't use this loss is that, we want to maximize the difference between individual models $\hat{f}_s$, while what this loss does is to ensure a smoother and more general $\hat{f}_s$ across different data domains by allowing larger variances $\hat{\sigma}$ for some samples.

4.2.3 Ensemble using Predicted Error

In the analysis above we ignored the correlation between the errors in individual model outputs. It may not be true that the errors are independent, so in this section we generalize the methods above to consider correlations between errors in different personalized model predictions.

Instead of learning $\hat{\sigma}_s(x)$, we use the same neural network structure as $\hat{f}_s(x)$ to predict $\hat{\varepsilon}_s(x)$ which approximates $y - \hat{f}_s(x)$. This allows us to calculate the covariance matrix $\Sigma = [\sigma_{ij}] = [\varepsilon_i \varepsilon_j]$ of the multivariate prediction.

To take covariance in prediction noise into consideration, we just need to replace Σ in section 4.2.2 by $\Sigma = [\sigma_{ij}] = [\varepsilon_i \varepsilon_j]$.

4.3 Experiments

4.3.1 Dataset

We developed our model based on the widely used UNBC-McMaster Shoulder Pain dataset [LCP$^+$11a]. It includes facial videos of participants suffering from shoulder pain while performing a series of active and passive range-of-motion tests to their affected and unaffected limbs on two separate occasions. The dataset has 25 subjects, 200 videos and 48,398 frames of size 320 x 240 pixels in total.

Algorithm 1: Pain Estimation Model Training

```
Data: $D = (X, Y)$, $D_s$ = data from subject $s$
Result: Model to predict $y \in Y$ given $x \in X$
/* Train Stage-1 personalized models                                  */
for s in training subjects do
    train $S1_s$ using $D_s$
    for x in D do
        get predictions $S1_s(x)$
    end
end
/* Ensemble learning on Stage-1 predictions                           */
Over $D_{train}$, learn $a_s$ to minimize the MSE of $S1(x) = \sum_s a_s S1_s(x)$
for x in D do
    get predictions $S1(x)$
end
/* Train Stage-2 personalized models                                  */
for s in training subjects do
    train $S2_s$ using $D_s$
    for x in D do
        get predictions $S2_s(S1(x))$
    end
end
/* Train Stage-2b variance/error prediction models                    */
for s in training subjects do
    train $S2b_s$ using $D_{train}$
    for x in D do
        get predictions $S2b_s(S1(x))$
    end
end
/* Ensemble learning on Stage-2 predictions using Stage-2b
   uncertainty estimations                                            */
Over $D_{training}$, learn error matrix to minimize the input dependent loss
```

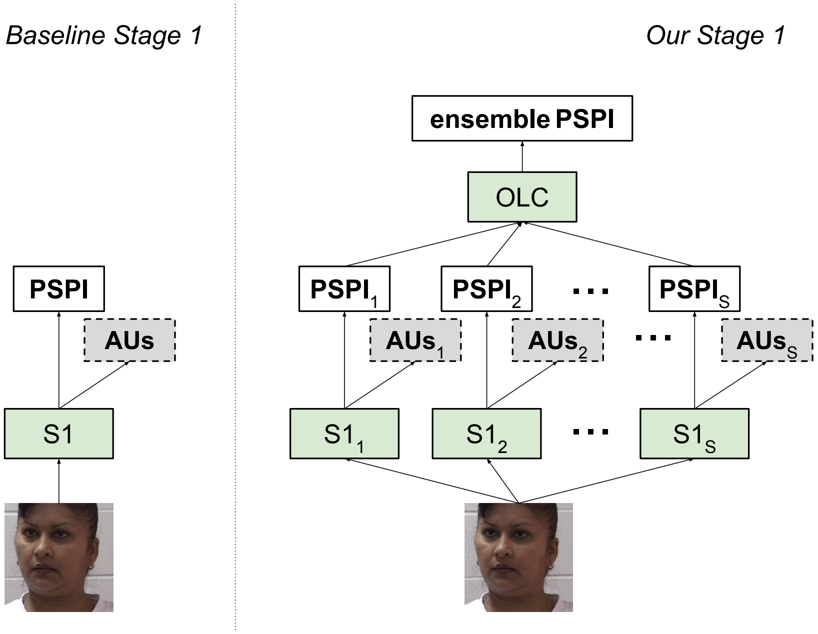


Figure 4.1. Stage 1 model structure. $S1$ and $S1_s$'s have a similar structure to VGG16. They are trained to predict PSPI and AUs. In our model $S1_s$ is trained with subject s, and OLC parameters are learned to combine predictions from individual models to get a better ensemble PSPI prediction.

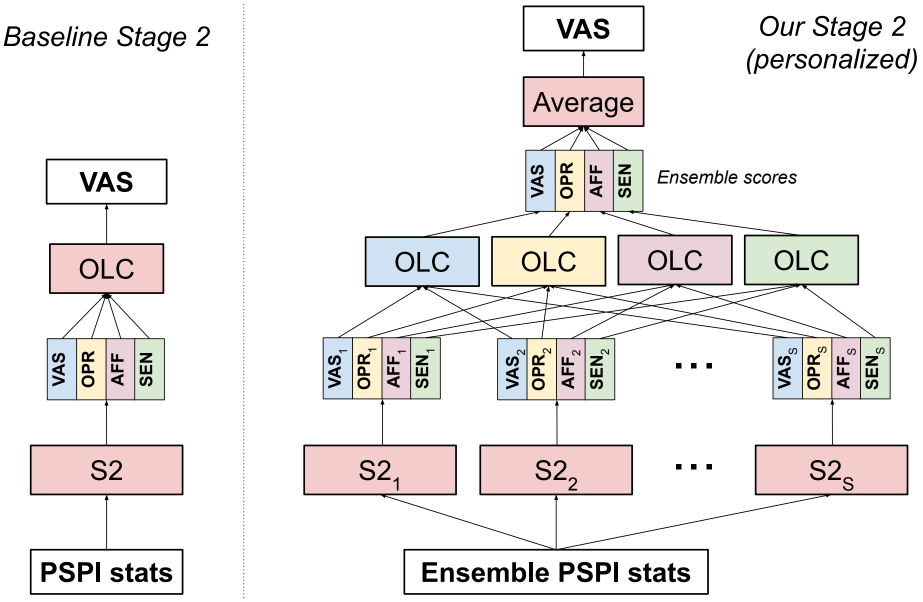


Figure 4.2. Stage 2 model structure using individual models. The baseline model [XHdS19] uses OLC to combine four pain score, and we use OLC to combine individual models, and then average the four scores to get the final estimation of VAS.

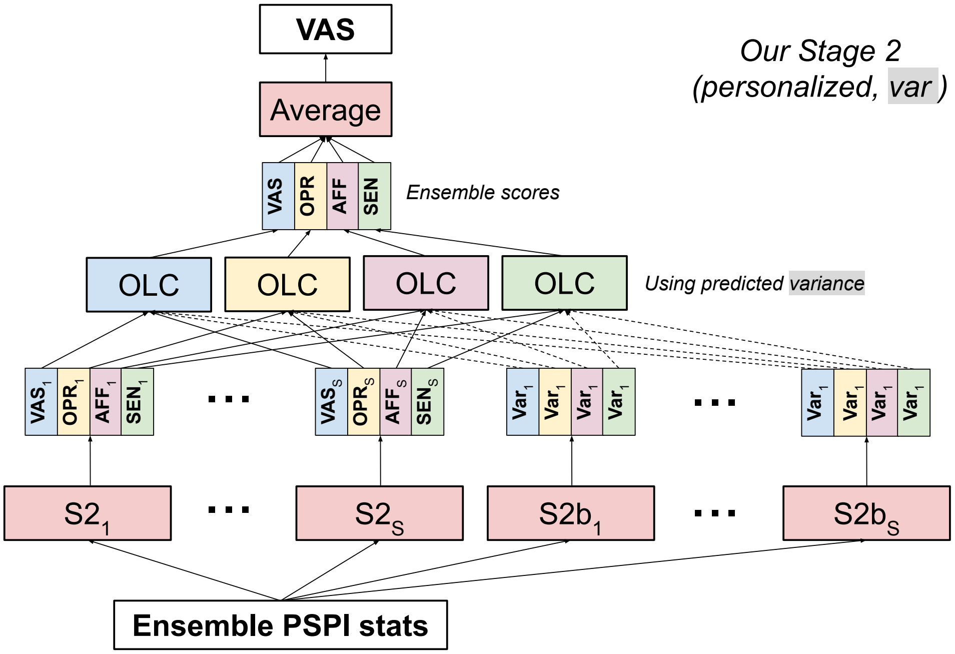


Figure 4.3. Stage 2 model structure using individual models and uncertainty estimation. $S2b_s$ models have the same structure as $S2_s$ and learns to predict $(y_s - S2_s(x))^2$ after the $S2_s$ models have been trained. This diagram uses variance predictions as an example. For error prediction models, we simply replace all the "Var" with "Error" in this figure.

Table 4.1. Frame-level PSPI Prediction

Stage 1 Model	MAE	MSE	ICC	PCC
Baseline	0.80 ± 0.07	1.53 ± 0.14	0.47 ± 0.04	0.49 ± 0.04
Personalized model	0.63 ± 0.05	1.28 ± 0.11	0.45 ± 0.05	0.50 ± 0.05

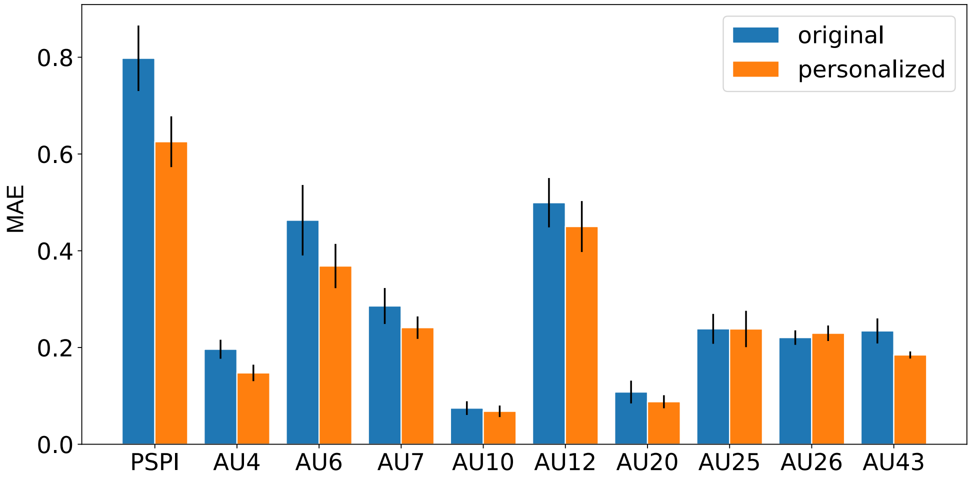


Figure 4.4. Stage 1 Performance.

The dataset has two types of labels: frame-level labels and sequence-level labels. Frame-level labels include 66 AAM landmarks, 11 facial action unit (AU) [EF76] intensities and 1 PSPI [PS08] score. In the first stage of our model, we train individual models to predict PSPI as well as AUs.

Sequence-level labels include the gold standard self-rating VAS pain score ranging from 0-10, as well as three other pain ratings: OPR (Observers Pain Rating - An estimate of the VAS given by a human observer of the video) 0-5, AFF (Affective-motivational scale) 0-15 and SEN (Sensory Scale) 0-15. The AFF and SEN measures are designed to separate the emotional and sensory aspects of pain. Their properties are discussed in more detail in [GMD78, HGDM80].

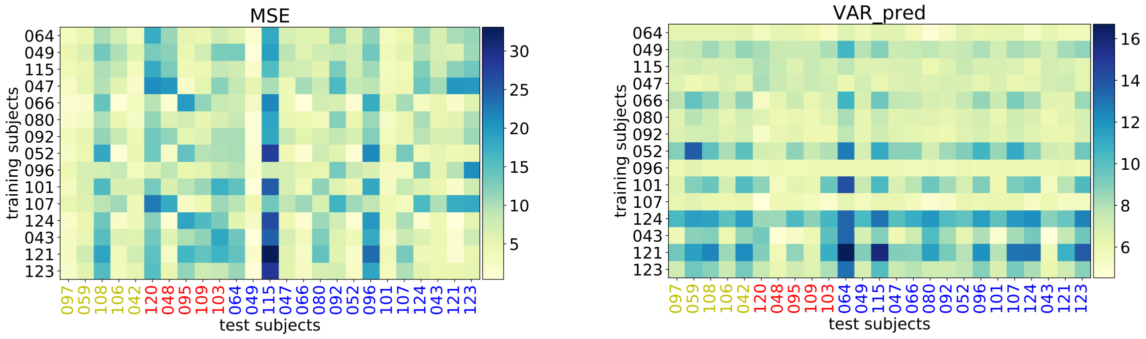
4.3.2 Algorithm, Model Training and Evaluation

Our model uses the EMTL model described in [XHdS19] as the baseline structure. In [XHdS19], Stage 1 fine-tunes a VGGFace neural network with the last layer replaced by a regression layer to predict frame-lavel PSPI and AUs from video frames, and Stage 2 uses a fully connected neural network to estimate sequence-level pain scores using 9 statistics of PSPI predictions in a video as features. The difference between our model and the EMTL model can be found in Figures 4.1, 4.2 and 4.3.

The training algorithm of our pain estimation model is shown in Algorithm 1. Implementation details such as image pre-processing and optimization methods are the same as [XHdS19].

Following [XHdS19], we performed 5-fold cross validation with each fold consisting of 5 subjects. We used the same training/test splits for all stages in each iteration. One of the 4 training folds is randomly selected as the validation set during neural network training. After 5 iterations, we concatenated all the test samples and calculated the Mean Absolute Error (MAE), Mean Squared Error (MSE), Intraclass Correlation Coefficient (ICC) and Pearson Correlation Coefficient (PCC).

For all models, we performed the 5-fold cross validation 5 times, and report mean and standard deviation of MAE, MSE, ICC and PCC over 5 runs of the 5-fold cross validation.



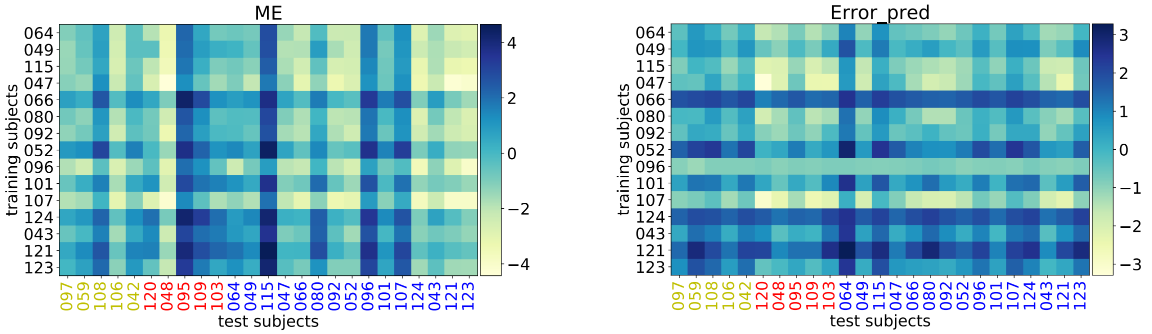
(a) $(y - \hat{f}_s)^2$

(b) $\hat{\sigma}_s^2$

Figure 4.5. Personalized Model MSE on Individuals. Actual (a) and Predicted (b)

Table 4.2. Frame-level PSPI Prediction

Stage 1 Model	Stage 2 Model	MAE	MSE	ICC	PCC
Baseline [XHdS19]	Baseline [XHdS19]	1.95 ± 0.06	5.90 ± 0.23	0.43 ± 0.03	0.55 ± 0.03
Personalized	Baseline	1.95 ± 0.07	5.66 ± 0.37	0.46 ± 0.03	0.57 ± 0.04
Personalized	Personalized	**1.88 ± 0.07**	5.70 ± 0.47	**0.50 ± 0.04**	0.57 ± 0.04
Personalized	Personalized, reg-variance	**1.88 ± 0.07**	**5.58 ± 0.37**	**0.49 ± 0.04**	**0.59 ± 0.04**
Personalized	Personalized, reg-error	**1.88 ± 0.07**	**5.57 ± 0.37**	**0.50 ± 0.04**	**0.59 ± 0.04**
Personalized	Personalized , MLE-variance	**1.87 ± 0.07**	**5.60 ± 0.40**	**0.50 ± 0.04**	**0.58 ± 0.04**
Personalized	Personalized , MLE-error	**1.88 ± 0.07**	5.69 ± 0.42	**0.50 ± 0.04**	0.57 ± 0.04



(a) $y - \hat{f}_s$

(b) $\hat{\varepsilon}_s$

Figure 4.6. Personalized Model Mean Error on Individuals. Actual (a) and Predicted (b)

4.3.3 Frame-level Pain using Individual Models

For the first stage, we train an individual VGGFace model for each subject. We didn't train from scratch but instead trained a Stage 1 model using all training subjects, and then fine-tuned it for 10 epochs using data from each subject to get the individual model for this subject. The model structure is shown in Figure 4.1.

The performance of individual models on their own subject's data is good. The training accuracy of individual models are higher on their own data than the training accuracy of the model trained on all subjects. But the test accuracy on subjects not used for training is lower for individual models. This is as expected because individual models can learn personalized distributions better, but won't work so well when used for other subjects.

We apply the optimal linear combination to individually trained models in section 4.2.1 to Stage 1, and show better performance on PSPI prediction (Table 4.1) and most AU predictions (Figure 4.4). We didn't use variance of S1 predictions based on inputs because looking at learning curves, we noticed that the squared error or error of S1 predictions can't be predicted using the same VGG16 structure based on image inputs. The validation error of learning $\hat{\sigma}^2$ or $\hat{\varepsilon}$ doesn't decrease while training.

4.3.4 Sequence-level Pain using Individual Models

After getting predictions of PSPI, we train individual Stage 2 models to predict VAS. The model structure is shown in Figure 4.2. The VAS prediction performance of the models is shown in Table 4.2.

The first row is the original model proposed in [XHdS19], and is the previous state-of-the-art. The second row uses personalized models for Stage 1, as described in section 4.3.3, and Stage 2 remains the same except using PSPI predictions learned with optimal linear combination on individual predictions. The performance is better than the first row, showing that learning models tuned to individual faces and combining the outputs with OLC at Stage 1 helps both PSPI prediction and VAS prediction.

The third row uses PSPI predictions based on OLC, as well as individual models in Stage 2 and OLC on top of individual VAS predictions. The performance is further improved. For Stage 2 individual models, each model is trained from scratch on one training subject.

This shows that, even without uncertainty estimation, simply learning individual models and running ensemble learning on top of the individual predictions can improve the performance of the model on unseen test subjects significantly.

In Figure 4.5(a) we take one fold in one iteration as an example, and plot the MAE of each individual model on each test subject. We can see that although clearly some subjects are generally good as training or test subjects, there are significant differences across subjects, e.g. subject 049 is easy to predict as a test subject, but its performance using the training subject 066 is not as good as subject 106 which is not performing as well using other training subjects. For some of the test subjects, such as subjects 048 and 121, the MAE varies a lot across training models.

It is also not true that a training subject always performs the best on itself. We don't see clear diagonal pattern in the square on the right side where the test subjects are in the same order as the training subjects. For example, subject 155 performs better on models trained with subject

047 and 096 than the model trained on itself.

We calculated a "cheating" MAE where we look at test performance in this plot and choose the best training subject for each test subject, and the MAE is 0.26, showing great potential for individual model ensembles.

4.3.5 Sequence-level Pain using Individual Models and Uncertainty Estimation

In this work, we use the same structure as the Stage 2 sequence-level prediction models to predict the error or variance of the predictions, and the final model is shown in Figure 4.3. For each personalized model $\hat{f}_s(x)$, we train models to predict $(y-\hat{f}_s(x))^2$, and $y-\hat{f}_s(x)$ using all training subjects. These models are denoted as $\hat{\sigma^2}_s(x)$ and $\hat{\varepsilon}_s(x)$ respectively, and we refer to them as variance predictions and error predictions.

Figure 4.5 plots the average squared error $(y-\hat{f}_s)^2$ and the average predicted squared error $\hat{\sigma}_s^2$ of each individual model on each test subjects. For a test subject, we'd like the variance prediction models $\hat{\sigma}_s^2$ to be able to predict, from the training data, which training models will be more reliable on test data, and they successfully recognize such differences. For example, test subject 115 picks out training subjects 107 and 096 and 047 as having low mean squared error predictions, and would weight them more using our input-dependent ensemble methods.

Similarly, Figure 4.6 plots the average (predicted) error. The error prediction models $\hat{\varepsilon}_s$ can not only learn the reliability of different individual models $\hat{f}_s$, but also their bias, e.g. they all learn that subject 066 generally overrates his pain level, or the person is more stoic in their facial expression of pain, and subject 107 tends to rate his VAS lower than shown in his facial expression.

We test our four methods using uncertainty estimations: regularization using variance, MLE using variance, regularization using error, MLE using error. For the regularization methods, in practice, if there is enough data, β can be decided using cross-validation. In this work we simply use the fixed $\beta = 1/|D_{training}|$.

The last four rows in Table 4.2 show the models using personalized, input-dependent uncertainty estimation proposed in section 4.2.2 and 4.2.3. They all improve the performance of the model.

The variance in samples is large, resulting in relatively large standard deviation in the performance metrics. However, as our train-test cross-validation splits were the same across all models, we can perform pairwise tests which are much more sensitive in this case. We performed a Wilcoxon signed-rank one-sided test and the p-value is < 0.0005 for all four methods, supporting our hypothesis that our personalized models are significantly better than the baseline model.

4.4 Conclusion

The relationship between perceived pain and facial expression of that pain is different for different people. In this work we addressed this issue by creating a method that learns data-dependent personalized models. Personalization is performed at stage one acting on video frames and also at stage two predicting VAS from statistics of the PSPI measure. Uncertainty estimation is used at the second stage to adjust ensemble weights to improve performance on new subjects. We showed on the UNBC-McMaster Shoulder Pain dataset that our method improves upon the non-personalized model and achieves the state-of-the-art performance.

Acknowledgments

Chapter 4, in part, has been submitted for publication of the material as it may appear in International Conference of the IEEE Engineering in Medicine and Biology Society, 2021. Xu, Xiaojing, and Virginia R. de Sa. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Summary

In this dissertation, we build a deep learning framework to estimate self-rated pain level from videos directly. We study the relationship between different levels of pain metrics, consider the multidimensional measurements of pain, apply transfer learning on small datasets, and use uncertainty estimation to optimize the prediction for a specific test sample.

Chapter 2 studies the relationship between sequence-level metrics and frame-level metrics. Specifically, we explore an extended multitask learning model to predict VAS from human-labeled AUs with the help of other sequence-level pain measurements during training. This model consists of two parts: a multitask learning neural network model to predict multidimensional pain scores, and an ensemble learning model to linearly combine the multidimensional pain scores to best approximate VAS. Starting from human-labeled AUs, the model outperforms provided human sequence-level estimates.

Chapter 3 learns sequence-level metrics based on frame-level automatically predicted AUs with a software called iMotions. We apply transfer learning by training another machine learning model to map iMotions AU codings to a subspace of manual AU codings to enable more robust pain recognition performance when only automatically coded AUs are available for the test data.

Chapter 4 learns a VGGFace neural network multitask learning model to predict AUs. Combining with the model structure in Chapter 2, we build a 3-stage multitask-learning multidimensional pain deep model to predict VAS from videos directly.

Chapter 5 improves the model further using individual models and uncertainty estimation. For a new test video, we jointly consider which individual models generalize well generally, and which individual models are more similar/accurate to this test video, in order to choose the optimal combination of individual models and get the best performance on new test videos. Our structure achieves state-of-the-art performance on two datasets.

We closed the gap between frame-level and sequence-level pain metrics. We designed a model that serves as a baseline of how well one can predict VAS using AUs. It can be combined with pain estimation work on images to achieve end-to-end VAS prediction. In this case, the performance shown by our model provided an upper bound on the accuracy that can be achieved when using automatically estimated AUs instead of manually labeled AUs.

We have also shown that if we can achieve better predictions of AUs, the final VAS prediction can also be improved. Although AU/PSPI prediction should be improved in a way that also improves VAS prediction, because more accurate AU/PSPI doesn't always leads to more accurate VAS.

Our model based on AUs beat human observers, and our model based on videos achieved the state-of-the-art performance on two datasets. In clinics, the model using manual AUs outperformed human observer and can be used to teach human which AUs to focus on. The model based on raw video was almost as good as human. So when a human observer is not present, the model can serve as a cheap, effective and consistent monitoring system that is almost as accurate as human observers. Moreover, when simply averaged with human predictors, the model beat human alone. The model provided additional information than human observers. So when a human observer is present, the model can be combined with human observers to provide better estimates of pain than human alone.

Further, we studied the subjective nature of pain, and propose to improve pain estimation using individual models. We combine individual models in a way that is not only optimal in general, but also optimized for a new test sample.

Bibliography

[ABR83] Tim A Ahles, Edward B Blanchard, and John C Ruckdeschel. The multidimensional nature of cancer-related pain. *Pain*, 17(3):277–288, 1983.

[AKRP$^+$15] Min SH Aung, Sebastian Kaltwang, Bernardino Romera-Paredes, Brais Martinez, Aneesha Singh, Matteo Cella, Michel Valstar, Hongying Meng, Andrew Kemp, Moshen Shafizadeh, et al. The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal emopain dataset. *IEEE transactions on affective computing*, 7(4):435–451, 2015.

[AKRP$^+$16] Min SH Aung, Sebastian Kaltwang, Bernardino Romera-Paredes, Brais Martinez, Aneesha Singh, Matteo Cella, Michel Valstar, Hongying Meng, Andrew Kemp, Moshen Shafizadeh, et al. The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal emopain dataset. *IEEE transactions on affective computing*, 7(4):435–451, 2016.

[ALC$^+$09] Ahmed Bilal Ashraf, Simon Lucey, Jeffrey F Cohn, Tsuhan Chen, Zara Ambadar, Kenneth M Prkachin, and Patricia E Solomon. The painful face–pain expression recognition using active appearance models. *Image and vision computing*, 27(12):1788–1796, 2009.

[ALS16] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[BLFL14] Marian Stewart Bartlett, Gwen C Littlewort, Mark G Frank, and Kang Lee. Automatic decoding of facial movements reveals deceptive pain expressions. *Current Biology*, 24(7):738–743, 2014.

[BMR15] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6. IEEE, 2015.

[Car97] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[CAW18] Zhanli Chen, Rashid Ansari, and Diana Wilkie. Automated pain detection from facial expressions using facs: A review. *arXiv preprint arXiv:1811.07988*, 2018.

[CDlTC17] Wen-Sheng Chu, Fernando De la Torre, and Jeffrey F Cohn. Learning spatial and temporal cues for multi-label facial action unit detection. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 25–32. IEEE, 2017.

[CJL$^+$19] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[Cle86] Robert T Clemen. Linear constraints and the efficiency of combined forecasts. *Journal of Forecasting*, 5(1):31–38, 1986.

[CPG11] Kenneth D Craig, Kenneth M Prkachin, and Ruth E Grunau. The facial expression of pain. 2011.

[CPS19] Evan David Campbell, Angkoon Phinyomark, and Erik Justin Scheme. Feature extraction and selection for pain recognition using peripheral physiological signals. *Frontiers in neuroscience*, 13:437, 2019.

[Cra92] Kenneth D Craig. The facial expression of pain better than a thousand words? *APS Journal*, 1(3):153–162, 1992.

[CYT$^+$02] W Crawford Clark, Joseph C Yang, Siu-Lun Tsui, Kwok-Fu Ng, and Susanne Bennett Clark. Unidimensional pain rating scales: a multidimensional affect and pain survey (maps) analysis of what they really measure. *Pain*, 98(3):241–247, 2002.

[DKD09] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[DLR$^+$78] WW Downie, PA Leatham, VM Rhind, V Wright, JA Branco, and JA Anderson. Studies with pain rating scales. *Annals of the rheumatic diseases*, 37(4):378–381, 1978.

[ECJ$^+$19] Itir Onal Ertugrul, Jeffrey F Cohn, László A Jeni, Zheng Zhang, Lijun Yin, and Qiang Ji. Cross-domain au detection: Domains, learning approaches, and measures. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[EF76] Paul Ekman and Wallace V Friesen. Measuring facial movement. *Environmental psychology and nonverbal behavior*, 1(1):56–75, 1976.

[For76] Wilbert Evans Fordyce. *Behavioral methods for chronic pain and illness*, volume 1. Mosby St. Louis, 1976.

[FQA$^+$90] M Bosi Ferraz, MR Quaresma, LR Aquino, E Atra, P Tugwell, and CH Goldsmith. Reliability of pain scales in the assessment of literate and illiterate patients with rheumatoid arthritis. *The Journal of rheumatology*, 17(8):1022–1024, 1990.

[G$^+$17] Anu George et al. Automatic recognition of facial expression using features of salient patches with svm and ann classifier. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, pages 908–913. IEEE, 2017.

[GC87] Ruth VE Grunau and Kenneth D Craig. Pain expression in neonates: facial action and cry. *Pain*, 28(3):395–410, 1987.

[GMD78] Richard H Gracely, Patricia McGrath, and Ronald Dubner. Ratio scales of sensory and affective verbal pain descriptors. *Pain*, 5(1):5–18, 1978.

[Has97] Sherif Hashem. Optimal linear combinations of neural networks. *Neural networks*, 10(4):599–614, 1997.

[HBN$^+$18] Mohammad A Haque, Ruben B Bautista, Fatemeh Noroozi, Kaustubh Kulkarni, Christian B Laursen, Ramin Irani, Marco Bellantonio, Sergio Escalera, Golamreza Anbarjafari, Kamal Nasrollahi, et al. Deep multimodal pain recognition: a database and comparison of spatio-temporal visual modalities. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 250–257. IEEE, 2018.

[HC04] Marilyn L Hill and Kenneth D Craig. Detecting deception in facial expressions of pain: accuracy and training. *The Clinical journal of pain*, 20(6):415–422, 2004.

[HGDM80] Marc W Heft, Richard H Gracely, Ronald Dubner, and Patricia A McGrath. A validation model for verbal descriptor scaling of human clinical pain. *Pain*, 9(3):363–373, 1980.

[HHG$^+$18] Kara Hawley, Jeannie S. Huang, Matthew Goodwin, Damaris Diaz, Virginia R. de Sa, Kathryn A. Birnie, Christine T. Chambers, and Kenneth D. Craig. Youth and parent appraisals of participation in a study of spontaneous and induced pediatric clinical pain. *Ethics & Behavior*, pages 1–15, 2018.

[HHP$^+$14] Thomas Hadjistavropoulos, Keela Herr, Kenneth M Prkachin, Kenneth D Craig, Stephen J Gibson, Albert Lukas, and Jonathan H Smith. Pain assessment in elderly adults with dementia. *The Lancet Neurology*, 13(12):1216–1227, 2014.

[HSDA10] DL Hoffman, A Sadosky, EM Dukes, and J. Alvir. How do changes in pain severity levels correspond to changes in health status and function in patients with painful diabetic peripheral neuropathy. *Pain*, 149(2):194–201, May 2010.

[JPP+15] Eun-Hye Jang, Byoung-Jun Park, Mi-Sook Park, Sang-Hyeob Kim, and Jin-Hun Sohn. Analysis of physiological signals for recognition of boredom, pain, and surprise emotions. *Journal of physiological anthropology*, 34(1):25, 2015.

[JV16] Shashank Jaiswal and Michel Valstar. Deep learning the dynamic appearance and shape of facial action units. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–8. IEEE, 2016.

[JVP11] Bihan Jiang, Michel F Valstar, and Maja Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 314–321. IEEE, 2011.

[JZHM75] CRB Joyce, DW Zutshi, V Hrubes, and RM Mason. Comparison of fixed interval and visual analogue scales for rating chronic pain. *European journal of clinical pharmacology*, 8(6):415–420, 1975.

[KG17] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[KL14] Miriam Kunz and Stefan Lautenbacher. The faces of pain: a cluster analysis of individual differences in facial activity patterns of pain. *European Journal of Pain*, 18(6):813–823, 2014.

[KVR19] Sudhakar Kumawat, Manisha Verma, and Shanmuganathan Raman. Lbvcnn: Local binary volume convolutional neural network for facial expression recognition from image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[LAZ17] Wei Li, Farnaz Abtahi, and Zhigang Zhu. Action unit detection with region adaptation, multi-labeling learning and optimal temporal fusing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2017.

[LAZY18] Wei Li, Farnaz Abtahi, Zhigang Zhu, and Lijun Yin. Eac-net: Deep nets with enhancing and cropping for facial action unit detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2583–2596, 2018.

[LCC06] Anne-Claire Larochette, Christine T Chambers, and Kenneth D Craig. Genuine, suppressed and faked facial expressions of pain in children. *Pain*, 126(1-3):64–71, 2006.

[LCP+11a] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *Face and Gesture 2011*, pages 57–64. IEEE, 2011.

[LCP11b] Mary E Lynch, Kenneth D Craig, and Philip WH Peng. *Clinical pain management: a practical guide*. Wiley Online Library, 2011.

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[LPS$^+$17] Dianbo Liu, Fengjiao Peng, Andrew Shea, Rosalind Picard, et al. Deepfacelift: interpretable personalized models for automatic estimation of self-reported pain. *arXiv preprint arXiv:1708.04670*, 2017.

[LWW$^+$11] Gwen Littlewort, Jacob Whitehill, Tingfan Wu, Ian Fasel, Mark Frank, Javier Movellan, and Marian Bartlett. The computer expression recognition toolbox (cert). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 298–305. IEEE, 2011.

[MAEAKAS20] Rasha M Al-Eidan, Hend Al-Khalifa, and AbdulMalik Al-Salman. Deep-learning-based models for pain recognition: A systematic review. *Applied Sciences*, 10(17):5984, 2020.

[MBML03] Paolo L Manfredi, Brenda Breuer, Diane E Meier, and Leslie Libow. Pain assessment in elderly patients with severe dementia. *Journal of Pain and Symptom Management*, 25(1):48–52, 2003.

[MBPVG14] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *European conference on computer vision*, pages 720–735. Springer, 2014.

[McG92] Deborah B McGuire. Comprehensive and multidimensional assessment and measurement of pain. *Journal of pain and symptom management*, 7(5):312–319, 1992.

[MR06] Md Maruf Monwar and Siamak Rezaei. Pain recognition using artificial neural network. In *Signal Processing and Information Technology, 2006 IEEE International Symposium on*, pages 28–33. IEEE, 2006.

[MRP$^+$17] Lopez Martinez, Daniel Rosalind Picard, et al. Personalized automatic estimation of self-reported pain intensity from facial expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 70–79, 2017.

[MVJP17] Brais Martinez, Michel F Valstar, Bihan Jiang, and Maja Pantic. Automatic analysis of facial actions: A survey. *IEEE Trans on Affective Computing*, 2017.

[NW94] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international* *conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[OBBMW15] Temitayo A Olugbade, Nadia Bianchi-Berthouze, Nicolai Marquardt, and Amanda C Williams. Pain level recognition using kinematics and muscle activity for physical rehabilitation in chronic pain. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 243–249. IEEE, 2015.

[PBB+01] Jean-Francois Payen, Olivier Bru, Jean-Luc Bosson, Anna Lagrasta, Eric Novel, Isabelle Deschaux, Pierre Lavagne, and Claude Jacquot. Assessing pain in critically ill sedated patients by using a behavioral pain scale. *Critical care medicine*, 29(12):2258–2263, 2001.

[Prk92] Kenneth M Prkachin. The consistency of facial expressions of pain: a comparison across modalities. *Pain*, 51(3):297–306, 1992.

[Prk09] Kenneth M Prkachin. Assessing pain by facial expression: facial expression as nexus. *Pain Res Manag.*, 14(1):53–58, 2009.

[PS08] Kenneth M Prkachin and Patricia E Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.

[PVZ15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[PY10] Sinno J. Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans on knowledge and data engineering*, 22(10):1345–1359, 2010.

[QSH15] Brenna L Quinn, Esther Seibold, and Laura Hayman. Pain assessment in children with special needs: A review of the literature. *Exceptional Children*, 82(1):44–57, 2015.

[RCC+20] Srinivasa N Raja, Daniel B Carr, Milton Cohen, Nanna B Finnerup, Herta Flor, Stephen Gibson, Francis J Keefe, Jeffrey S Mogil, Matthias Ringkamp, Kathleen A Sluka, et al. The revised international association for the study of pain definition of pain: concepts, challenges, and compromises. *Pain*, 161(9):1976–1982, 2020.

[RCG+17] Pau Rodriguez, Guillem Cucurull, Jordi Gonzàlez, Josep M Gonfaus, Kamal Nasrollahi, Thomas B Moeslund, and F Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE transactions on cybernetics*, 2017.

[RLA18] Andrés Romero, Juan León, and Pablo Arbeláez. Multi-view dynamic facial action unit detection. *Image and Vision Computing*, 2018.

[RMZ+20] Sia Rezaei, Abhishek Moturu, Shun Zhao, Kenneth M Prkachin, Thomas Hadjistavropoulos, and Babak Taati. Unobtrusive pain monitoring in older adults with dementia using pairwise and contrastive training. *IEEE Journal of Biomedical and Health Informatics*, 2020.

[RPP13] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Automatic pain intensity estimation with heteroscedastic conditional ordinal random fields. In *International Symposium on Visual Computing*, pages 234–243. Springer, 2013.

[RRBP16] Adria Ruiz, Ognjen Rudovic, Xavier Binefa, and Maja Pantic. Multi-instance dynamic ordinal random fields for weakly-supervised pain intensity estimation. In *Asian Conference on Computer Vision*, pages 171–186. Springer, 2016.

[RRM+07] Anne-Sylvie Ramelet, Nancy Rees, Susan McDonald, Max Bulsara, and Huda Huijer Abu-Saad. Development and preliminary psychometric testing of the multidimensional assessment of pain scale: Maps. *Pediatric Anesthesia*, 17(4):333–340, 2007.

[RTK+21] Ognjen Rudovic, Nicolas Tobis, Sebastian Kaltwang, Björn Schuller, Daniel Rueckert, Jeffrey F Cohn, and Rosalind W Picard. Personalized federated deep learning for pain estimation from face images. *arXiv preprint arXiv:2101.04800*, 2021.

[SAD+15] Karan Sikka, Alex A Ahmed, Damaris Diaz, Matthew S Goodwin, Kenneth D Craig, Marian S Bartlett, and Jeannie S Huang. Automated assessment of children's postoperative pain using computer vision. *Pediatrics*, 136(1):e124–e131, 2015.

[SDB13] Karan Sikka, Abhinav Dhall, and Marian Bartlett. Weakly supervised pain localization using multiple instance learning. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8. IEEE, 2013.

[SFV+17] Kamal Kaur Sekhon, Samantha R Fashler, Judith Versloot, Spencer Lee, and Kenneth D Craig. Children's behavioral pain cues: Implicit automaticity and control dimensions in observational measures. *Pain Res Manag.*, 2017.

[SH19] Ali Sellami and Heasoo Hwang. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Systems with Applications*, 122:75–84, 2019.

[SLC+19] Zhiwen Shao, Zhilei Liu, Jianfei Cai, Yunsheng Wu, and Lizhuang Ma. Facial action unit detection using attention and relation learning. *IEEE transactions on affective computing*, 2019.

[SLCM18] Zhiwen Shao, Zhilei Liu, Jianfei Cai, and Lizhuang Ma. Deep adaptive attention for joint facial action unit detection and face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 705–720, 2018.

[TH18] Mohammad Tavakolian and Abdenour Hadid. Deep spatiotemporal representation of the face for automatic pain intensity estimation. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 350–354. IEEE, 2018.

[TL86] G Trenkler and EP Liski. Linear constraints and the efficiency of combined forecasts. *Journal of Forecasting*, 5(3):197–202, 1986.

[TZY$^+$17] Chuangao Tang, Wenming Zheng, Jingwei Yan, Qiang Li, Yang Li, Tong Zhang, and Zhen Cui. View-independent facial action unit detection. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 878–882. IEEE, 2017.

[VB09] Carl L Von Baeyer. Children's self-report of pain intensity: what we know, where we are headed. *Pain Research and Management*, 14(1):39–45, 2009.

[vBVvdS$^+$17] Regina LM van Boekel, Kris CP Vissers, Rob van der Sande, Ewald Bronkhorst, Jos GC Lerou, and Monique AH Steegers. Moving beyond pain scores: Multidimensional pain assessment is essential for adequate pain management after surgery. *PLoS One*, 12(5):e0177345, 2017.

[VGL$^+$16] Maria Velana, Sascha Gruss, Georg Layher, Patrick Thiam, Yan Zhang, Daniel Schork, Viktor Kessler, Sascha Meudt, Heiko Neumann, Jonghwa Kim, et al. The senseemotion database: A multimodal database for the development and systematic validation of an automatic pain-and emotion-recognition system. In *IAPR Workshop on Multimodal Pattern Recognition of Social Signals in Human-Computer Interaction*, pages 127–139. Springer, 2016.

[VP06] Michel Valstar and Maja Pantic. Fully automatic facial action unit detection and temporal analysis. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 149–149. IEEE, 2006.

[WAHLE$^+$16] Philipp Werner, Ayoub Al-Hamadi, Kerstin Limbrecht-Ecklundt, Steffen Walter, Sascha Gruss, and Harald C Traue. Automatic pain assessment with facial activity descriptors. *IEEE Transactions on Affective Computing*, 8(3):286–299, 2016.

[WGE$^+$13] Steffen Walter, Sascha Gruss, Hagen Ehleiter, Junwen Tan, Harald C Traue, Philipp Werner, Ayoub Al-Hamadi, Stephen Crawcour, Adriano O Andrade, and Gustavo Moreira da Silva. The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *2013 IEEE international conference on cybernetics (CYBCO)*, pages 128–131. IEEE, 2013.

[WHM11] Lior Wolf, Tal Hassner, and Itay Maoz. *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011.

[Wil02] Amanda C de C Williams. Facial expression of pain: an evolutionary account. *Behavioral and brain sciences*, 25(4):439–455, 2002.

[WVW07] Jeremy West, Dan Ventura, and Sean Warnick. Spring research presentation: A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences*, 1, 2007.

[WXL$^+$17] Feng Wang, Xiang Xiang, Chang Liu, Trac D Tran, Austin Reiter, Gregory D Hager, Harry Quon, Jian Cheng, and Alan L Yuille. Regularizing face verification nets for pain intensity regression. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1087–1091. IEEE, 2017.

[XCD$^+$18] Xiaojing Xu, Kenneth D. Craig, Damaris Diaz, Matthew S. Goodwin, Murat Akcakaya, Büşra Tuğçe Susam, Jeannie S. Huang, and Virginia R. de Sa. Automated pain detection in facial videos of children using human-assisted transfer learning. In *Joint Workshop on Artificial Intelligence in Health*, pages 10–21. CEUR-WS, 2018.

[XCD$^+$19] Xiaojing Xu, Kenneth D. Craig, Damaris Diaz, Matthew S. Goodwin, Murat Akcakaya, Büşra Tuğçe Susam, Jeannie S. Huang, and Virginia R. de Sa. Automated pain detection in facial videos of children using human-assisted transfer learning. In *Artificial Intelligence in Health. AIH 2018. Lecture Notes in Computer Science*, volume 11326, pages 162–180. Springer International Publishing, Cham, 2019.

[XdS20] Xiaojing Xu and Virginia R. de Sa. Exploring multidimensional measurements for pain evaluation using facial action units (in press). In *International Workshop on Automated Assessment for Pain at 15th IEEE International Conference on Face and Gesture Recognition (FG2020)*. IEEE, 2020.

[XHdS19] Xiaojing Xu, Jeannie S. Huang, and Virginia R. de Sa. Pain evaluation in video using extended multitask learning from multidimensional measurements. In *Machine Learning for Health ML4H at NeurIPS 2019*, Proceedings of Machine Learning Research. PMLR, 2019.

[XSN$^+$18] Xiaojing Xu, Büsra Tugçe Susam, Hooman Nezamfar, Damaris Diaz, Kenneth D Craig, Matthew S Goodwin, Murat Akcakaya, Jeannie S Huang, and Virginia R de Sa. Towards automated pain detection in children using facial and electrodermal activity. In *Joint Workshop on AI in Health*, pages 208–211. CEUR-WS, 2018.

[ZCZ16] Kaili Zhao, Wen-Sheng Chu, and Honggang Zhang. Deep region and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3391–3399, 2016.

[ZGKS18] Ghada Zamzmi, Dmitry Goldgof, Rangachar Kasturi, and Yu Sun. Neonatal pain expression recognition using transfer learning. *arXiv preprint arXiv:1807.01631*, 2018.

[ZHSZ16] Jing Zhou, Xiaopeng Hong, Fei Su, and Guoying Zhao. Recurrent convolutional neural network regression for continuous pain intensity estimation in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 84–92, 2016.

[ZPG$^+$16] Ghada Zamzmi, Chih-Yun Pai, Dmitry Goldgof, Rangachar Kasturi, Yu Sun, and Terri Ashmeade. Machine-based multimodal pain assessment tool for infants: a review. *preprint arXiv:1607.00331*, 2016.

[ZYC$^+$14] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.