Imperial College London
Department of Computing

Regression-based Estimation of Pain and Facial Expression Intensity

Sebastian Kaltwang

June, 2015

Supervised by Prof. Maja Pantic

Submitted in part fulfilment of the requirements for the degree of PhD in Computing and the Diploma of Imperial College London. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

Abstract

Human inner feelings and psychological states like pain are subjective states that cannot be directly measured, but can be estimated from non-verbal behaviour such as spontaneous facial expressions. Since these expressions are typically characterized by subtle movements of facial parts, analysis of the facial details is required. The contribution of this thesis is two-fold. First, we propose a novel set of Bayesian regression-based learning methods for intensity estimation of facial expressions. Second, we create and publicly release the first multi-modal database of patients experiencing chronic pain, in order to facilitate further research into machine learning for automated analysis of pain.

We formulate three novel regression methods for continuous estimation of the intensity of facial expressions of pain and facial muscle groups (AUs). The first regression model treats the observed face holistically and estimates the intensity of target expressions using the framework of Relevance Vector Machine (RVM) and the newly proposed fusion of the shape and appearance features. This is the first method in the field that addresses automated continuous intensity estimation of facial expressions of pain. We then extend this approach to the Doubly Sparse RVM (DSRVM) that automatically learns the importance of various facial parts for the target task at hand. DSRVM achieves this by enforcing double sparsity by jointly selecting the most relevant training examples (a.k.a. relevance vectors) and the most important kernels associated with the informative facial parts for estimation of facial expression intensity. This advances prior work on multiple-kernel learning, where the kernel sparsity is typically ignored. Lastly, we introduce a regression-based approach that jointly learns the inter-dependence of facial parts and multiple AU or pain targets. This is accomplished by a newly formulated latent tree (LT) model, that efficiently learns a hidden inference structure between features and targets. The proposed approach is the first that addresses the joint estimation of continuous intensity of multiple AU outputs in a principled manner. We show that this joint approach achieves better intensity estimation of AUs compared to existing methods, especially in the presence of noisy inputs.

The proposed regression methods have been evaluated on two established datasets of naturalistic facial expressions, i.e., DISFA and ShoulderPain, and our newly created dataset, named EmoPain. The new database consists of spontaneously displayed pain-related facial expressions and body movements recorded by multiple modalities, while patients with chronic back-pain were performing instructed physical exercises. Facial expression videos have been annotated frame-wise in terms of the continuous pain intensity. We empirically show that the proposed methods, which model the face explicitly as the sum of its parts, outperform the existing state-of-the-art methods for the target tasks. This supports the findings in psychology research which suggest that only components of expressions rather than the holistic face play the key role in interpretation of human facial expression interpretation, and, in particular, its intensity estimation.

Acknowledgements

I would like to thank Prof. Maja Pantic for her supervision and support, which made this thesis possible. I would also like to thank Prof. Sinisa Todorovic for the collaboration and guidance. Furthermore, I am grateful for the discussions and suggestions from my colleagues at the iBug group, especially Ognjen Rudovic, Stavros Petridis and Brais Martinez. Many thanks goes to the team of UCL Interaction Centre for the collaboration on the data collection. I am lucky to have received unconditional support from my family, including Maria, Hans-Jürgen and Michael. This dates back to the inspiration from my grandfather "Opa" Paul, without his "Funkbud" I might have never chosen the path of computer science.

This thesis has been funded in part by the European Community Horizon 2020 [H2020/2014-2020] under grant agreement no. 645094 (SEWA) and by the EPSRC Emotion & Pain Project [EP/H016988/1].

Contents

Chapter 1

Introduction

Contents

Spontaneous facial expressions are a window to our inner feelings and thoughts. Charles Darwin already noted in 1872:

> "They (the expressions in the face and body) reveal the thoughts and intentions of others more truly than do words, which may be falsified." [34]

Facial expressions communicate emotions, clarify and stress what is being said, and signal comprehension, disagreement and stances [45]. It is not surprising then that machine understanding of human facial expressions could revolutionize the way we interact with computers, robots and cars; such technology would enable these artifacts to react properly when their users are tired, stressed and bored. Hence, machine understanding of facial expressions has recently become a hot research topic.

Most work on the topic to date focused on detection of the presence or absence of a certain facial expression (e.g., prototypic facial expression of happiness) or of a certain facial action (e.g., a smile, which is coded as AU12 in FACS [44]), instead on their full range intensity estimation [201]. Yet, the meaning and function of spontaneous facial expressions depends largely on their intensity. For example, the smiles of enjoyment are full-blown smiles, while the "fake happiness smiles" (as in sarcasm) may be asymmetric and are usually less in intensity when

observed in naturalistic social settings [48]. As noted in [70], "most of the smile genuineness impression is created by the intensity of the smile".

The importance of analyzing facial behavior intensity rather than in terms presence/absence is supported by the relevant research in psychology [45, 76, 77, 118]. That research has found that the intensity of spontaneous facial expressions are proportional to the intensity of underlying affective states, modulated by a particular social situation. For example, the vigor of spontaneous eye squints and brow scowls reveals the intensity of the felt pain [31]. Consequently, the intensity of people's affective and physiological states (e.g., pain) – which cannot be directly measured – can be effectively estimated from facial behavior estimation. In fact, the professional and scientific literature has demonstrated many limitations and biases of verbal self-reports, and great benefits of measures based on nonverbal facial behavior [30]. That research has found that a fine-grained coding of facial activity provides a more consistent mechanism for understanding biological, behavioral, cognitive, and social parameters of pain than self-reported verbal measures. This is currently the most prominent line of research in psychological and clinical studies of pain [31, 194]. This also explains why machine understanding of pain intensity from facial expressions would be beneficial in those studies.

Facial behavior and affect intensity can be either measured continuously or by discrete ordinal levels. Continuous measures are described by real-valued intensities (e.g., pain at level 2.7 on a scale from 0 to 3), while ordinal measures assign a value from a countable ordered set (e.g., pain at level 2, out of the levels 0, 1, 2 or 3). Each method has its individual advantages and disadvantages and both have been used in previous work. It has been shown that the perceived affective state is linearly related to the observed facial expression intensity [77], and thus a continuous intensity scale is better representing the affective state. Furthermore, continuous measures are more sensitive than discrete alternatives when labeled by human raters [181,192]. On the downside, continuous measures might exceed what scorers can reliably differentiate [31, 45] and the rating procedure is more time consuming [181]. In contrast to human observers, todays' computer vision algorithms can easily track and estimate very fine differences within the face (e.g., see [183]). Thus, harnessing the power of automated systems, in this work we focus on the more difficult but also more principled task of *continuous-valued* affect recognition, which has long been hoped for by psychologists [31].

Many available expression databases provide only discrete intensity levels and in these cases our regressor training is performed using discrete outputs (e.g., AU intensity labels from 0 to 5). However, during inference the regressors provide a continuous estimation of the intensity. The discrete levels are merely an artifact of the labeling procedure and are not justified from a

psychological point of view (see paragraph above). Therefore, we use a continuous regression model without discretization. Although the ground-truth is discrete, we find the best fit of a continuous function that passes through the discrete training data points. This means, that our model imitates a discrete function at the training data points, but interpolates for unseen data.

Suffering from pain is a common experience in human life. In the more severe cases, pain management is neccessary: either by medication (common for acute pain) or rehabilitative programs (common for chronic pain). However, pain management programs suffer from two shortcomings: (1) there are too few resources in the health care system to treat all patients face-to-face; (2) current approaches fail to integrate treatment of interrelated physiological and psychological factors. Consequently, the creation of methods for automatically recognising pain intensity could facilitate the development of an interactive computer system that will provide appropriate feedback and prompts to the patient based on his/her measured pain behaviour. In this work, we focus on pain intensity estimation from facial expressions, which have proven to be highly informative for the target task [194].

Most of the past work on facial expression recognition treats the observed facial features holistically rather than a sum of its part [201]. Yet componential facial emotion theory, which suggests that only components of facial expressions (facial actions) are universally displayed, and that only components of expressions play a role in facial expression interpretation, not full expressions [137, 163]. This explains further why humans can "fill in" the missing parts of an occluded facial expression and judge expressed emotional states even though just some facial actions are visible / present. Note that in this work, 'holistic' always refers to facial features and not to expressions.

In contrast to earlier work in machine understanding of facial expressions, we study spontaneous facial behavior in video for identifying the intensity levels of: (1) expression components, i.e., Facial Action Units (AUs) of the Facial Action Coding System (FACS) [44], in response to various emotions including pain, and (2) one prototypical expression, i.e., the facial expression of pain, in response to pain induction.

Since our goal is to identify the intensity of (1) and (2), our problem is that of continuous estimation of spontaneous facial behavior. This problem is challenging for a number of reasons. In general, spontaneous facial expressions are characterized by subtle, minimal facial deformations that are difficult to track, and frequent out-of-plane head movements whose effects are difficult to remove. For patients with pain, considered in this work, their facial expressions

are typically subdued, due to a long-term exposure to pain. Moreover, near-by intensity levels of emotional experience (or pain) are typically manifested by very small differences in facial expressions. All these challenges require a *fine-grained* approach which would be capable of identifying the most relevant facial details and their subtle movements for *continuous* facial behavior estimation.

We propose here a method for automated estimation of facial expression components, i.e., AUs. FACS defines 32 AUs, considered to be the smallest visually discernible facial movements directly related to contractions of the underlying muscles. FACS has been developed for human observers and it provides the rules for the recognition of these 32 AUs and their intensity, which is defined using the ordinal five level FACS model (A<B<C<D<E) [44]. This 5-level FACS model for AU intensity scoring is useful for human annotators who then do not have to depict finer differences between the intensity of the observed facial movements. But, as mentioned above, this model does not incorporate the continuous movement of facial muscles, which can be estimated from state-of-the-art computer vision algorithms. Hence, in this work, we approach the problem of FACS coding and pain estimation as a continuos-value estimation problem.

1.1 Problem Space

Although continuous valued estimation of facial expressions seems to be a promising line of research, many arising challenges are still unaddressed. The two main problems are the lack of available data and the insufficiency of current machine learning methods, which are specified below.

One is tempted to assume the data acquisition problem could be solved easily due to the ubiquity of facial expression display in everyday life. However it is non-trivial to record a genuine display of emotion with a non-occluded and stable view of the face in high quality. Additional to the recording apparatus and emotion elicitation procedure, it is necessary to obtain reliable annotations of the recorded data. Up to date, only few databases are available and even fewer provide intensity annotations (see Chap. 4). Machine learning algorithms crucially depend on data, and thus the provision of a new dataset for the research community is urgently needed.

As mentioned above, studies in psychology suggest that faces are interpreted by humans as the set of its parts, and also current description methods decompose facial expressions in terms of localized muscle movements (see Sec. 2.1). This is in contrast to most machine learning

methods, which treat the facial features holistically. New algorithms, that are able to focus on certain facial regions and interpret the face similar to humans as a joint part-based model (where each part is a facial feature), could provide improved performance and interpretability. The research goal can be summarized as obtaining a AU and pain intensity estimation method that (a) treats the face component wise and (b) produces outputs on a continuous scale.

1.2 Potential Applications

The applications for AU and pain intensity estimation cover many areas related to human behavior monitoring and human-computer interaction, such as healthcare, advertising, automotive industry, security and robotics. In the following, we describe some example applications.

Pain intensity estimation has the potential to improve healthcare practices. Patients in intensive care units need to be monitored, especially if they are unable to communicate. Currently, this is carried out by nurses. However, there is evidence of pain rating discrepancy between patients and physicians [114], which is caused by individual factors like gender and experience. Machines have the potential to rate pain in a standardized way, judging from either the patient's or the physician's perspective, depending on the training data. Additionally, the monitoring could be carried out over a longer time without interruption, instead of the nurse checking in intervals.

Apart from monitoring, pain intensity estimation could also be applied in more interactive scenarios, e.g., by building a 'virtual physiotherapist'. Patients suffering from chronic lower back pain are usually guided through exercises by a human physiotherapist. However, this is only feasible for a limited amount of time and then the patients need to continue with the exercises on their own. This is an opportunity for automated systems that aim to provide similar guidance as the physiotherapist [168]. Pain monitoring is one important task of the physiotherapist, in order to adapt further movement instructions or stop the exercise, depending on the pain level. While pain can be well estimated from facial behavior, the information could be further integrated with systems that monitor different cues, like non-verbal vocalizations and body gestures, see also Sec. 2.2.

Another promising area for AU intensity estimation is the improvement of advertising campaigns. McDuff et al. [124] showed that it is possible to detect ad preference from facial expressions. Intensity estimation could provide more fine grained feedback and subsequently interactive ads could be created that adapt to the subjects reaction. The companies Affectiva Inc. [1] and Realeyes OU [2] focus already on facial expression analysis for advertising and the

field is likely to grow.

The automotive industry is also starting to use facial expression recognition, specifically for monitoring drivers. Fatigue is a frequent cause of accidents and Horng et al. [79] showed the feasibility of fatigue detection from facial expressions. Automated systems have the potential to warn the driver or stop the vehicle to prevent accidents. Currently, Seeing Machines Ltd. [3] provides already fatigue and distraction detection for mining vehicles.

Further applications include security and surveillance, e.g., by recognizing threatening or dangerous situations by detecting related feelings and emotions like anger, fear or pain. Facial expressions can also reveal deception, and prototype recognition machines haven been build for border control and automated screening [50].

Human-robot interaction could be improved by building emotion-aware robots. One example is the TERESA project by Shiarlis et al. [165], where a telepresence system for interaction with elderly people is developed, that is able to understand emotional reactions.

1.3 Contributions

There are four main contributions presented in this thesis: (1) development of a baseline method for automatic estimation of pain / AU intensity based on existing regression techniques, (2) development of a new regression approach to the target problem that finds out informative facial parts and exploits that knowledge, (3) development of a new regression approach that combines part-based focus with joint target inference, and (4) creation of a new database. In what follows, each of the contributions is described in more detail.

Our initial baseline approach to AU and pain intensity estimation consists of three steps. In the first step, we extract shape-based features (i.e, locations of characteristic facial points) and appearance-based features (Local Binary Patterns (LBPs) [135] and Discrete Cosine Transform (DCT) [4]) from facial images of subjects displaying different intensities of pain. In the second step, for each set of features we train separate regression models (we employ Relevance Vector Regression (RVR) [174]) for prediction of the pain intensity levels. Finally, the outputs of the regressors trained using different feature sets are combined in two ways: (i) by computing the mean estimate of the regressors, and (ii) by using the outputs of separate regressors as an input to another RVR, which gives a single estimate for the pain intensity. The proposed approach is one of the first works that perform pain intensity estimation. Furthermore, we show that the proposed feature-fusion scheme outperforms the separately trained RVRs on different feature

sets, whereby the combination of appearance features (DCT and LBP) performs best. We also demonstrate the performance of the proposed approach in the task of continuous intensity estimation of the facial AUs. More details are provided in Chap. 7. The initial approach has been published as:

> S. Kaltwang, O. Rudovic, and M. Pantic. Continuous Pain Intensity Estimation from Facial Expressions. In volume 7432 of *Lecture Notes in Computer Science*, pages 368–377, Heidelberg, 2012. Springer.

We developed further a novel multiple-kernel regression approach to the target problem, called Doubly Sparse Relevance Vector Machine (DSRVM). DSRVM identifies the most relevant training examples of face snapshots – termed relevance vectors – which improve regression. Simultaneously, DSRVM also identifies the most informative parts of relevant training faces. To this end, DSRVM uses a bank of kernel functions and the selection of important facial parts is formalized as a selection of optimal kernel functions from the bank. To avoid overfitting, and reduce computation complexity, we regularize DSRVM to be twofold sparse in terms of both relevance vectors and kernels. DSRVM simultaneously learns multiple kernels within a probabilistic framework. This allows *computationally efficient* EM learning and *doubly sparse* solutions, where the learned DSRVM uses only a few kernels and a few relevance vectors. This advances related multiple-kernel learning (MKL) methods [66, 145, 147, 170]. They are typically specified within the max-margin framework, where enforcing sparsity in both primal and dual domains is computationally intractable, and thus requires approximations [206]. The existing MKL methods enforce sparsity only by selecting a few relevance vectors; however, the resulting number of relevant kernels can be prohibitively large.

We present empirical evaluation on a number of benchmark datasets. The experiments demonstrate many advantages of DSRVM, in comparison with competing approaches, in terms of higher accuracy and reduced computation complexity. Additionally, we show that the learned kernels correspond well with the AU region definitions and we are able to identify the important facial regions for pain recognition. Further details are provided in Chap. 8 and the following DSRVM paper has been accepted for publication:

> S. Kaltwang, S. Todorovic, and M. Pantic. Doubly Sparse Relevance Vector Machine for Continuous Facial Behavior Estimation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 2015 (to appear).

Our latest approach to the target problem is not only learning the important facial parts, but also estimates various AUs and facial expressions together. We consider a Bayesian generative framework and formalize our problem as that of jointly predicting multiple AU targets, given a set of image features. Every target can be defined as a vector of various attributes associated with each AU, and in a special case for our problem as AU intensities. Image features are defined as local descriptors of the face, which can be appearance based (e.g., patches) or locations of facial landmarks detected in a video frame. We specify a graphical model for representing the joint distribution of targets and features, and use the Bayes' rule to derive the AU intensities. Our formulation has a number of advantages over existing approaches [86, 121, 152, 158], which typically adopt the discriminative framework for directly predicting AU intensities given the features. While discriminative approaches are generally robust, we experimentally demonstrate that they underperform in challenging real-world situations. In particular, due to frequent partial occlusions of the face or large out-of-plane head movements in non-staged video, some input features might be missing or very unreliable. Our results show that our model can robustly handle missing input features by marginalizing them out, unlike the competing discriminative approaches. Also, our model is less likely to overfit to training human subjects, due to the joint modeling of all AUs and features.

For effectively capturing statistical dependencies among targets and features, our model organizes them in a tree structure with hidden variables, and hence we call our model Latent Tree (LT). LT structure is unknown a priori and we specify a new algorithm for efficient learning of both model parameters and model structure on training data. For AU intensity estimation, we derive closed-form expressions of posterior marginals of all variables in LT, and specify an efficient inference of the targets given the features.

We have evaluated LT on several benchmark datasets. In comparison with baselines and the state-of-the-art methods, the results demonstrate our superior performance, even under significant noise introduced to facial landmark points. We also demonstrate effectiveness of our structure learning by probabilistically sampling locations of facial landmark points, conditioned on a given AU intensity. Our generative sampling produces plausible facial expressions. More details can be found in Chap. 9 and the LT model has led to the following publication:

> S. Kaltwang, S. Todorovic, and M. Pantic. Latent Trees for Estimating Intensity of Facial Action Units. In *IEEE Conference on Computer Vision Pattern Recognition (CVPR'15)* IEEE, 2015.

In order to address the lack of data, we created the EmoPain database in collaboration with University College London and the University of Leicester. EmoPain contains multi-modal recordings of patients suffering from chronic lower-back pain performing several movement exercises. The data includes 7 camera views, 2 audio-channels, full body motion-capture, and 4 Electroencephalography (EEG) channels of the back muscles. Additional, the videos have been annotated by multiple observers in terms of the continuous pain intensity. The database is the first to include multi-modal chronic pain behavior. The author of this thesis contributed to EmoPain by recording the video and audio signals, setting up a system for synchronization between all modalities, as well as organizing the pain annotations. More details of the database are provided in Chap. 10. EmoPain will be available online[1] and has let to the publication of the paper:

> M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, M. Shafizadeh, A. C. Elkins, N. Kanakam, A. de Rothschild, N. Tyler, P. J. Watson, A. C. de C. Williams, M. Pantic, and N. Bianchi-Berthouze. The automatic detection of chronic pain-related expression: requirements, challenges and a multimodal dataset. *IEEE Transactions on Affective Computing*, 2015 (to appear).

1.4 Thesis Outline

The rest of the thesis is structured as follows. Chap. 2–6 provide an overview of the background, current work and prerequisites for the following novel work. Chap. 2 is focusing on the problem domain targeted by this work, i.e., facial expressions intensity estimation. Specifically, Facial Action Units (AUs) and expressions of pain are explained. Chap. 3 provides an overview of current AU and pain recognition frameworks, while explaining each of the included stages. Chap. 4 introduces the currently available databases, with a focus on the data used in this work. Chap. 5 explains the evaluation procedure to compare different intensity estimation methods. Chap. 6 describes the pre-processing for experiments within this thesis.

Chap. 7 describes our initial approach to pain and AU intensity estimation, using a variation of Relevance Vector Regression (RVR) for holistic facial expression recognition. Chap. 8 explains the novel Doubly Sparse Relevance Vector Machine (DSRVM) that is able to learn a sparse set of relevant local facial features for a specific regression task. The focus on local parts is continued with the Latent Tree (LT) model in 9, which is able to learn a hidden structure

[1] The EmoPain database will be available at www.emo-pain.ac.uk

that combines local facial features with multiple AU and pain recognition targets. Chap. 10 introduces the new EmoPain database for research on pain behavior. Finally, Chap. 11 concludes the thesis.

Chapter 2

Background

Contents

Facial expressions are a specific form of non-verbal communication, which also includes other forms like vocal intonations and body gestures. There are two main ways to describe facial expressions: *sign* and *judgement* based [23, 24] approaches. Both are grounded on the non-verbal communication model by Rosenthal [150], which assumes communication between two human entities, the subject and the observer. The subject experiences an internal state (e.g., pain or other emotions), which is expressed through external features (e.g., facial muscles). These features are then recognized and interpreted by the observer.

The judgement based approach to describe facial behaviour takes the role of the observer and how he interprets the expression. It tries to decode meaning, e.g., by assigning one of the six basic emotions [47] and/or emotion intensities. In contrast to that, the sign based approach analyzes the physical communication channel, i.e., the facial muscles. It analyzes how parts of the face move, e.g., raising of the brows or stretching of the mouth. The sign based approach is purely descriptive and thus leads to improved objectivity.

In the following, we focus on one sign based approach, the Facial Action Coding System, in Sec. 2.1, and one judgment based approach for recognizing pain in Sec. 2.2.

2.1 Facial Action Coding System

The Facial Action Coding System (FACS) [43] is a method for measuring facial movement and thus a sign based approach to describe facial expressions. To date, it is the most widely adopted, standardized and easy-to-use approach to describe facial movement [23]. Previous attempts were usually made up ad-hoc, not building on a common description standard [23].

FACS is based on the muscles that move the facial parts. Each human has the same muscles (with few exceptions) and thus each facial expression can be described as the sum of the muscle contractions. Most muscles cannot be moved independently, but rather move in groups. FACS uses the groups of facial muscles, that can be moved independently, as atomic building blocks, called Facial Action Units (AUs). Each group is assigned a unique number, e.g., the group of the inner brows is defined as facial action unit 1 (abbreviated as AU1). A list of the most common AUs is provided in Fig. 2.1. Each facial expression can be described as a well defined set of AUs, see Fig. 2.2 as an example for the expression of pain.

An exact description of each AU and a guide how to recognize it, is provided within the FACS manual [44]. Additional to the presence or absence of AUs, the manual defines intensity codings on a five point scale from A to E, which are commonly noted as a suffix of the AU identifier (e.g., AU 1A). The range of the different intensities are not covering the full expression range equidistantly, see Fig. 2.3. The low intensities A and B, as well as the highest intensity E cover small ranges, while the medium intensities C and D cover a large range. This means that most of the expressions are coded with C or D.

FACS coding requires profound knowledge of the manual and additional training by FACS experts in order to reach a high scoring accuracy, i.e., it is not possible to use naïve coders. Additionally, AUs are usually coded per image, i.e., a video needs to be coded for each frame separately, which is very time consuming. The limited availability of coders and high time demand makes it difficult to code large datasets.

2.2 Facial Expression of Pain

Pain is a human sensation which is subjectively interpreted, and thus its measuring is judgment based. In order to quantify the facial expression of pain, we first explain what is pain in general and then how to measure it based on facial expressions.

Pain is an inner feeling that attracts attention and focuses on escape, recovery and healing [194]. In order to describe pain, it is possible to use an extension to the non-verbal

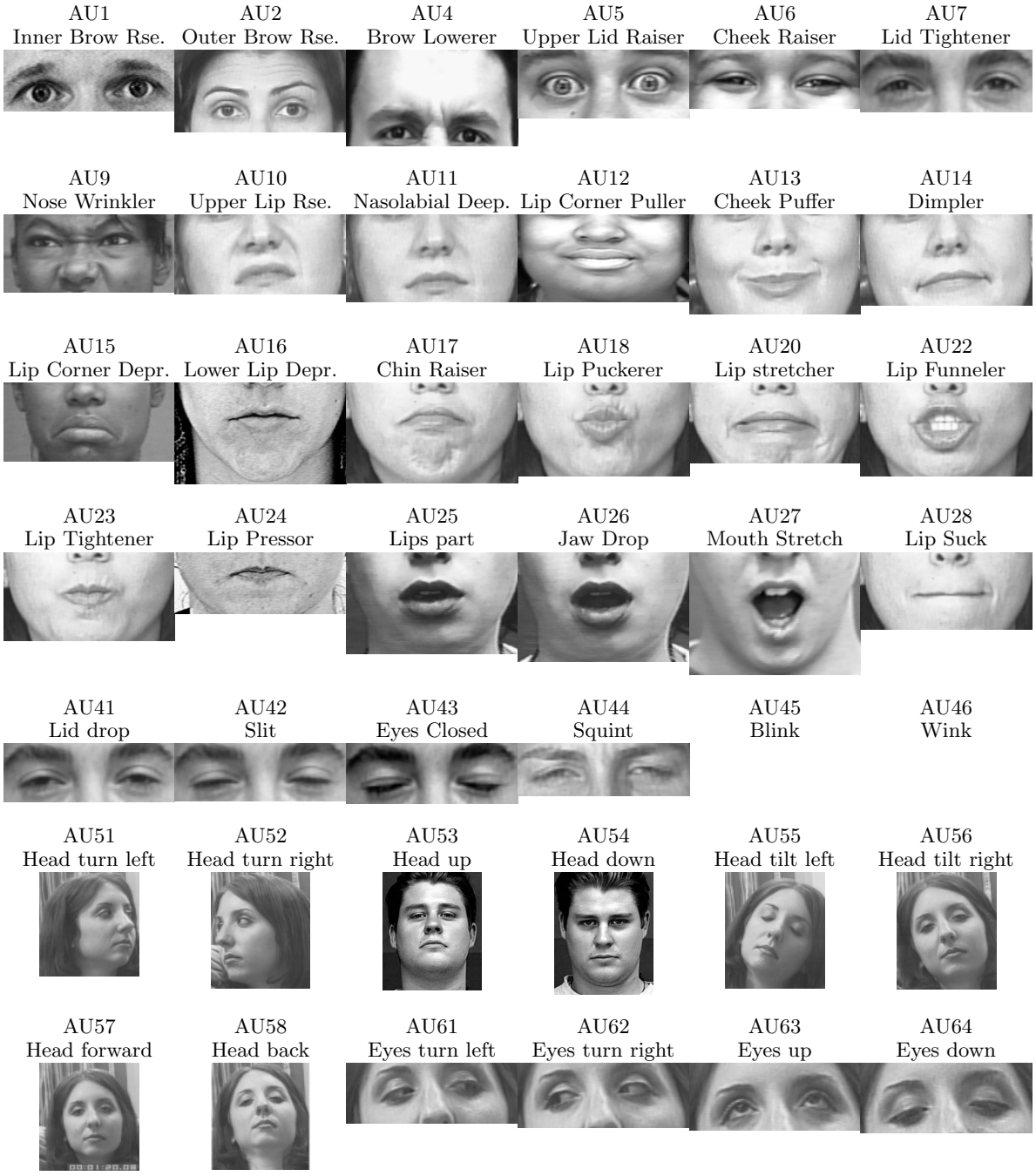


Figure 2.1: List of the most common Facial Action Units (AUs) including example images. (image source: www.cs.cmu.edu/~face/facs.htm, CK+ [107])

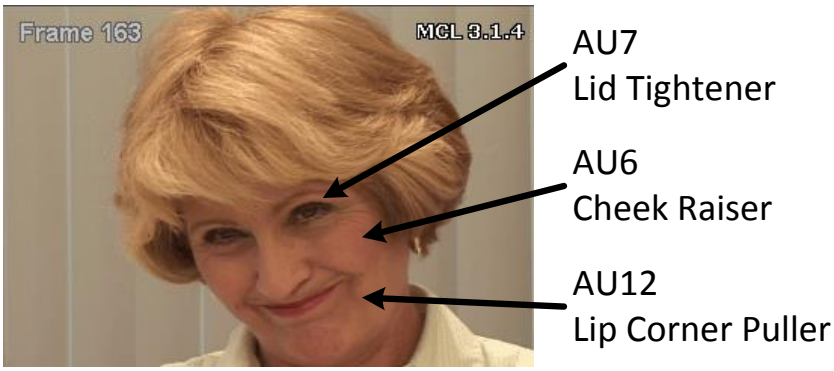


Figure 2.2: Subject showing the expression of pain with annotated AUs. (image source: ShoulderPain [109])

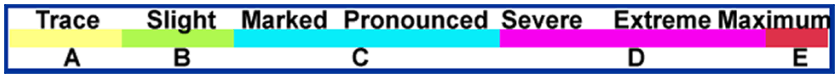


Figure 2.3: Visualization of the defined AU intensity range for the levels A to E as color bars. The bar symbolizes the range from the slightest muscle movement (level A, Trace) to the most extreme contraction (level E, Extreme Maximum). (image source: [44])

communication model of Rosenthal [150], developed by Prkachin and Craig [143]. The model is based on pain as an inner experience, which is encoded into external features. The features are then decoded by an observer. Pain can be encoded via facial expressions, non-verbal vocalizations, speech, body gestures and physiological measures (like heart-rate, EEG, fMRI). In contrast to the other cues, physiological measures cannot be naturally interpreted by human observers. In principle, pain could be recognized from any of these measures, but the focus of this work are facial expressions, which are unobtrusive and have proven to be highly informative regarding pain [143, 156, 194]. Previous work tried to recognize pain from physiological measures [12, 15, 130], but these methods are not automatized and the measurement procedure is intrusive.

According to the model, the inner pain sensation is influenced by (1) the pain stimulus (e.g., the amount of tissue damage, the voltage of electric shock), (2) intrinsic factors (e.g., age, mood, gender, personality) and (3) extrinsic factors (e.g., safe/dangerous environments, influence of drugs). All these factors influence the perceived pain and the communicated pain. The inner experience of pain and the outwards communicated pain are not the same and they might even be inversely correlated. For example, in a situation where relatives or potential helpful people are present, the subject expresses its pain stronger, in order to show the need for help. Conversely, the situation is less threatening because help is available and thus less pain is felt by the subject.

In order to measure pain, there are several possibilities: (1) measure the factors that influence the pain sensation (which include stimulus, intrinsic and extrinsic factors), (2) ask the subject, (3) ask an observer. Regarding (1), the factors can only partly be measured, e.g., by measuring the voltage of an electro-shock stimulus. However, often the stimulus cannot be quantified or compared with other stimuli. E.g. when the stimulus is the movement of a painful joint, then the pain depends on the severity of the medical condition and there is no obvious way to measure it. Other intrinsic factors like mood and personality are even harder to be quantified and thus do not lead to a reliable pain measure. Asking the subject is also inaccurate [30] and difficult for dynamic situations that require a continuous intensity measurement over time. Therefore this work focuses on measuring the pain reaction of the subject in terms of facial expressions, rather than measuring the pain cause. The pain reaction is measured by an observer. However, observer ratings are highly subjective as well. In order to gain a more reliable measure, it is possible to combine several observers and thus obtain a more robust and reproducible result.

As an objective alternative to subjective observer ratings, Prkachin and Solomon developed the Prkachin and Solomon Pain Intensity Scale (PSPI) [144]. The main idea is that all the information about pain from facial expressions should be included in the AU scores (see Sec. 2.1), since AUs provide a complete description of the face. They found out that brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43) encode most of the pain information. Following this insight, they defined pain intensity as the sum of the relevant AU intensities:

$$\text{PSPI} = \text{AU4} + max(\text{AU6}, \text{AU7}) + max(\text{AU9}, \text{AU10}) + \text{AU43} \tag{2.1}$$

Each AU intensity is in the range from 0-5, where 0 represents the AU absence and 1-5 corresponds to the levels A-E (see Fig. 2.3). Thus, the PSPI has a range from 0-15. Prkachin and Solomon show that PSPI correlates well with observer rated pain intensity levels [144].

The main advantage of PSPI is that the subjective part of the judgment-based pain rating is eliminated, and directly mapped to the sign-based FACS, making the results easily reproducible. On the downside, PSPI misses several factors that are relevant for pain rating. The survey of Williams [194] identified a larger set of AUs that all have been linked to the expression of pain and thus are likely to influence the intensity: AUs 4, 6, 7, 9, 10, 11, 12, 14, 20, 24, 25, 26, 27, 41, 43 and 45. PSPI ignores many of these AUs and only uses the subset that has the strongest relation with pain [111, 144]. The problem is that the correlation with pain has been tested for each AU separately, but co-occurrences of AUs have been ignored. Indeed, each separate AU would barely give any evidence for pain, since they are present in many other

expressions. However in combination, they are likely to contribute to the recognition of pain intensity. PSPI suffers from the same co-occurrence issue, since AU intensities are summed independently from each other. This can lead to major misinterpretations, since the same AUs occur in other expressions than pain: E.g. AU6 is commonly present in the expression of happiness, but according PSPI it would be scored as pain.

In summary, there are four possibilities to quantify pain, which are illustrated in Fig. 2.4. In this work, we focus on the observer rating and the objective alternative PSPI.

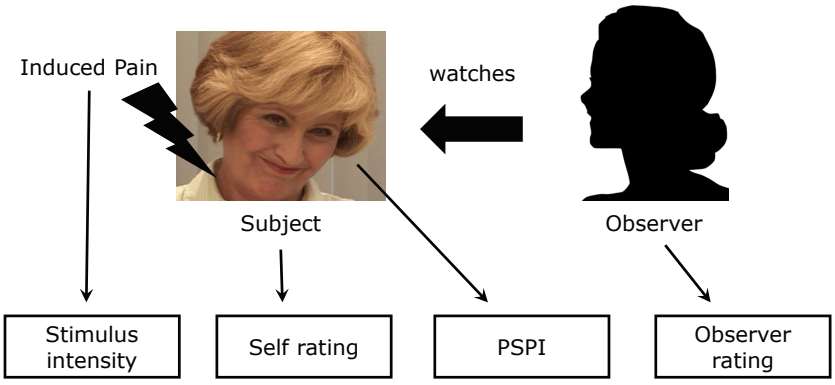


Figure 2.4: Overview of the different options to quantify pain.

Chapter 3

State-of-the-Art

Contents

This chapter reviews prior work on automatic detection and intensity estimation of facial expressions. Sec. 3.1 surveys video pre-processing methods, which are common to any recognition approach. The focus of this thesis is facial expression intensity estimation, which emerged out of earlier works on facial expression detection. Thus, we will provide a brief overview of the more established field on facial expression detection first in Sec. 3.2 and move to a detailed review for intensity estimation in Sec. 3.3.

Fig. 3.1 shows a generic overview of the typical stages needed for pain and AU recognition from facial expressions. The visualization is kept as generic as possible and some implementations leave stages out or have additional stages.

3.1 Pre-processing

A typical recognition system needs to pre-process the video input first before applying the actual detection or intensity estimation algorithm in the last stage. Typical pre-processing includes face detection, facial landmark localization, face registration, feature extraction and dimensionality reduction. The following sections review different pre-processing methods.

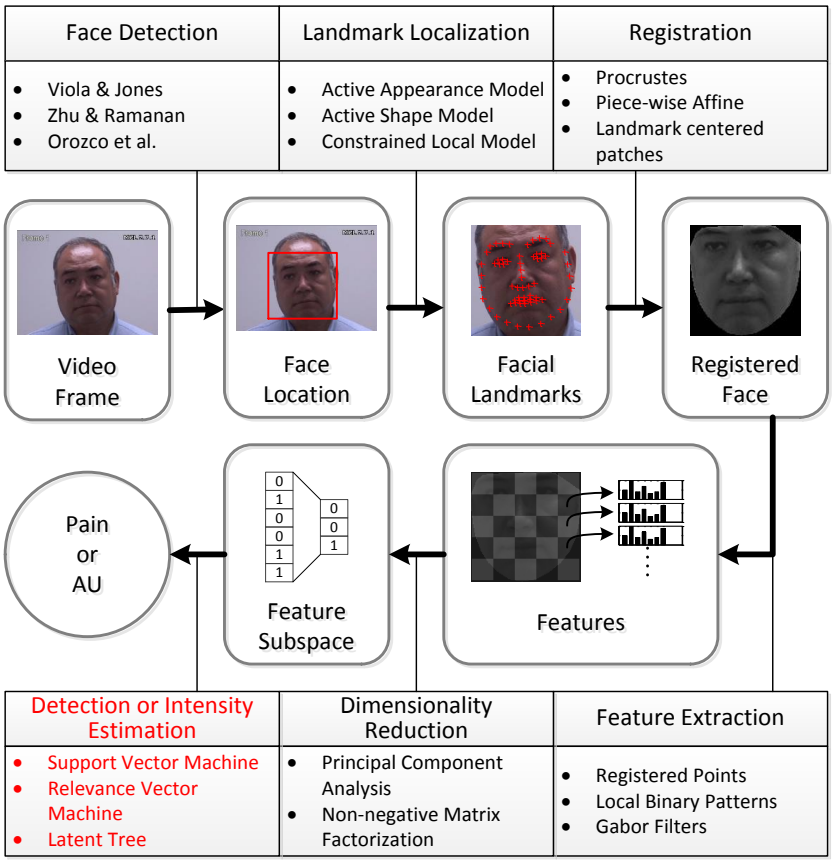


Figure 3.1: Generic overview of the typical stages within a pain and AU recognition pipeline. The rounded boxes visualize the occurring data types, starting from the input video frame and ending with the output pain or AU recognition. The tables describe the transition algorithms for transforming the data types. The table caption names the algorithm family and the lists below contain example instances. For implementation examples, see Tab. 3.1. For details of all pre-processing stages, see Sec. 3.1. For details on detection and intensity estimation, see Sec. 3.2 and 3.3. The last stage is highlighted, since the later parts of this thesis focus on intensity estimation algorithms.

3.1.1 Face Detection

A face detection algorithm localizes the coordinates of one or more faces within an image frame. This usually includes not only the face center, but also the bounding box. The bounding box provides additional information on how to rescale and crop the image for further processing.

Face detection is a relatively mature machine learning problem, i.e., many algorithms exist that solve the problem robustly and efficiently. The first widely adopted algorithm was presented by Viola and Jones [190], who use a cascade of AdaBoost classifiers and Haar features. The algorithm is targeted on frontal and near-frontal faces, but extensions for multiple views exist. Zhu and Ramanan [207] implemented Deformable Part Models (DPM) [54] to jointly detect the face, the head-pose and facial landmarks, even during non-frontal poses. Orozco et

al. [136] follow a similar idea by adopting DPM, but specifically optimize the model for face detection only. A comparative review of the state of the art is provided by Mathias et al. [117], who show that new variations based on the original Viola & Jones algorithm and DPM reach top performance in comparison with commercial and research systems.

3.1.2 Facial Landmark Localization

Facial Landmark Localization (FLL) algorithms detect a set of predefined facial points within either a single image or a sequence of images. The specific set of landmarks depends on the training data used, but usually includes points around the eyebrows, eyes, nose, mouth and the face contour. In case of an image sequence, the locations of the current landmarks can be initialized with the previous frame and thus the optimization is faster (also called "Tracking"). In contrast to face detection, current FLL algorithms still struggle with difficult cases, especially out-of-plane head rotations, changing lighting conditions and partial occlusions. Nevertheless, state-of-the-art methods provide good landmarks in controlled environments and are still usable in real-world conditions if the edge-cases are filtered out.

In the following, "shape" means a set of landmark points, where each point is either a 2D or 3D coordinate. This set of points defines the location of the main facial features, i.e., the "shape" of the face. In contrast to that, "appearance" means a set of image pixel intensities, usually arranged within a 2D rectangular grid.

One of the early successful approaches to FLL is the Active Shape Model (ASM) by Cootes and Taylor [26]. Most of the newer and more sophisticated approaches build on their success. ASM fits a learned Point Distribution Model (PDM) [27] to an image according to single point adjustments. The PDM is a parametric linear model for a set of points, learned from annotations. During inference, the ASM determines the optimal adjustment of each single point by a local search within the image, e.g., by finding the closest edge that is perpendicular to the shape. Then the ASM computes the optimal change for all points together according the PDM. These two steps are repeated until convergence. The underlying idea of iteratively refining shape (i.e., the PDM) and image (in this case, the distance of points to edges) constraints is used in many of the more recent methods as well.

The next major improvement has been achieved by Active Appearance Models (AAM) [25, 119]. As the ASM, the AAM uses a PDM for modelling the points. However, the image constraints are also implemented by a trained parametric linear model, the appearance model. It models pixel intensities within a reference shape, usually the mean shape of the PDM.

In order to compare the pixel intensities of the current shape and appearance instantiation with the actual image, the pixels within the current shape need to be transformed into the reference shape. This is usually done by a piece-wise affine warp (PWA), where each region of the triangulated mesh defined by the reference shape is warped separately. Image pixels are organized on a rectangular grid and therefore the warping between regions needs to interpolate between neighbouring pixels. The same PWA can also be used for face registration (see Sec. 3.1.3) and an example of a warped face is shown in Fig. 3.1 as "Registered Face". AAMs are holistic models, since the appearance and shape are fit jointly for the whole face.

Part-based models take a different approach and fit the regions of the face separately. Constrained Local Models (CLM) [160] use regressors to estimate the current image patch displacement to the target landmark. Then a response map is build from different patches and the final output is obtained by Regularized-Landmark Mean-Shift.

Valstar and Pantic [184] first detect facial point hypothesis for each frame using classifiers and locally extracted features. Then Particle Filtering with Factorized Likelihoods (PFFL) [141] is applied to get more reliable estimates over sequences of points. The disadvantage of this model is the estimation of each point from local image evidence only. Newer methods, like the Supervised Descent Method (SDM) [196], iteratively use regression and appearance features from all previous point estimates to get a more accurate point location.

An open-source framework of different FLL implementations and an extensive comparison of the state-of-the-art is provided by Alabort et al. [5].

3.1.3 Face Registration

Face Registration is an intermediate step to prepare the shape or appearance for further feature extraction. It aims on reducing the unwanted variation within the data that occurs due to varying face pose, varying camera position and varying anthropomorphic differences between subjects. Registration is achieved by finding a separate mapping for each face instance, which transforms the face into a common reference space.

The face shape is usually normalized regarding translation, rotation and scale, i.e., regarding a linear similarity transform. The most simple way is to select a set of fixed anchor points that define the linear transform and thus also the common reference space. A 2D translation, rotation and scale transform is fully defined by 2 anchor points, while 3 points are sufficient for a 3D transform. The selected points should not be influenced by facial expressions, and thus common choices are either the corner of the eyes or the eye centers. In case of 3D points,

the nose center could be added. Although these points are stable within the face, in practice this is not the case due to errors during landmark localization.

A more robust method is Procrustes analysis [68], which takes all points into account for alignment. Procrustes iteratively estimates the reference shape and the frame-wise alignment transform until convergence. The reference shape is initialized by the mean of all points and then iteratively updated with the mean of all aligned points. The transforms are obtained by minimizing the squared differences between the actual shapes and the mean shape.

Additionally to the shape normalization regarding a similarity transform, some methods normalize regarding the subject-specific neutral face shape by subtracting it from each frame.

The most common appearance registration methods are the piece-wise affine warp (PWA) and extracting pixel regions from patches centered around landmarks (LCP). PWA is the same warp applied by AAMs (see Sec.3.1.2), while using either the AAM reference shape as warping target, or a mean shape obtained by Procrustes. LCP simply defines patch regions around the landmark points where features are extracted from. In some cases, the same translation, rotation and scale transform obtained by Procrustes is applied to the pixels before extracting the patches.

3.1.4 Feature Extraction

In order to infer facial expressions, the used machine learning algorithm requires a vector containing information about the face as input. This vector is commonly called "feature". The registered shape or appearance could be directly used as features, but it is common to apply a feature transform. This transform usually incorporates desired properties, like lowering the dimensionality or invariances regarding illumination and rotation. We can differentiate between features extracted from the shape on one hand and appearance on the other hand.

The shape points (PTS) can be used as features directly, i.e., their (x, y) coordinates for 2D shapes and (x, y, z) coordinates for 3D shapes. Other possibilities include pairwise distances between points or the parameters of a learned PDM (see Sec. 3.1.2).

The appearance pixel intensities (PX) can be used as features as well. A face image with the resolution of 100×100 pixels would lead to a feature dimension of $10,000$, therefore the image is often downscaled to reduce the complexity.

Over the last years, a large set of appearance features has been developed by the research community, each with its own pros and cons. We describe here the subset that has been

commonly used for facial expression intensity estimation.

Gabor filter banks [61] are inspired by the primal visual cortex. A Gabor filter is defined by its orientation and frequency. Usually the filter response of an image regarding a set of different orientations and frequencies is used as features [52]. Essentially a Gabor filter is a robust edge detector, where the orientation and frequency specify the edge orientation and sharpness. Log-normal filters are similar to Gabor, but overcome some of their drawbacks, like the inseparability regarding orientation and frequency [116].

Discrete Cosine Transform (DCT) [4] features are based on a frequency analysis of the image and are the pendant to the Discrete Fourier Transform (DFT) used for processing signals like audio or EEG. The only difference is the cosine basis, which makes it more suitable for non-negative signals like pixel intensities. The DCT of an image results in an equally sized representation in the frequency domain and usually most of the higher frequencies are discarded. This leads to a low dimensional representation that is invariant to high frequency changes and thus can be used as holistic face descriptor. For a more extensive description and comparison of features based on image filters (like Gabor and DCT), please see Randen and Husoy [148].

Another group of appearance features is based on histograms of quantized local descriptors. A local descriptor uses the image intensities within a small neighbourhood, with only a few pixels in diameter. The quantized local descriptor response is accumulated over a larger image region within a histogram. This process discards spatial information and thus provides a compressed descriptor that is invariant regarding small translations.

One of the histogram based features is the Histogram of Gradients (HOG), which uses the intensity gradients as local descriptor. A detailed example implementation is provided in [19]. Another instance are Local Binary Patterns (LBP) [135], which use the sign of the intensity difference between the center pixel and circular surrounding pixels as local descriptor. Similar to that, the scale invariant feature transform (SIFT) [106] uses a weighted 3D histogram of gradient locations and orientations. For a more extensive description and comparison of local descriptors, please see Mikolajczyk and Schmid [128].

Any of the above described features can be applied in combination, also called feature fusion. We can differentiate between three fusion types: early, mid-level and late fusion. Early fusion is applied before the recognition algorithm by concatenating the feature vectors and treating the combination as a single feature. Mid-level fusion is applied during the recognition algorithm, i.e., the algorithm explicitly handles two or more sets of features as separate inputs while

estimating a single target output. Late fusion is applied after multiple recognition algorithms, i.e., a separate algorithm is applied for each feature and then their results are combined, e.g., by majority voting or averaging.

3.1.5 Dimensionality Reduction

Dimensionality reduction (DR) is an optional step after feature extraction. It aims at reducing the complexity of the data by removing irrelevant or redundant information. The extracted features can have many dimensions, sometimes more than several thousand. Some classifiers and regressors cannot handle that many dimensions, caused either by exceeding computational limits or through over-fitting, i.e., they suffer from the "curse of dimensionality". DR methods provide a mapping from the original features to a feature subspace, either by selecting a subset of dimensions or by mapping to a new space of reduced dimensionality. In the following, we provide an overview of the common DR methods.

A relatively simple DR method is Vector Quantization (VQ), which involves defining a template set of data vectors and then mapping each input vector to its closest template, i.e., the only information kept is the discrete template id. The set of templates can be either hand-crafted or learned by clustering algorithms, like k-means clustering.

One of the oldest and most studied DR methods is Principal Component Analysis (PCA) [85], which calculates a linear mapping into a space where all dimensions are uncorrelated. This is achieved by eigenvalue decomposition of the data matrix. Each of the new dimensions corresponds to an eigenvalue and thus the dimensions can be ranked according to the size of the respective eigenvalue. Usually only a subset of the dimensions with the largest eigenvalues are retained and the other ones discarded.

Newer matrix factorization methods also calculate linear projections of the data like PCA, but imposing different constraints. Independent Component Analysis (ICA) [80] constrains the new dimensions to be not only uncorrelated but statistically independent. This problem is not convex any more and thus needs to be solved by iterative methods. Non-negative matrix Factorization (NMF) [99] restricts all values to be greater or equal to zero and is thus well suited for pixel intensities. NMF provides a part-based decomposition of the data, i.e., most of the new component weights are zero. Both, ICA and NMF lead to sparse subspace weights.

Other DR methods are supervised and additionally includes the target information (like AU or pain). The goal is to find a feature subspace with highly discriminative information regarding the target. An early approach is Linear Discriminant Analysis (LDA) [55], which

finds a subspace that minimizes the within-class variance and maximizes the between-class variance of the targets. A more recent approach is Spectral Regression (SR) [17], which first performs spectral analysis on the Laplacian matrix, followed by learning a linear projection through least squares regression. The target information is encoded within in the Laplacian.

Instead of finding a new subspace, DR methods can also select a discriminative subset of the original dimensions. The Minimal-Redundancy-Maximal-Relevance Criterion (mRMR) [142] provides a ranking of the feature dimensions according to their mutual information (MI) with the target. Only a fixed number of top ranked features are used and the rest is discarded. Since the MI is difficult to calculate for continuous variables, an extension has been proposed based on the Pearson correlation coefficient [127].

Some algorithms perform DR jointly with the target prediction and thus it is possible to use these classification or regression methods for DR. E.g. AdaBoost [56] and GentleBoost [57] have been successfully used for DR, since they construct a classifier from a discriminative subset of features and thus this set can be used as DR subset.

An extensive comparative review of DR techniques is provided by Van der Maaten et al. [188].

3.2 Expression Detection

Detecting the presence or absence of facial expressions is a binary classification problem. This means that the ground-truth needs to be binarized for databases with annotated intensities (see Sec. 4). Usually the AU and pain intensity level of zero is defined as absence and levels greater than zero as presence.

To date, few works have addressed the problem of automatic pain detection [14, 63, 105, 108, 109]. Brahnam et al. [14] used Principal Component Analysis, Linear Discriminant Analysis and Support Vector Machines (SVMs) for binary classification of pain images (i.e., pain vs. no pain). Gholami et al. [63] used intensities from facial images to train a Relevance Vector Machine (RVM) classifier for pain detection. Littlewort et al. [105] proposed a two-layer SVM-based approach for the classification of image sequences in terms of real pain and posed pain. In their approach, the presence of Facial Action Units (AUs) (see [43] for AU description) per frame is detected with a set of AU-specific SVM classifiers based on Gabor features. The outputs of the AU-specific SVMs are then temporally filtered and used as an input to the SVM classifier. The work by Lucey et al. [109] also addresses AU and pain detection based

on SVMs. They detect pain either directly using image features or by applying a two-step approach, where first AUs are detected and then this output is fused by Logistical Linear Regression in order to detect pain.

3.3 Expression Intensity Estimation

Inferring the intensity of facial expressions involves predicting either 3 or more discrete levels or fully continuous values. Most detection algorithms can easily be extended to multi-class classification and thus the commonly used methods for inferring discrete intensities are similar to their detection counterparts. In contrast to that, continuous intensity estimation requires regression algorithms, which mainly differ in the optimized loss or target probability distribution. Regression algorithms are also better suited for discrete intensities, if there are only few training samples available per intensity.

Recognition models can be divided into two groups depending on their temporal inference. Static models perform inference on time points of fixed length and each point is treated independently of the others. A time point can be either a single video frame or a pre-defined window of frames. In contrast to that, dynamic models perform inference over series of time points, where the dependence of consecutive points is modeled explicitly.

For an overview of Pain and AU intensity estimation methods, see Tab. 3.1. In the following, we first provide an overview of the static methods for facial expression intensity estimation and then describe the dynamic models.

Table 3.1: Overview of Pain and AU intensity estimation methods. For database descriptions, please see Sec. 4. The landmark localization (Land. Local.) is shown in brackets if it was performed by the database creators rather than the paper authors. Methods that perform pain and AU intensity estimation are listed twice. The publications highlighted in bold resulted from of this thesis.

Pain Intensity Estimation

Name	Year	Database	Land. Local.	Registration	Features	Dim. Reduction	Recognition	Type	Time	Levels	Measures	Subj. indep.
Hammal and Cohn [74]	2012	ShoulderPain	AAM	Procrustes, PWA	PTS, LNF	-	SVC	class.	stat.	4	Acc, ICC	both
Kaltwang et al. [86]	2012	ShoulderPain	(AAM)	Procrustes, PWA	DCT, LBP, PTS	-	RVR	regr.	stat.	16	CORR, MSE	yes
Rudovic et al. [151]	2013	ShoulderPain	(AAM)	PWA	LBP	-	CORF	regr.	dyn.	6	ICC, MAE	yes
Zafar and Khan [199]	2014	ShoulderPain	PFFL	-	PTS	-	k-NN	class.	stat.	16	MSE	no
Rudovic et al. [152]	2015	ShoulderPain	(AAM)	Procrustes	PTS	PCA	CORF	regr.	dyn.	6	ICC, MAE	yes

AU Intensity Estimation

Name	Year	Database	Land. Local.	Registration	Features	Dim. Reduction	Recognition	Type	Time	Levels	Measures	Subj. indep.	AUs
Fasel and Luettin [51]	2000	(non-public)	manual	-	PX	ICA, PCA	k-NN	class.	stat.	6	Acc	no	16
Bartlett et al. [10]	2006	(non-public)	-	Eye enters	Gabor	AdaBoost	SVC	class.	stat.	2	CORR	yes	8
Mahoor et al. [112]	2009	(non-public)	AAM	Procrustes, PWA	PX, PTS	SR	SVC	class.	stat.	6	ICC	no	2
Hamm et al. [73]	2011	(non-public)	ASM	Procrustes	PTS, Gabor	-	AdaBoost	class.	stat.	2	-	no	15
Jeni et al. [82]	2012	BU-4DFE, CK-Enh.	CLM	Procrustes	PTS	-	SVR	regr.	stat.	3	-	no	14
Kaltwang et al. [86]	2012	ShoulderPain	(AAM)	Procrustes, PWA	DCT, LBP, PTS	-	RVR	regr.	stat.	6	CORR, MSE	yes	10
Savran et al. [162]	2012	Bosphorus	manual	-	Gabor, 3D shape	AdaBoost	SVR	regr.	stat.	6	CORR	yes	25
Mavadati et al. [121]	2013	DISFA	AAM	LCP	LBP, HOG, Gabor	SR	SVC	class.	stat.	6	Acc, ICC	no	11
Li et al. [102]	2013	DISFA	(AAM)	LCP	HOG, Gabor	SR	SVC+DBN	class.	dyn.	6	ICC	no	11
Jeni et al. [81]	2013	BP4D-Spont., CK-Enh.	CLM	LCP	PX	NMF	SVR	regr.	stat.	3, 6	CORR, MSE	yes	15
Sandbach et al. [158]	2013	DISFA	(AAM)	Eye, Nose centers	LBP	GentleBoost	SVR+MRF	regr.	stat.	6	CORR, MSE	yes	6
Baltrušaitis et al. [8]	2014	DISFA	(AAM)	LCP, Eye centers	PTS, PX	NMF	CCNF	regr.	dyn.	6	CORR, MSE	yes	12
Mavadati et al. [122]	2014	DISFA	(AAM)	LCP	Gabor	VQ	HMM	class.	dyn.	6	Acc, ICC	both	12
Zafar and Khan [199]	2014	ShoulderPain	PFFL	-	PTS	-	k-NN	class.	stat.	6	Acc, MSE	no	6
Zhang et al. [203]	2015	Bosphorus	AAM	-	PTS	mRMR	NNet, SVR	regr.	stat.	6	CORR, MSE	no	16
Rudovic et al. [152]	2015	DISFA, ShoulderPain	(AAM)	Procrustes	PTS	PCA	CORF	regr.	dyn.	6	ICC, MAE	yes	15
Kaltwang et al. [87]	2015	DISFA, ShoulderPain	(AAM)	Procrustes	PTS	-	LT	regr.	stat.	6	CORR, ICC, MSE	yes	15
Girard et al. [64]	2015	BP4D-Spont.	prop.	LCP	Gabor, SIFT	SR, PCA	SVC, SVR	both	stat.	6	ICC	no	1
Valstar et al. [186]	2015	BP4D-Spont.	CLM	Eye, Nose centers	PTS, LBP, Gabor	PCA	SVR	regr.	stat.	6	CORR, ICC, MSE	yes	5

Abbreviations: **Landmark Localization (Land. Local.)** – AAM: Active Apparance Model, ASM: Active Shape Model, CLM: Constrained Local Model, PFFL: Particle Filtering with Factorized Likelihoods; prop.: proprietary software; **Registration** – LCP: landmark centered patches, PWA: Piece-wise Affine; **Features** – DCT: Discrete Cosine Transform, HOG: Histogram of Gradients, LBP: Local Binary Patterns, LNF: Log-normal Filters, PTS: Landmark Points, PX: Pixel Intensities, SIFT: Scale Invariant Feature Transform; **Dimensionality (Dim.) Reduction** – ICA: Independent Component Analysis, mRMR: Minimal-Redundancy-Maximal-Relevance Criterion, NMF: Non-negative Matrix Factorization, PCA: Principal Component Analysis, SR: Spectral Regression, VQ: Vector Quantization; **Recognition** – CCNF: Continuous Conditional Neural Fields, CORF: Conditional Ordinal Random Field, DBN: Dynamic Bayesian Network, HMM: Hidden Markov Model, k-NN: k Nearest Neighbour, LT: Latent Tree, MRF: Markov Random Field, NNet: Neural Network, RVR: Relevance Vector Regression, SVC: Support Vector Classification, SVR: Support Vector Regression; **Time** – stat.: static, dyn.: dynamic; **Measures** – Acc: Classification Accuracy, CORR: Pearson Correlation Coefficient, ICC: Intra-class Correlation Coefficient, MAE: Mean Absolute Error, MSE: Mean Squared Error; **Subject independent (Subj. indep.)**.

3.3.1 Static Models

Fasel and Luettin [51] presented an early work on AU intensity estimation based on a proprietary database, manually annotated landmarks, PCA and ICA features from pixel intensities and the K nearest neighbour classifier (k-NN) [29]. k-NN is one of the most simple classifiers. It assigns the target class by taking a majority vote over the k nearest training samples within the feature space. The features contain still images from a single subject only showing posed expressions in a very restricted environment that ensures the head stays at the same place. k-NN has been recently used by Zafar and Khan [199] as well for AU and pain intensity estimation. They use PFFL tracked landmarks on ShoulderPain with point features.

One of the most widely used static models is the Support Vector Machine, which can be used for classification (SVC) [189] or regression (SVR) [41]. It has been frequently used for AU and pain intensity estimation [10, 74, 81, 82, 102, 112, 121, 158, 162, 203]. SVC is a max-margin classifier, i.e., it learns the decision boundary by maximizing the margins between the classes. SVR maps regression to a classification problem, by defining a tube around the target function as the correct class and then applying the same max-margin framework as SVC. SVC/SVR solutions are sparse, since only data samples that lie within the margin contribute to the solution. When combined with a kernel (e.g. Gaussian kernel), then the decision boundary is obtained in kernel space instead of the feature space, and thus each of the dimensions corresponds to a training data sample. Solutions are sparse in kernel space as well, i.e., the weight of most training data samples will be zero.

SVC was first used for recognizing AUs by Bartlett et al. [10] on non-public data. They extracted a large set of Gabor features and selected the most informative ones by AdaBoost. Their method was trained for AU detection, but they showed that the intensities are correlated with the distance from the SVC margin. SVC was also used by Hammal and Cohn [74] for one of the first methods on pain intensity estimation. They tracked AAM landmarks on ShoulderPain and used log-normal filter features to recognize 4 discrete pain levels.

Mahoor et al. [102, 112, 121] used supervised Spectral Regression (SR) and SVC for AU intensity estimation. The early work [112] is based on shape and appearance features while using a proprietary database containing facial expressions of infants and mothers engaging in face-to-face interaction. The following work [121] has been done in conjunction with the DISFA database release and focuses on various appearance features. Additionally, it has been extended to a dynamic model [102], see the section below.

Jeni et al. [81, 82] tracked landmarks with CLMs and used 3D shape [82] and appearance

features followed by NMF [81] on the CK-Enhanced, BU-4DFE and BP4D-Spontaneous data.

Zhang et al. [203] compares SVR with Artificial Neural Networks (NNet) [72] on the Bosphorus data using AAM tracked point features and mRMR feature selection. NNets are organized in layers of numeric variables, called 'neurons'. Calculation of the target prediction starts with an input layer for the features, followed by one or more hidden layers and finishes with a layer of one or more targets. The values of each layer depend on the previous layer through a parameterized transition function, which commonly consists of a linear function of all previous neurons followed by a non-linearity (e.g. sigmoid or rectifier). The weights are usually optimized on the training data by a dynamic-programming gradient-decent, also called back-propagation.

Savran et al. [162] applied Adaptive Boosting (AdaBoost) [56] feature selection and SVR to Bosphorus 3D data, combining 3D shape and Gabor wavelets. AdaBoost, and its improved version GentleBoost [57], learn an ensemble of weak classifiers. Weak classifiers are often simple linear classifiers that only depend on a subset of the features. AdaBoost defines a learning routine, that iteratively improves the combined result of weak classifiers, by emphasizing previously misclassified training samples.

Hamm et al. [73] used ASM tracking, shape and Gabor features and AdaBoost to infer AU intensities within a proprietary facial expression database showing neuropsychiatric patients and controls. Their work focuses on the psychology side and shows qualitative results only.

Sandbach et al. [158] applies LBP features, GentleBoost feature selection and a Markov Random Field (MRF) [91] to the DISFA data for inferring upper face AU intensities. A MRF is a graph consisting of nodes and edges, where each node corresponds to a random variable and each edge corresponds to a parameterized potential function. The potential is a non-normalized distribution and thus induces a distribution on the connected nodes (which can be derived by calculating the normalization constant). Nodes follow the Markov property, i.e. the distribution of each node only depends on its neighbors. Various algorithms for inference and parameter optimization have been developed, see e.g., [101]. Commonly inference is only exact for tree structured models, if the model contains loops we need to resort to approximate algorithms. Sandbach et al. [158] learns a tree structured MRF from the AU distributions. Through the MRF, all AU intensities are estimated jointly. This is different from most other approaches, which train a separate model for each AU.

Girard et al. [64] analyzes different ways to apply SVC and SVR to infer smile intensity, i.e., AU12, on the BP4D-Spontaneous and non-public Spectrum databases. They found that

regression is better than multi-class classification, and the distance from the decision boundary of a binary classifier performed worst.

Recently, the second facial expression recognition and analysis challenge (FERA 2015) [186] included AU intensity estimation as a sub-challenge. The challenge uses an extension of the BP4D-Spontaneous data, which includes intensity estimation of AUs 6, 10, 12, 14 and 17. Baseline results have been reported using SVR and PTS, LBP and Gabor features. At the time of submission of this thesis, the participating methods have not been published yet.

3.3.2 Dynamic Models

Only few dynamic models for AU and pain intensity estimation exist. Mavadati et al. [122] proposed Hidden Markov Models (HMM) [146] applied to quantized Gabor features from the DISFA data. HMMs are probabilistic generative graphical models, that generate the observed data from a singe hidden variable at each time frame. The hidden variables are connected with each other and fulfill the markov property, i.e., they are only connected to their immediate predecessor or successor in time. HMMs are a specific type of MRF and exact inference is achieved by the forward-backward algorithm. Parameters are usually optimized by EM, where forward-backward is the E-step. Li et al. [102] extended their earlier static model [121] based on SVC by adding a Dynamic Bayesian (DBN) [131] on top. DBNs are a generalization of HMMs that allow multiple hidden variables per time-frame and thus all AU intensities are inferred together. Several works [8, 151, 152] have been proposed that use extensions of Conditional Random Fields (CRF) [95] for temporal inference. CRFs are similar to HMMs, the main difference is their discriminative nature instead of generative. I.e. the model is conditioned on the observed features, but their distribution is not explicitly modeled. Baltrušaitis et al. [8] extend CRFs to Continuous Conditional Neural Fields (CCNF) that replace the linear factor of the observed features with an artificial neural network. The CCNF is applied to shape and NMF appearance features on the DISFA data for AU intensity estimation. Rudovic et al. [151, 152] extended the Conditional Ordinal Random Field (CORF) [90] to incorporate heteroscedastic noise [151] and context modeling [152]. CORF is a CRF extension for ordinal regression, i.e., the discrete AU intensity classes can be modeled according their ordinal relations. The experiments include AU intensity estimation on DISFA as well as pain intensity estimation on ShoulderPain.

Chapter 4

Databases

Contents

This chapter provides an overview of the publicly available databases for AU and pain recognition. First, we introduce the databases mainly used in our experiments: ShoulderPain (Sec. 4.1) and DISFA (Sec. 4.2). Then we describe other available databases in Sec. 4.3.

4.1 ShoulderPain

The *UNBC-McMaster Shoulder Pain Expression Archive Database* (ShoulderPain) [109] contains face videos of patients suffering from shoulder pain while performing range-of-motion tests of their arms. The participants were recruited from 3 physiotherapy clinics and from the McMaster University campus. The inclusion criterium was self-identification with shoulder pain, which included different medical conditions such as arthritis, bursitis, tendonitis, subluxation, rotator cuff injuries, impingement syndromes, bone spur, capsulitis and dislocation. Two different movements are recorded: (1) the subject moves the arm himself, and (2) the subject's arm is moved by a physiotherapist. Only one of the arms is affected by pain, but movements of the other arm are recorded as well as a control set. 200 sequences of 25 subjects were recorded (in total 48,398 frames). For each frame, the intensities of pain related AU's 4, 6, 7, 9, 10, 12, 20, 25, 26, 27 and 43 are provided on a 0-5 discrete intensity scale, except for AU 43, which is binary. The number of frames available per AU intensity level is shown in Table 4.1. The AU labels were obtained by one of three certified FACS coders and an

inter-observer agreement of 95% (according the Ekman-Friesen formula [44]) was reached on a small subset of the data which was labeled by all three coders. Additionally to the AU

Table 4.1: ShoulderPain: The number of video frames for each AU and its intensity level from 0 to 5.

Intensity	0	1	2	3	4	5
AU4	47324	202	509	225	74	64
AU6	42841	1776	1663	1327	681	110
AU7	45034	1360	991	608	305	100
AU9	47975	93	151	68	76	35
AU10	47873	171	208	63	61	22
AU12	41511	2145	1799	2158	736	49
AU20	47692	286	282	118	0	20
AU25	45992	766	803	611	138	88
AU26	46306	430	918	265	478	1
AU27	48380	6	3	3	6	0
AU43	45964	2434	-	-	-	-

annotations, the database creators provide discrete pain intensities according to Prkachin and Solomon method [144] (see also Sec. 2.2). The pain intensities are quantified into 16 discrete levels (0 to 15) and their distribution is shown in Fig. 4.1.

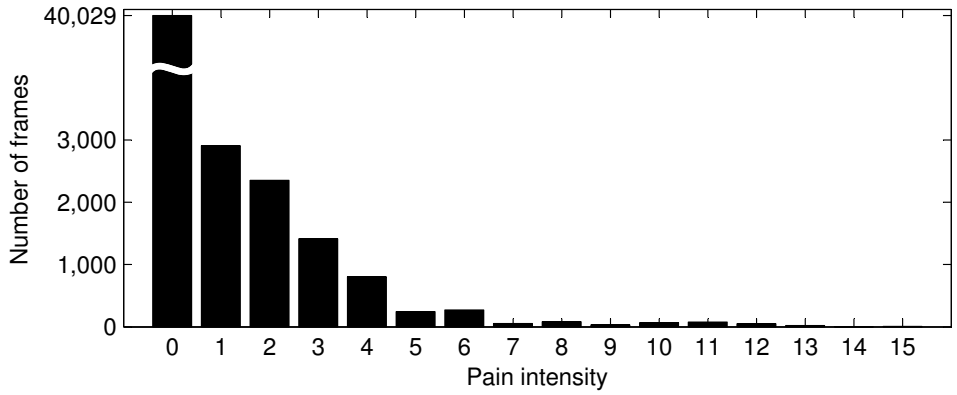


Figure 4.1: ShoulderPain: Frame distribution over pain intensity levels

We chose the ShoulderPain, since it is the only database that focuses solely on pain and that provides detailed pain intensities per frame. Some other databases also include pain expressions (see the following part), but only for few sequences and detailed annotations are missing.

The imbalanced pain intensity distribution can influence model training as well as the evaluation scores. In order to better balance the different pain intensities during model training, some of the work on ShoulderPain [74, 151, 152] aggregates pain levels. This is important for

algorithms that model each intensity as a separate class, since otherwise there are almost no training samples for certain classes. Since we treat pain intensity estimation as a continuous regression problem, the imbalanced data does not pose problems during training. However, due to the lack of data, the model accuracy will be relatively low for high intensity levels.

The evaluation scores are also influenced by imbalanced data. All metrics CORR, MSE and ICC, are second order measures, i.e. the influence of the error from single data samples increases quadratically with the distance from the overall error mean. This means that higher pain intensities (which are prone to a higher error) have a greater influence on the score and thus the evaluated models are heavier penalized for missing the high intensities. Therefore the aggregation of higher levels would probably increase the performance score. However, since the higher levels are rare, the influence on the score is limited. Furthermore, since there are many possibilities to aggregate levels and each of them will lead to different scores, we believe the best method is to keep the data processing as simple as possible and thus without aggregation.

4.2 DISFA

The *Denver Intensity of Spontaneous Facial Action Dataset* (DISFA) [121] contains spontaneous facial expressions of subjects while watching a stimulating video. The stimulus video contains 9 short clips from youtube which are related to the emotions happiness, surprise, fear, disgust and sadness. The participants were 27 adults with an age between 18 and 50 years. The face of each subject was recorded with a resolution of 1024x768 pixels, and a frame-rate of 20 frames-per-second, resulting in a total number of 130,754 frames. Each of these frames has been annotated with AU's and their corresponding intensity on a 0-5 discrete scale by a single expert FACS rater. In order to validate the inter-observer reliability, 10 randomly selected videos were annotated by a second FACS rater and the inter-rater ICC for different AUs ranges from 0.80 to 0.94. The following AU's are annotated: 1, 2, 4, 5, 6, 9, 12, 15, 17, 20, 25, and 26. The intensity distribution for each AU is shown in Tab. 4.2. In addition to the annotations, the database creators provided 66 active-appearance-model (AAM) tracked facial landmark points.

We chose DISFA, since it is one of the few naturalistic databases which provide per-frame annotated videos for 12 AU intensities. Other databases only contain very few AUs (BP4D-Spontanous) or only posed facial expressions, see Tab. 4.3.

Table 4.2: DISFA: The number of video frames for each AU and its intensity level from 0 to 5.

Intensity	0	1	2	3	4	5
AU1	112286	2272	1749	2809	1393	555
AU2	99165	1720	934	3505	836	369
AU4	106160	4661	7636	6586	4328	1383
AU5	99015	1579	719	293	104	34
AU6	106425	9157	5986	3599	601	141
AU9	99458	1659	2035	3045	316	77
AU12	99987	13943	6869	7233	2550	172
AU15	108358	5180	1618	1017	47	0
AU17	117824	6342	4184	2281	112	11
AU20	121377	1591	1608	1305	28	0
AU25	84721	9805	13935	15674	5580	1039
AU26	105778	13443	7473	3529	314	217

4.3 Other Databases

Besides ShoulderPain and DISFA, there exist more facial expression databases. This section provides an overview of all publicly available AU annotated databases and selected data with continuous facial expression annotations. Since the AU annotation process is time consuming and requires trained experts, only few databases exist. Furthermore, many databases do not have intensity annotations and omit a subset of AUs. Databases that are not publicly available are left out. Tab. 4.3 provides an overview of all databases, for more details see the respective section. Some databases are only partly annotated, thus all statistics shown here describe the part of the database that has AU or pain annotations. We differentiate within 4 database categories: Pain (Sec. 4.3.1), detection of AUs (Sec. 4.3.2), intensity estimation of AUs (Sec. 4.3.3) and intensity estimation of other expressions (Sec. 4.3.4). Obviously, all databases for AU intensity estimation can also be used for AU detection. Note that BP4D-Spontanous is explicitly listed in both categories, since only a subset of the AUs for detection is annotated with intensities.

4.3.1 Pain

The *High Resolution 4 Dimensional Database from the Applied Digital Signal and Image Processing Research Centre* (Hi4D-ADSIP) [120] is besides ShoulderPain (see Sec. 4.1) and BP4D-Spontanous (see Sec. 4.3.2) the only database that contains the facial expression of pain. It consists of 3360 sequences from 80 subjects recorded at 60 fps showing different facial expressions. Each frame within the sequence is a 3D triangular mesh covering 180° of the frontal side of the head. Each subject performs 7 facial expressions (anger, disgust, fear happiness,

Table 4.3: Overview of the facial expression databases. The table shows the abbreviated name (Name), the number of ground-truth intensity levels (Levels), the release year (Year), the number of included subjects (#Subjects), the time capturing type (Time), the naturalness of the expression (Expression) and the number of annotated AUs (#AUs). Levels can be either binary (bin.), an integer or continuous (cont.). Time can be either static for images (Stat.) or dynamic for videos (Dyn.). Expression can be either posed or spontaneous (Spont.). ShoulderPain appears twice, since it is annotated for pain and AUs. BP4D-Spontanous appears three times, since the intensities are only available for a subset of the binary annotated AUs and the data contains pain as well. The database highlighted in bold resulted from of this thesis.

Pain (see Sec. 4.3.1)

Name	Levels	Year	#Subjects	Time	Expression
BP4D-Spontanous [202]	bin.	2013	41	Dyn.	Spont.
EmoPain [7]	cont.	2015	21	Dyn.	Spont.
Hi4D-ADSIP [120]	4	2012	80	Dyn.	Posed
ShoulderPain [109]	16	2011	25	Dyn.	Spont.

Detection of AUs (see Sec. 4.3.2)

Name	Levels	Year	#Subjects	Time	Expression	#AUs
AM-FED [123]	bin.	2013	242	Dyn.	Spont.	10
BP4D-Spontanous [202]	bin.	2013	41	Dyn.	Spont.	27
CASME [198]	bin.	2013	35	Dyn.	Spont.	all
CASME II [197]	bin.	2014	26	Dyn.	Spont.	all
CK [88]	bin.	2000	97	Dyn.	Posed	all
CK+ [107]	bin.	2010	26	Dyn.	Spont.	8
D3DFACS [28]	bin.	2011	10	Dyn.	Posed	all
GEMEP-FERA [187]	bin.	2011	10	Dyn.	Posed	12
ISL Frontal [177]	bin.	2007	10	Dyn.	Posed	14
ISL Multi-View [176]	bin.	2010	8	Dyn.	Posed	15
MMI (Part I-III) [138]	bin.	2005	210	Dyn.	Posed	all
MMI (Part IV-V) [185]	bin.	2010	25	Dyn.	Spont.	all
SAL [40]	bin.	2008	20	Dyn.	Spont.	10

Intensity Estimation of AUs (see Sec. 4.3.3)

Name	Levels	Year	#Subjects	Time	Expression	#AUs
Bosphorous [161]	6	2008	105	Stat.	Posed	25
BP4D-Spontanous [202]	6	2013	41	Dyn.	Spont.	2
CK-Enhanced [177]	3	2007	97	Dyn.	Posed	14
DISFA [121]	6	2012	27	Dyn.	Spont.	12
ICT-3DRFE [171]	cont.	2011	23	Stat.	Posed	all
ShoulderPain [109]	6	2011	25	Dyn.	Spont.	11

Intensity of Other Expressions (see Sec. 4.3.4)

Name	Levels	Year	#Subjects	Time	Expression
SEMAINE [126]	cont.	2012	150	Dyn.	Spont.

sadness, surprise and pain) with 3 intensities (mild, normal and extreme). The subjects were instructed to show the expression and thus the data is posed rather than spontaneous. There are no annotations per frame, only the expression is given per sequence. The limited number of pain intensities (3 and neutral), missing dynamic annotations per frame, and the artificial posed nature of the data makes Hi4D-ADSIP not suitable for our goal of continuous pain intensity estimation and thus this work focuses on the ShoulderPain database.

4.3.2 Detection of AUs

The *Affectiva-MIT Facial Expression Dataset* (AM-FED) [123] contains 242 sequences from different subjects watching three funny superbowl commercials. The facial video is recorded by webcam with varying camera type, viewing angle, frame-rate and lighting, which is a difficult, real application scenario. The data is annotated for the presence of 10 AUs: 2, 4, 5, 9, 12, 14, 15, 17, 18 and 26. Additional, 22 tracked facial landmarks are provided.

The *Binghamton–Pittsburgh 4D Spontaneous Expression Database* (BP4D-Spontanous) [202] contains 328 sequences of facial 3D images and 2D texture recorded at 25 fps from 41 subjects. During each sequence, the subject is recorded while performing one of 8 interaction tasks with an experimenter. Each task is designed to elicit one of the emotions happiness, sadness, surprise, embarrassment, fear, pain, anger and disgust. Each sequence has been AU annotated for the most expressive 20 sec period. Onset and offset is annotated for 27 AUs: 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 27, 28, 30, 32, 38 and 39. Additional, the 6 level discrete intensities (0-5) of the AUs 12 and 14 are provided. Furthermore, the database creators include 49 2D and 83 3D tracked facial landmarks and head pose information per frame, as well as the self-reported emotion intensity per sequence on a 6 point Likert scale. The emotion intensity and pain label is not provided on a frame level, and thus this database is not suitable for per-frame pain intensity estimation.

The *Chinese Academy of Sciences Micro-expression Database* (CASME) [198] contains 195 sequences of micro-expressions from 35 subjects recorded at 60 fps. Micro-expressions are short and involuntary facial movements that occur when expressions are concealed [46]. The suppressed emotions are elicited by showing strongly positive and negative videos while given the additional instruction to maintain a neutral face. The sequences are annotated with the onset, apex and offset of all AUs.

The CASME II [197] database is an extension of CASME and is thus recorded and annotated within the same setting, see above. It has a higher frame-rate of 200 fps and contains 247

sequences of 26 subjects.

The *Cohn-Kanade Facial Expression Database* (CK) [88] contains 486 facial image sequences from 97 subjects recorded at 12 fps. The subjects have been instructed to show one of 23 different expressions and thus the data is posed. Each sequence starts with a neutral face and finishes with the apex of the expression. Only the apex frame is AU coded and the coding comprises all AUs.

The *Extended Cohn-Kanade Dataset* (CK+) [107] is based on the CK recordings (see above) and includes sequences that were not annotated previously. The data is divided in two parts, one posed and one spontaneous. The posed part has the same properties and annotations as the CK data and contains 107 sequences from 26 subjects. The spontaneous part contains 122 sequences from 66 subjecs smiling at the instructor in-between the instructed recordings. These sequences where annotated for presence/absence of the 8 AUs: 6, 12, 15, 17, 23, 24, 25 and 26.

The *Dynamic 3D FACS Dataset* (D3DFACS) [28] containes 519 sequences from 10 subjects recorded at 60 fps. The data includes a 3D mesh and 2D texture for each frame. The subjects were instructed to perform certain AU combinations and only the frames showing a peak expression have been AU coded.

The AU detection sub-challenge of the *GEMEP - Facial Expression Recognition Challenge Dataset* (GEMEP-FERA) [187] is part of the *Geneva Multimodal Emotion Portrayal* (GEMEP) [9] video corpus and includes 158 sequences from 10 subjects. Each subject is a professional actor and recorded while beeing instructed to speak several pseudolinguistic phoneme sequences. The presence/absence of 12 AUs have been annotated: 1, 2, 4, 6, 7, 10, 12, 15, 17, 18, 25 and 26.

The *Intelligent Systems Lab Facial Expression Frontal Database* (ISL Frontal) [177] includes 42 sequences from 10 subjects recorded at 30 fps, where the face is shown from a nearly frontal viewing angle. The subjects have been instructed to perform single AUs, certain AU combinations and the 6 basic emotions. The annotation include 14 AUs: 1, 2, 4, 5, 6, 7, 9, 12, 15, 17, 23, 24, 25 and 27.

The *Intelligent Systems Lab Facial Expression Multi-View Database* (ISL Multi-View) [176] is a different subset of the ISL Frontal data (see above) that includes 4 additional camera viewing angles. The recordings comprises 40 sequences from 10 subjects recorded at 30 fps. Furthermore, the others provide 15 annotated AUs and 34 facial landmarks for selected frames.

The *Maja and Michel Initiative Facial Expression Database Part I-III* (MMI Part I-III) [138] includes 169 AU codes sequences from 19 subjects. The subjects have been instructed to show single AUs, certain AU combinations and certain emotions. The annotations include all AUs and onset, apex and offset times.

The *Maja and Michel Initiative Facial Expression Database Part IV-V* (MMI Part IV-V) [185] includes 383 sequences from 25 subjects, which have been recorded while watching emotion inducing videos. The sequences are fully AU coded including onset, apex and offset.

4.3.3 Intensity Estimation of AUs

The *Bosphorus Database for 3D Face Analysis* (Bosphorous) [161] includes 4666 static 3D coordinates and 2D texture images from 105 subjects. The subjects have been instructed to display 23 separate AUs (1, 2, 4, 9, 10, 12, 12L, 12R, 12LW, 14, 15, 16, 17, 18, 20, 22, 23, 24, 25, 26, 27, 28, 34, 43 and 44) and 3 AU combinations. Additionally, 24 manually labelled facial landmarks are provided.

The *Intelligent Systems Lab Enhanced Cohn-Kanade AU-coded Facial Expression Database* (CK-Enhanced) [177] is based on the CK database (see above) and contains additional annotations for 14 AUs on a 3-level intensity scale: absent, present with low intensity and present. In contrast to the CK annotations which are only provided for the peak-frame, the CK-Enhanced annotations include all frames of the sequence. The newly annotated AUs are: 1, 2, 4, 5, 6, 7, 9, 12, 15, 17, 23, 24, 25 and 27.

The *Institute for Creative Technologies 3D Relightable Facial Expression Database* (ICT-3DRFE) [171] consist of static 3D meshes and 2D texture from 23 subjects, including photometric information that allow for photorealistic rendering. The subjects have been instructed to show 15 different expressions and thus the databases includes 345 faces in total. Annotations include all AUs with continuous intensities between 0 and 1.

4.3.4 Intensity Estimation of Other Expressions

The *Sustained Emotionally coloured Machine-human Interaction using Nonverbal Expression Dataset* (SEMAINE) [126] contains spontaneous facial expressions of users having a conversation with an operator. The operator talks about a topic that is relevant to the user, and tries to elicit different emotions. The face of the user has been recorded with a resolution of 780x580 pixels, and a frame-rate of 50 frames per second. We use a subset of the SEMAINE dataset which has been fully annotated and tracked. This subset contains 43 video sequences

of 10 subjects (between 22 and 60 years old), and a total number of 582,235 frames. The SEMAINE dataset provides annotations for the intensity of several affect dimensions on a continuous scale between -1 and 1. We use the annotations of valence and arousal, since they are relevant for discrimination between many affective states [155]. Each video is annotated per frame by six raters. The mean of the six raters is used as ground truth, leading to valence and arousal intensity distributions that are close to a Gaussian distribution with mean 0 and variance 1, as shown in Fig. 4.2. For localizing facial landmarks, the face has been tracked by the AAM model described in [39].

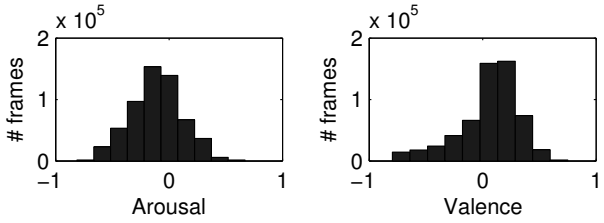


Figure 4.2: SEMAINE: Frame distribution over arousal and valence intensity levels.

Chapter 5

Evaluation Procedure

Contents

This chapter explains how we evaluate proposed recognition systems. First, Sec. 5.1 provides details on how to obtain a recognition score given predictions and ground-truth. Then the partitioning of training and testing is given in Sec. 5.2. Finally, Sec. 5.3 explains how the performance of multiple algorithms are statistically compared based on scores from multiple databases.

5.1 Metrics

In order to evaluate a recognition system, it is necessary to compare its prediction output from testing data with the given ground-truth. The better the recognition system, the closer the predictions should match the ground-truth. The distance between ground-truth and predictions is measured quantitatively by the evaluation metric. This section provides an overview of the most common evaluation metrics for continuous targets, which are the Pearson product-moment correlation coefficient (CORR), the mean squared error (MSE) and the Intra-class correlation coefficient (ICC). We include CORR and MSE, since they have been the most common metrics evaluating affect analysis [69], and MSE is the most common metric for evaluating regression algorithms (e.g. [147,174]). Recently, the Intra-Class Correlation Coefficient ICC(3,1) [166] has been proposed for evaluating approaches related to automatic human behaviour analysis (e.g. [74,121]) and thus we include this measure as well. First, we define each

of the metrics and then discuss their differences.

Given N ground-truth targets $\mathbf{t} = [t_1, ..., t_N]^\top$, $t_n \in \mathbb{R}$ and N regressor predictions $\mathbf{y} = [y_1, ..., y_N]^\top$, $y_n \in \mathbb{R}$, then the CORR is defined as:

$$\text{CORR}(\mathbf{t}, \mathbf{y}) = \frac{\text{cov}[\mathbf{t}, \mathbf{y}]}{\sigma_\mathbf{t} \sigma_\mathbf{y}}, \tag{5.1}$$

where $\sigma_\mathbf{t}$ is the standard deviation of $\mathbf{t}$ and $\text{cov}[\mathbf{t}, \mathbf{y}]$ is the covariance between $\mathbf{t}$ and $\mathbf{y}$. The MSE is defined as the Expected value of the squared error:

$$\text{MSE}(\mathbf{t}, \mathbf{y}) = \frac{1}{N} \sum_n (t_n - y_n)^2 \tag{5.2}$$

Note that some authors additionally apply the square-root to the MSE (RMSE) (e.g., [158]), but this does not change the performance ranking between algorithms, since the square-root is strictly monotonically increasing. If we assume the error to be a continuous random variable, then the MSE is its variance, while the RMSE is its standard deviation. Thus the RMSE has the same unit as the error, while the MSE has the original unit squared.

In order to compare the (R)MSE across different datasets, the ground-truth targets should have the same standard deviation. Otherwise, we could scale the targets and predictions of one dataset with an arbitrary constant c and thus both scale the MSE and RMSE arbitrarily (the only difference is that the RMSE would be scaled linearly in c, while the MSE would be scaled by c^2). A standardized scaling is implicitly performed for CORR and ICC, and thus CORR and ICC are better measures for comparison across datasets than MSE and RMSE.

The ICC [166] originates from behavioural psychology and measures the agreement between two or more raters. It is based on quantities obtained by the Analysis of Variance (ANOVA) framework. Several types of ICC have been defined, each one differing by the data model, see [125, 166]. The ICC of concern in this work is noted as ICC(3,1) according [166] and ICC(C,1) Case 3 according [125], since it is the commonly used ICC for evaluating AU intensity estimation (e.g., in [121, 122, 152]). All further mentioning of ICC is referring to this specific ICC type. The ICC is defined as $\text{ICC} = \frac{\text{BMS}-\text{EMS}}{\text{BMS}+(K-1)\text{EMS}}$, where K is the number of raters, BMS are the between target mean squares and EMS are the residual mean squares, as defined by ANOVA. We use the ICC as evaluation metric and thus $K = 2$, since $\mathbf{t}$ and $\mathbf{y}$ correspond to one rater each. In this case, the formula can be simplified to:

$$\text{ICC}(\mathbf{t}, \mathbf{y}) = \frac{2\text{cov}[\mathbf{t}, \mathbf{y}]}{\sigma_\mathbf{t}^2 + \sigma_\mathbf{y}^2} \tag{5.3}$$

CORR is a linearity index, since it measures the degree to which $y = at + b$ holds for arbitrary constants a and b [125]. In contrast to that, ICC measures the degree to which

$y = t + b$ holds, and thus is an additivity index [125]. MSE measures the degree to which the identity mapping $y = t$ holds.

To better understand the differences between the metrics, we describe their respective invariances, i.e., how $\mathbf{t}$ and $\mathbf{y}$ can change without changing the value of the metric. Furthermore, we demonstrate an equivalence transform between the metrics, i.e., how $\mathbf{t}$ and $\mathbf{y}$ can be normalized in order to obtain equivalent metric values.

From the functional mapping between ground-truth and targets above, it follows that (1) CORR is invariant regarding additive and multiplicative constants, (2) ICC is invariant regarding additive constants and (3) MSE is not invariant regarding any constants.

The ICC is equivalent to CORR if we normalize $\mathbf{t}$ and $\mathbf{y}$ regarding their standard deviation, i.e., for $\hat{\mathbf{t}} = \frac{\mathbf{t}}{\sigma_{\mathbf{t}}}$ and $\hat{\mathbf{y}} = \frac{\mathbf{y}}{\sigma_{\mathbf{y}}}$, the following holds:

$$\mathrm{ICC}(\hat{\mathbf{t}}, \hat{\mathbf{y}}) = \frac{2\mathrm{cov}[\hat{\mathbf{t}}, \hat{\mathbf{y}}]}{\sigma^2_{\hat{\mathbf{t}}} + \sigma^2_{\hat{\mathbf{y}}}} = \frac{2\mathrm{cov}\left[\frac{\mathbf{t}}{\sigma_{\mathbf{t}}}, \frac{\mathbf{y}}{\sigma_{\mathbf{y}}}\right]}{1+1} = \frac{\mathrm{cov}[\mathbf{t}, \mathbf{y}]}{\sigma_{\mathbf{t}}\sigma_{\mathbf{y}}} = \mathrm{CORR}(\mathbf{t}, \mathbf{y}) \tag{5.4}$$

Analogous, the MSE is equivalent to CORR if we normalize $\mathbf{t}$ and $\mathbf{y}$ regarding their mean and standard deviation, i.e., for $\hat{\mathbf{t}} = \frac{\mathbf{t}-\mu_{\mathbf{t}}}{\sigma_{\mathbf{t}}}$ and $\hat{\mathbf{y}} = \frac{\mathbf{y}-\mu_{\mathbf{y}}}{\sigma_{\mathbf{y}}}$ where $\mu_{\mathbf{t}}$ is the mean of $\mathbf{t}$, the following holds:

$$\mathrm{MSE}(\hat{\mathbf{t}}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_n \left(\hat{t}^2_n - 2\hat{t}_n\hat{y}_n + \hat{y}^2_n\right) = 1 - 2\frac{\mathrm{cov}[\mathbf{t}, \mathbf{y}]}{\sigma_{\mathbf{t}}\sigma_{\mathbf{y}}} + 1 = -2\mathrm{CORR}(\mathbf{t}, \mathbf{y}) + 2 \tag{5.5}$$

CORR, MSE and ICC all measure different aspects of the distance between ground-truth and predictions. Which measure is preferred depends highly on the application domain and thus we usually show the results of all three measures.

5.2 Division of Training and Testing Data

When evaluating a recognition system, we are interested in a performance estimate for unseen data, i.e. data that has not been used during training. This makes it necessary to divide the available data in non-overlapping sets of training and testing data. When applying the same principle to data from human subjects, we can extend the requirement to performance estimates from unseen subjects, i.e. subjects thats has not been used during training.

Thus for dividing the data into training and testing sets, we use the subject-independent setting, where the videos of selected subjects are left out for testing, and the videos of all other subjects in the dataset are used for training. This process is repeated with different

subjects, until all subjects have been used for testing. The results are combined by calculating the weighted average across all subjects left out for testing. The weight of each subject corresponds to the number of frames each subject occurs in.

We always use the subject-independent setting within this thesis for all datasets, except the results in Tab. 8.4, where we compare our work to previously published subject-dependent results.

5.3 Statistical Comparison of Algorithms

When evaluating different AU and pain recognition methods regarding the performance metrics explained in Sec. 5.1, we usually obtain one score per algorithm, per target and per database. Multiple scores make it difficult to directly compare and rank algorithms, since usually a single algorithm is not consistently performing significantly better than all others. Therefore we perform statistical comparison tests of algorithms over multiple data sets, as suggested by [38]. First, the Friedman test [58] is applied to obtain a score rank and to detect whether all algorithms are statistically the same. If the null-hypothesis is rejected, then the Hommel [78] post-hoc procedure is applied to detect which pairs of algorithms are different. Both tests are performed with a significance value $p = 0.05$. Since a larger set of databases (2 or 3 is not sufficient) is needed to produce a meaningful result, we assume each target to be a different database and thus obtain an overall ranking of algorithms across targets and databases. E.g. when comparing algorithms on DISFA with 12 AUs and ShoulderPain with 10 AUs, then we apply the Friedman test over a total of $12 + 10 = 22$ databases.

The Friedman results are reported as a ranked list of algorithms. Each algorithm subset which has equal score rank (according the Hommel procedure) is indicated by a black bar on the right side which spans the rows of the included algorithms. An example result is shown in Tab. 5.1: in this case, the algorithm pairs (A,B), (B,C) and (C,D) are not statistically different, but there is a difference between the pairs (A,C), (A,D), and (B,D). Thus, no single algorithm is clearly the best, but we know that the best must be either A or B (including the option that A and B are equally good).

Table 5.1: Example representation of the Friedman test [58] rank results and equal-performance subsets obtained by Hommel's dynamic procedure [78]. The different algorithms are ranked by their expected performance rate. The subsets of algorithms which have statistically equal performance are indicated by a black bar on the right side.

Rank	Method	
1	Algorithm A	▌
2	Algorithm B	▌▌
3	Algorithm C	▌▌
4	Algorithm D	▌

Chapter 6

Pre-processing

Contents

This chapter describes the pre-processing methods used for experiments in this thesis. For a general overview of methods, please see Sec. 3.1. First, we show an overview of the used pre-processing methods, including a motivation for each specific choice, in Sec. 6.1. Then we present processing details for each method within Sec. 6.2, 6.3 and 6.4.

6.1 Overview

The following chapters use three sets of features for recognizing facial expressions: Facial Landmark Points (PTS), Local Binary Patterns (LBP) and Discrete Cosine Transform (DCT). We employ these three sets of features because they contain different types of information. PTS are geometric features, and are robust to illumination changes. However, they cannot accurately capture subtle facial movements (e.g., the eye wrinkles). This can be well described by the appearance features (i.e., DCT and LBP) that are derived from pixel intensities of an image. Compared to PTS, DCT and LBP are much more sensitive to skin color variation, and texture variation due to the illumination changes. Note, however, that DCT and LBP capture different characteristics of the texture changes. Specifically, DCT features describe image appearance on a large scale, since it is equivalent to a low-pass filer. The overall image structure is still preserved, but sharp edges are lost. Conversely, LBP features are local

descriptors that model statistics of the gradient orientations within a small pixel neighborhood, i.e., they describe the edges.

In regard to past literature, PTS has been used abound for AU and pain intensity estimation [8,73,74,82,112,151,152,186,199,203] and has been the most used feature, see Tab. 3.1. LBP as a histogram based edge descriptor has been widely used for AU and pain recognition as well [121,151,158,186], although alternative local descriptors like HOG, SIFT or Gabor would probably lead to similar results. DCT has been only used for pain detection [13,110] and face recognition [42,49]. However, for the reasons explained above, we expect it to work well in combination with LBP, which has already been proven for face recognition [49].

Each of the following chapters (Chap. 7–10) focuses on a different aspect of recognizing facial expressions, which also influences the choice of features. Tab. 6.1 is showing an overview of the different features and pre-processing methods. Chap. 7 focuses on the evaluation of different features and their combinations and thus uses shape features (PTS) and complementary appearance features (LBP and DCT). Chap. 8 assess the importance of different facial regions and thus uses the best appearance feature identified in the previous chapter (i.e., LBP). The region importance is assessed in the spatial and temporal domain and thus uses multiple patch resolutions and time windows. Chap. 9 provides initial experiments with both, PTS and LBP features. However, the main experiments focus on PTS only, due to poor LBP results. The model in this chapter had problems with the large dimensionality (59 dimensions per patch) and thus we used 6x6 patches instead of 9x9. Chap. 10 uses all types of features, PTS, LBP and DCT, since it includes the best methods from the previous chapters. Furthermore, Chap. 10 uses LCP for appearance registration instead of PWA, since no facial contour points are provided by the used tracker and thus PWA cannot be applied.

Table 6.1: Overview of pre-processing methods used for the experiments in this thesis. Each column represents one of the following chapters.

	Chap. 7	Chap. 8	Chap. 9	Chap. 10
# Landmarks	66	66	66	49
Shape Registration	Procrustes	-	Procrustes	Procrustes
App. Registration	PWA	PWA	PWA	LCP
App. # Patches (S)	9x9	6x6, 9x9	6x6	30
App. Time Window (T)	1	1, 10, 20	1	1
Features	PTS, LBP, DCT	LBP	PTS, LBP	PTS, LBP, DCT

Abbreviations: App.: Appearance; **Registration** – LCP: Landmark Centered Patches, PWA: Piece-wise Affine; **Features** – DCT: Discrete Cosine Transform, LBP: Local Binary Patterns, PTS: Facial Landmark Points.

In the following, we provide details for the feature extraction methods, PTS, LBP and DCT.

Each of them has been applied in the same way to different databases. The only exception is Chap. 10, which uses an additional shape normalization step that includes subtracting the individual mean shape for each subject to improve the subject-independent results.

6.2 Facial Landmark Points (PTS)

Each of the used datasets provides annotated facial points. Details about the annotation process are explained in the corresponding database description, see Sec. 10.4 for the EmoPain data and Chap. 4 for all other data. In order to obtain local point features (PTS) of the face, we follow the same procedure for all databases: the 66 (see Fig. 6.1) or 49 tracked facial landmarks are aligned by Procrustes analysis to the mean shape, which removes translations and in-plane rotations of the face. Then each of the x and y landmark coordinates are normalized by subtracting the mean and dividing by the standard deviation. The coordinates are stacked together into the final 132 or 98 dimensional feature vector.

Figure 6.1: Example frame showing 66 facial landmark points from the ShoulderPain database.

6.3 Local Binary Patterns (LBP)

LBP features were obtained by extracting Local Binary Pattern [135] from registered facial images. These registered images are obtained by a piece-wise affine (PWA) warp to a base shape using the standard active appearance model (AAM), equivalent to [109]. The base shape has a size of 128×118 pixels for ShoulderPain and DISFA, and 128×155 pixels for SEMAINE. The AAM used for tracking SEMAINE had a different aspect ratio and we scaled the base shape to match vertically. A concatenation of all registered frames within a video sequence of L frames results in a space-time volume of the size $128 \times 118 \times L$ (or $128 \times 155 \times L$), as illustrated in Fig. 6.2. We divide the space-time volume into subvolumes, and extract video features from each subvolume. In this way, we enforce that our video features are local, extracted from relatively small spatiotemporal supports, rather than from the entire face. As

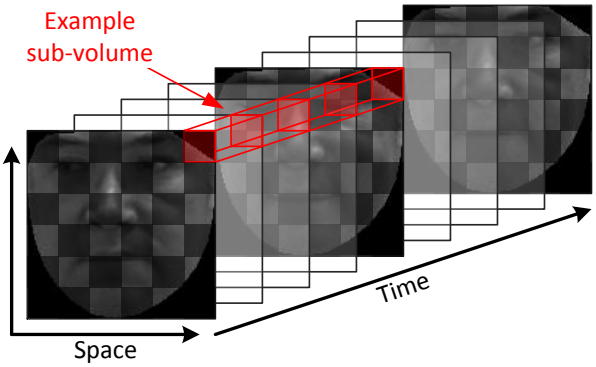


Figure 6.2: Video data represented as space/time pixel volume. One of the sub-volumes is highlighted for the space scale of 6x6 patches and time scale of 5 frames.

mentioned in Chap. 1, our local extraction of video features is motivated by a number of psychological studies [118, 137, 154] which argue that facial expressions are characterized by distinct combinations of local AUs, rather than global features extracted from the entire face. We partition each video frame into a regular grid of $S \times S$ patches. In our experiments, we use $S \in \{6, 9\}$, i.e., the face is divided into 36 or 81 patches. Note that each of these patches defines a video subvolume with L frames. For analyzing various temporal scales, we scan these subvolumes along the time axis. The scan has a step size of one frame and a window size of $T \in \{1, 10, 20\}$ frames. Thus, we extract features from a total of $S \times S \times (L - T + 1)$ subvolumes per video and each feature vector includes information from a window of T consecutive frames.

LBP features are defined for image patches (time-scale $T = 1$). For video subvolumes with $T > 1$, we use the temporal extension: LBP in three orthogonal space-time planes [204]. LBPs and their temporal extensions have been demonstrated useful for facial expression recognition [84]. Temporal extensions of LBP typically improve performance in comparison to the static LBP [84].

The EmoPain data (see Chap. 10) does not include the facial contour landmarks and thus the PWA registration is not possible. As alternative, we extract features from 30 landmark centered patches (LCP) as depicted in Fig. 6.3. 19 of the full 49 landmarks have been removed to reduce the overlap between patches.

6.4 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) [4] features are obtained from the same registered facial images explained above in Sec. 6.3. The 2-dimensional DCT was applied to the full images, and the first 500 coefficients were used as features, which were selected based on the *zig-zag*

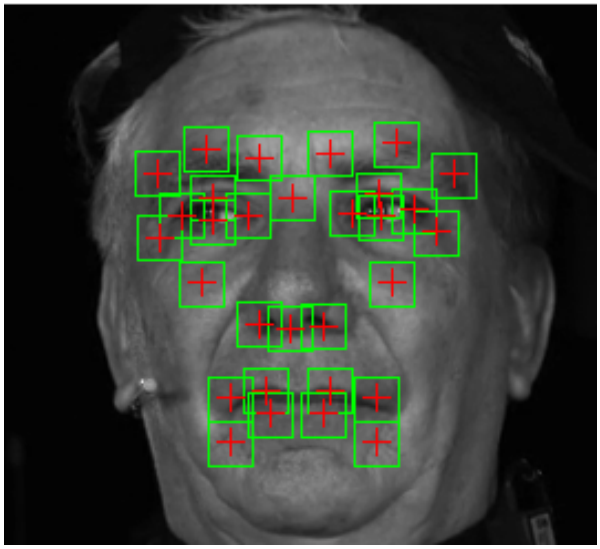


Figure 6.3: Example of a normalized face image with highlighted regions (green) around the 30 facial landmarks (red) from which the features were extracted on the EmoPain data.

scheme [193]. Since PWA was not possible for the EmoPain data, DCT was applied to the LCP patches by selecting the first 59 coefficients of each patch, which results in the same dimensionality as LBP features.

Chapter 7

Relevance Vector Machine Feature Fusion

Contents

This chapter describes our first approach to continuous facial expression intensity estimation. The observed face is treated holistically and we learn a set of independent regression functions using different shape (PTS) and appearance (DCT and LBP) features, and then perform their late fusion. We show on the ShoulderPain database that late fusion of the afore-mentioned features leads to better pain intensity estimation compared to feature-specific estimation.

7.1 Introduction

We propose a three-step approach to continuous pain intensity estimation per video frame (in contrast to [108], which estimates pain for a whole video sequence only). The outline of the proposed approach is depicted in Fig. 7.1. In the first step, we extract shape-based features (i.e, locations of characteristic facial points) and appearance-based features (Local Binary Patterns (LBPs) [135] and Discrete Cosine Transform (DCT) [4]) from facial images of subjects displaying different intensities of pain. The PSPI pain intensity (see Sec. 2.2) was annotated by the database creators using sixteen discrete values (0 to 15), with 0 meaning no pain and 15 meaning its peak. In the second step, for each set of features we train separate

Relevance Vector Regression (RVR) [174] models for prediction of the pain intensity levels. RVR models the target function by selecting representative cases, the so called 'Relevance Vectors', which are used in the model during inference of a query image. We use RVR instead of the popular Support Vector Regression (SVR) in the target task because it usually results in a more sparse model, i.e., fewer relevance vectors are selected than support vectors for the same task [174]. In our case, this is important since we deal with image sequences.

Finally, the outputs of the regressors trained using different feature sets are combined in two ways: (i) by computing the mean estimate of the regressors, and (ii) by using the outputs of separate regressors as an input to another RVR, which gives a single estimate for the pain intensity. In contrast to the aforementioned methods which deal with pain detection only (i.e., pain vs. no pain), the proposed approach is the first one that performs continuous pain intensity estimation. Furthermore, we show that the proposed feature-fusion scheme outperforms the separately trained RVRs on different feature sets, whereby the combination of appearance features (DCT and LBP) performs best. We also demonstrate the performance of the proposed approach in the task of continuous intensity estimation of the facial AUs.

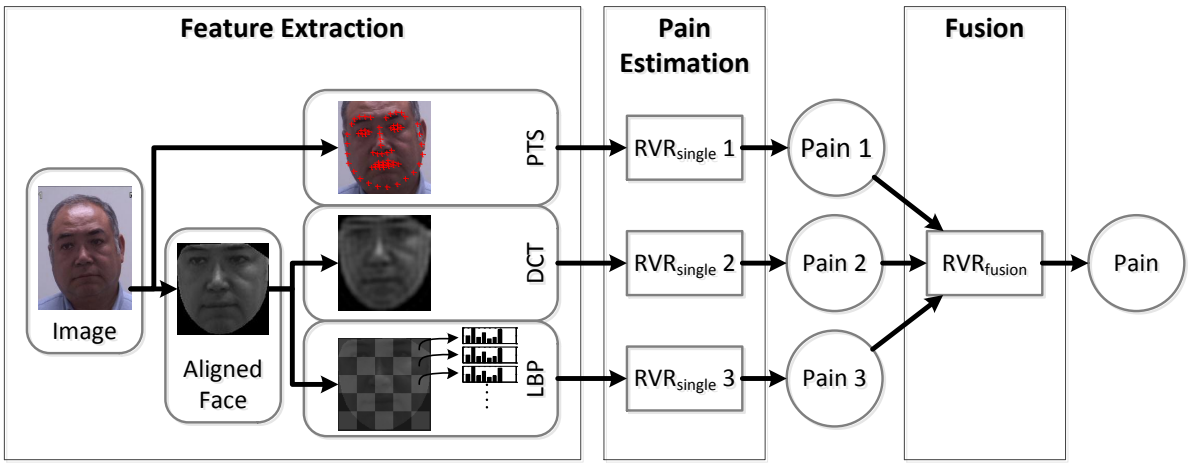


Figure 7.1: Overview of the proposed approach to continuous pain intensity estimation. We first extract three feature sets from a face image: facial landmarks (PTS), Discrete Cosine Transform coefficients (DCT) and Local Binary Patterns (LBP). We then use Relevance Vector Regression (RVR) to learn the feature-specific functions, which independently estimate the pain intensity from each feature set. In the final step, we use a second layer RVR to perform the fusion of the pain intensity estimations obtained by the feature-specific functions

The rest of the chapter is organized as follows. Sec. 7.2 presents the regression-based approach to continuous pain intensity estimation. Then Sec. 7.3 shows the experiments and discusses the results. Finally, Sec. 7.4 concludes the chapter.

7.2 The Model

To perform continuous pain intensity estimation from a single feature set, we learn a regression function that maps the features to the corresponding (discrete) pain intensity levels. This function is learned by means of the Relevance Vector Regression (RVR) model [174]. Formally, for each feature set we model the outputs (y) of the target function as:

$$y(\mathbf{x}; \mathbf{w}, \gamma, \sigma) = \sum_{n=1}^{N} w_n \kappa(\mathbf{x}, \mathbf{x}_n) + \epsilon, \tag{7.1}$$

where $\mathbf{x}$ is the input feature vector, $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ are N possible relevance vectors (in our case the same as the training data points) and $\mathbf{w} = (w_1, ..., w_N)$ are the weight parameters. Here, the sparsity of the model comes from the fact that most of the weights parameters tend to go to zero, thus, the corresponding training samples are not used for inference. This is enforced by a sparse prior probability on the $\mathbf{w}$. As the kernel function κ, we use the standard Radial Basis Function (RBF) kernel with the length scale parameter γ. The noise ϵ on the outputs is modeled as a Gaussian with zero mean and the standard deviation σ. During the experiments, the length scale parameter γ was calculated heuristically from the training features.

Once the feature-set-specific target functions are learned, we perform late fusion of their outputs. This fusion is performed in two ways: (i) mean fusion and (ii) RVR fusion. In the mean fusion approach, we calculate the mean of the outputs, obtained by the feature-set-specific target functions $\{y_1, ..., y_L\}$, as $y_f = \frac{1}{L} \sum_{l=1}^{L} y_l$, where y_f is the mean fusion output and L is the number of the feature sets. RVR fusion is performed by learning another RVR model that uses the outputs of the feature-set-specific target functions as an input, i.e., $\hat{\mathbf{y}} = (y_1, ..., y_L)$, which are continuous estimates of the pain level intensities, and the (discrete) pain level intensities as outputs. This fusion function is given by

$$y_f(\hat{\mathbf{y}}; \mathbf{w}^f, \gamma^f, \sigma^f) = \sum_{m=1}^{M} w_m^f \kappa^f(\hat{\mathbf{y}}, \hat{\mathbf{y}}_m) + \epsilon^f, \tag{7.2}$$

where $\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_M\}$ are M training inputs, obtained from the first-layer outputs, and $\mathbf{w^f} = (w_1^f, ..., w_M^f)$ are the weight parameters, γ^f is the length scale of the Radial Basis Function kernel κ^f and ϵ^f is the noise, as defined above. Note that the training samples used to learn the feature-set-specific target functions may differ from the samples used to learn the fusion function.

7.3 Results

We performed two sets of experiments. In the first set of experiments we evaluated the performance of the proposed approach in the task of continuous AU intensity estimation. In the second set, we evaluated the performance in the task of continuous pain intensity estimation. Both sets use different combinations of PTS, LBP and DCT features, see Chap. 6 for details. In all our experiments we applied a leave-one-subject-out cross-validation procedure. Specifically, we used facial images of 24 subjects for training and one subject for testing. The feature-specific target functions were trained using the same training data as for the fusion functions. Note that in terms of generalization performance, the performance of the proposed 2-layer approach is expected to be better if the feature-specific target functions and the fusion function are trained using data corresponding to different subjects. However, we found that this strategy results in worse performance than using the same training data to train both layers. This could be due to the limited number of available subjects: if the subjects are split between the first and the second layer, then each of the layers is trained on less subjects, and hence the performance decreases. Note also that AU27 was left out, since only few examples with intensities greater than zero are present in the dataset (see Table 4.1).

7.3.1 Single-Feature

Table 7.1 shows the results for the feature-specific target function learned in the task of continuous pain/AU intensity estimation. In the case of pain intensity estimation, the ground truth contains 16 discrete intensity levels, while in the case of AUs there are 6 discrete intensity levels. In addition, we show the results of two methods for pain intensity estimation: Pain (I) is directly estimated from the features as described in Section 7.2, where Pain (II) is calculated from the estimated AU intensities by using the PSPI score, see Eq. 2.1. As can be seen, the results obtained by the PSPI method are in some cases outperformed by the direct estimation. This is a consequence of the error propagation in the AU estimation, since for some AUs only few positive examples (i.e., with intensity level greater than zero) were available during training. Since the Pain (II) is computed by using a deterministic formula, the inaccuracies in the estimation of each AU are added in the final estimate of the pain intensity. Note also from Table 7.1 that for AU intensity estimation, LBP features outperform PTS and DCT features. This is because the LBPs are local descriptors and are able to better capture appearance variation caused by changes in AU intensities, since different AUs are located in different regions of a facial image. The accuracy in AU intensity estimation attained by using LBPs directly translates into the accuracy attained by the Pain (II) approach, which outperforms

Pain (I) in the case of the LBPs. On the other hand, in the case of DCT features, which capture global changes in appearance, the Pain (I) is more accurate than the Pain (II) approach. This again shows that estimating the pain intensity level from AU intensities is sensitive to the errors in AU intensity estimation. We also observe that in the case of predicting AU20 with LBP features, the MSE can be misleading: the result of 0.103 seems better than the MSE of other AUs, but CORR and ICC are very low at 0.092 and 0.087. This is due to the imbalanced data used for training (see Table 4.1 and Fig. 8.2) where the vast majority of the frames have the zero intensity.

CORR and ICC show the same trend, however the ICC score is in general lower than CORR. This is probably caused by scale differences between the predictions and ground-truth: CORR is invariant to these differences, while ICC treats it as an error.

Overall, LBP features perform best in terms of the MSE measure, while in the case of the CORR and ICC measures, the difference is not that apparent, though LBPs are still the best in most cases. On the other hand, DCT features perform best in the task of pain intensity estimation. Overall, appearance features (DCT and LBP) work better than shape features (PTS). However, the poor performance of shape features might be caused by registration errors, because the Procrustes alignment cannot cope properly with out-of-plane rotations. A better registration will likely improve the single shape and the fusion results, therefore we would not suggest to rely on appearance features alone.

7.3.2 Mean Feature Fusion

The results for the mean-fusion approach are shown in Table 7.2. In most cases, MSE, CORR and ICC improve over the results obtained with single features only. This shows that the employed features contain complementary information. Based on the CORR results, the DCT+LBP fusion gives the best results in most cases. This is not surprising, because DCT and LBP, although both being appearance-based features, capture different information: DCT captures global, while LBP captures local appearance variation.

7.3.3 RVR Feature Fusion

The results for the RVR feature fusion are shown in Table 7.3. The results are similar to those obtained by the mean fusion in the sense that almost all values improve over the single feature results, as expected. However, the improved performance of DCT+LBP features is even more pronounced in the case of the RVR fusion approach, giving the best CORR results overall. Although we would expect the RVR fusion to perform at least as good as the mean fusion in

Table 7.1: **Single feature** results for Facial Action Unit (AU) and pain intensity estimation, measured by the mean squared error (MSE), the Pearson correlation coefficient (CORR) and the intra-class correlation coefficient (ICC). Pain (I) is estimated directly from the features and Pain (II) is calculated from the estimated AU intensities using the PSPI score (see Eq. 2.1). The best result for each target and each measure is printed in bold letters

Measure	MSE			CORR			ICC		
Features	PTS	DCT	LBP	PTS	DCT	LBP	PTS	DCT	LBP
AU4	0.341	0.254	**0.204**	.096	**.140**	.133	.042	.039	**.070**
AU6	0.906	0.592	**0.590**	.385	**.528**	.527	.357	.401	**.435**
AU7	0.806	0.504	**0.379**	.120	.303	**.342**	.090	.194	**.206**
AU9	0.119	**0.119**	0.113	**.246**	.224	.190	**.125**	.118	.105
AU10	0.084	**0.079**	0.097	.171	**.203**	.169	**.061**	.054	.042
AU12	1.010	0.717	**0.600**	.330	.484	**.548**	.300	.373	**.409**
AU20	0.505	0.158	**0.103**	.012	**.092**	**.092**	.014	**.043**	.024
AU25	0.707	0.579	**0.486**	.130	.104	**.204**	.095	**.125**	.117
AU26	0.896	0.834	**0.475**	.013	.016	**.111**	.056	.033	**.071**
AU43	0.300	0.273	**0.176**	.240	.291	**.465**	.160	.234	**.307**
Pain (I)	2.592	**1.712**	1.812	.363	**.528**	.483	.295	.417	**.421**
Pain (II)	2.532	1.716	**1.484**	.348	.480	**.518**	.310	.399	**.445**

Table 7.2: **Mean feature fusion** results for Facial Action Unit (AU) and pain intensity estimation, measured by the mean squared error (MSE), the Pearson correlation coefficient (CORR) and the intra-class correlation coefficient (ICC). The best results are shown in bold letters.

Measure	MSE				CORR				ICC			
Features	PTS DCT	PTS LBP	DCT LBP	all	PTS DCT	PTS LBP	DCT LBP	all	PTS DCT	PTS LBP	DCT LBP	all
AU4	0.224	0.201	0.206	**0.191**	.205	.260	.294	**.295**	.053	**.071**	.061	.067
AU6	0.543	0.544	0.496	**0.472**	.500	.508	.526	**.543**	.445	.443	**.448**	.467
AU7	0.479	0.429	**0.361**	0.376	.276	.276	**.376**	.343	.159	.164	**.215**	.188
AU9	0.087	0.091	0.096	**0.083**	.370	.339	.323	**.382**	.146	.139	.128	**.149**
AU10	0.064	0.070	0.075	**0.064**	.371	.312	.334	**.370**	**.066**	.056	.050	.060
AU12	0.656	0.625	0.568	**0.563**	.529	.545	.582	**.588**	.388	.395	**.424**	.421
AU20	0.177	0.179	**0.103**	0.119	.103	.095	**.133**	.129	.034	.025	**.038**	.035
AU25	0.474	0.449	0.455	**0.415**	.212	.213	**.264**	.252	.129	.123	.135	**.137**
AU26	0.622	**0.482**	0.557	0.493	.090	.118	.090	**.120**	.068	**.082**	.058	.078
AU43	0.232	**0.184**	0.191	0.187	.360	.396	**.462**	.439	.233	.261	**.301**	.282
Pain (I)	1.469	1.642	1.508	**1.373**	.489	.481	**.554**	.547	.417	.413	**.458**	.454
Pain (II)	1.928	1.850	**1.368**	1.480	.395	.403	**.529**	.494	.405	.429	**.465**	.453

all tasks, this does not seem to be the case. A reason for this could be the fact that both layers in the proposed approach are trained on the same data (because of the limited training data), which could have led the 2nd-layer RVR to over-fit the data.

Table 7.3: **RVR feature fusion** results for Facial Action Unit (AU) and pain intensity estimation, measured by the mean squared error (MSE), the Pearson correlation coefficient (CORR) and the intra-class correlation coefficient (ICC). The best results are shown in bold letters.

Measure	MSE				CORR				ICC			
Features	PTS DCT	PTS LBP	DCT LBP	all	PTS DCT	PTS LBP	DCT LBP	all	PTS DCT	PTS LBP	DCT LBP	all
AU4	0.264	0.248	**0.242**	0.274	.209	.199	**.243**	.177	.044	.042	**.051**	.042
AU6	0.539	0.550	**0.480**	0.549	.487	.514	**.533**	.502	.411	**.444**	.442	.437
AU7	0.423	0.428	**0.343**	0.400	.248	.321	**.402**	.314	.127	.165	**.192**	.171
AU9	0.132	0.233	**0.120**	0.201	.401	.326	**.479**	.414	.116	.114	**.151**	.142
AU10	0.087	0.074	0.071	**0.070**	.080	.243	**.424**	.294	.035	.024	**.064**	.047
AU12	0.782	0.713	**0.617**	0.657	.507	.542	**.576**	.545	.386	.401	**.433**	.413
AU20	0.140	**0.088**	0.109	0.147	.049	.059	**.086**	.049	.006	.005	**.016**	.007
AU25	0.669	**0.538**	0.572	0.762	.106	.199	**.235**	.090	.048	**.095**	**.095**	.037
AU26	0.604	**0.414**	0.490	0.582	.005	.060	**.090**	.015	.001	**.042**	.039	.010
AU43	0.243	**0.158**	0.179	0.182	.352	.512	**.516**	.437	.224	**.351**	.340	.294
Pain (I)	1.801	1.567	**1.386**	1.804	.489	.485	**.590**	.502	.396	.406	**.470**	.417
Pain (II)	1.867	1.899	**1.633**	1.770	.342	.345	**.471**	.369	.383	.432	**.444**	.434

Tab. 7.4 shows the statistical comparison results across all AU and pain (I) targets according the Friedman test [58] and Hommel procedure [78] (see Sec. 5.3 for details). DCT+LBP has the best rank for all measures, but it is only significantly better than PTS regarding MSE and PTS and PTS+DCT regarding ICC. CORR results show more significant differences: all LBP combinations (DCT+LBP, LBP and PTS+LBP) are in the top group and thus we can conclude that including LBP provides significantly better results.

Table 7.4: Rank comparison of single feature and RVR fusion algorithms over all AU and pain (I) targets obtained by the Friedman test [58] and Hommel procedure [78]. The different features are ranked by their expected performance rate for each of the measures MSE, CORR and ICC. The subsets of features which have statistically equal performance are indicated by a black bar on the right side.

Rank	MSE	CORR	ICC
1	DCT+LBP	DCT+LBP	DCT+LBP
2	LBP	LBP	LBP
3	PTS+LBP	PTS+LBP	DCT
4	all	DCT	all
5	DCT	all	PTS+LBP
6	PTS+DCT	PTS+DCT	PTS
7	PTS	PTS	PTS+DCT

Fig. 7.2 shows an example of the pain intensity estimation from one test image sequence. The estimation is based on our best model, i.e., DCT+LBP RVR fusion (the Pain (I) approach). In most cases, the continuous pain intensity estimation is close to the ground-truth. Note, however, the peaks around the frames 95, 120 and 336, which are all caused by the eye blinks. This is a consequence of the fact that the proposed approach is static (i.e., it is trained per frame), and therefore, it cannot differentiate between an eye blink (short time) and eye closure (long time). During the training stage, the model has learned that the closed eyes are related to pain, and that is why the eye blinks result in sudden peaks in the estimated pain intensity, as shown in Fig. 7.2.

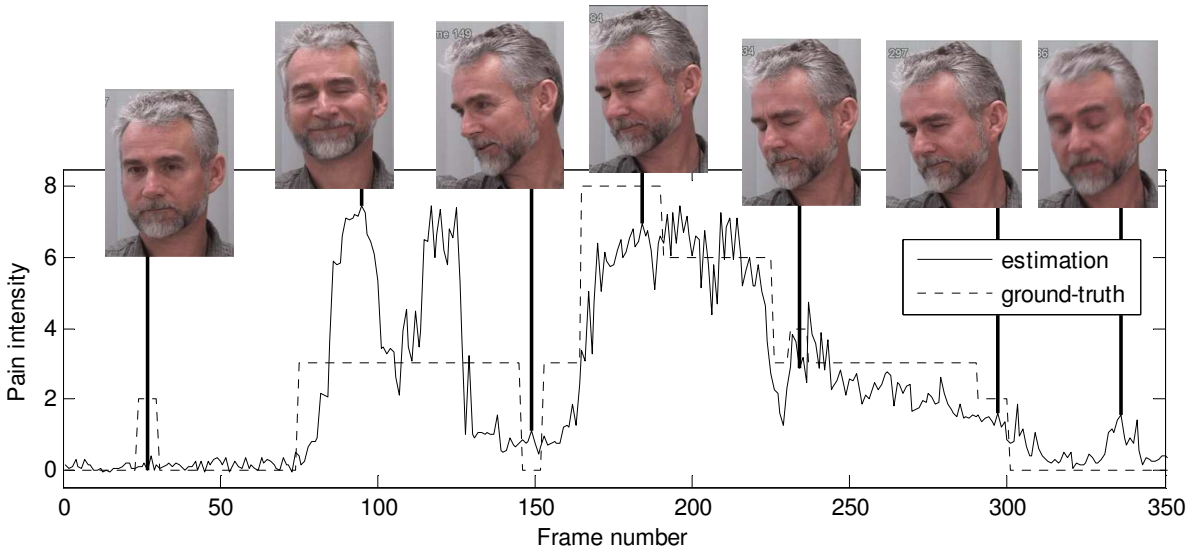


Figure 7.2: Example pain estimation sequence for DCT+LBP RVR fusion

7.4 Conclusion

We have proposed a three-step approach to continuous pain intensity estimation based on Relevance Vector Regression. We have shown that for the task of continuous pain and AU intensity estimation, the proposed approach achieves better results when trained using appearance-based features (either DCT or LBP) than with the shape features (PTS). Also, when used as single input features, LBPs worked best in most cases. Furthermore, we showed that the fusion of DCT and LBP features gives the best performance in the target task. However, we believe that by a proper alignment of the shape-based features (e.g., by using [153]), the overall performance attained by the fusion of these three feature sets should improve. We also showed that direct pain estimation can be more accurate than calculation from the estimated AUs,

which is probably due to the inaccuracies in AU intensity estimation. The approach presented in this chapter estimates the AU intensities independently and does not exploit information about their co-occurrences. Furthermore, the current approach is holistic and does not focus on certain regions of the face. These limitations of the proposed approach are the focus of the following chapters.

Chapter 8

Doubly Sparse Relevance Vector Machine

Contents

Since inner feelings and physiological states like pain are typically characterized by subtle movements of facial parts, the analysis of the facial details could improve the holistic approach introduced in the previous chapter. Here we introduce the Doubly Sparse Relevance Vector Machine (DSRVM), which is able to focus on informative facial regions.

8.1 Introduction

Automatic facial expression recognition has mainly focused on the face as holistic entity, rather than on specific facial parts [201]. This is contrary to research in psychology, which suggest that only components of facial expressions are universally displayed, rather then the whole face [137, 163]. Motivated by these insights into human perception of faces, we introduce a component based approach for automatic facial expression recognition.

Most approaches to automatic facial behavior estimation typically analyze the face as a whole [201]. They usually estimate temporal changes of facial appearance or facial feature points extracted from the entire face (e.g., [71, 86, 108, 184]). The only exceptions include the part-based methods for detecting facial actions units (AUs) [92, 159], classifying basic emotion categories [200, 205], and pain classification [105, 132]. However, these approaches are not suitable for our problem due to the following limitations. None of the works except [86] performs intensity estimation. The methods of [92, 159] identify important facial parts for detecting AUs, but they do not account for interactions between the parts. Consequently, they underperform in the case when two (or more) AUs simultaneously co-occur — which is quite frequent in spontaneous facial expressions — since this modifies the appearance of facial parts relative to single AU occurrences. Also, the work of [200] seems inappropriate for our purposes, because of its poor trade-off between complexity and accuracy. It uses a computationally expensive graph matching for identifying relevant facial parts and their relationships for emotion categorization. [205] is a stage-wise approach, which first selects the patches using Multi-task sparse learning and then does classification using SVM. Finally, the methods in [105, 132] select features for pain classification. However, the selection is done independently as a preprocessing step and features are selected arbitrarily rather than in spatial groups.

Kernel-based approaches to event recognition use a kernel function that maps the original feature space to a more suitable, "kernelized" space for recognition. Standard kernel-based methods, including the RVM [174] used in Chap. 7, pre-define the kernel function before learning. Our DSRVM is related to recent methods for Multiple Kernel Learning (MKL), where the goal is to learn an optimal way to combine several kernel functions [66]. We define each kernel based on a different facial feature and thus MKL is able to learn relevant facial parts. Existing MKL methods are mainly aimed at classification problems. Only a few MKL methods address regression [145, 147, 170]. These regression MKL methods use Support Vector Regression (SVR) as a base learner, and seek to learn the weights of a linear combination of the kernels. To this end, the method of [145] uses a domain-specific heuristic, which is not generalizable to other domains, and thus seems unsuitable for our purposes. The methods of [147, 170] jointly learn SVR and kernel weights via semi-infinite linear programming [170], and gradient descent [147], and thus induce prohibitively long running times. By contrast, our DSRVM uses a computationally efficient EM algorithm, and significantly reduces running time of learning relative to the existing regression MKL methods. In addition, the above related work enforces sparsity only in the primal domain, without regularizing the total number of resulting relevance vectors. Our DSRVM is doubly sparse by identifying only a few relevant

kernels and a few relevance vectors. The additional sparsity in the kernel domain leads to (1) improved runtime, since fewer kernels need to be evaluated and (2) improved generalization ability, since potentially uninformative kernels can be pruned out.

In the sequel, Sec. 8.2 formalizes DSRVM; Sec. 8.3 explains our differences from RVM; Sec. 8.4 presents our differences from related MKL methods; Sec. 8.5 specifies kernels that we use for DSRVM regression; Sec. 8.6 describes the experimental setup and results; and Sec. 8.7 presents our concluding remarks.

8.2 The Model

This section specifies our DSRVM which is aimed at the following regression problem. Suppose we are given training video frames showing spontaneous facial expressions, $\mathcal{D} = \{(\mathbf{x}_n, t_n)\}_{n=1}^{N}$, where $\mathbf{x}_n$ is a feature vector, and t_n is the associated target value corresponding to the intensity level of a person's emotional experience (e.g., real-valued pain). Our goal is to find a function, y, that models $y(\mathbf{x}) = t$ for any $(\mathbf{x}, t)$ pair.

For regression, DSRVM uses a weighted sum of M basis functions, $y(\mathbf{x}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x})$, where $\mathbf{w} = [w_1 ... w_M]^\top$ weight the contribution of basis functions $\{\phi_m\}_{m=1}^{M}$ in the sum. The mth basis is computed by centering a kernel $\boldsymbol{\kappa}$ at the mth training data point, $\phi_m(\mathbf{x}) = \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_m)$ and $M = N$. The kernel $\boldsymbol{\kappa}$ is defined as

$$\boldsymbol{\kappa} = \sum_{k=1}^{K} v_k \kappa_k, \tag{8.1}$$

where $\{\kappa_k\}_{k=1}^{K}$ is a set of predefined kernels, and $\mathbf{v} = [v_1 ... v_K]^\top$ are their corresponding weights. κ_k could be any kernel function, like Radial Basis Function (RBF), and histogram intersection kernels. There is no restriction to Mercer kernels, as in Support Vector Machines [169].

Thus, DSRVM defines the regression function y as

$$y(\mathbf{x}; \mathbf{w}, \mathbf{v}) = \sum_{m=1}^{M} \sum_{k=1}^{K} w_m v_k \kappa_k(\mathbf{x}, \mathbf{x}_m). \tag{8.2}$$

DSRVM is a *doubly sparse* model, because learning seeks to identify a small subset of non-zero weights w_m and v_k, whereas the remaining weights are set to zero. This means that a sparse set of basis functions ϕ_m will be used for regression. Since each ϕ_m is associated with $\mathbf{x}_m$, the training data with nonzero weights in (8.2) are called the relevance vectors (RV). Following this convention, the κ_k with non-zero weights are called the *relevance kernels* (RK).

DSRVM solves the regression model defined by (8.2) in a Bayesian way, and therefore the next step is to define probability distributions for the error and the parameters of (8.2). We

assume an additive Gaussian error ϵ with zero mean and variance σ^2, i.e., $t = y(\mathbf{x}; \mathbf{w}, \mathbf{v}) + \epsilon$ and thus:

$$p(t|\mathbf{w}, \mathbf{v}, \sigma^2, \mathbf{x}) \sim \mathcal{N}\left(t; y(\mathbf{x}; \mathbf{w}, \mathbf{v}), \sigma^2\right), \tag{8.3}$$

where $\mathcal{N}\left(t; y, \sigma^2\right)$ denotes the Gaussian distribution over the variable t with mean y and variance σ^2. Note that (8.3) holds for all training $(\mathbf{x}_n, t_n)$ pairs as well as the test data $(\mathbf{x}_{\text{new}}, t_{\text{new}})$. Furthermore, we assume independence between the observations, i.e., $p(\mathbf{t}|\mathbf{w}, \mathbf{v}, \sigma^2, \mathbf{X}) = \prod_n p(t_n|\mathbf{w}, \mathbf{v}, \sigma^2, \mathbf{x}_n)$, where $\mathbf{t} = [t_1, ...t_N]^\top$ and $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$. In order to enforce sparse weights, we define a hierarchical Gaussian prior over $\mathbf{w}$ and $\mathbf{v}$:

$$p(\mathbf{w}|\boldsymbol{\alpha}) \sim \mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{A}^{-1}) \tag{8.4}$$

$$p(\mathbf{v}|\boldsymbol{\beta}) \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{B}^{-1}) \tag{8.5}$$

with the hyper-parameters $\mathbf{A} = \text{diag}(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_M]^\top$ and $\mathbf{B} = \text{diag}(\boldsymbol{\beta})$, $\boldsymbol{\beta} = [\beta_1, ..., \beta_K]^\top$. Furthermore, we assume a uniform prior distribution for the hyper-parameters. When integrated out, the hierarchical prior leads to a sparse distribution over $\mathbf{w}$ and $\mathbf{v}$ [174]. A plates diagram of the model is depicted in Fig. 8.1. A full Bayesian treatment of the model

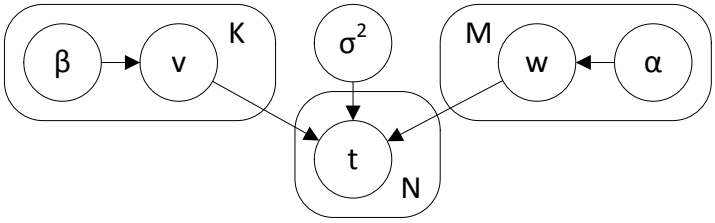


Figure 8.1: Plates diagram of the model.

would lead to the predictive distribution for a new target t_{new}, given the features $\mathbf{x}_{\text{new}}$:

$$p(t_{\text{new}}|\mathbf{t}, \mathbf{X}, \mathbf{x}_{\text{new}}) = \int p(t_{\text{new}}|\boldsymbol{\Omega}, \mathbf{x}_{\text{new}}) p(\boldsymbol{\Omega}|\mathbf{t}, \mathbf{X}) d\boldsymbol{\Omega} \tag{8.6}$$

where $\boldsymbol{\Omega} = (\mathbf{w}, \mathbf{v}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2)$ is the set of all parameters. Hence, the training procedure needs to find the posterior distribution $p(\boldsymbol{\Omega}|\mathbf{t}, \mathbf{X})$. Since this posterior is intractable without further assumptions, we employ a type-II maximum likelihood (ML) estimate of the hyper-parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and a ML estimate for σ^2. To improve readability, we leave out the conditioning on $\mathbf{X}$ in the following. The posterior composes into $p(\boldsymbol{\Omega}|\mathbf{t}) = p(\mathbf{w}, \mathbf{v}|\mathbf{t}, \boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2) p(\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2|\mathbf{t})$, where $\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2$ are the ML estimates of the corresponding parameters. The joint posterior of the weight parameters $(\mathbf{w}, \mathbf{v})$ cannot be explicitly calculated and hence is approximated by a variational distribution that factorizes regarding $\mathbf{w}$ and $\mathbf{v}$:

$$p(\mathbf{w}, \mathbf{v}|\mathbf{t}, \boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2) \approx q(\mathbf{w}) q(\mathbf{v}), \tag{8.7}$$

where $q(\mathbf{w})$ and $q(\mathbf{v})$ are arbitrary distributions whose explicit form is derived in the following part. Since t does not depend on $\boldsymbol{\alpha}_*$ and $\boldsymbol{\beta}_*$ if the posterior of $\mathbf{w}$ and $\mathbf{v}$ is given, the predictive distribution (8.6) can be approximated by

$$p(t_{\text{new}}|\mathbf{t}, \mathbf{x}_{\text{new}}) \approx \int p(t_{\text{new}}|\mathbf{w}, \mathbf{v}, \sigma_*^2, \mathbf{x}_{\text{new}}) q(\mathbf{w}) q(\mathbf{v}) d\mathbf{w} d\mathbf{v} \tag{8.8}$$

and the variational lower bound of the marginal log-likelihood $p(\mathbf{t}|\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2)$ is

$$\mathcal{L} = \int q(\mathbf{w}) q(\mathbf{v}) \log \left(\frac{p(\mathbf{w}, \mathbf{v}, \mathbf{t}|\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2)}{q(\mathbf{w}) q(\mathbf{v})} \right) d\mathbf{w} d\mathbf{v}. \tag{8.9}$$

The DSRVM training algorithm maximizes the approximated log-likelihood $\mathcal{L}$ by repeating 5 update steps:

Step 1: Re-estimate $q(\mathbf{w})$
Step 2: Re-estimate $q(\mathbf{v})$
Step 3: Optimize $\boldsymbol{\alpha}_*$
Step 4: Optimize $\boldsymbol{\beta}_*$
Step 5: Optimize σ_*^2

These steps are a mix of Variational Inference and Expectation-Maximization, since we optimize jointly $q(\mathbf{w})$ and $q(\mathbf{v})$ with variational methods and maximize $\boldsymbol{\alpha}_*$, $\boldsymbol{\beta}_*$ and σ_*^2 given the expectations of $q(\mathbf{w})$ and $q(\mathbf{v})$. In the following, we derive the update formulas for each step of the DSRVM algorithm:

Step 1: In order to derive a variational update formula for $q(\mathbf{w})$, we need to solve the expectation of the joint distribution with respect to $q(\mathbf{v})$, since $\log q^\star(\mathbf{w}) \sim \mathbb{E}_{q(\mathbf{v})} \left[\log p(\mathbf{w}, \mathbf{v}, \mathbf{t}|\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2) \right]$, where $q^\star(\mathbf{w})$ is the optimal distribution over all possible $q(\mathbf{w})$. This leads to the explicit solution:

$$\begin{aligned} q^\star(\mathbf{w}) &\sim \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \\ \boldsymbol{\mu} &= \sigma^{-2} \boldsymbol{\Sigma} \left(\textstyle\sum_k \mathbb{E}[v_k] \, \boldsymbol{\Phi}_k^\top \right) \mathbf{t}, \\ \boldsymbol{\Sigma} &= \left(\mathbf{A} + \sigma^{-2} \textstyle\sum_{k_1} \sum_{k_2} \mathbb{E}[v_{k_1} v_{k_2}] \, \boldsymbol{\Phi}_{k_1}^\top \boldsymbol{\Phi}_{k_2} \right)^{-1}, \end{aligned} \tag{8.10}$$

where $\mathbb{E}[\cdot]$ is the expectation regarding the posterior distribution in (8.7), and $\boldsymbol{\Phi}_k \in \mathbb{R}^{N \times M}$ is the kth matrix slice along the 3rd dimension of the multi-kernel design tensor $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N \times M \times K}$ with $\boldsymbol{\mathcal{K}}(n, m, k) = \kappa_k(\mathbf{x}_n, \mathbf{x}_m)$. Since $q(\mathbf{v})$ is Gaussian as well, computation of $\mathbb{E}[v_k]$ and $\mathbb{E}[v_{k1} v_{k2}]$ is straightforward.

Step 2: Analogously, we derive the update formula for $q(\mathbf{v})$:

$$\begin{aligned} q^\star(\mathbf{v}) &\sim \mathcal{N}(\mathbf{v}; \boldsymbol{\nu}, \boldsymbol{\Lambda}), \\ \boldsymbol{\nu} &= \sigma^{-2}\boldsymbol{\Lambda}\left(\textstyle\sum_m \mathbb{E}\left[w_m\right] \boldsymbol{\Psi}_m^\top\right)\mathbf{t}, \\ \boldsymbol{\Lambda} &= \left(\mathbf{B} + \sigma^{-2}\textstyle\sum_{m_1}\sum_{m_2}\mathbb{E}\left[w_{m_1}w_{m_2}\right]\boldsymbol{\Psi}_{m_1}^\top\boldsymbol{\Psi}_{m_2}\right)^{-1}, \end{aligned} \tag{8.11}$$

where $\boldsymbol{\Psi}_m \in \mathbb{R}^{N\times K}$ is the mth matrix slice along the 2nd dimension of the kernel design tensor $\boldsymbol{\mathcal{K}}$.

Step 3: In order to get an efficient update rule for $\boldsymbol{\alpha}_*$, we further approximate $q(\mathbf{v})$ with a delta function at its mode $\boldsymbol{\nu}$. Thus, the marginal likelihood is approximated by:

$$p(\mathbf{t}|\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2) \approx p(\mathbf{t}|\mathbf{v}, \boldsymbol{\alpha}_*, \sigma_*^2)\delta(\mathbf{v}). \tag{8.12}$$

Taking into account a uniform prior for $\boldsymbol{\alpha}$ and σ^2, and following the same update formula as for the original RVM [174], the solution is a convolution of Gaussians and can be expressed in closed form

$$p(\mathbf{t}|\mathbf{v}, \boldsymbol{\alpha}_*, \sigma_*^2) \sim \mathcal{N}(\mathbf{t}; \mathbf{0}, \mathbf{C}_\mathbf{v}), \tag{8.13}$$

where $\mathbf{C}_\mathbf{v} = \boldsymbol{\Phi}_\mathbf{v}\mathbf{A}^{-1}\boldsymbol{\Phi}_\mathbf{v}^\top + \sigma^2\mathbf{I}$ and $\boldsymbol{\Phi}_\mathbf{v} \in \mathbb{R}^{N\times M}$ with $\boldsymbol{\Phi}_\mathbf{v}(n,m) = \sum_k v_k\kappa_k(\mathbf{x}_n, \mathbf{x}_m)$. As described in [175], we can derive an optimal update for each α_m separately as

$$\alpha_m^* \begin{cases} = \frac{a_m^2}{b_m^2 - a_m} & \text{if } b_m^2 > a_m \\ \to \infty & \text{otherwise} \end{cases}, \tag{8.14}$$

where $a_m = \phi_m^\top\mathbf{C}_{-m}^{-1}\phi_m$, $b_m = \phi_m^\top\mathbf{C}_{-m}^{-1}\mathbf{t}$, $\mathbf{C}_{-m} = \sigma^2\mathbf{I} + \sum_{i\neq m}\alpha_i^{-1}\phi_i\phi_i^\top$ and ϕ_m is the mth column of $\boldsymbol{\Phi}_\mathbf{v}$ with $\phi_m(n) = \sum_k v_k\kappa_k(\mathbf{x}_n, \mathbf{x}_m)$. Note that in each iteration only α_m with the largest likelihood increase is updated, for details see [175]. Setting α_m to infinite effectively prunes out the corresponding basis function ϕ_m, i.e., only the basis with $\alpha_m < \infty$ are used for inference.

Step 4: For deriving an update formula for $\boldsymbol{\beta}_*$, we follow the same reasoning as in step 3. We approximate $q(\mathbf{w})$ with a delta function at its mode $\boldsymbol{\mu}$. Then, we approximate the marginal likelihood as

$$p(\mathbf{t}|\boldsymbol{\alpha}_*, \boldsymbol{\beta}_*, \sigma_*^2) \approx p(\mathbf{t}|\mathbf{w}, \boldsymbol{\beta}_*, \sigma_*^2)\delta(\mathbf{w}). \tag{8.15}$$

and analogously to step 3, we need to maximize

$$p(\mathbf{t}|\mathbf{w}, \boldsymbol{\beta}_*, \sigma_*^2) \sim \mathcal{N}(\mathbf{t}; \mathbf{0}, \mathbf{D}_\mathbf{w}), \tag{8.16}$$

where $\mathbf{D}_\mathbf{w} = \mathbf{\Psi}_\mathbf{w}\mathbf{B}^{-1}\mathbf{\Psi}_\mathbf{w}^\top + \sigma^2\mathbf{I}$ and $\mathbf{\Psi}_\mathbf{w} \in \mathbb{R}^{N\times K}$ with $\mathbf{\Psi}_\mathbf{w}(n,k) = \sum_m w_m\kappa_k(\mathbf{x}_n,\mathbf{x}_m)$. As above, we can derive an optimal update for each β_k separately as

$$\beta_k^* \begin{cases} = \frac{c_k^2}{d_k^2 - c_k} & \text{if } d_k^2 > c_k \\ \to \infty & \text{otherwise} \end{cases}, \tag{8.17}$$

where $c_k = \psi_k^\top \mathbf{D}_{-k}^{-1}\psi_k$, $d_k = \psi_k^\top \mathbf{D}_{-k}^{-1}\mathbf{t}$, $\mathbf{D}_{-k} = \sigma^2\mathbf{I} + \sum_{i\neq k}\beta_i^{-1}\psi_i\psi_i^\top$ and ψ_k is the kth column of $\mathbf{\Psi}_\mathbf{w}$ with $\psi_k(n) = \sum_m w_m\kappa_k(\mathbf{x}_n,\mathbf{x}_m)$. Again, note that only β_k with the largest likelihood increase is updated in each iteration.

Step 5: We further derive an update for the noise variance σ_*^2, by solving the derivative regarding the marginal likelihood $\frac{\partial\mathcal{L}}{\partial\sigma^2} = 0$. This leads to the formula:

$$\sigma_*^2 = \tfrac{1}{N}\mathbb{E}_{\mathbf{w},\mathbf{v}}\left[\|\mathbf{t} - y(\mathbf{X};\mathbf{w},\mathbf{v})\|^2\right], \tag{8.18}$$

$$\begin{aligned} &\mathbb{E}_{\mathbf{w},\mathbf{v}}\left[\|\mathbf{t} - y(\mathbf{X};\mathbf{w},\mathbf{v})\|^2\right] \\ &= \|\mathbf{t} - y(\mathbf{X};\mathbb{E}[\mathbf{w}],\mathbb{E}[\mathbf{v}])\|^2 + \sum_{m_1,m_2,k_1,k_2} \mathbf{\Sigma}(m_1,m_2)\mathbf{\Lambda}(k_1,k_2)\sum_n \kappa_{n,m_1,k_1}\kappa_{n,m_2,k_2} \end{aligned} \tag{8.19}$$

with $y(\mathbf{X};\mathbf{w},\mathbf{v}) = [y(\mathbf{x}_n;\mathbf{w},\mathbf{v})]_{n=1}^N$ and $\kappa_{n,m,k} = \kappa_k(\mathbf{x}_n,\mathbf{x}_m)$.

Summary: Our DSRVM algorithm is summarized in Alg. 8.1. Interleaving the updates of $q(\mathbf{w})$ and $q(\mathbf{v})$ will improve the approximation in (8.7). We first update $\boldsymbol{\alpha}$, followed by r updates of $q(\mathbf{w})$ and $q(\mathbf{v})$. Then we update $\boldsymbol{\beta}$, followed by r updates of $q(\mathbf{v})$ and $q(\mathbf{w})$. Each of the above $q(\mathbf{w})$ and $q(\mathbf{v})$ updates is followed by a σ^2 update. Any other order of the updates would be a valid algorithm, however this order has been chosen for several reasons: (1) Part of the statistics that is neccessary for updating σ^2 is already calculated at the $q(\mathbf{w})$ and $q(\mathbf{v})$ steps, therefore we can follow up with a σ^2 update at low additional cost. (2) The $\boldsymbol{\alpha}$ step depends only on $(q(\mathbf{v}),\sigma^2)$ and not on $(q(\mathbf{w}),\boldsymbol{\beta})$. Therefore any $(q(\mathbf{w}),\boldsymbol{\beta})$ update immediately before the $\boldsymbol{\alpha}$ step would be inefficient. The same reasoning holds for any $(q(\mathbf{v}),\boldsymbol{\alpha})$ update immediately before the $\boldsymbol{\beta}$ step. (3) Interleaving r updates of $q(\mathbf{w})$ and $q(\mathbf{v})$ between the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ updates improves robustness, because it improves the approximation in (8.7) and hence the approximation of $\delta(\mathbf{w})$ and $\delta(\mathbf{v})$.

Initialization: First we initialize σ_*^2 with the variance of the targets $\mathbf{t}$. Then we select a single basis and a single kernel, i.e., setting all α_m and β_k to infinite except one and thus initially $M_\text{rel} = K_\text{rel} = 1$. The selection process first calculates the inner product between $\mathbf{t}$ and all possible single basis/kernel combinations. Then we select randomly from the 50% of

(m, k) pairs with the largest inner product. The optimal α_m for the selected (m, k) pair can be calculated in closed form when assuming $v_k = 1$ and β_k can be calculated for $w_m = 1$.

Complexity: The space and time complexity of the DSRVM algorithm depends highly on the number of relevance vectors $M_{\text{rel}} = |\{\alpha_m : \alpha_m < \infty\}|$ and the number of relevance kernels $K_{\text{rel}} = |\{\beta_k : \beta_k < \infty\}|$. Due to the sparse hierarchical prior on $\mathbf{w}$ and $\mathbf{v}$, it follows that $M_{\text{rel}} \ll \max(M, K)$ and $K_{\text{rel}} \ll \max(M, K)$. Then the time complexity of all five training steps is in $O(M_{\text{rel}}^3 + K_{\text{rel}}^3 + M_{\text{rel}}^2 K_{\text{rel}}^2 N + M^2 N + K^2 N)$. The space complexity is in $O(MKN + M_{\text{rel}}^2 K_{\text{rel}}^2)$, i.e., it is mainly influenced by the $M \times K \times N$ gram matrix. Testing only involves the evaluation of (8.2) once, i.e., the time and space complexity is both in $O(M_{\text{rel}} K_{\text{rel}})$. Note that only the update steps 3 and 4 can change M_{rel} and K_{rel} and the difference is at most ± 1, i.e. the growth is at most linear in the number of iterations.

Algorithm 8.1. DSRVM learning algorithm

```
initialize q(w), q(v), α, β, σ²
while not converged do
    update α as in Step 3
    update q(w) as in Step 1 and σ² as in Step 5
    for r times do
        update q(v) as in Step 2 and σ² as in Step 5
        update q(w) as in Step 1 and σ² as in Step 5
    end for
    update β as in Step 4
    update q(v) as in Step 2 and σ² as in Step 5
    for r times do
        update q(w) as in Step 1 and σ² as in Step 5
        update q(v) as in Step 2 and σ² as in Step 5
    end for
end while
return q(w), q(v)
```

Next, we derive the predictive distribution for new data. Therefore, we need to solve (8.8), which is possible because it is a convolution of Gaussians:

$$p(t_{\text{new}}|\mathbf{t}, \mathbf{x}_{\text{new}}) \approx \mathcal{N}(t_{\text{new}}; y_{\text{new}}, \sigma_{\text{new}}^2), \tag{8.20}$$

where $y_{\text{new}} = y(\mathbf{x}_{\text{new}}; \mathbb{E}[\mathbf{w}], \mathbb{E}[\mathbf{v}])$. We need only y_{new} to make predictions and do not compute σ_{new}^2.

8.3 DSRVM vs. RVM

Our DSRVM extends RVM [174, 175]. RVM cannot identify and account for relevant facial parts. Consequently, RVM is bound to confuse distinct facial expressions sharing the same movements of specific facial parts.

The key difference between our DSRVM and RVM is that RVM uses a single, unique kernel for regression, centered at each training data point:

$$y_{\mathrm{RVM}}(\mathbf{x};\mathbf{w}) = \sum_{m=1}^{M} w_m \kappa(\mathbf{x},\mathbf{x}_m). \tag{8.21}$$

RVM seeks to learn a small subset of non-zero weights w_m associated with relevance vectors $\mathbf{x}_m$. By comparing (8.2) and (8.21), it follows that RVM does not have an explicit mechanism for additionally enforcing sparsity over the features of relevance vectors.

The RVM assumes a Gaussian distributed noise and independently distributed targets as in (8.3), while y is defined as in (8.21). The prior of $\mathbf{w}$ is defined as in (8.4). SVR has the same regression function as RVM (8.21), but does not assume any prior over $\mathbf{w}$. SVR achieves sparsity from the loss function, which is defined by the ϵ-insensitive loss (an adaption of the Hinge-loss for classification) instead of Gaussian noise and thus results in zero weights for all support vectors outside the ϵ-boundary.

The RVM kernel is fixed and hence there are no kernel weights $\mathbf{v}$ included in the model. As a result, learning of RVM simplifies only to maximizing the marginal likelihood $\mathcal{L}_{\mathrm{RVM}} = p(\boldsymbol{\alpha}_*, \sigma_*^2|\mathbf{t})$, under the assumptions that the prior of $(\boldsymbol{\alpha}_*, \sigma_*^2)$ is uniform. To this end, learning of RVM iterates three steps until convergence of $\mathcal{L}_{\mathrm{RVM}}$:

Step a: Re-estimate $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_*, \sigma_*^2)$
Step b: Optimize $\boldsymbol{\alpha}_*$
Step c: Optimize σ_*^2

In the following, we describe each of the RVM steps in detail and compare it with our DSRVM update steps.

Step a: $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_*, \sigma_*^2)$ is a convolution of two Gaussians and can hence be calculated in

closed form:

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_*, \sigma_*^2) &\sim \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \\
\boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top \mathbf{t}, \\
\boldsymbol{\Sigma} &= \left(\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right)^{-1},
\end{aligned}
\tag{8.22}
$$

where $\boldsymbol{\Phi}$ is the kernel basis matrix with $\boldsymbol{\Phi}(n, m) = \kappa(\mathbf{x}_n, \mathbf{x}_m)$. It follows that RVM uses a linear function of the targets to estimate its parameters $\mathbf{w}_{\text{RVM}} = (\sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top)\mathbf{t} = L\mathbf{t}$. While the convexity and closed-form of such linear RVM formulation is appealing, the use of the linear function strongly restricts the complexity of data that RVM can represent. By contrast, our DSRVM introduces kernel weights $\mathbf{v}$ that play a role of hidden variables in the estimation of $\mathbf{w}_{\text{DSRVM}} = \sigma^{-2}\boldsymbol{\Sigma}_\mathbf{v}\boldsymbol{\Phi}_\mathbf{v}^\top\mathbf{t}$ (see (8.10)), where each configuration of $\mathbf{v}$ values corresponds to a particular component in the exponentially large mixture of distributions of $\mathbf{w}_{\text{DSRVM}}$. This significantly extends the modeling capacity of our DSRVM relative to that of RVM.

Step b: The marginal likelihood $\mathcal{L}_{\text{RVM}}$ can be maximized regarding $\boldsymbol{\alpha}_*$ by the same update formula as in (8.14), except that $\phi_m(n) = \kappa(\mathbf{x}_n, \mathbf{x}_m)$ due to the single fixed kernel.

Step c: Optimizing σ_*^2 by taking the corresponding derivative of $\mathcal{L}_{\text{RVM}}$ leads to the update formula:

$$
\sigma_*^2 = \frac{\|\mathbf{t} - y_{\text{RVM}}(\mathbf{X}; \mathbb{E}[\mathbf{w}])\|^2}{N - M + \sum_m \alpha_m \boldsymbol{\Sigma}(m, m)}.
\tag{8.23}
$$

The DSRVM σ_*^2 update (8.18) has a different form than (8.23), because the DSRVM target $\mathcal{L}$ is a variational approximation while $\mathcal{L}_{\text{RVM}}$ is Gaussian.

When comparing both sets of update formulas, we see that the RVM update steps a, b and c correspond to the DSRVM update steps 1, 3 and 5, and indeed our DSRVM algorithm includes the RVM algorithm as a special case for a single kernel, i.e., for $K = 1$.

The RVM predictive distribution is a convolution of two Gaussians and thus can be computed in closed form:

$$
p(t_{\text{new}}|\mathbf{t}, \mathbf{x}_{\text{new}}) = \mathcal{N}(t_{\text{new}}; y_{\text{new}}, \sigma_{\text{new}}^2),
\tag{8.24}
$$

with $y_{\text{new}} = y_{\text{RVM}}(\mathbf{x}_{\text{new}}; \mathbb{E}[\mathbf{w}])$. From comparing (8.21) and (8.2) we see that the RVM predictive function y_{RVM} is linear in $\mathbf{w}$, while the DSRVM predictive function y is multi-linear in $\mathbf{w}$ and $\mathbf{v}$.

8.4 DSRVM vs. Related MKL Methods

This sections explains differences between DSRVM and related MKL methods, including SMKL [147] and multi-kernel RVM (mRVM) [22, 33].

SMKL [147] defines the regression function as in (8.2), with the additional constraint of convex kernel combinations, i.e., $\sum_m v_m = 1$ and $v_m \geq 0$. In contrast to DSRVM, the SMKL method does not optimize the basis weights $\mathbf{w}$ and $\mathbf{v}$ within a Bayesian setting, but rather solves a max-margin formulation equivalent to a SVM. The SVM algorithm provides an optimal solution for the basis weights $\mathbf{w}$ given a fixed kernel and the kernel weights $\mathbf{v}$ are optimized by steepest descend. Unfortunately, the evaluation of the descend direction involves repeated executions of the SVM algorithm. Therefore SMKL repeats the SVM algorithm within a nested loop, leading to a large number of repetitions and thus a long training time. Furthermore, the sparsity of the kernel weights $\mathbf{v}$ is only encouraged through the convexity constraint, which is weaker than the hierarchical prior of the DSRVM. The SMKL training step includes a gradient evaluation with $O(KM_{\text{Rel}}^2)$ and SVM solving with $O(M_{\text{Rel}}^3 + M^2N)$ (see e.g., [11]). As for DSRVM, the complexity highly depends on the number of support vectors M_{Rel} and if $M_{\text{rel}} \ll M$, then the dominating term is $O(M^2N)$, which is similar to DSRVM, see Sec. 8.2. In practice, M_{Rel} for DSRVM is lower than for SMKL, and thus the DSRVM training is faster, see the results in Tab. 8.3.

The multi-class and multi-kernel RVM (mRVM) [33] is a RVM extension for classification that defines a shared hierarchical prior $\boldsymbol{\alpha}$ over the basis weights $\mathbf{w}$ for each class. Additionally, mRVM learns the kernel weights $\mathbf{v}$ for a convex combination of kernels. mRVM uses similar update formulas as RVM for $\mathbf{w}$ and additionally optimizes $\mathbf{v}$ by maximizing the marginal log-likelihood with a Quadratic Programming algorithm. As in the case of SMKL, the sparseness of $\mathbf{v}$ is only weakly enforced by the convexity constraint, in contrast to the Bayesian formulation of our DSRVM.

The multi-kernel RVM approaches of [22] and [182] use the same regression MKL formulation as ours (8.2), but combine the basis and kernel weights so that there is a separate weight for each basis and kernel combination. This leads to a large number of weights to learn (MK in comparison to our $M + K$), which makes the method more prone to overfitting and slower to train. We compare our method to this kernel formulation, see Sec. 8.5. While [22] uses the standard RVM to learn the weights, [182] formulates an efficient computation in the Fourier domain for circulant gram matrices. However, this is only possible because their particular application domain is significantly different from ours, since they seek to predict pixel values

from a single image. In the application of this chapter however, the features of each training instance stem from different images and thus the resulting gram matrices are not circulant.

8.5 DSRVM Kernels

In order to define kernels over local facial parts, we use the LBP features described in Chap. 6, which are descriptors of local edge distributions. Since the right space-time location and scale of subvolumes that are relevant for facial behavior estimation are not known a priori, we extract the video subvolumes from a range of spatial and temporal scales. Specifically, we use $S \in \{6, 9\}$ and $T \in \{1, 10, 20\}$, i.e., the face is divided into 6x6 or 9x9 patches while applying time windows of 1, 10 or 20 frames. Thus we obtain the feature vectors $\{\mathbf{x}_k : k = 1, \ldots, K\}$ locally extracted from $K = S^2$ video subvolumes per time window. In this section, we specify how to kernelize these features for DSRVM regression.

From (8.2), given a window with features $\mathbf{x}$ and mth training window with features $\mathbf{x}_m$, DSRVM uses K RBF kernels defined for each of their respective subvolumes $k \in \{1, ..., K\}$:

$$\kappa_k(\mathbf{x}, \mathbf{x}_m; \gamma_k) = \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_{mk}\|^2}{2\gamma_k}\right), \tag{8.25}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{mk}$ are the local features extracted from kth subvolume of the two frame windows.

Each kernel parameter γ_k is estimated independently on training videos over the corresponding features $\mathbf{x}_k$ by a heuristic. For fair comparison of DSRVM with alternative algorithms, we use the same set of kernels.

It is possible to use each of the K kernels as a separate basis function for RVM like in [22,182], which results in MK basis functions and thus in a kernel gram matrix of size $N \times (MK)$. We compare with this approach and call it RVM separate (RVM sep). Note that RVM sep uses the standard RVM algorithm as in [22], since the gram matrix is not circulant and thus the more efficient method of [182] cannot be applied.

8.6 Results

This section presents the experimental setup and shows empirical results.

We evaluate our DSRVM on four datasets: (1) the artificial dataset used for benchmark evaluation of regressors (see Sec. 8.6.1); (2) the ShoulderPain database (see Sec. 4.1); (3)

the DISFA database (see Sec. 4.2); and (4) the SEMAINE database (see Sec. 4.3.4). We additionally chose (1) because it has been used by competing MKL regression methods.

Furthermore, we present experiments on the continuous artificial and SEMAINE datasets that show the advantage of regression models over classifiers when dealing with continuous data in Sec. 8.6.7.

For the non-artificial datasets, we use all frames of testing videos. For training, we equidistantly sub-sample 2000 frames to reduce the training time and memory to a reasonable amount. For the AU recognition experiments, we additionally assure that at least 25% of the training data contains the specific AU which we train for. Since the vast majority of frames within ShoulderPain (40,029) contains no pain (level 0), in training, we remove most frames annotated with pain level 0 from the beginning and the end of each video sequence and keep only 11 frames with pain level 0 before and after each video sequence. The resulting distribution of pain intensity levels is shown in Fig. 8.2.

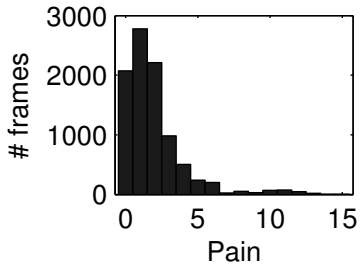


Figure 8.2: ShoulderPain: Frame distribution over pain intensity levels after re-balancing pain level 0. Note the remaining imbalance towards the lower pain levels.

For a pair-wise comparison of our DSRVM with RVM and other methods, we estimate significance of the results using a two-tailed Student's t-test. We report the t-test probability value – p-value – that is minimally needed to reject the null hypothesis (i.e., DSRVM and the comparing method are the same), where low p-values correspond to high significance.

For each database, we measure the performance of DSRVM and competing methods and provide further statistics: the number of selected basis (# RV, for Relevance Vectors), the number of selected kernels (# RK, for Relevant Kernels), and the running times for training (TRN) and testing (TST). TRN is the time needed for training the model on 2,000 data points and TST is the time needed for testing the model. Since the number of testing samples varies between folds and datasets, we divide TST by the number of samples per fold and multiply by 10,000 to represent the time for 10,000 data points. The running times are evaluated on a single core of an Intel Xeon E5-2640 CPU with 64 GB RAM. Note that DSRVM training needs

significantly less memory than 64 GB, usually 4GB is sufficient for 2000 training examples. In order to get a robust result regarding local minima, we report the average result of 10 random initializations as explained in the initialization paragraph of Sec. 8.2. Additionally, we visualize the selected kernels for an intuitive interpretation of the learned model.

DSRVM training is iterative. When we evaluate performance w.r.t. MSE (or CORR, ICC), we use the MSE-based (or CORR-based, ICC-based) convergence criterion for stopping the iterations in training (see Alg. 8.1, line 2). This gives three variants of our DSRVM. The results for all other metrics – namely, #RV, #RK, TRN and TST – are obtained by averaging the different variants of DSRVM. In order to have a fair comparison with the other models, each of them is separately optimized regarding CORR, MSE and ICC.

For the ShoulderPain dataset we do a full evaluation of all space-time scales $S = \{6, 9\}$ and $T = \{1, 10, 20\}$, while for the DISFA and SEMAINE datasets we use a simple baseline and do not account for the temporal extent of changes in facial features (i.e., $S = 6$ and $T = 1$).

8.6.1 Artificial Data

The Sonnenburg et al. [170, p. 1548] regression experiment tests an algorithm's choice of kernels in terms of accuracy and sparsity. The task is to learn the target function $t = \sin(fx) + \sin(x) + x + \epsilon$, where f is the frequency of a varying Sine function with $f \in \{1, 2, \ldots, 20\}$, and ϵ is white Gaussian noise with variance 0.3. The set of kernels consists of 10 RBF with the length-scale parameters $\gamma_k \in \{0.001, 0.005, 0.01, 0.05, 0.1, 1, 10, 50, 100, 1000\}$. The range of length-scale values for the RBF is chosen to correspond to different frequencies of the Sine function, so that an optimal MKL algorithm needs to adapt the used Kernel to the current frequency. Each feature-target pair (x, t) with $x \in \mathbb{R}$ is constructed by randomly sampling x from a uniform distribution in $[0, 10]$. We use 2000 feature-target pairs (x, t), where one half serves for training, and the other half for testing. For evaluation, we use the standard setting of [41, 170, 174]. Specifically, we randomly sample artificial data in order to form 10 sets of data. Each set (a.k.a, fold) is split in half for training and testing. The reported results are averaged across the 10 folds.

8.6.2 Baseline Methods

We compare our DSRVM with three baselines — namely, RVM [174], SMKL [147] and mRVM [33].

RVM is specified in Sec. 8.3. RVM uses a single kernel, and thus we cannot use the expression

for DSRVM kernels given by (8.25). Hence, we use three strategies to compute the RVM kernel. The first strategy, called *RVM-all*, computes the kernel as a sum of *all* DSRVM kernels given by (8.25) with kernel weights $v = 1$. The second strategy, called *RVM-best*, sets the kernel as one of the DSRVM kernels given by (8.25) that gives the best CORR result — it sets the corresponding weight in v to 1 and all others to 0. The third strategy, called *RVM-sep*, sets one dimension in v to 1 and all others to 0 for all possible K dimensions. This leads to MK basis functions, in contrast to M basis functions of the other approaches.

SMKL is well suited for our comparison, since its inference model is the same as that for DSRVM: given the kernel gram-matrix, the estimated target is calculated in a multi-linear operation weighted by the basis and the kernel weights. Furthermore, SMKL is based on support vector regression (SVR) [169], the main competing regression method for RVM [174]. The SVR regression parameter ϵ and cost C have been optimized by a grid-search on training data. For implementing RVM we use the SparseBayes Matlab toolbox [174], and for implementing SMKL we use the SimpleMKL toolbox [147], which uses the SVR solver from the SVM-KM toolbox [18].

mRVM is a multi-class multi-kernel classifier and thus this experiment compares a classifier with continuous regression models. Specifically, we compare with the mRVM-1 as defined by [33], since it is rather similar to DSRVM due to the constructive approach that starts from a single basis function. The targets of the SEMAINE and artificial data are continuous and thus mRVM requires to convert them into classes. We discretize the targets into c classes by dividing the range into c equidistant bins. An inverse transform from the predicted class to a continuous value is needed for evaluation and thus we map each predicted class to the center value of the corresponding bin. The mRVM results are obtained by using the optimal c. Additionally the results for varying c are provided in Sec. 8.6.7. In contrast to that, the targets of the ShoulderPain and DISFA data are discrete and thus no further discretization is needed.

8.6.3 Results on the Artificial Dataset

We conduct the Sonnenburg et al. [170, p. 1548] regression experiment for comparing the kernel choices made by DSRVM and those made by SMKL. For the target artificial dataset, both DSRVM and SMKL use 10 RBF kernels, whose widths γ_k are specified in Sec. 8.6.1. The results are shown in Fig. 8.3. As the frequency of the target function changes, DSRVM adapts the kernel weights so as to tune to the particular frequency. As can be seen, DSRVM learns both positive and negative kernel weights. For DSRVM, negative weights (blue) are usually

paired with positive weighted kernels (red) of similar width γ. Starting with the frequency 1, DSRVM chooses kernel widths 1 and 100. As the frequency increases, the kernel width is shifted toward lower values until the main width is 0.01 for the frequency 20. In contrast, kernel weights learned by SMKL are only positive. From Fig. 8.3, SMKL always selects the smallest width of 0.001.

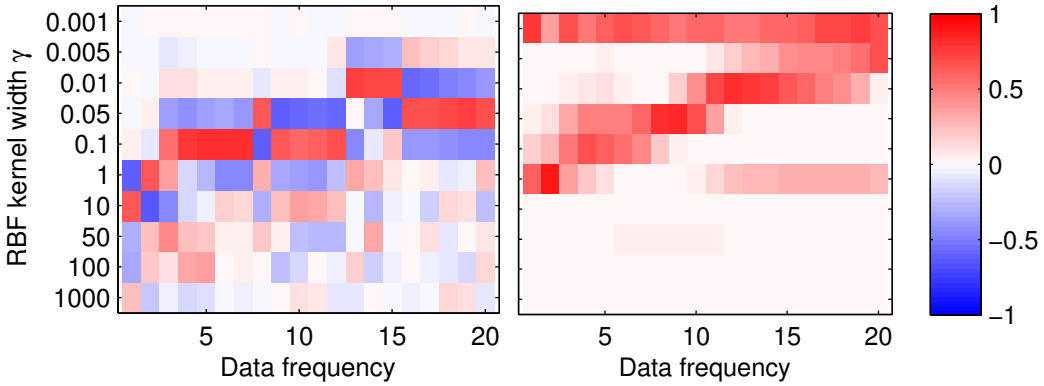


Figure 8.3: Kernel weights learned by DSRVM (left) and SMKL (right) on the Sonnenburg artificial data [170] for the varying frequency of the target function.

Fig. 8.4 compares DSRVM with SMKL, RVM-all and RVM-sep, in terms of their MSE and CORR rates, as well as the number of selected relevance vectors (# RV) and relevance kernels (# RK) on the Artificial dataset. We leave out the only single kernel approach RVM-best to improve the visibility. As can be seen, DSRVM yields better MSE, CORR and ICC rates, and selects significantly fewer RV's than SMKL and RVM-all. The MSE measures error in terms of absolute values of the prediction and ground-truth. Therefore, MSE depends on the scale of the targets, and consequently the artificial data has in general lower MSE results than the ShoulderPain data. Note that the range of the MSE results for the artificial data at frequency 20 is from 0 to 0.06, and thus the DSRVM improvement of 0.02 accounts for 33% of the overall range, which makes the improvement significant.

SMKL selects fewer RKs, but its performance is worse than that of DSRVM, since it selects the smallest kernel width (see Fig. 8.3). Selecting smaller kernel widths allows for more fine grained modeling, but then more RVs are needed. An optimal algorithm selects the width just small enough to model the target function. If unnecessary small widths are selected, then too many RVs are needed and the algorithm is prone to overfitting. DSRVM selects in this case more RKs than SMKL, but the kernel widths are better adjusted to the data, as can be seen from the results.

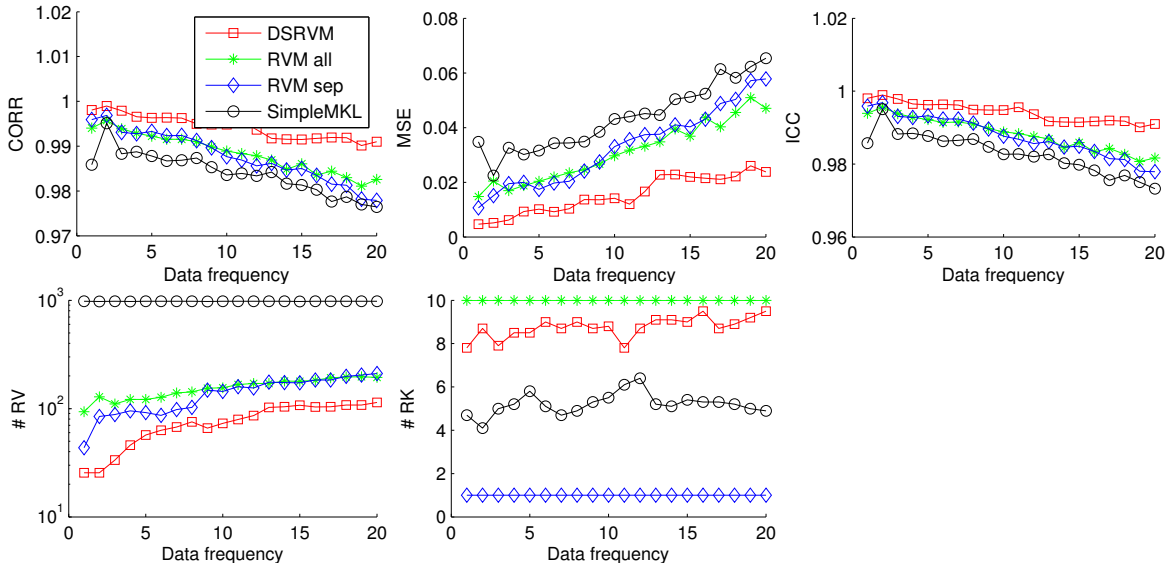


Figure 8.4: Results on the artificial data for a varying frequency of the target function: CORR (top left), MSE (top middle), ICC (top right), the number of selected relevance vectors (#RV) (bottom left), and the number of selected relevant kernels (#RK) with non-zero kernel weights (bottom middle). Note that #RV is shown on the logarithmic scale.

8.6.4 Results on the ShoulderPain Dataset

We carry out two sets of experiments on the ShoulderPain data aimed at testing how (i) the number of training examples, and (ii) changes in space-time scale, affect the performance of the tested models.

In order to show how different models scale with training data, Fig. 8.5 compares results to those of SMKL, RVM-all and RVM-sep for a varying number of training examples. The space-time scale of extracting video features is fixed at a regular grid of 6×6 patches ($S = 6$) per frame, and temporal window of 1 frame ($T = 1$). As can be seen, our accuracy is better in terms of MSE than for the competing approaches, and the CORR is on par with SMKL. The sparse kernel prior of DSRVM brings less advantage in this case, since the facial expression of pain involves both, the upper and lower face and thus is less localized than e.g., specific AUs. Regarding ICC, DSRVM is among the best performing models together with RVM-sep. RVM-all performs better than DSRVM for lower number of training examples, which is expected since it has fewer parameters to learn and thus is less likely to over-fit.

Fig. 8.5 also shows the number of relevance vectors (RV), and the number of relevant kernels (RK) with non-zero kernel weights learned by DSRVM, SMKL, RVM-all and RVM-sep, as the number of training examples increases. Note that RVM cannot select kernels, therefore the

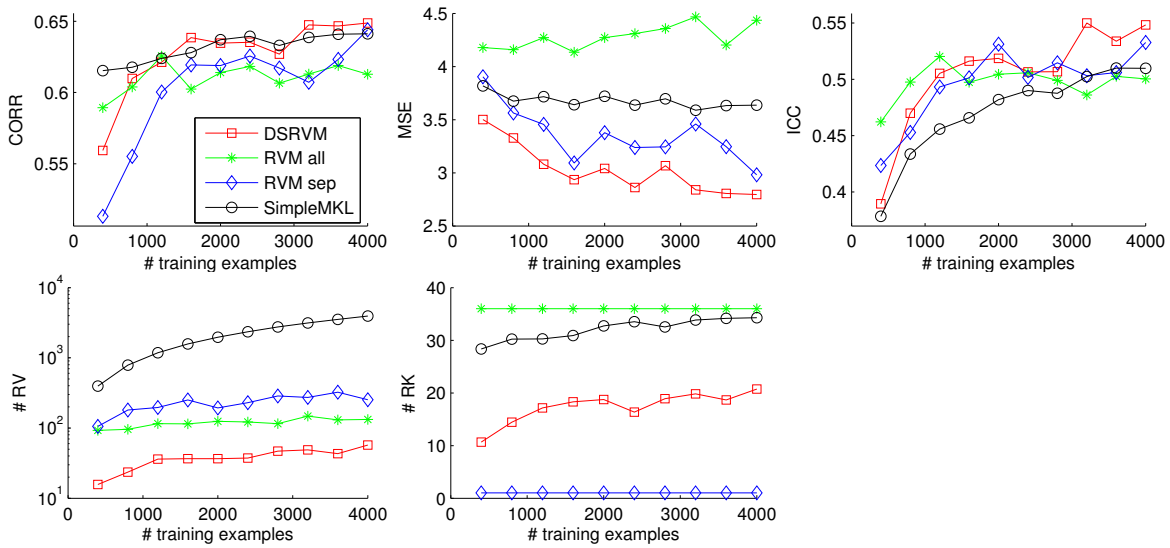


Figure 8.5: Results on the ShoulderPain data for the pain targets and a varying number of training examples: CORR (top left), MSE (top middle), ICC (top right), the number of selected relevance vectors (#RV) (bottom left), and the number of selected relevant kernels (#RK) with non-zero kernel weights (bottom middle). Note that #RV is shown on the logarithmic scale.

graphs for RVM-all and RVM-sep stay constant at the total number of kernels. As can be seen, DSRVM consistently selects fewer RVs and RKs than the other methods. This suggests that the doubly sparse formulation of DSRVM achieves greater sparsity of kernels than the compared methods. In addition, since DSRVM selects significantly fewer RV's than SMKL, DSRVM regression is more computationally efficient than that of SMKL.

Tab. 8.1 compares the results by DSRVM to those by RVM, SMKL and mRVM for different space-time scales. The number of training examples is fixed at 2000. In terms of CORR, DSRVM outperforms SMKL and all RVM variants for the spatial scale set to 9×9 patches ($S = 9$), and all temporal scales $T = \{1, 10, 20\}$. When the spatial scale is set to 6×6 patches ($S = 6$), DSRVM yields a comparable performance to that of SMKL while the RVM variants perform worse. RVM-best is the worst performing with high significance (low p-values). This demonstrates that a single kernel is not sufficient for regression on the ShoulderPain dataset, i.e., a specific face patch is not sufficient to recognize the pain level. In terms of MSE, DSRVM outperforms all methods. Regrading ICC, DSRVM performs best in most of the cases, with some exceptions where it is on par with RVM-all and RVM-sep.

Tab. 8.1 also shows that all regression methods (except the single kernel RVM-best) perform better than the classification mRVM. Classification methods are disadvantaged when applied to intensity estimation, since the inherent value of intensities and their "greater than" and

Table 8.1: Results on the ShoulderPain data for the pain targets. DSRVM is compared to different RVM and SMKL. Video features are extracted at different space-time (S-T) scales. p-value measures significance of the result in comparison to DSRVM. The table shows mean squared error (MSE), the correlation with the targets (CORR) and the Intra-class Correlation Coefficient (ICC). The best results are marked bold. If two results differ by at most 0.01 and the p-value is greater than 0.05, we mark both results bold.

	Method	S6x6 T1	S6x6 T10	S6x6 T20	S9x9 T1	S9x9 T10	S9x9 T20
CORR	DSRVM	**0.63**	**0.66**	**0.63**	**0.63**	**0.63**	**0.62**
	RVM all	0.61	0.62	0.61	0.60	0.59	0.58
	p-value	.22	.07	.18	.03	.08	.10
	RVM best	0.29	0.45	0.45	0.39	0.49	0.46
	p-value	.00	.00	.00	.00	.00	.00
	RVM sep	0.62	0.62	0.60	0.62	0.56	0.59
	p-value	.16	.10	.11	.37	.02	.13
	SMKL	**0.64**	**0.66**	**0.63**	**0.62**	0.60	0.59
	p-value	.49	.20	.14	.23	.07	.04
	mRVM	0.53	0.52	0.45	0.49	0.50	0.50
	p-value	.00	.00	.00	.05	.02	.04
MSE	DSRVM	**3.04**	**2.86**	**3.22**	**3.00**	**3.15**	**3.08**
	RVM all	4.27	4.37	4.55	3.72	4.00	3.98
	p-value	.02	.01	.01	.03	.02	.01
	RVM best	9.85	4.90	3.79	4.16	4.32	4.84
	p-value	.03	.00	.01	.00	.00	.00
	RVM sep	3.38	2.89	4.00	3.37	3.99	4.61
	p-value	.64	.50	.30	.10	.14	.03
	SMKL	3.72	3.69	3.85	3.67	3.78	3.93
	p-value	.03	.02	.03	.03	.01	.01
	mRVM	4.30	4.46	4.49	4.41	4.49	4.58
	p-value	.01	.01	.01	.01	.01	.01
ICC	DSRVM	**0.52**	**0.53**	**0.47**	**0.49**	**0.48**	**0.46**
	RVM all	0.50	0.51	**0.48**	**0.48**	0.45	0.44
	p-value	.75	.89	.63	.20	.46	.26
	RVM best	0.15	0.34	0.38	0.32	0.40	0.37
	p-value	.00	.00	.13	.00	.01	.00
	RVM sep	**0.53**	0.51	0.49	0.47	0.43	0.43
	p-value	.59	.85	.49	.55	.26	.96
	SMKL	0.48	0.49	0.46	0.47	0.46	0.44
	p-value	.03	.06	.07	.13	.10	.07
	mRVM	0.40	0.40	0.36	0.37	0.40	0.41
	p-value	.00	.01	.12	.22	.45	.81

"equal" relationships are not incorporated in a classification model and thus each intensity is modeled as a class on its own.

From Tab. 8.1, for all methods, we observe that the temporal scales of $T = 1$ and $T = 10$ video frames give better results than $T = 20$. This can be explained by research findings in psychology, which suggest that the intensity of pain experience can be encoded from the number of facial actions recruited and their vigor – lower levels of pain are manifested in brow lowering and narrowing of the eyes, while higher levels of pain are manifested by these actions expressed more vigorously and recruiting additional (lower face) actions [31]. Given

that ShoulderPain videos have been recorded at 25 fps, and that facial muscle activation is relatively rapid (onset ranging from 1/16 seconds to 1/3 seconds [149]), a temporal window of 9–10 frames covers the onset of even the slowest facial change. Hence, longer temporal windows (say $T = 20$) cover not only the current pain level but the subsequent one(s) too. Hence, using longer temporal windows leads to more frequent confusion between successive pain levels, as temporally consistent features are learned covering multiple pain levels rather than a single one (as clearly observable from Fig. 8.6 too).

Overall, the spatial scale of 6×6 patches gives the best results. This is, because a patch at the finer spatial scale of 9×9 patches may not provide sufficiently rich spatial information for facial expression estimation. For all space-time scales, DSRVM pain level estimation results are better than those presented in our preliminary work [86], where we reported CORR of 0.59.

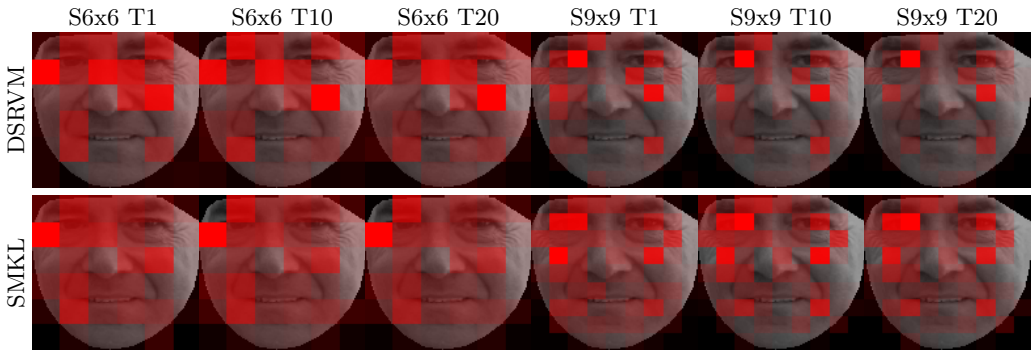


Figure 8.6: ShoulderPain dataset: The values of kernel weights v learned by DSRVM (top row) and SMKL (bottom row) for various spatial (S) and temporal (T) scales are indicated by the intensity of color red of the corresponding patches. Each patch corresponds to one kernel, and the larger the kernel weight the higher is the intensity of the color red per patch. The reddest patches correspond well with the AU definitions presented in [43].

Fig. 8.6 shows a sample video frame from the ShoulderPain dataset, and kernel weights v learned for different space-time scales by DSRVM and SMKL. Each patch of the video frame corresponds to one kernel. We observe that both DSRVM and SMKL select similar patches with large kernel weights as relevant for shoulder-pain-level estimation. These patches fall mainly on the facial areas around the eyes, nose and mouth corners. As already explained above, these results agree with the well-known definition of the facial expression of pain [194], Including brow lowering (AU4) and narrowing of the eyes (AU7) as well as additional facial action such as upward lip pull (AU12). Fig. 8.6 also shows that DSRVM learns sparser kernel weights than SMKL (i.e., fewer patches are selected as relevant for pain-level estimation).

8.6.5 Results on the DISFA Dataset

Tab. 8.2 shows the results attained by DSRVM, SMKL, RVM-all, RVM-best and RVM-sep on the DISFA dataset, for different AUs. Additionally, we compare to the method of [121] on our feature set. [121] learns a low-dimensional manifold with Spectral Regression (SR) [17], followed by SVM classification [20]. The SR step includes training and testing subjects and thus is not subject-independent. In order to have a fair comparison, we use the same subject-independent setting as for the other methods. We run SR followed by SVM (SR+SVM) and the results are shown in Tab. 8.2. For completeness, we also included a comparison to DSRVM within the subject-dependent setting of [121] in Tab. 8.4.

Within Tab. 8.2, DSRVM gives the best CORR for most AUs, except for AU 5 and 15. For AU 2 and 12, the DSRVM is on par with the best result, since the p-value is large in both cases. RVM-best yields better CORR for AU5, because the "upper eye lid raise" occurs at a small facial area, unlike other AUs, which can be "covered" with a single kernel. Either DSRVM or mRVM are the best regarding MSE for most AUs, while DSRVM is better than mRVM on average. Again, DSRVM selects fewer RV's and RK's than SMKL. Specifically, it selects half of the kernels selected by SMKL as being relevant for regression, thereby achieving twice greater sparsity than SMKL. This leads to 6–7 times faster training, and 20 times faster test runtimes of DSRVM relative to those of SMKL. SR+SVM training is fast since the SVM is applied to the low-dimensional manifold. However, the performance is relatively low, which is probably caused by overfitting the manifold to the training subjects. The SR+SVM results are much lower than in [121], due to the subject-independent evaluation.

The best MSE score is reached for DSRVM at AU5, however the CORR for the same AU is relatively low with 0.17. This difference within CORR and MSE stems from the bias of the AU intensity distribution within the DISFA data. AU5 occurs rarely within the data in comparison with e.g., AU4. Therefore a model can reach a good MSE by conservatively rating closer to intensity 0, even if a few high intensity AU events are missed. In contrast to that, CORR is a relative measure and highly penalizes the score if high intensity AU events are missed. Therefore it is possible that CORR and MSE show different trends, since they measure different aspects of the differences between predictions and targets. The same effect can be seen in Fig. 8.5, where DSRVM is on par with SimpleMKL regarding CORR, but outperforms SimpleMKL regarding MSE.

Regarding ICC, DSRVM is on par with RVM-sep. Each of the methods is the best for half of the AU targets. On average they perform similar, followed by RVM-all and SMKL. Although

Table 8.2: Results on the DISFA data for different AU targets. See the caption of Tab. 8.1. AVG is the average results of all AUs.

	Method	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	AVG
CORR	DSRVM	0.31	**0.28**	**0.54**	0.17	**0.57**	**0.43**	**0.80**	0.32	**0.40**	**0.23**	**0.66**	**0.42**	**0.43**
	RVM all	0.26	**0.29**	0.41	0.12	0.46	0.32	0.75	0.33	**0.40**	0.19	0.62	0.40	0.38
	p-value	.11	1.00	.00	.05	.02	.00	.05	.49	.80	.07	.34	.59	.01
	RVM best	0.18	0.15	0.46	**0.19**	0.40	0.30	0.73	0.18	0.21	0.16	0.61	0.17	0.31
	p-value	.00	.01	.07	.16	.03	.03	.01	.01	.00	.00	.11	.00	.00
	RVM sep	**0.35**	0.27	0.50	0.17	0.47	0.34	0.78	**0.34**	0.38	0.21	0.61	0.40	0.40
	p-value	.42	.24	.18	.68	.03	.03	.06	.16	.61	.14	.08	.49	.04
	SMKL	0.26	0.20	0.45	0.17	0.49	0.36	**0.81**	0.26	0.36	0.22	**0.66**	0.40	0.39
	p-value	.06	.01	.02	.94	.06	.02	.26	.05	.14	.43	.93	.47	.00
	mRVM	0.21	0.16	0.32	0.10	0.39	0.40	0.69	0.23	0.32	**0.24**	0.59	0.23	0.32
	p-value	.25	.17	.02	.22	.02	.57	.00	.22	.17	.97	.00	.03	.00
	SR+SVM	0.13	0.13	0.33	0.04	0.40	0.39	0.75	0.21	0.29	0.18	0.47	0.36	0.31
	p-value	.01	.00	.00	.02	.01	.24	.04	.08	.08	.14	.00	.09	.00
MSE	DSRVM	**0.66**	0.59	**0.84**	0.15	**0.42**	**0.42**	0.39	0.19	**0.31**	0.26	1.01	**0.45**	**0.47**
	RVM all	1.20	0.85	1.11	0.27	0.59	0.59	0.42	0.25	0.43	0.35	1.06	0.57	0.64
	p-value	.01	.02	.01	.03	.02	.00	.43	.01	.01	.03	.66	.03	.00
	RVM best	1.28	1.21	1.05	0.30	0.60	0.78	0.48	0.33	0.51	0.42	1.14	0.70	0.73
	p-value	.01	.01	.07	.11	.17	.08	.02	.07	.02	.09	.13	.00	.00
	RVM sep	1.10	0.85	1.09	0.40	0.64	0.67	0.42	0.26	0.45	0.43	1.15	0.73	0.68
	p-value	.01	.00	.10	.10	.00	.04	.06	.00	.01	.00	.24	.00	.00
	SMKL	0.73	0.69	0.92	0.17	0.49	0.46	**0.35**	0.23	0.34	0.24	**0.97**	0.50	0.51
	p-value	.22	.04	.07	.34	.03	.14	.06	.28	.14	.26	.55	.08	.03
	mRVM	**0.66**	**0.57**	1.35	**0.10**	0.57	**0.41**	0.56	**0.17**	0.33	**0.19**	1.36	0.62	0.58
	p-value	.99	.95	.03	.10	.04	.94	.00	.48	.36	.01	.00	.12	.07
	SR+SVM	1.00	1.13	1.52	0.20	0.69	0.50	0.59	0.52	0.45	0.44	1.66	0.58	0.77
	p-value	.13	.07	.01	.34	.01	.13	.00	.28	.02	.08	.00	.05	.00
ICC	DSRVM	0.26	0.22	**0.47**	0.14	**0.47**	**0.40**	**0.75**	0.28	0.34	**0.19**	**0.58**	0.35	**0.37**
	RVM all	0.24	**0.27**	0.34	0.10	0.42	0.30	0.72	0.30	**0.38**	0.17	**0.59**	0.36	0.35
	p-value	.25	.09	.00	.06	.30	.00	.24	.41	.22	.34	.94	.51	.17
	RVM best	0.17	0.14	0.41	**0.16**	0.37	0.29	0.69	0.15	0.19	0.14	0.56	0.14	0.29
	p-value	.06	.06	.24	.15	.21	.07	.04	.01	.00	.07	.52	.01	.00
	RVM sep	**0.33**	0.24	0.45	0.14	0.44	0.31	**0.74**	**0.31**	0.37	**0.19**	**0.59**	**0.38**	**0.37**
	p-value	.22	.19	.62	.65	.46	.03	.54	.06	.40	.67	.73	.29	.74
	SMKL	0.30	0.21	0.44	0.12	0.41	0.29	0.73	0.28	0.34	0.15	0.58	0.34	0.35
	p-value	.56	.88	.40	.03	.26	.02	.30	.91	.94	.08	.85	.77	.05
	mRVM	0.19	0.15	0.24	0.04	0.35	0.38	0.65	0.19	0.27	**0.19**	0.57	0.16	0.28
	p-value	.40	.40	.05	.04	.04	.59	.01	.19	.23	.88	.64	.01	.00
	SR+SVM	0.10	0.10	0.30	0.03	0.37	0.36	0.74	0.18	0.27	0.15	0.46	0.33	0.28
	p-value	.01	.02	.00	.03	.07	.27	.70	.09	.19	.16	.03	.44	.00

DSRVM and RVM-sep perform similar regarding ICC, DSRVM selects less RV and the testing time is about 8 times faster, as can be seen in Tab. 8.3.

Tab. 8.3 provides statistics about the learned models. Each value is the average over all cross-validation folds and all AU targets. The table shows the number of selected relevance vectors (#RV), the number of selected relevant kernels (#RK), training runtime (TRN) in sec$\times 10^2$, and test runtime (TST) in sec. #RK is only shown for models that adjust the kernel weights. Tab. 8.3 shows the advantages of DSRVM relative to SMKL and RVM, in terms of the selected numbers of RV's and RK's. DSRVM and mRVM select significantly fewer

Table 8.3: Model statistics on the DISFA data averaged over all AU targets. DSRVM is compared to all baseline models including SR+SVM. The table shows the number of selected relevance vectors (#RV), the number of selected relevant kernels (#RK), training runtime (TRN) in sec$\times 10^2$, and test runtime (TST) in sec. #RK is only shown for models that adjust the kernel.

Method	#RV	#RK	TRN	TST
DSRVM	43.5	17.4	21.3	1.8
RVM all	111.5	-	1.2	6.6
RVM best	71.4	-	1.3	0.7
RVM sep	207.2	-	36.2	15.5
SMKL	1913.9	32.8	149.9	38.8
mRVM	47.0	36.0	78.0	6.0
SR+SVM	463.4	-	0.1	0.7

RV's than RVM and SMKL. In terms of RK's, DSRVM uses fewer kernels than SMKL and mRVM, while simultaneously keeping a sufficiently large number of kernels so as to yield good performance. Note that the sparse kernel and basis selection of DSRVM directly affects the test running time (TST). DSRVM regression is more than 5 times faster than that by SMKL, and even more faster than RVM-all. Moreover, DSRVM also has 7 times faster training time (TRN) in comparison to SMKL. As expected, the training time of RVM-all and RVM-best is lower than that for DSRVM, because these methods learn only the basis weights, whereas the kernel weights are fixed.

Table 8.4: ICC results on the DISFA data for different AU targets. SR+DSRVM is compared to SR+SVM within a subject-dependent setting that corresponds to the same evaluation procedure as in [121]. The last column shows the average results over all AUs (AVG).

Method	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	AVG
SR+DSRVM	**0.83**	0.81	**0.87**	**0.69**	**0.83**	**0.83**	**0.88**	**0.80**	**0.79**	**0.73**	0.89	0.75	**0.81**
SR+SVM [121]	0.80	**0.83**	**0.87**	0.58	0.81	0.80	0.84	0.71	0.69	0.54	**0.94**	**0.79**	0.77

Tab. 8.4 shows the ICC results on the DISFA data while using the same evaluation procedure as in [121]. The first step is to train a SR subspace [17] with data from all subjects and thus the results are subject-dependent. Then the DSRVM model is trained on 3000 samples with features from the learned subspace and a single Gaussian kernel, while using a leave-one-subject-out cross-validation procedure. Thus the combined method is shown as SR+DSRVM. The results show a better DSRVM performance in the majority of cases, including the average of all AUs (AVG).

Fig. 8.7 shows the kernel weights v learned by DSRVM and SMKL for different AUs on DISFA data. For AUs 17 and 20, we see that DSRVM is more sparse than SMKL, although the emphasize lies on similar regions. The facial regions selected as relevant for AU detec-

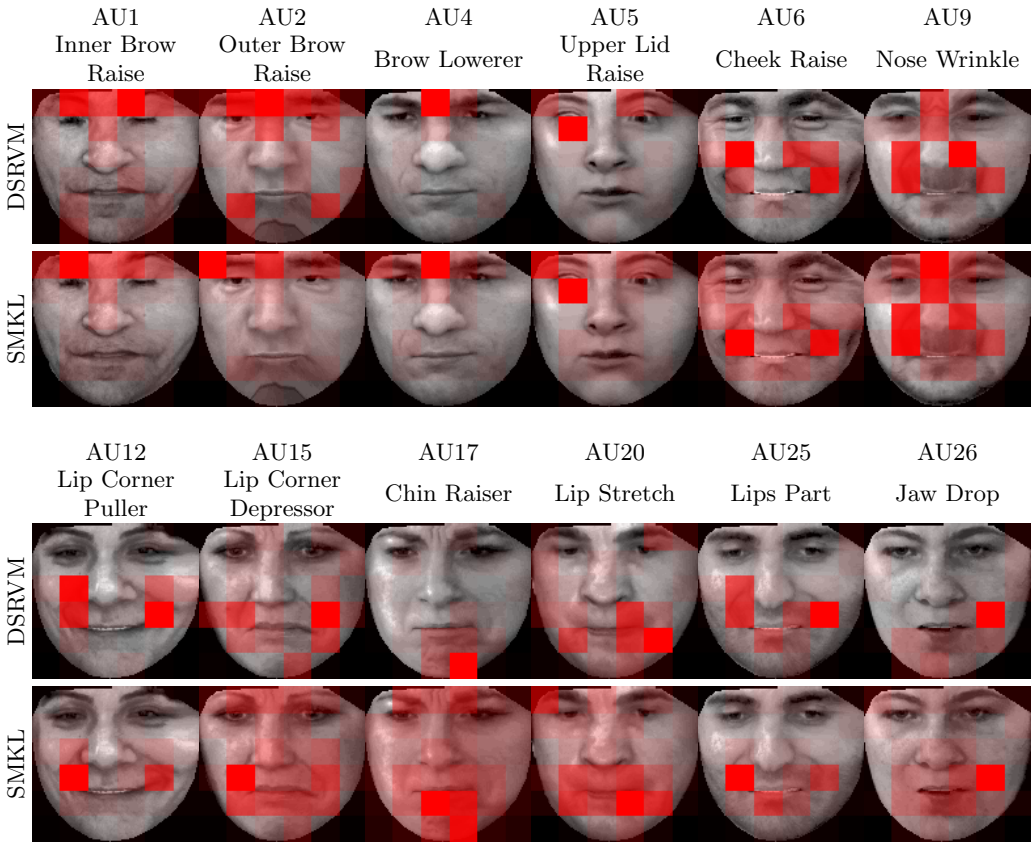


Figure 8.7: DISFA data: The values of kernel weights v learned by DSRVM (1st and 3rd row) and SMKL (2nd and 4th row) for different AUs. See the caption of Fig. 8.6. Note: each of the facial images was selected to show the highest activation of the respective target AU, but additionally other AUs may be present.

tion correspond well with the AU definitions presented in [43]. Additionally, we learn which changes to focus on. E.g. according [43], the appearance change of AU4 includes: (1) eyebrow lowering, (2) pushing the eye cover fold downward, (3) pulling the eyebrows together, (4) vertical wrinkles between the eyebrows and (5) oblique wrinkle across the forehead (optional). The learned patches focus only on the region between the eyes, which means that the outer parts of the brows are less relevant for AU4. A strong focus is also learned for AU6, which is estimated only from the region left and right to the nose, i.e. mainly from the deepening of the nasolabial furrow and not from the wrinkling next to eyes. This could be due to two reasons: either (1) the nasolabial furrow is highly correlated with the wrinkles next to the eyes, and thus only one of these regions is needed to recognize AU6 or (2) the nasolabial furrow is more informative than the eye wrinkles. The hypothesis could be tested by separately running DSRVM on the upper and lower face. If the recognition results are similar for both regions, then this would indicate that there is indeed a high correlation. On the other hand, if the

lower face performs better, then this would indicate that the information content is different.

8.6.6 Results on the SEMAINE Dataset

Tab. 8.5 shows results attained by DSRVM, SMKL, RVM-all, RVM-best and RVM-sep on the SEMAINE dataset for the two target functions: arousal and valence. DSRVM significantly outperforms the other methods for both valence and arousal.

Table 8.5: Results on the SEMAINE data for the arousal (Ar.) and valence (Val.) targets. DSRVM is compared to different RVM and SMKL. The p-value measures significance of the result in comparison to DSRVM. The table shows mean squared error (MSE), the correlation with the targets (CORR) and the Intra-class Correlation Coefficient (ICC). The best results are marked bold. If two results differ by at most 0.01 and the p-value is greater than 0.05, we mark both results bold.

	CORR		MSE		ICC	
	Ar.	Val.	Ar.	Val.	Ar.	Val.
DSRVM	**0.31**	**0.31**	**0.042**	**0.058**	**0.21**	**0.20**
RVM all	0.25	0.23	**0.046**	0.065	**0.20**	0.17
p-value	.02	.02	.15	.01	.45	.24
RVM best	0.20	0.18	0.057	0.073	0.14	0.13
p-value	.03	.06	.00	.00	.10	.39
RVM sep	0.21	0.23	0.049	0.073	0.18	**0.19**
p-value	.01	.00	.03	.00	.37	.36
SimpleMKL	0.22	0.22	0.051	0.070	0.18	0.16
p-value	.01	.00	.00	.00	.11	.07
mRVM	0.10	0.12	0.059	0.072	0.05	0.05
p-value	.00	.00	.00	.02	.00	.00

Fig. 8.8 shows a sample video frame from the SEMAINE dataset, and kernel weights v learned by DSRVM and SimpleKLM, for a given value of arousal and valence. For arousal, DSRVM focuses more on the facial area around the nose and below the eyes. This can be explained by the fact that high arousal (such as in surprise, disgust and happiness) is characterized by vertical facial motions in those areas (e.g., nose wrinkling in disgust and raised cheeks in happiness). For valence, DSRVM focuses on the inner eyebrows, the nasolabial furrow and the eye corners. Again, this can be explained by the facial motion being typical for positive valence (happiness, characterized by smiles that affect the mouth corners, the nasolabial furrow and the eye corners) and for negative valence (e.g., frowns, deepened nasolabial furrow like in anger). SimpleMLK is less sparse, and regards almost all patches on the entire face as relevant for regression, including the patches learned by DSRVM. The focus areas are different, which can be caused by the non-sparse weights.

Tab. 8.6 compares DSRVM with prior work [134, 164] on the SEMAINE datasets, in terms of CORR. Note that the comparison in Tab. 8.6 is not standardized, since prior work uses different subsets of SEMAINE. But, since each subset is supposed to represent the entire

Figure 8.8: SEMAINE data: The values of kernel weights v learned by DSRVM (left) and SMKL (right) for Arousal and Valence targets, which show that DSRVM learns sparser kernel weights. See the caption of Fig. 8.6.

dataset reasonably well, the results in Tab. 8.6 can be viewed as a reasonably good estimate of a standard comparison. In particular, [134] uses tracked facial points as features, and an output-associative RVM for regression. Results are reported separately for positive (pos) and negative (neg) arousal/valence sequences, in contrast to our setting which includes both, positive and negative. [164] uses LBP histograms as features, a different face alignment from ours, and SVR for regression. Tab. 8.6 shows that CORR of DSRVM is the best for arousal. Regarding valence, DSRVM outperforms [164]. However, the valence results of DSRVM are not directly comparable with those of [134], since [134] separately predicts positive and negative valence values, whereas DSRVM predicts all valence values.

Table 8.6: CORR results on the SEMAINE dataset. [134] reports separate results for positive (pos) and negative (neg) arousal/valence.

Method	DSRVM	[134] pos	[134] neg	[164]
Arousal	0.31	0.16	0.27	0.08
Valence	0.31	0.43	0.27	0.13

Tab. 8.7 shows the statistical comparison results across all targets from all datasets according the Friedman test [58] and Hommel procedure [78] (see Sec. 5.3 for details). The targets include 12 AUs from DISFA, pain from ShoulderPain (features S6x6T10) and arousal and valence from SEMAINE. DSRVM has the best rank for all measures, but it has equal scores as RVM-sep, RVM-all and SMKL regarding CORR and ICC. MSE results show more significant differences: only DSRVM, SMKL and mRVM are in the top group. Overall, the score difference between DSRVM and SMKL is not significant, but DSRVM has the better runtime performance, as can be seen from Tab. 8.3. Since DSRVM is additionally able to adapt the kernel weights $\mathbf{v}$, we expect it to have a better rank than RVM. SMKL has the same power, but is more likely to overfit due to less sparsity. The rank results confirm the ranking, but statistical significance is only given for MSE results, probably the advantage is not sufficiently pronounced for CORR and ICC.

Table 8.7: Rank comparison of the different models over all targets from all datasets (ShoulderPain, DISFA and SEMAINE) obtained by the Friedman test [58] and Hommel procedure [78]. The different features are ranked by their expected performance rate for each of the measures MSE, CORR and ICC. The subsets of features which have statistically equal performance are indicated by a black bar on the right side.

Rank	CORR	MSE	ICC
1	DSRVM	DSRVM	DSRVM
2	RVM sep	SMKL	RVM sep
3	RVM all	mRVM	RVM all
4	SMKL	RVM all	SMKL
5	mRVM	RVM sep	mRVM
6	RVM best	RVM best	RVM best

8.6.7 Comparison with Classification

To demonstrate the advantage of continuous regression models over classifiers, we compare the continuous DSRVM with the discrete mRVM [33] on the artificial and the SEMAINE dataset. Fig. 8.9 (top left) shows the mRVM results for discretizing the targets into different numbers of classes on the artificial data. An advantage of regression methods is the ability to naturally handle continuous data without the need of discretization. Thus the DSRVM results are constant because the targets are not discretized into classes. We see that the optimum result for mRVM is reached at 8 classes, but DSRVM results in superior performance at all times.

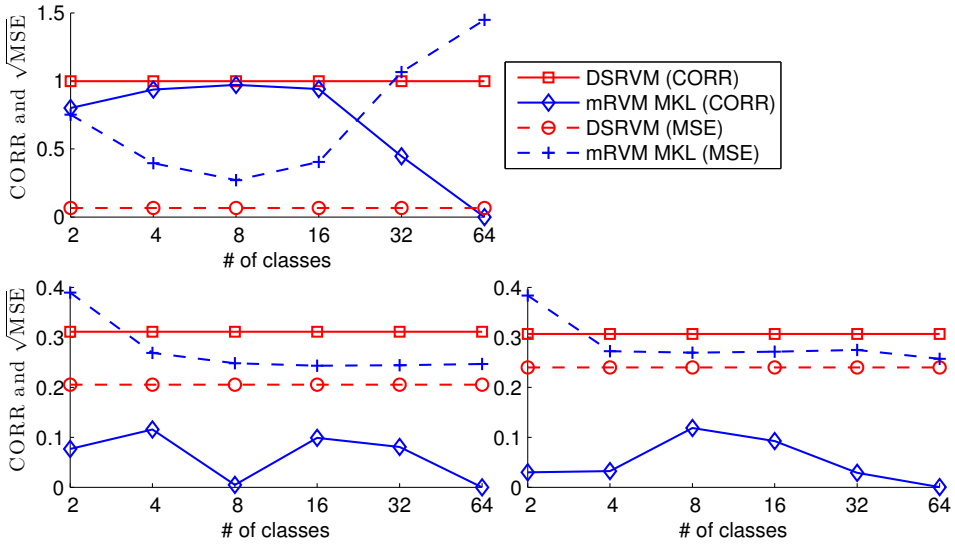


Figure 8.9: Results on the artificial data (top left) and on the SEMAINE data for the arousal (bottom left) and the valence (bottom right) target: Comparison of the continuous DSRVM with the discrete mRVM method [33] by discretizing the targets into different number of classes. (Note: we use the square root of the MSE for better axis scaling)

As can be seen from Fig. 8.9 (bottom left and right), we have a similar case for the SEMAINE data. Again DSRVM outperforms mRVM, independently of the number of classes the targets are divided into. The MSE becomes constant as the number of classes increases, since the mRVM cannot train properly due to too few data per class and therefore assigns the majority class to all instances. Due to the bias in the target distribution, this leads to a low MSE since most instances are close to the majority value around 0. However, the CORR as a relative measure takes the variance of predictions into account and clearly states that the target is modeled badly.

8.7 Conclusion

Motivated by psychological studies on the importance of local facial features for defining facial behavior, we have specified a new regression method – called Doubly Sparse Relevance Vector Machine. DSRVM generalizes RVM by jointly choosing a sparse set of relevant kernels associated with face parts, and a sparse set of relevance vectors (i.e., training data) for modeling facial expressions. This also advances related multiple-kernel learning (MKL) methods, typically specified within the max-margin framework, where enforcing joint sparsity of kernel weights and relevance vectors is typically ignored. DSRVM uses computationally efficient EM algorithm for learning relevant kernels and relevance vectors, and thus achieves about 20 times faster training than one of the latest MKL methods, called SMKL. Also, due to achieving higher sparsity, DSRVM has more than 3 times faster test runtimes, and more economic memory usage than SMKL.

We have evaluated DSRVM on challenging benchmark datasets and in most cases DSRVM yields better results than RVM and SMKL. In addition, DSRVM can be used to provide insights in the nature of facial expressions, since it learns which face parts provide the most relevant visual cues for estimating the target facial behavior and thus narrowing the focus of the broad appearance description provided in [43].

Chapter 9

Generative Multi-Output Latent Trees

Contents

In the previous chapters we explored facial expression intensity estimation of *single targets* (pain or AUs) by holistic (Chap. 7) and part-based (Chap. 8) methods. In this chapter, we advance towards joint intensity estimation for *multiple targets*, while still treating the face as a sum of its parts. Specifically, the Latent Tree (LT) model is introduced, which learns a hidden structure that governs the dependence between facial parts and the prediction targets. Additionally, we show that the inference through a hidden structure is able to provide good recognition performance even in the presence of missing or noisy features.

9.1 Introduction

In the previous chapter we introduced the DSRVM model, which is able to select and weight facial parts in order to address a specific target AU or pain. In this chapter, we still want to keep the ability to weight facial parts, but do this jointly for all AU and pain targets. This poses additional problems, since for recognizing all AUs, (almost) all facial parts are relevant and thus a model that is able to focus on specific regions would just select the whole face. In order to obtain different weightings for different targets in a joint manner, we propose a

latent tree (LT) model as solution, which is able to learn a joint model of facial landmarks and targets. Facial parts are implicitly weighted by their relative position to the targets within the tree.

To address the problem of part-based multi-target facial expression intensity estimation, we consider a Bayesian generative framework. We formalize our problem as that of jointly predicting multiple AU targets, $\mathbf{T} = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$, given a set of image features, $\mathbf{F} = \{\mathbf{x}_{T+1}, ..., \mathbf{x}_{T+F}\}$. Every target $\mathbf{x}_m \in \mathbf{T}$ can be defined as a vector of various attributes associated with mth AU, and in a special case for our problem as AU intensities. Image features $\mathbf{x}_m \in \mathbf{F}$ are defined as local descriptors of the face, which can be appearance based (e.g., patches) or locations of facial landmarks detected in a video frame.

We specify a graphical model for representing the joint distribution of targets and features, $p(\mathbf{T}, \mathbf{F})$, and use the Bayes' rule to derive an elegant solution to AU intensity estimation as

$$\hat{\mathbf{T}} = \max_{\mathbf{T}} \frac{p(\mathbf{T}, \mathbf{F})}{\sum_{\mathbf{T}'} p(\mathbf{T}', \mathbf{F})}. \quad [1] \tag{9.1}$$

Our formulation has a number of advantages over existing approaches [86, 121, 152, 158]. They typically adopt the discriminative framework for directly predicting AU intensities given the features, e.g., using Support Vector Classification (SVC) [121], Relevance Vector Machine (RVM) [86], AdaBoost [177], or ordinal Conditional Random Fields (CRF) [152]. While discriminative approaches are generally robust, we experimentally demonstrate in this chapter that they underperform under noisy features. In particular, due to frequent partial occlusions of the face or large out-of-plane head movements in non-staged video, some input features might be missing or very unreliable. Our results show that our model can robustly handle missing input features by marginalizing them out, unlike the competing discriminative approaches. Also, our model is less likely to overfit to training human subjects, due to the joint modeling of all AUs $\mathbf{T}$ and features $\mathbf{F}$.

For effectively capturing statistical dependencies among $\mathbf{T}$ and $\mathbf{F}$, our model has hidden (latent) random variables. Also, for ensuring modeling efficiency (e.g., few model parameters) and efficient inference of $\hat{\mathbf{T}}$, we organize the hidden variables in a tree structure, and hence call our model Latent Tree (LT). In LT, leaf nodes represent $\mathbf{T}$ and $\mathbf{F}$, and all other nodes correspond to the hidden variables (also called hidden nodes). Importantly, no other restrictions are placed on the model structure beyond the tree structure, defined by the total number of hidden nodes and edges.

[1] We always use the sum symbol for marginalization, even for continuous variables, for simplicity.

LT structure is unknown a priori. We specify a new algorithm for efficient learning of both model parameters and model structure on training data. Our structure learning iteratively builds LT by introducing either new parent nodes or new connections between existing hidden nodes, depending on the resulting increase in the joint likelihood $p(\mathbf{T}, \mathbf{F})$. Our key contribution here is a heuristic algorithm for efficiently computing the maximum likelihood increase.

For AU intensity estimation, we derive closed-form expressions of posterior marginals of all variables in LT, and specify an efficient inference of $\hat{\mathbf{T}}$ given $\mathbf{F}$ in two passes – bottom-up and top-down.

Common graphical models (e.g., naive Bayes and artificial neural networks) usually place the target variable at the root and the features at the leaves. However, this implicitly assumes that: (1) the dependence between the target and each of the features is roughly the same, since the path length to each feature is the same, and (2) there is only a single target. If multiple targets are placed at the root, then the model implicitly assumes that the dependence between the targets is high.

In a generative model like LT, it is not important which node is at the root. By rotating the tree, we can create a new tree that induces the same distribution on its variables, but with an arbitrary variable as root. Therefore, the structure that we impose is a singly connected graph where all inner nodes are latent and outer nodes are either targets or features.

By placing all targets and features at outer nodes, we relax both of the above assumptions from other graphical models, i.e., some targets can depend highly on a subset of features and other targets can depend on a different set, defined by the distances in the graph. The dependence within targets and features can also be intuitively interpreted from the graph structure, please see Sec. 9.5.3 for an example.

The common structure (with features as leaves, latent nodes in-between and targets at the top) is included as a special case of our model: if we learn the structure to have equal distances between each feature-target pair, then this model is equivalent to the common model.

We have evaluated LT on several benchmark datasets. In comparison with baselines and the state-of-the-art methods, the results demonstrate our superior performance, even under significant noise introduced to facial landmark points. We also demonstrate effectiveness of our structure learning by probabilistically sampling locations of facial landmark points, conditioned on a given AU intensity. Our generative sampling produces plausible facial expressions.

9.1.1 Closely Related Work

The literature abounds with various formulations of generative models and their structure learning [93]. The two unique aspects of our approach, suited to AU intensity estimation, include tying latent AU intensities and observable features together at the leaf level of LT, and a novel formulation of efficient graph edits for structure learning based on the Bayesian structural Expectation-Maximization (EM) [59].

Recent work on the Binary Latent Tree (BLT) [75] puts the restrictive constraint on the model structure that every non-leaf node cannot have more than two children. Our structure learning is more efficient, and significantly differs from the way they build BLT as trading-off the Mutual Information score and the Bayesian Information Criterion (BIC). We experimentally demonstrate that their binary-tree restriction leads to poor (BLT) performance in our domain.

Structural learning of latent trees can also be formulated by grouping random variables according to their information distance [21]. However, we treat feature and target dimensions equally (leading to a minimum of 142 nodes, without including hidden nodes) and thus the space of possible grouping combinations is very large, which leads to an inefficient algorithm.

A Bayesian net can be learned for modeling dependencies between AUs [102] with the structure learning algorithm of [35]. However, a direct comparison with this approach would be unfair to us as they treat AUs as observable variables predicted via SVMs and subject-dependent Spectral Regression (as in [121]).

The field of learning phylogenetic trees (PT) from DNA or protein data (see e.g., [53]) is also related to LT, although the problem formulation is different. The data consists of aligned nucleotide sequences of fixed length, where each sequence corresponds to a certain species. Each nucleotide belongs to a pre-defined set of possible nucleotide-types. The target is to find the best ancestral tree structure, regarding some optimality measure. Each node within the tree corresponds to a nucleotide sequence and each node is the ancestor of its children. Usually the inner-nodes are not known, only the species on the tree leaves are observed. Each ancestor should be as similar as possible to its children, and the similarity is measured by the branch length and is associated with expected nucleotide changes. Note that all nodes are categorical and have the same number of states.

A wide range of algorithms have been developed to infer PT, see [53]. Due to the similarities with latent trees, some of the algorithms are also very similar: neighbor joining [157] is one of

the most popular PT algorithms and proceeds by greedily adding a new ancestor to the two most similar species, which results in a binary tree. The algorithm uses the same strategy as BLT [75], which adds a new parent node to the two most similar nodes according the mutual information criterion. Also, the structural EM [59] has been applied to phylogeny [60]. The main difference of our LT to other PT algorithms is the application goal: PT is applied to cases where the data truly is derived from a tree and the goal is to recover this tree as good as possible. On the other hand, our LT algorithm is applied to approximate the data by a tree although the original distribution probably does not follow a tree factorization. The LT goal is not related to the tree structure, but to predict its unobserved leave variables.

In the following, Sec. 9.2 specifies LT; Sec. 9.3 formulates our inference; Sec. 9.4.1 presents our model parameter learning; Sec. 9.4.2 specifies our model structure learning; and Sec. 9.5 presents our results.

9.2 The Model

This section specifies our LT for modeling $p(\mathbf{T}, \mathbf{F})$, where $\mathbf{T}$ and $\mathbf{F}$ are introduced in Sec. 9.1. Let $\mathbf{X} = \{\mathbf{T}, \mathbf{F}\} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$, $M = T + F$. To model $p(\mathbf{X})$, we use a tree that includes, in addition to $\mathbf{X}$, also L hidden discrete variables $\mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_L\}$, each with the same number of states K. The tree is aimed at efficiently representing joint distributions of various subsets of $\mathbf{X}$ as follows. Leaf nodes of the tree correspond to every $\mathbf{x}_m \in \mathbf{X}$, and nodes at levels closer to the root correspond to every $\mathbf{h}_l \in \mathbf{H}$. The nodes are hierarchically connected in the tree to represent that the distribution of every node $\mathbf{x}_m \in \mathbf{X}$ (or $\mathbf{h}_l \in \mathbf{H}$) is conditioned on its parent node in the tree $\mathbf{h}_{P(m)} \in \mathbf{H}$ (or $\mathbf{h}_{P(l)} \in \mathbf{H}$). Thus, the tree structure is defined by the function $P(\cdot)$ which assigns the parent to each node, or the empty set $\emptyset$ if the node is a root. A non-leaf node in the tree may have arbitrary many children nodes.

The conditional distribution between hidden nodes $\mathbf{h}_l$ and $\mathbf{h}_{P(l)}$ is categorical, since both nodes are discrete:

$$p(\mathbf{h}_l | \mathbf{h}_{P(l)} = k) = \text{Cat}(\mathbf{h}_l; \boldsymbol{\mu}_{k,l}), \tag{9.2}$$

where $k \in \{1, ..., K\}$, $\text{Cat}(\mathbf{h}; \boldsymbol{\mu})$ is the categorical distribution over $\mathbf{h} \in \{1, ..., K\}$ with the parameter $\boldsymbol{\mu} \in \mathbb{R}^K$, $\forall k : \boldsymbol{\mu}_k \geq 0$, and $\sum_{k=1}^K \boldsymbol{\mu}_k = 1$. The annotated AU targets are discrete and thus the conditional distribution between a target $\mathbf{x}_m$ and its parent $\mathbf{h}_{P(m)}$ is categorical as well, i.e., equivalent as in (9.2).

The conditional distribution for continuous features $\mathbf{x}_m^{(\text{cont.})}$ is Gaussian:

$$p(\mathbf{x}_m|\mathbf{h}_{P(m)} = k) = \mathcal{N}(\mathbf{x}_m; \boldsymbol{\mu}_{k,m}, \boldsymbol{\Sigma}_{k,m}), \tag{9.3}$$

with the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^d_+ \times \mathbb{R}^d_+$, where d is the dimensionality of $\mathbf{x}_m$. The tree root $\mathbf{h}_r$ has no parent, and thus is not conditioned on another node. Its distribution is defined as a prior:

$$p(\mathbf{h}_r|\mathbf{h}_{P(r)}) = p(\mathbf{h}_r|\emptyset) = \text{Cat}(\mathbf{h}_r; \boldsymbol{\mu}_r). \tag{9.4}$$

From (9.2)–(9.4), the joint distribution of all variables can be expressed as

$$p(\mathbf{X}, \mathbf{H}) = \textstyle\prod_{m,l} p(\mathbf{x}_m|\mathbf{h}_{P(m)})p(\mathbf{h}_l|\mathbf{h}_{P(l)}). \tag{9.5}$$

We use (9.5) to define the marginal log-likelihood of a given set of data points $\{\mathbf{X}^{(1)}, ..., \mathbf{X}^{(N)}\}$ as

$$\mathcal{L} = \textstyle\sum_{n=1}^{N} \ln \sum_{\mathbf{H}} p(\mathbf{X}^{(n)}, \mathbf{H}). \tag{9.6}$$

To learn LT parameters and structure, we maximize $\mathcal{L}$, given by (9.6), on training data using an EM algorithm. As inference is an integral part of learning, in the sequel, we first specify our inference in Sec. 9.3, and then present our learning of LT parameters in Sec. 9.4.1 and LT structure in Sec. 9.4.2.

9.3 Bottom-up/Top-down Inference on LT

We use the MAP criterion, given by (9.1), to predict discrete AU intensities $\hat{\mathbf{T}} = \{\hat{\mathbf{x}}_m : m = 1, ..., T\}$, given all input features $\mathbf{F}$. From our specification of LT, presented in Sec. 9.2, the MAP estimation of (9.1) can be decomposed for every individual target $\mathbf{x}_m \in \mathbf{T}$ as

$$\hat{\mathbf{x}}_m = \max_{\mathbf{x}_m} \textstyle\sum_{\mathbf{h}_{P(m)}} p(\mathbf{x}_m|\mathbf{h}_{P(m)})p(\mathbf{h}_{P(m)}|\mathbf{F}). \tag{9.7}$$

From (9.7), our inference problem amounts to finding the posterior $p(\mathbf{h}_{P(m)}|\mathbf{F})$.

In the following, we explain how to compute the marginal posteriors $p(\mathbf{h}_l|\mathbf{S})$ for all hidden nodes $\mathbf{h}_l \in \mathbf{H}$ using the standard bottom-up/top-down inference (a.k.a., the inside-outside algorithm) on trees [93], where $\mathbf{S}$ can be an arbitrary subset of $\{\mathbf{T}, \mathbf{F}\}$. The resulting posteriors of parents of leaf nodes in LT can then be used for AU intensity estimation in (9.7).

The bottom-up/top-down inference on LT efficiently computes the marginal posteriors $p(\mathbf{h}_l|\mathbf{S})$ in two passes – bottom-up and top-down, as illustrated in Fig. 9.1. In particular, for every $\mathbf{h}_l$, the algorithm defines the set of *inside* variables $\mathbf{x}_{in(l)} = \{\mathbf{x}_m :$

$\mathbf{x}_m \in \mathbf{S}$ is descendant of $\mathbf{h}_l\}$, and the set of *outside* variables $\mathbf{x}_{out(l)} = \{\mathbf{x}_m : \mathbf{x}_m \in \mathbf{S}$ is not descendant of $\mathbf{h}_l\}$, and their distributions

$$\boldsymbol{\beta}_l = p(\mathbf{x}_{in(l)}|\mathbf{h}_l), \quad \boldsymbol{\alpha}_l = p(\mathbf{h}_l|\mathbf{x}_{out(l)}). \tag{9.8}$$

From (9.8), it is straightforward to derive that for all $\mathbf{h}_l \in \mathbf{H}$:

$$p(\mathbf{h}_l|\mathbf{S}) = \frac{\boldsymbol{\beta}_l \boldsymbol{\alpha}_l}{\sum_{\mathbf{h}_l} \boldsymbol{\beta}_l \boldsymbol{\alpha}_l}. \tag{9.9}$$

Bottom-up. The algorithm first computes the likelihoods $\boldsymbol{\beta}_l$ in the bottom-up pass starting from the leaves as

$$\boldsymbol{\beta}_l = \prod_c (\sum_{\mathbf{h}_c} \boldsymbol{\beta}_c p(\mathbf{h}_c|\mathbf{h}_l)), \tag{9.10}$$

where $\{\mathbf{h}_c\}$ are children of $\mathbf{h}_l$. Note: If some $\mathbf{x}_m$ are unobserved, i.e., if $\mathbf{S}$ is a strict subset of $\{\mathbf{T}, \mathbf{F}\}$, then the unobserved $\boldsymbol{\beta}_m$ are uniform.

Top-down. Then, the algorithm computes the distributions $\boldsymbol{\alpha}_l$ starting from the root as

$$\boldsymbol{\alpha}_l = \sum_{\mathbf{h}_{P(l)}} p(\mathbf{h}_l|\mathbf{h}_{P(l)}) \boldsymbol{\alpha}_{P(l)} \prod_s \left(\sum_{\mathbf{h}_s} \boldsymbol{\beta}_s p(\mathbf{h}_s|\mathbf{h}_{P(l)}) \right), \tag{9.11}$$

where $\{\mathbf{h}_s : \mathbf{h}_{P(l)}{=}\mathbf{h}_{P(s)}, \mathbf{h}_s{\neq}\mathbf{h}_l\}$ are the siblings of $\mathbf{h}_l$.

In summary, for AU intensity estimation, we first run the upward pass (9.10) and then the downward pass (9.11) to compute the distributions of inside and outside variables, $\boldsymbol{\beta}_l$ and $\boldsymbol{\alpha}_l$, for all hidden variables $\mathbf{h}_l \in \mathbf{H}$, and then estimate the specific marginal posterior $p(\mathbf{h}_{P(m)}|\mathbf{F})$ as in (9.9) required for estimating the AU intensity $\hat{\mathbf{x}}_m$ as in (9.7).

As explained in the sequel, we also use the bottom-up/top-down inference algorithm as an integral part of learning model parameters. For this learning, we will be required to compute both marginal posteriors $p(\mathbf{h}_l|\mathbf{X})$ and pairwise posterior marginals $p(\mathbf{h}_l, \mathbf{h}_{P(l)}|\mathbf{X})$. Fortunately, due to the tree structure of our model, they can be computed exactly as in (9.9), and as

$$p(\mathbf{h}_l, \mathbf{h}_{P(l)}|\mathbf{X}) \sim \boldsymbol{\beta}_l p(\mathbf{h}_l|\mathbf{h}_{P(l)}) \prod_s (\sum_{\mathbf{h}_s} \boldsymbol{\beta}_s p(\mathbf{h}_s|\mathbf{h}_{P(l)})) \boldsymbol{\alpha}_{P(l)}, \tag{9.12}$$

where $\{\mathbf{h}_s\}$ are the siblings of $\mathbf{h}_l$.

9.4 Learning LT

Given a set of training data $\{\mathbf{X}^{(1)}, ..., \mathbf{X}^{(N)}\}$, we learn LT parameters and LT structure by maximizing $\mathcal{L}$, given by (9.6). This maximization is conducted iteratively by alternating two

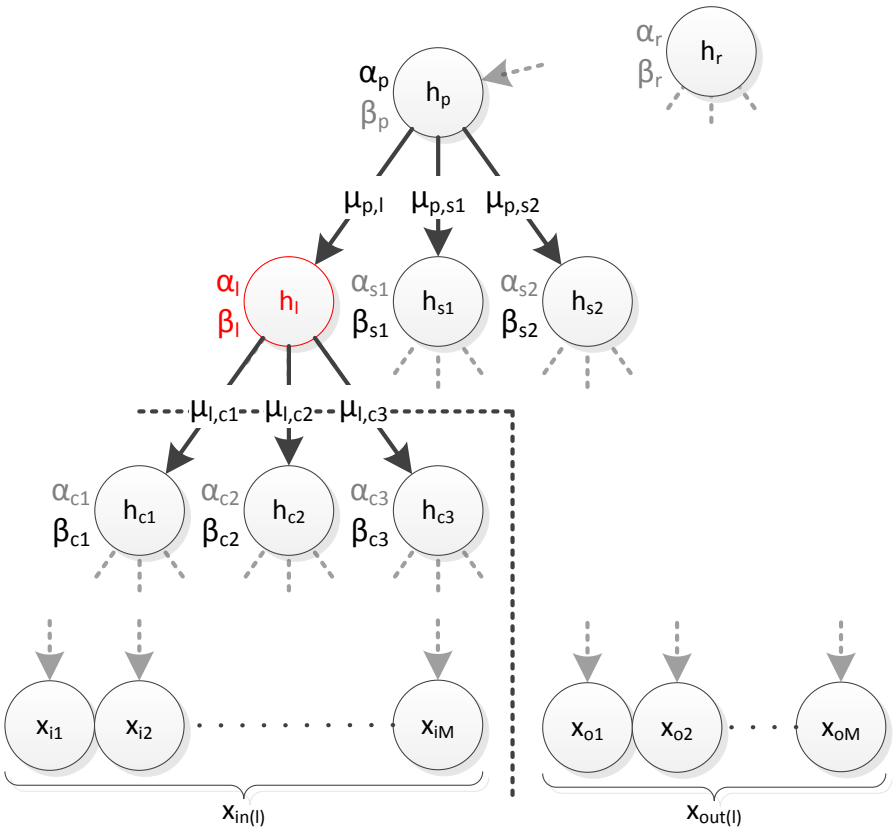


Figure 9.1: The inside-outside algorithm for computing the marginal posterior of node $\mathbf{h}_l$ (red) with parent $\mathbf{h}_p$, children $\{\mathbf{h}_{c1}, \mathbf{h}_{c2}, \mathbf{h}_{c3}\}$ and siblings $\{\mathbf{h}_{s1}, \mathbf{h}_{s2}\}$. $\mathbf{x}_{in(l)}$ and $\mathbf{x}_{out(l)}$ are two complementary sets of leaf nodes, where $\mathbf{x}_{in(l)}$ consists of descendants of $\mathbf{h}_l$.

steps. First, for a given current estimate of LT structure, we compute the updates of model parameters. Second, for a given current estimate of LT parameters, we conduct graph-edits for revising the LT structure. The two steps are iterated until $\mathcal{L}$ stops increasing, or the maximum number of iterations is reached. In the following, we first describe our parameter learning, and then specify our structure learning.

9.4.1 Learning LT Parameters

During learning of model parameters, $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, defined in Sec. 9.2, we assume that the LT structure is given. Maximizing $\mathcal{L}$ does not lend itself to a closed-form solution, because the sum over $\mathbf{H}$ appears inside the logarithm in (9.6). Therefore, we resort to an EM algorithm, which iteratively estimates the joint posterior $q^{(n)} = p(\mathbf{h}_1, ..., \mathbf{h}_L | \mathbf{X}^{(n)})$, and uses $q^{(n)}$ to update the model parameters by maximizing expected log-likelihood with respect to $q^{(n)}$ as

$$(\boldsymbol{\mu}, \boldsymbol{\Sigma})^{\text{new}} = \arg\max_{(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \sum_{n=1}^{N} \mathbb{E}_{q^{(n)}}[\ln p(\mathbf{X}^{(n)}, \mathbf{H})]. \tag{9.13}$$

Following the standard steps of finding a derivative of the expectation term in (9.13) with respect to each model parameter gives the well-known update equations for the Gaussian and categorial distributions of LT. Specifically, solving (9.13) regarding the parameters $(\boldsymbol{\mu}_{k,l})^{\text{new}}$ for the distributions defined in (9.2) leads to the update:

$$\mu_{k,l}^{\text{new}}(k1) = \frac{\sum_n q^{(n)}(\mathbf{h}_l = k1, \mathbf{h}_{P(l)} = k)}{N_{k,P(l)}}, \tag{9.14}$$

with $N_{k,P(l)} = \sum_n q^{(n)}(\mathbf{h}_{P(l)} = k)$.

Furthermore, solving (9.13) regarding the parameters $(\boldsymbol{\mu}_{k,m}, \boldsymbol{\Sigma}_{k,m})^{\text{new}}$ for the distributions defined in (9.3) leads to the update:

$$\boldsymbol{\mu}_{k,m}^{\text{new}} = \frac{\sum_n q^{(n)}(\mathbf{h}_{P(m)} = k)\mathbf{x}_m^{(n)}}{N_{k,P(m)}} \tag{9.15}$$

$$\boldsymbol{\Sigma}_{k,m}^{\text{new}} = \frac{\sum_n q^{(n)}(\mathbf{h}_{P(m)} = k)(\mathbf{x}_m^{(n)} - \boldsymbol{\mu}_{k,m}^{\text{new}})(\mathbf{x}_m^{(n)} - \boldsymbol{\mu}_{k,m}^{\text{new}})^\top}{N_{k,P(m)}}, \tag{9.16}$$

with $N_{k,P(m)} = \sum_n q^{(n)}(\mathbf{h}_{P(m)} = k)$.

Finally, solving (9.13) regarding the parameters $(\boldsymbol{\mu}_r)^{\text{new}}$ for the distributions defined in (9.4) leads to the update:

$$\mu_r^{(\text{new})}(k) = \frac{\sum_n q^{(n)}(\mathbf{h}_r = k)}{N}. \tag{9.17}$$

Importantly, the update equations of parameters associated with the hidden nodes $\mathbf{h}_l$ can be expressed in terms of pairwise posterior marginals $p(\mathbf{h}_l, \mathbf{h}_{P(l)}|\mathbf{X}^{(n)})$. Also, in case of the root $\mathbf{h}_r$ and leaf nodes $\mathbf{x}_m$, the parameter update equations are expressed in terms of posterior marginals $p(\mathbf{h}_r|\mathbf{X}^{(n)})$ and $p(\mathbf{h}_{P(m)}|\mathbf{X}^{(n)})$, respectively. These posteriors can be computed exactly in (9.9) and (9.12) using the bottom-up/top-down inference algorithm.

9.4.2 Learning LT Structure

Given an estimate of LT parameters, our goal is to find an optimal tree structure that would maximize $\mathcal{L}$, given by (9.6). Finding an optimal tree in the space of all possible trees is intractable. Therefore, we specify a heuristic algorithm for structure learning. A common approach is to start from a trivial initial tree which has no connections between nodes and no hidden nodes. From there, the tree is successively altered according to an optimization criterion, e.g., using Mutual Information (MI) [75] or information distance [21], until convergence. Rather than adopting a new information-theoretic criterion for structure learning, we use the very

same log-likelihood $\mathcal{L}$ for learning the tree as when learning model parameters. Our unified framework of parameter and structure learning allows us to derive an efficient algorithm for revising the tree so as to maximize log-likelihood gains.

Another common issue in structure learning is regularization of model complexity, typically addressed by using the Bayesian information criterion (BIC) [75]. As one of our contributions, we regularize our tree learning by favoring those structure changes that: 1) Minimally increase model complexity, while at the same time 2) Maximally increase the gain in the conditional likelihood of all descendant nodes under the introduced structural change. The latter regularization condition is motivated by the generative properties of our model: if we add a new child to a node, then we require that the conditional likelihoods of all its siblings be improved – not only the overall joint likelihood. This effectively means that the newly added child needs to contribute information to all its siblings. The regularization condition also helps to avoid connecting all leaf nodes to a single latent node (equivalent to a mixture model), since this would mean to repeatedly add children to the same node. However, when adding more siblings, the regularization condition becomes harder to fulfill.

Algorithm. Our structure learning iteratively revises candidate trees, starting from the initial forest of trivial trees wherein all $\mathbf{x}_m \in \mathbf{X}$ are independent, and paired with the corresponding hidden root, $\mathbf{h}_r$. In this initial forest, the joint log-likelihood $\mathcal{L}$ of $\mathbf{X}$ is equal to the sum of the tree specific log-likelihoods. Our structure learning then proceeds by introducing either a new edge in the tree, or a new hidden node and appropriately connecting it to the existing ones. In particular, we consider two types of graph-edit operations:

(1) Add new edge (l', l) between two existing nodes $\mathbf{h}_{l'}$ and $\mathbf{h}_l$;

(2) Add new parent $\mathbf{h}_{l'}$ to existing nodes $\mathbf{h}_{l_1}, \mathbf{h}_{l_2}$ including edges (l', l_1) and (l', l_2).

The graph-edit operations (1) and (2) yield increases in log-likelihood, $\Delta\mathcal{L}$. Our goal is to identify the operation that produces the highest $\Delta\mathcal{L}$ and simultaneously meets the regularization constraints. One constraint is to maintain the tree structure. Another is the regularization constraint that requires, for all siblings $\{\mathbf{h}_s\}$ of the newly added child $\mathbf{h}_l$, that the difference $\Delta\mathcal{C}_s = \mathcal{C}_s^{(\text{new})} - \mathcal{C}_s^{(\text{old})}$ in the conditional likelihood $\mathcal{C}_s = p(\mathbf{x}_{in(s)}|\mathbf{x}_{out(s)})$ must be greater than the threshold $t_{\mathcal{C}}$. The structure learning terminates if there are no graph revisions to perform, i.e., when the tree becomes rooted at a single root, or all possible graph-edits would lead to a log-likelihood increase that is smaller than the threshold $t_{\mathcal{L}}$. Our LT structure learn-

ing algorithm is summarized in Alg. 9.1 and maximizes $\mathcal{L}$ (defined in (9.6)) as each step is guaranteed to increase $\mathcal{L}$.

Algorithm 9.1. LT structure learning algorithm

initialize forest
while $\Delta\mathcal{L} \geq t_{\mathcal{L}}$ **and not** single root **do**
  Try:
  (1) Add new edge
    (a) select edge (l', l) with $\max \Delta\mathcal{L}$
    (b) require $\Delta\mathcal{C}_s \geq t_{\mathcal{C}}$ for all siblings $\{\mathbf{h}_s\}$ of $\mathbf{h}_l$
  Otherwise:
  (2) Add new parent with $\max \Delta\mathcal{L}$
end while

Efficiency. In the above algorithm, Δ has to be calculated for all possible pairs $(\mathbf{h}_l, \mathbf{h}_{l'})$, which is quadratic in the number of roots. We specify two mechanisms to achieve efficiency. The first mechanism concerns our observation that adding new nodes by graph-edit operation (2) will increase model complexity more than operation (1), since (1) adds just an edge, whereas (2) adds a node and two edges. Therefore, we specify a heuristic procedure to first evaluate the Δ's of all possible operations of type (1), and start considering (2) only if none of operations of type (1) meet the above algorithm's criteria and constraints. This additionally helps to avoid a maximum depth binary tree (equivalent to the structure of a Markov chain), since it can only be created by repeatedly choosing (2) while including the newest latent node as child. The second mechanism concerns our efficient evaluation of Δ. Specifically, after the structural change, we perform a single M-step to compute only the model parameters for the newly added connection between $\mathbf{h}_l$ and $\mathbf{h}_{l'}$, while using the joint posterior q from the previous E-step, before the structural change. It is straightforward to show that this approximate procedure is guaranteed to increase the log-likelihood in the M-step [37, 97].

9.5 Results

In order to evaluate the LT model for AU recognition, we design the experiments to contain local targets and local features, so that out LT model can discover the hidden structure that governs the dependencies of the input, which in this application leads to a joint generative model of the facial points and the AUs. In the following, we describe the evaluation setting, the models that we compare to (Sec. 9.5.1), followed by quantitative (Sec. 9.5.2) and qualitative (Sec. 9.5.3) results.

In this chapter, we focus on the DISFA [121] and ShoulderPain [109] datasets, since they

both provide per-frame AU intensity labels for spontaneous facial expressions, see Chap. 4 for details. We exclude AU 27 from the experiments, since it is present for 18 frames only. The main features used were facial landmark points (PTS), see Chap. 6 for feature extraction details. We included initial experiments with LBP features, which lead to poor results with our LT model.

Each of the PTS feature dimensions is continuous and thus modeled in the LT with a Gaussian node. The AU targets are discrete and thus modeled with a Categorical node, where each category corresponds to one AU intensity level. The LBP features consist of histograms with 59 bins and therefore we model each of them with a Categorical node having 59 states. For prediction, we use the expected value of the AU intensity, given the corresponding posterior node distribution. This means the predictions are continuous, but restricted to the interval 0-5. The data is grouped into cross-validation folds with no more than 3 subjects per fold, which leads to 9 folds for DISFA and 8 folds for ShoulderPain.

The LT model supports the prediction of multiple targets at the same time, but in order to determine if multiple targets improve the performance, we evaluate our model for different settings: (1) LT-all, which includes all targets for training; (2) LT-sep which trains a separate model for each target; and (3) LT-single, which is limited to a single hidden variable and trained separately per target as LT-sep.

Additionally to the evaluation on clean data, we create random noise to corrupt the test features. The noise is created with different severity levels: 50% noise features means that for every testing instance, we randomly select 50% of the feature dimensions and replace them with a randomly sampled value from a Gaussian distribution that has the same mean and variance as the overall training dataset. The noise is only influencing the test data, i.e., the models are trained on clean data. Furthermore, we compare our LT-all model to all other methods by the pair-wise Student's t-test with a p-value of 0.05 and mark all significantly different results with ' * '.

We did not include experiments where noise influences the training data, however we expect training noise to influence both, the parameter and structure learning. The parameters will tend towards less discriminative distributions, i.e. the categorical distributions will tend towards the uniform distribution and the Gaussian distributions will tend towards the mean and variance of the overall data. The influence on structure learning might be more severe: structure learning depends mainly on the log-likelihood increases of all possible structure changes. Noise will lower all likelihoods differences; this will not be a problem for low and moderate

levels of noise, as long as the noise is equally distributed and thus the original ranking order of likelihoods is preserved. However, if the noise becomes too severe, structure learning would only be able to recover the strongest node dependencies and weaker ones are missed out.

9.5.1 Baseline Methods

We compare our method to Support Vector Classification (SVC), Support Vector Regression (SVR) (both using the LIBSVM [20] implementation) and Binary Latent Trees (BLT) [75].

SVC has been used for the baselines of DISFA [121] and ShoulderPain [109], by treating each of the intensity levels as a separate class and applying the one-vs-one approach. SVR is similar, but it treats all target intensities on a continuous scale, rather than separate categories. We apply the Gaussian kernel to SVC and SVR and optimize all hyper-parameters by a grid search. SVC and SVR support only a single target, therefore we train a separate model per target.

BLT has not been used in a supervised context, but the inference step can infer the unobserved targets given observed features. Furthermore, BLT allows only categorical nodes, therefore we first apply k-means clustering with $K = 10$ to each of the continuous feature dimensions and then use the assigned cluster as categorical feature. We use the BLT implementation provided by the authors of [75].

Note that the training and testing data is the same for all models across all folds, but there is only one LT-all and BLT model trained for all targets while there is a separate SVC, SVR, LT-sep and LT-single model per target.

9.5.2 Quantitative Results

First, the DISFA results for different feature combinations are shown, followed by detailed shape feature results on the DISFA and ShoulderPain databases.

DISFA Data. Tab. 9.1 shows the average CORR over all AUs on the DISFA data for different feature combinations. The models have been evaluated using point (PTS), appearance (LBP) and the combination of both (PTS+LBP). The reported results for LBP are consistently lower than the ones for PTS and SVC/SVR beat our approach for LBP features. This can be explained with the nature of LBPs: since they aggregate information within a histogram, the locality of the data is lost and thus it is difficult for our model to learn local distributions, represented by branches of the tree. Combining LBP and PTS gives no improvement over PTS alone, therefore all following results are shown for PTS features only. We were not able

to obtain LBP and PTS+LBP results for BLT, since the algorithm did not terminate after running for 72 hours due to the high dimensionality of the data.

Table 9.1: Average results over all AU targets measured by the correlation coefficient (CORR) on DISFA data for different features. We compare facial landmark points (PTS) with local binary pattern (LBP) features and also show the combined results (PTS+LBP).

Feature	PTS	LBP	PTS+LBP
LT-all	0.43	0.14	0.43
LT-sep	0.41	0.10	0.40
LT-single	0.33	0.12	0.33
SVC	0.23	0.21	0.28
SVR	0.43	0.34	0.43

Tab. 9.2 shows the results on the DISFA dataset for PTS features. LT-all is on average the best for ICC and on par with SVR for CORR and MSE. The results for LT-sep are slightly lower, which shows that it is beneficial to learn all AUs together. SVC is significantly inferior than LT-all in almost all cases. This is probably due the discriminative nature of the SVC approach, which is better suited for binary classification. LT-all is significantly better than SVR regarding most measures for the AUs 5, 12, and 25 and LT-all is significantly worse than SVR regarding the most measures for the AUs 9 and 15. This can be explained by the fact that AUs 5, 12, and 25 are pronounced in the localized model, since they elicit large local variation in the points which is more difficult to be captured by the SVR that uses a kernel over all dimensions. In contrast to that, AUs 9 and 15 are barely recognizable with facial landmarks since they induce very little movement over a larger set of points, which is not captured in our generative model. BLT is significantly inferior to LT-all in most cases, because it creates a tree with more hidden nodes, and thus the features and targets are farther apart in the tree, which leads to lower dependence. Often BLT will not connect all data input nodes, which leaves a forest with mutual independent sets of variables.

The last two columns of Tab. 9.2 show the average results for 10% (a[10]) and 20% (a[20]) added noise. Although the CORR of LT-all and SVR is on par for 0% noise, the LT-model does better as the noise increases. It is not statistically significant for CORR, probably because the performance is still too similar, however it is significant for ICC. This effect is even more pronounced in Fig. 9.3, which shows the CORR results as the noise level varies. The result at 0% noise is the same value as in Tab. 9.2 and the performance deteriorates as the noise increases. The performance drop of our LT model is slower than the other models, and it is even possible to beat other models as the noise increases, see AU17: at about 50% noise, out performance is better than SVR, although SVR has the better performance on clean data.

Table 9.2: Results on the DISFA data for different AU targets and PTS features. Different LT models are compared to SVC, SVR and BLT. The table shows CORR, MSE and ICC measures. The best results per AU and per measure are marked in bold. The results that are statistically different to LT-all are marked with *. Additionally we show the average performance over all AUs (a[0]), as well as the average performance for 10% and 20% added noise (a[10] and a[20]).

	AU	1	2	4	5	6	9	12	15	17	20	25	26	a[0]	a[10]	a[20]
CORR	LT-all	.41	**.44**	.50	.29	.55	.32	.76	.11	.31	.16	**.82**	.49	**.43**	**.40**	**.36**
	LT-sep	.41	**.44**	.47	**.34***	.55	.27	**.77**	.09	.18	.10	**.82**	.47	.41		
	LT-single	.30*	.29*	.27*	.12*	**.56**	.21*	.74	.09	.16	.12	.76*	.39*	.33		
	SVC	.19*	.19*	.33*	.01*	.10*	.12*	.60*	.02*	.14*	.04	.71*	.33*	.23*	.20*	.17*
	SVR	**.42**	**.44**	**.53**	.15*	.47	**.43***	.70*	**.21***	**.32**	**.21**	.76*	**.51**	**.43**	.39	.34
	BLT	.04*	.05*	.20*	.00*	.55	.18	.73	.01*	.04*	.02	**.82**	.26*	.24*	.23*	.22*
MSE	LT-all	.44	.39	.96	**.07**	.41	.31	.40	.17	.33	.16	.61	.46	**.39**	**.42**	**.46**
	LT-sep	**.41**	.37	1.00	**.07**	**.40**	.31	**.39**	.16	.33	**.15**	**.58**	.46	**.39**		
	LT-single	.47	.41	1.21*	**.07**	.41	.32	.44	.16	.32	**.15**	.76*	.50	.43		
	SVC	.51	.43	1.21	.08	.65*	.34	.65*	.18	.35	.16	.99*	.56*	.51*	.53*	.55*
	SVR	.42	**.35**	**.87**	**.07**	.45	**.27***	.50*	**.15***	**.29***	**.15**	.76*	**.41***	**.39**	**.42**	**.46**
	BLT	.53*	.45	1.27*	**.07**	**.40**	.31	.47	.16	.33	**.15**	.63	.56*	.45*	.45	.47
ICC	LT-all	**.32**	**.37**	.41	**.18**	**.46**	.23	**.73**	.07	**.23**	.09	**.80**	.39	**.36**	**.33**	**.31**
	LT-sep	.26*	.29*	.39	.15	.44	.18	**.73**	.06	.11	.03	**.80**	.39	.32		
	LT-single	.15*	.15*	.17*	.04*	.45	.12*	.70	.04	.07*	.06	.74*	.30*	.25		
	SVC	.12*	.11*	.31	.00*	.09*	.08*	.58*	.01*	.11*	.02	.70*	.28*	.20*	.17*	.14*
	SVR	.28	.30*	**.44**	.09*	.36	**.29**	.62*	**.13***	**.23**	**.12**	.71*	**.42**	.33	.28*	.23*
	BLT	.03*	.03*	.12*	.00*	.45	.08*	.68*	.00*	.01*	.00	**.80**	.17*	.20*	.19*	.19*

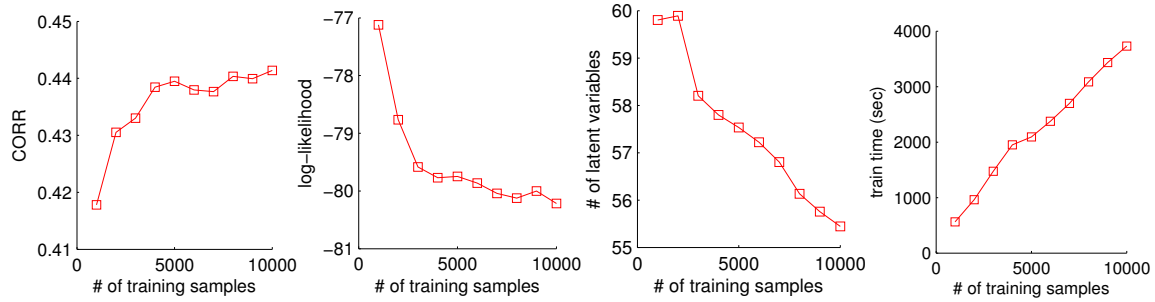


Figure 9.2: Results on the DISFA data while training our LT-all model on different number of training samples. Shown is the CORR performance, the joint log-likelihood on training data, the number of latent variables of the trained model and the training time.

This is due to our generative model, which is able to ignore noisy data that is not consistent with the clean dimensions. This effect is also pronounced with the BLT model: BLT has the same average performance on clean data as SVM. However, with increasing noise, BLT performs clearly better than SVM. BLT handles the noisy data of AU6 and AU12 very well, even better than LT, probably because it does not learn a full tree of all landmarks, but infers them from a small subset of landmarks and is thus less influenced by noise.

Fig. 9.2 shows the CORR, joint likelihood on training data, number of latent variables and the training time for our LT-all model as the number of training samples varies. The CORR

starts to level out after 4000 samples . The likelihood and the number of latent variables both decrease when the training samples increase. This shows that the model overfits with few examples, i.e., it has a high likelihood and many latent variables. However, with increasing number of training samples, the overfitting vanishes and thus the likelihood decreases. The training time shows a clear linear trend. Furthermore, the LT inference time for recognizing all 12 AUs in 14,535 testing samples is 17.9 sec, where SVR takes 106.2 sec for the same task.

ShoulderPain Data. Tab. 9.3 shows the results for the ShoulderPain data for different AU and pain targets. The LT model does best in most of the cases, except for AU 7 and 26. The appearance of these AUs is only barely present within the points and thus a generative model will assume theses changes to be noise, whereas discriminative models can learn them. The recognition results are in general lower than for the DISFA data, which is due to less frequent AU occurrences in the data and larger head movements.

Table 9.3: Results on the ShoulderPain data for different AU and pain targets. The L-all model is compared to SVC, SVR and BLT. The table shows CORR, MSE and ICC measures. The best results per AU and per measure are marked in bold. The results that are statistically different to LT-all are marked with *.

	AU	4	6	7	9	10	12	20	25	26	43	PSPI	avg
CORR	LT-all	.03	**.60**	.11	**.10**	**.15**	**.60**	**.09**	**.18**	.01	**.44**	**.48**	**.25**
	SVC	.04	.45	.25	.02	.06	.45	.00	.13	.07	.30	.37	.19
	SVR	**.05**	.48	**.26**	.09	.10	.44	.03	.17	**.10**	**.44**	.30	.22
	BLT	.03	.55	.06	.05	.05	.55	.00	.07	.06	.21	.43	.19
MSE	LT-all	.51	**1.06**	**1.19**	**.27**	**.28**	**1.12**	**.19**	.72	.50	**.14**	**3.51**	**.86**
	SVC	.76	1.74*	1.59	.48	.32	1.54	.36	1.24	.56	.17	5.00	1.25
	SVR	.65	1.44	1.40	.40	.36	1.35	.30	.76	.76	.15	4.93	1.13
	BLT	**.48**	1.15	1.29	**.27**	.31	1.17	**.19**	**.66**	**.41**	.18	3.61	.88
ICC	LT-all	.01	**.52**	.05	.02	**.08**	**.49**	**.02**	.11	.01	**.40**	**.38**	**.19**
	SVC	.03	.44	.22	.01	.03	.43	.00	.10	.04	.29	.35	.18
	SVR	**.04**	.42	**.23***	**.07**	.06	.39	**.02**	**.14**	**.09**	.39	.28	**.19**
	BLT	.01	.46	.01	.04	.07	.47	.00	.10	.02	.10*	.32	.14

Fig. 9.4 shows the CORR results on ShoulderPain for varying feature noise. Again, the LT model can handle the noise well and stays above the competing methods in the most cases. The advantage is less pronounced than in the DISFA data, which can be due to the less descriptive clean data, i.e., the facial movements are less pronounced than in DISFA. Only AUs 6, 12 and 43 are reasonably well recognized, results for the other AUs are low. This is in line with previous results on ShoulderPain, see Sec. 7.3, and is probably due to few training instances for the other AUs (see Tab. 4.1) in comparison to DISFA.

Tab. 9.4 shows the statistical comparison results across all AU targets from DISFA and ShoulderPain according the Friedman test [58] and Hommel procedure [78] (see Sec. 5.3 for

details). The targets include 12 AUs from DISFA and 10 AUs from ShoulderPain. LT and SVR are within the best performing subset for all measures, while SVC is in the worst performing subset.

Table 9.4: Rank comparison of the different models over all AUs from DISFA and ShoulderPain obtained by the Friedman test [58] and Hommel procedure [78]. The different features are ranked by their expected performance rate for each of the measures MSE, CORR and ICC. The subsets of features which have statistically equal performance are indicated by a black bar on the right side.

Rank	CORR	MSE	ICC
1	SVR	LT-all	LT-all
2	LT-all	BLT	SVR
3	SVC	SVR	SVC
4	BLT	SVC	BLT

9.5.3 Qualitative Results

Fig. 9.5 shows an example LT structure learned from the DISFA data. The tree has a relatively deep structure and includes 59 latent nodes. Each hidden node has on average 3.39 children and cardinality K=10 (K is optimized by a grid search). The mapping from leaf-node id numbers to landmark coordinates is shown in Fig. 9.7. The number of intermediate nodes on a path between two nodes can be seen as a dependence measure: e.g., if two nodes have the same parent, then they are highly dependent, and if two nodes are only connected over multiple higher levels, then their dependence is weak. Thus the learned tree structure can be interpreted as a hierarchical grouping of the features and targets according their dependence. We see that the AUs are grouped together with relevant landmark coordinates: AU6 (Cheek Raise) has a common parent (id 161) with landmarks from the eyebrow and nose (ids 49-51, 57, 60, 61, 66), AU12 (Lip Corner Puller) has a common parent (id 179) with landmarks around the lip (ids 115, 121), etc. The grouping seems to be sensible on higher levels as well, e.g. the latent node (id 180) that models the mouth AUs 17, 20 and 26, has a parent (id 188) that models further mouth related latent nodes (ids 178, 182, 183). As expected, the face contour landmarks (ids 1-34) are barely dependent on any AUs, since they are separately modeled on the right side of the tree and only connected to the AUs through the root (id 203). In addition, we discover some unexpected dependencies: AU1 (Inner Brow Raise) and AU2 (Outter Brow Raise) are far from the brow landmarks (ids 35-53), but rather close to the eyes (ids 84-93). This is probably due to the fact that the brows are often poorly tracked and thus the eyes might provide better information. AU9 (Nose Wrinkle) is grouped together with the eye landmarks, which might explain the poor performance in comparison to SVR (see Tab. 9.2). AU5 (Upper Lid Raise) barely depends on the eyes, but rather on the left lip

corner (ids 110 and 111), and shows a twice as good performance than SVR according CORR and ICC in Tab. 9.2. This indicates that AU5 might often co-occur with lower face AUs and thus is reliably detected by the mouth.

Fig. 9.6 shows an example LT structure learned from the ShoulderPain data. As in Fig. 9.5, AU4 and AU9 are grouped together, but this time with points around the nose (ids 56, 64 and 72) instead of the eye. We see that the same AUs show different correlations with landmark locations for different datasets. This might be caused by different frequencies of AU combinations, since AUs usually do not occur on their own but in combination with other AUs. Depending on which other AUs are active, the overall landmarks show a different pattern. ShoulderPain contains mostly AU combinations related to pain, while DISFA contains the emotions happiness, surprise, fear, disgust and sadness. Thus, AUs 4 and 9 might be better recognized during pain by points around the nose, while the eye provides more information for other emotions.

The PSPI is grouped together with AUs 6, 7 and 12. AUs 6 and 7 are expected, but AU 12 is not included in the PSPI formula. However, AU12 is related to pain as well (see [194]) and it seems that LT has learned this connection. Furthermore, AUs 4, 9, 10 and 43 are part of the PSPI formula, but not close to PSPI within the LT structure. This might be caused by the lower representation of these AUs within the dataset: the ShoulderPain database statistics (Tab. 4.1) show that AUs 6 and 7 are the most frequent ones and thus stronger correlated with PSPI.

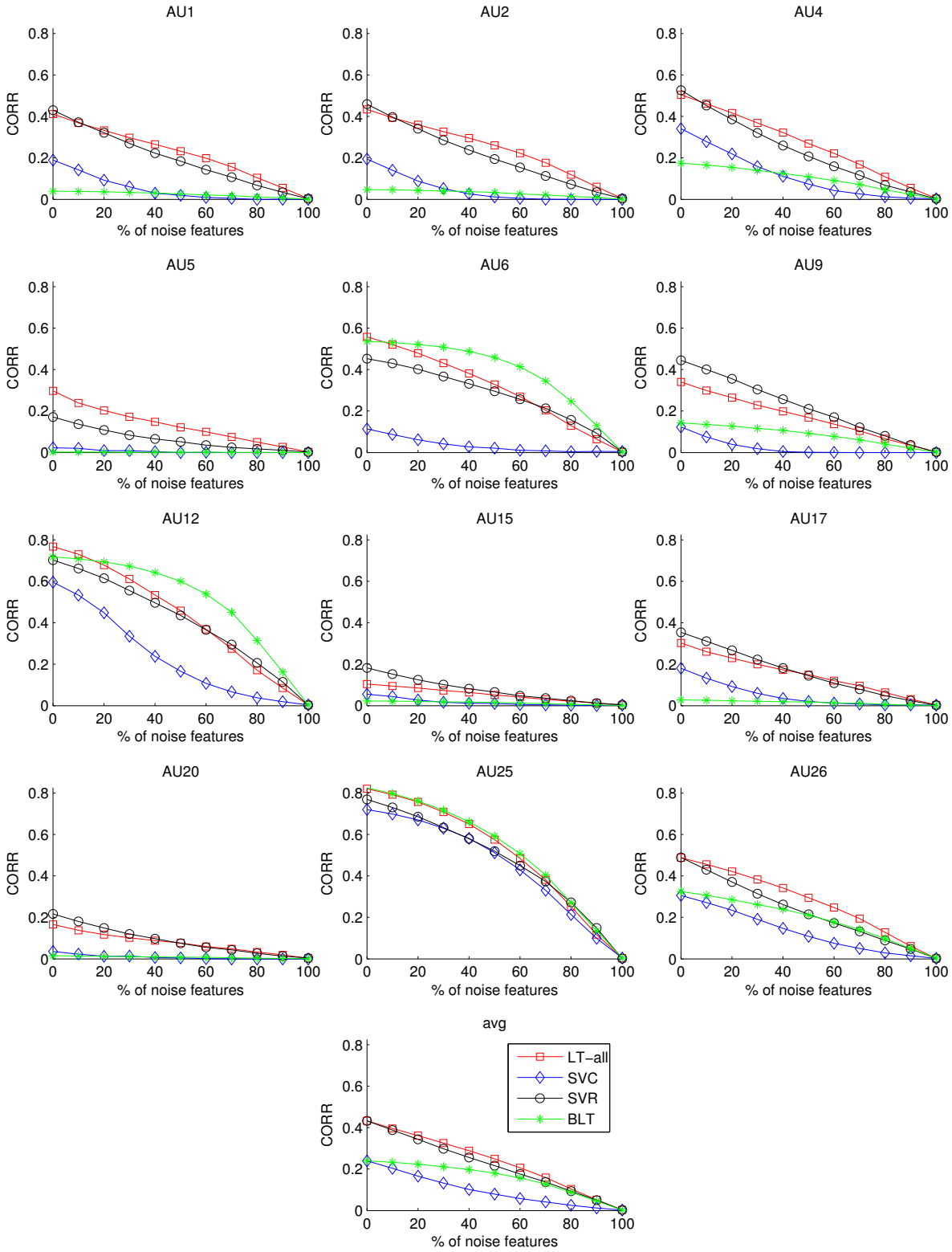


Figure 9.3: Results on the DISFA data for different AU targets. The LT-all model is compared to SVC, SVR and BLT. Each graph shows the correlation (CORR) as the percentage of noise feature varies.

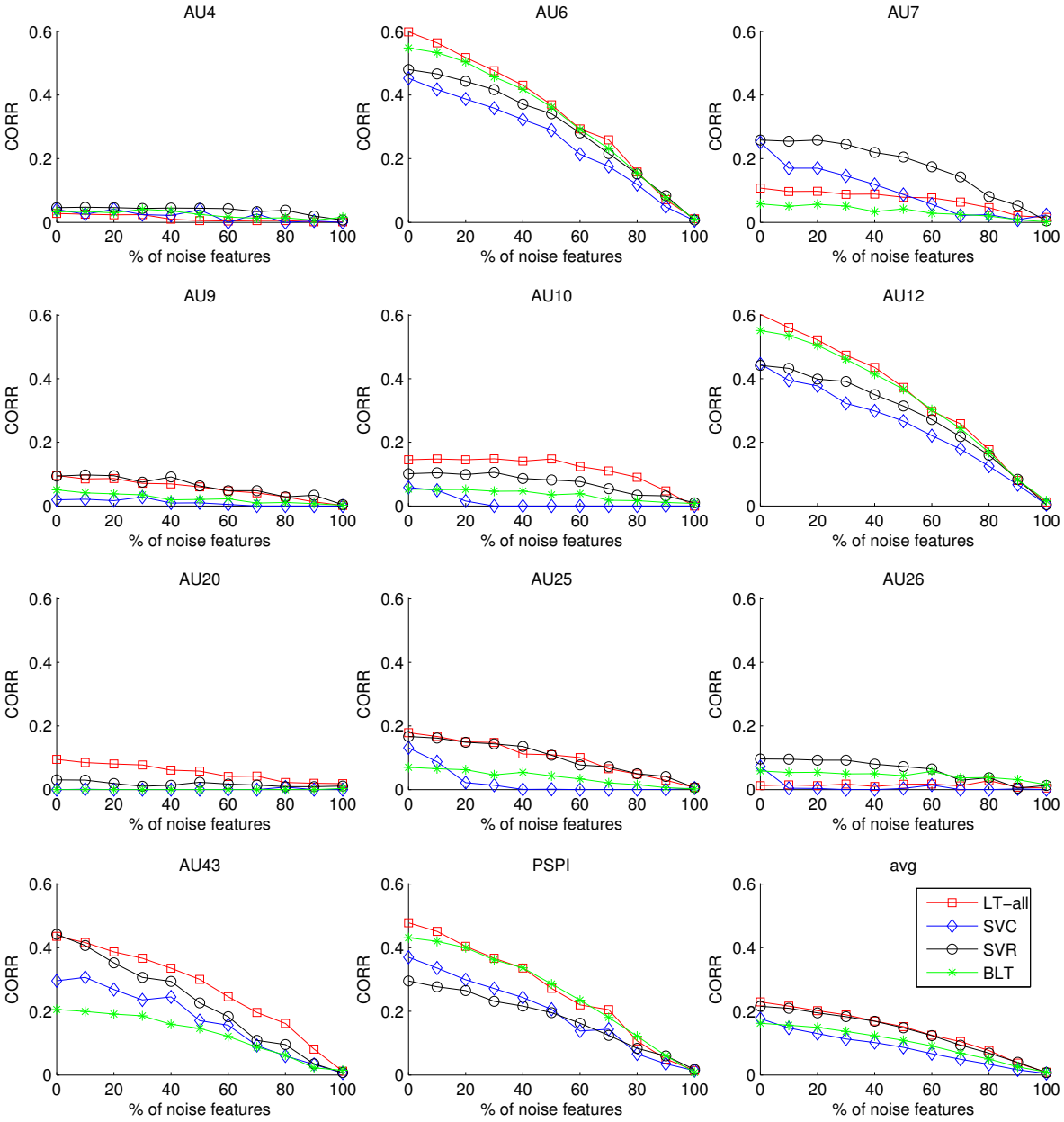


Figure 9.4: Results on the ShoulderPain data for different AU and pain targets. The LT-all model is compared to SVC, SVR and BLT. Each graph shows the correlation (CORR) as the percentage of noise feature varies.

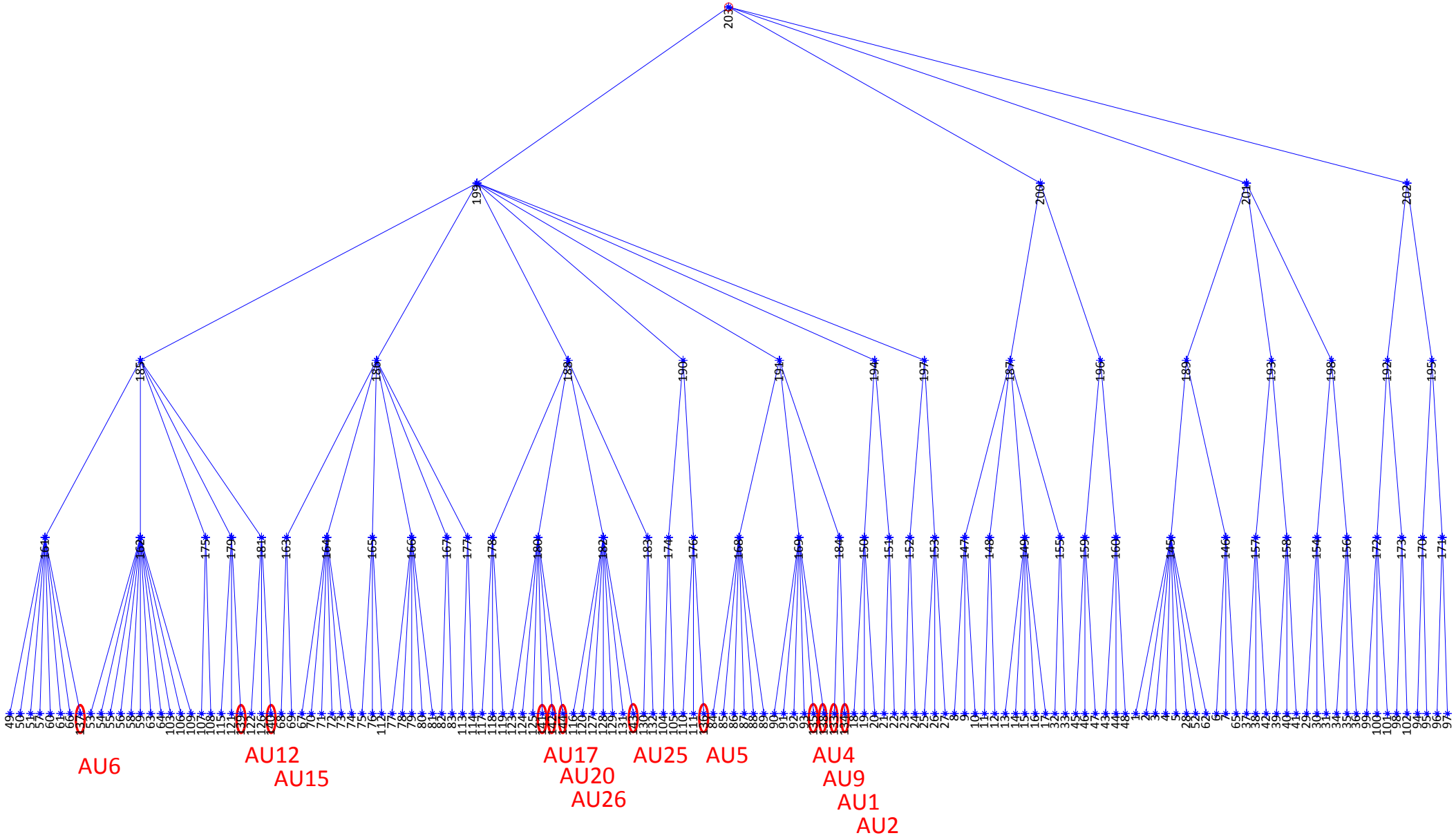


Figure 9.5: Example LT structure learned from training on the DISFA data. The leaf nodes are input variables that either correspond to landmark coordinates or AU targets. All AU target nodes are circled in red and annotated with the corresponding AU number. All non-leaf nodes are hidden variables. The LT has 203 nodes in total, which include 132 landmark coordinates (from 66 2-D landmarks), 12 AUs and 59 hidden variables. The nodes with ids 1-132 correspond to landmark coordinates (their mapping to face locations is shown in Fig. 9.7), ids 133-144 correspond to AUs and ids 145-203 correspond to latent variables.

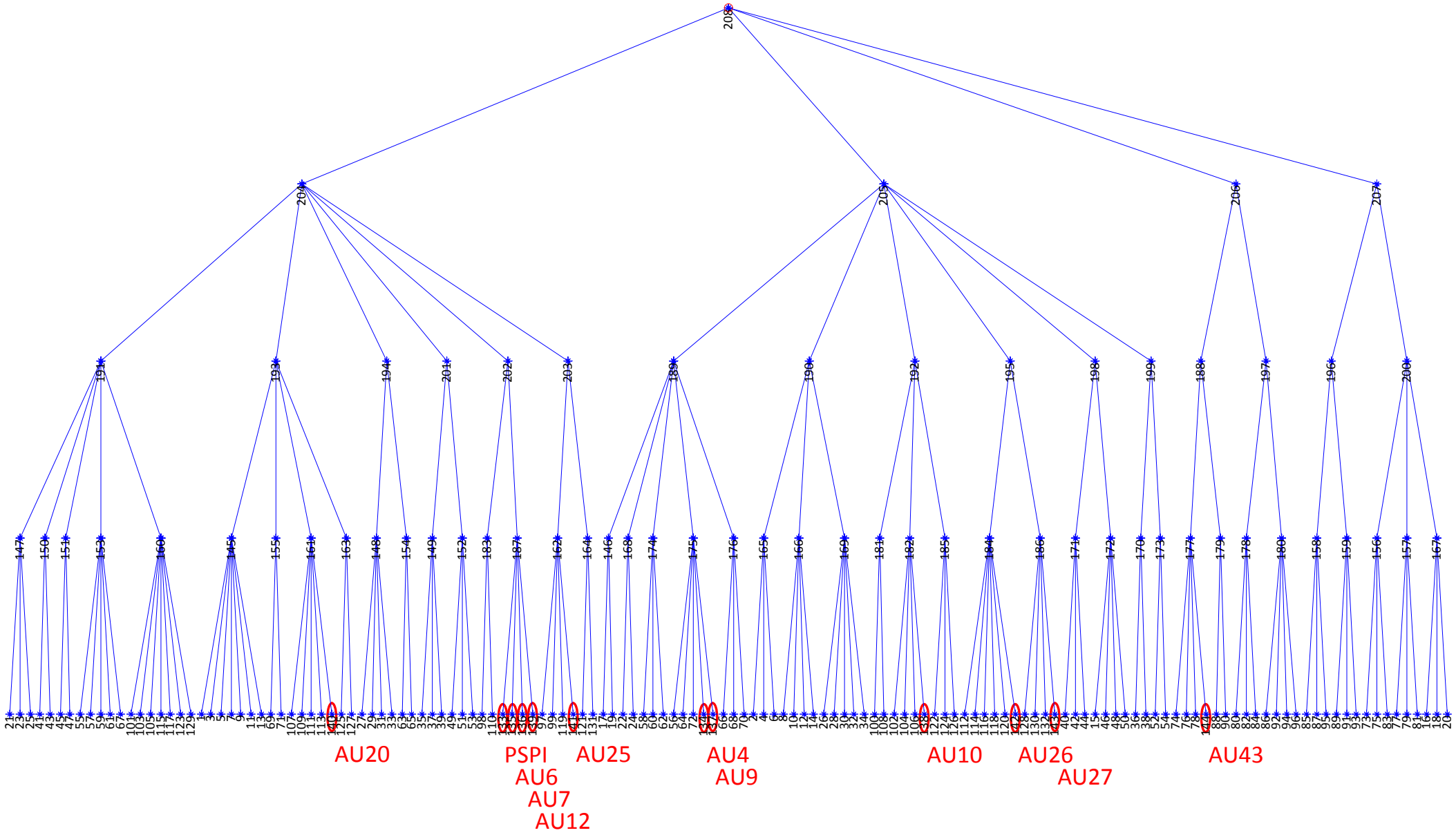


Figure 9.6: Example LT structure learned from training on the ShoulderPain data. The leaf nodes are input variables that either correspond to landmark coordinates or targets. All target nodes are circled in red and annotated with the corresponding AU number or PSPI for pain. All non-leaf nodes are hidden variables. The LT has 208 nodes in total, which include 132 landmark coordinates (from 66 2-D landmarks), 11 AUs, 1 PSPI and 64 hidden variables. The nodes with ids 1-132 correspond to landmark coordinates (their mapping to face locations is shown in Fig. 9.7), ids 133 corresponds to PSPI, ids 133-144 correspond to AUs and ids 145-208 correspond to latent variables.

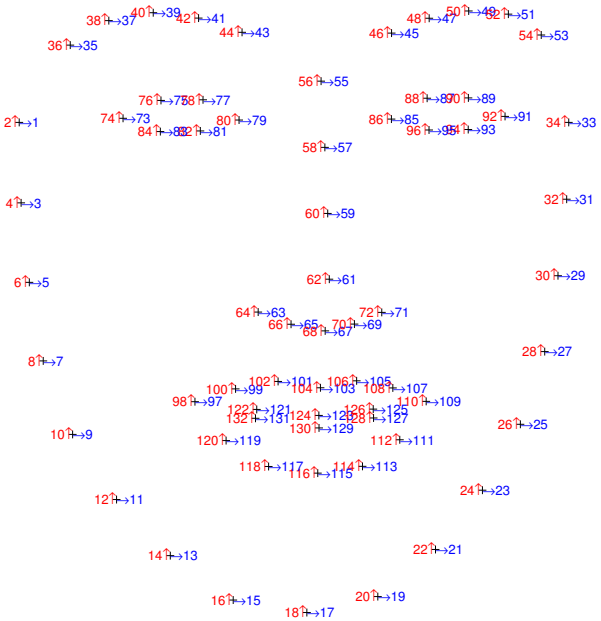


Figure 9.7: Mapping of facial landmark coordinates to the leaf-node id numbers of Fig. 9.5 and 9.6. The horizontal landmark coordinates are shown in blue and the vertical ones are shown in red. E.g. the horizontal landmark coordinate (blue) with the id number 5 corresponds to the leaf node with id number 5 in Fig. 9.5 and 9.6.

Fig. 9.8 shows the face model generated by LT on the DISFA data. Given that the LT model gets all AUs zero as input (i.e., the neutral face), we plot the inferred model output distribution in black. Then we overlay in red the changed landmark distribution if the model gets the maximum AU intensity as input. We can clearly see the correspondence between the distribution differences and the facial regions influenced by the related AU, e.g., AU1 mainly influences the eyebrow landmarks while AU12 mainly influences the landmarks around the mouth. The landmarks for AU15 have almost no difference to the neutral face, which shows that the model was not able to learn the subtle movements. This explains why SVR is significantly better in Tab. 9.2 for AU15.

SVR is also significantly better than LT for AU9. As can be seen in Fig. 9.8, LT is not detecting AU9 from the nose, but rather trough the eyebrows, which leads to a low performance. This is probably caused by co-occurrences of AU9 and AU4.

Fig. 9.9 shows the face model generated by LT on the ShoulderPain data. As for DISFA, we can observe meaningful regional emphasis for the different targets. The model tries to recognize AU4 through the lower nose, probably because AU4 is correlated with AU9. However, the correlation is not very strong, which explains the poor results. The results show that AU6 is recognized well, and the discriminant region seems to lie in the outer eye and mouth corners,

although it is not very pronounced. AU9 is recognized from the points around the nose and inner eye corners, which seems to make sense although the results are low. AU10 is very pronounced at the upper lip and eye lids, however the recognition results are low as well. AU25 is oddly shown on the left side of the face only, which might be due to out of plane rotations of the head. Pain is mainly recognized from the mouth region.

When comparing the common AUs from both databases within Fig. 9.8 and 9.9, we can see that AU4 is related to the brows in Fig. 9.8, while LT failes to detect the proper region within Fig. 9.9. And indeed, the results for AU4 are good on DISFA and poor on ShoulderPain. The situation is the other way round for AU6, which can be well detected within ShoulderPain, but there is no discriminative power within DISFA. AU26 shows importance around the mouth for both databases, but DISFA distributions are far more discriminative, which is again seen within the results.

9.5.4 Comparison with Prior Work

Several previous publications have already addressed the AU intensity estimation problem within the DISFA and ShoulderPain databases.

A dynamic ordinal regression framework is developed in [152], which reaches an average ICC of 0.58 on the DISFA and 0.62 on the ShoulderPain data. However, the model is evaluated on pre-segmented video sequences that both start and end with a neutral face, i.e., the facial expression events are known a-priori and only the exact intensity development needs to be inferred. Thus the results cannot directly compete within our setting, which does per-frame AU recognition without prior knowledge.

The work of [158] estimates the AU intensities by SVR and as a second step models the dependencies between AUs by a Markov random field. This work also evaluates the performance of a tree structure, reaching an average CORR result of 0.34 for the AUs 1, 2, 4, 5, 6 and 9, where our average is 0.42.

Furthermore, [121] learns a manifold of facial features and uses SVC on the learned manifold to classify different AU intensities, reaching an average ICC of 0.77. However the comparison to our method is not fair, since the supervised AU specific manifold learning includes the test subjects and thus the method is not subject-independent.

Different features and relevance vector regression is used within [86] on the ShoulderPain data, reaching an average CORR of 0.36 by fusing different appearance features. However the

points alone, which is equivalent to our setting, reach only an average CORR of 0.17.

9.6 Conclusion

This chapter introduced the novel LT model for joint estimation of multiple facial expression intensity targets. LT is able to model complex dependencies between features and targets through a set of hidden variables organized within a tree structure. We have formulated an efficient structure and parameter learning algorithm that iteratively maximizes the log-likelihood of training data while limiting the model complexity. Due to its generative framework, LT is able to provide more robust intensity estimations than competing methods, especially in the presence of noisy feature inputs.

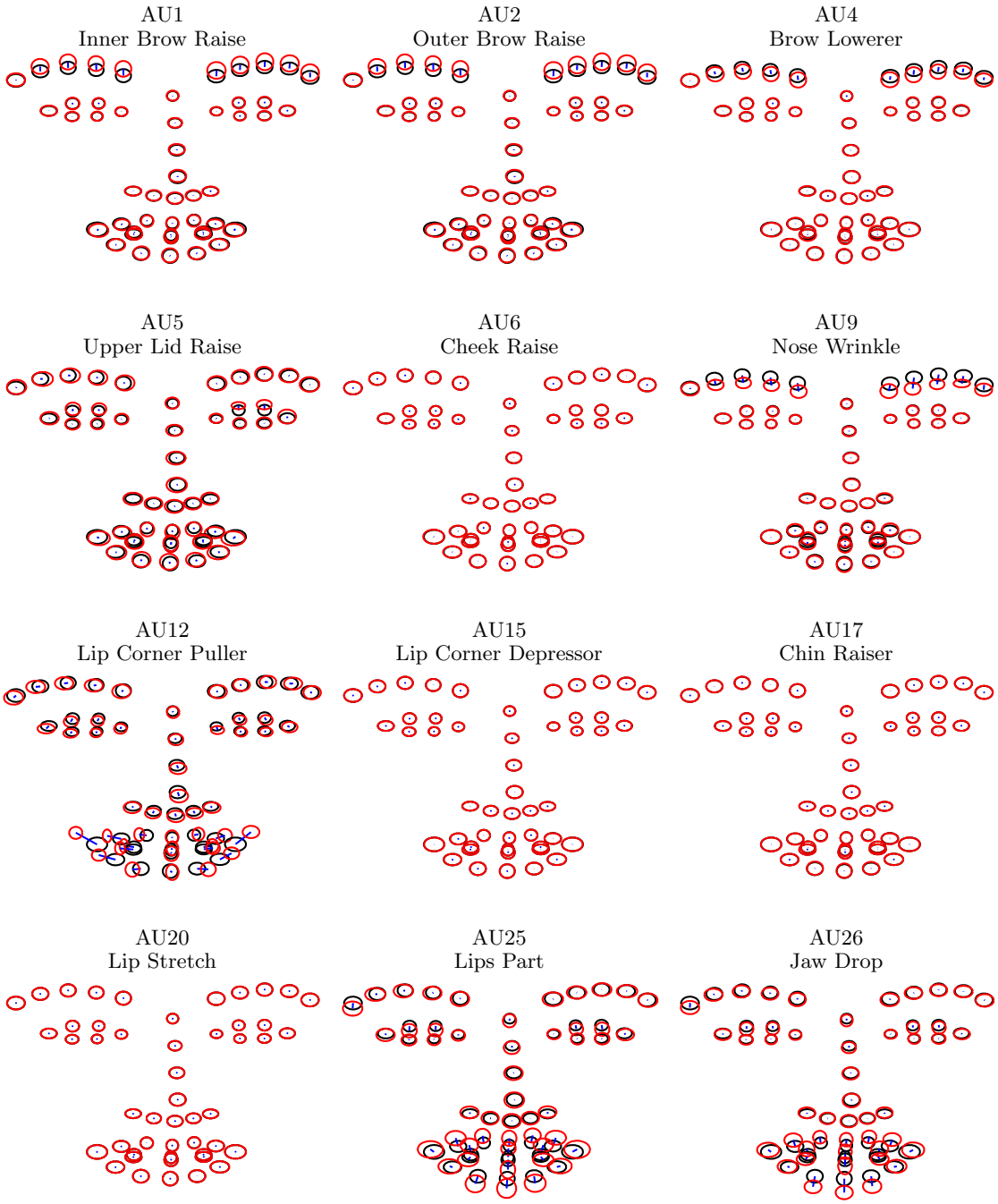


Figure 9.8: Landmark locations probabilistically sampled from an LT model trained on the DISFA data. The points along the face boundary have been excluded. For each AU, the corresponding face figure depicts the standard deviation of landmark locations for both the zero intensity (black ellipses) and highest intensity levels (red ellipses). Each of the ellipses is drawn with the corresponding distribution mean as center. Also, every face figure shows the difference between the mean of zero intensity and highest intensity (blue lines). If there is no difference, then only a small blue dot is visible. The ellipses cover each other, i.e., if the distribution does not change, then just a red ellipse is visible.

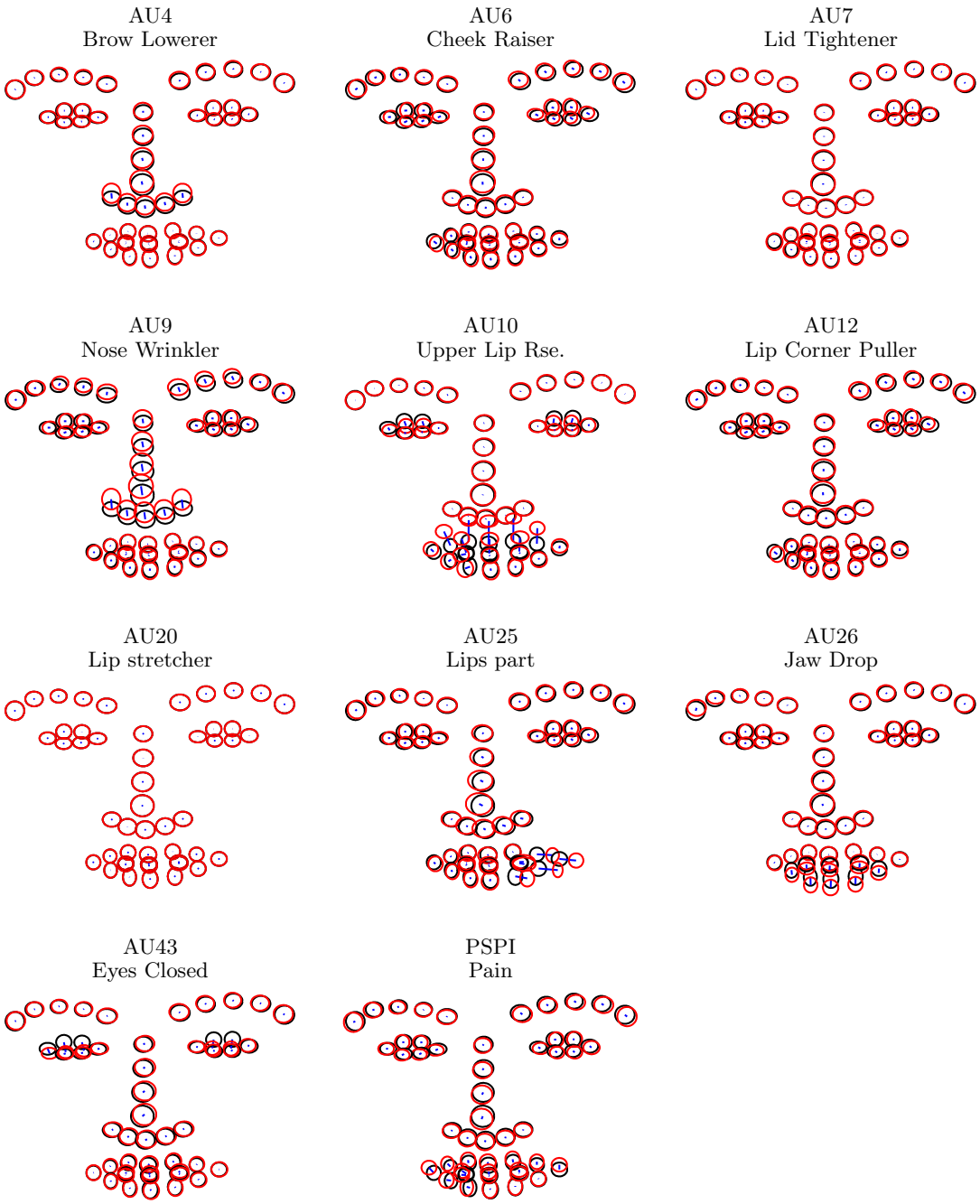


Figure 9.9: Landmark locations probabilistically sampled from an LT model trained on the Shoulder-Pain data. The points along the face boundary have been excluded. For each AU, the corresponding face figure depicts the standard deviation of landmark locations for both the zero intensity (black ellipses) and highest intensity levels (red ellipses). Each of the ellipses is drawn with the corresponding distribution mean as center. Also, every face figure shows the difference between the mean of zero intensity and highest intensity (blue lines). If there is no difference, then only a small blue dot is visible. The ellipses cover each other, i.e., if the distribution does not change, then just a red ellipse is visible.

Chapter 10

The EmoPain Database

Contents

This chapter describes the EmoPain database, which resulted from the project "Pain rehabilitation: E/Motion-based automated coaching"[1], in short "Emopain". It was a joint project of Imperial College London, University College London and University of Leicester and has been funded by EPSRC under the grants EP/H017178/1, EP/H017194/1 and EP/H016988/1. The author of this thesis was responsible for the recoding process of video and audio signals, the synchronization between modalities, the labeling of facial expressions, and the automatic pain intensity estimation from facial expressions, which are also the focus of this chapter. More information about the electromyography and motion capture modalities, as well as the body behaviour labeling, is provided by Aung et al. [7].

10.1 Introduction

In recent years there has been a drive toward accurate sensing and robust interpretation of activity within exercise and physical rehabilitation systems [67, 139, 140]. In part, this has been done to alleviate the high demands placed upon a limited number of healthcare staff as

[1]Project website: www.emo-pain.ac.uk

well as to make rehabilitation more enjoyable (e.g., through the use of games). This has led research and industry to develop systems deployable in non-clinical settings such as the home or workplace, many with the objective of providing corrective biomechanical feedback [140]. However, in such systems factors relating to the emotional states of the user have been largely ignored. For certain chronic conditions this is a major shortcoming since emotions are a major factor in impeding rehabilitation and directly affects the efficacy of long term management strategies where a user can become anxious, discouraged and ultimately demotivated [100].

A particular case where emotional factors undermine adherence to successful rehabilitation is chronic pain. Chronic pain is defined as pain that persists despite the resolution of injury or pathology or with no identified lesion or pathology [180]. It is attributed to changes in the central and peripheral nervous system resulting in amplified or uninhibited pain signals [16, 178]. These changes are closely linked with distress and affect behaviour, quality of life and daily function which can further result in depression, anxiety and social isolation [115].

Although management of all chronic conditions are generally subject to moderating factors that affect adoption and adherence to their respective therapies [94], chronic pain differs in that pain conveys threat [32]. Emotionally, this generates anxiety as well as contributing to catastrophic thinking. Untempered levels of anxiety can cause marked reluctance to undertake therapies which are perceived as potentially exacerbating pain to the extent of avoiding them [89, 191].

In this chapter, we focus on chronic musculoskeletal pain which affects an estimated one in ten adults globally [65]. For this common form of chronic pain, avoidance results in a reduction of beneficial physical activity as well as the overuse of alternative parts of the body due to the reluctance in moving perceived painful body regions. This could even lead to impairment in motor control where there is proprioceptive dysfunction [36]. The benefits of adherence to activity in rehabilitation are also well understood. It protects against weakening, stiffness and inhibits the neurophysiological mechanisms underlying the spread of pain. It also increases confidence in physical capacity, underpins achieving valued goals [62] and improves quality of life [195].

Qualitative studies [168] showed how physiotherapists with cognitive behavioural training make use of patients' behaviour to decide upon the type and timing of encouragement during therapy. Such interventions can vary from breathing prompts to the partitioning of an activity into less daunting piecemeal tasks, or simply providing information and reassurance. Physiotherapists were also reported to use behaviour as a measure of a person's progress in learning

to manage their own condition and use it to pace the transfer of management responsibilities from the physiotherapist to the individual; this eventually leading to fully effective self management.

Currently, experts are unable to provide the ideal amount of continuous long-term monitoring and motivation given the large number of people with chronic pain. This leads to a reliance on unsupervised self-management [179] which lacks encouraging feedback and therefore risks limiting or even reversing treatment gains. Clearly, the deployment of automated systems with the capacity to recognise pain related expressions and behaviours would be a major step toward in fulfilling this requirement gap. In principle, affect awareness integrated into self-use rehabilitation systems would allow for the development of systems that can provide tailored support and feedback during physical rehabilitation sessions.

We fill an empirical gap by supplying a multimodal fully labelled dataset for the most common musculoskeletal form of chronic pain, namely Chronic Lower Back Pain (CLBP). This is a disabling condition and often coincides with high levels of chronicity [6]. We focus on one form of musculoskeletal chronic pain as mixed data from different types of musculoskeletal chronic pain (e.g., neck or shoulder) would introduce added complexities and potential confounds within the dataset. However, it should be noted that, once a person has CLPB, the use of maladaptive body behaviour may lead to the emergence of pain in other parts of the body. The fully labelled multimodal dataset (named 'EmoPain') contains naturalistic pain-related affective expressions (facial and vocal expressions) and behaviours (movement and muscle activity) of people with CLBP while carrying out physical activity. Finally, we present the results of an analysis of the facial expression data by the previously introduced automatic pain intensity estimation methods.

This chapter is organized as follows: in Sec. 10.2, we present our data collection procedure, including details regarding patient recruitment, sensor set up and trial procedure. Sec. 10.3 describes two labelling procedures for face expression and body related behaviours. In Sec. 10.4, we describe the landmark localization procedure and Sec. 10.5 reports results on automatic pain recognition. Finally, Sec. 10.6 concludes by discussing the findings and possible directions on how these could be addressed.

10.2 Data Collection

In this section we detail the acquisition and resultant content of the EmoPain dataset. We aimed to maximize naturality of the elicited data as well as resolution, quality and synchron-

ization accuracy.

10.2.1 Patient Recruitment

Potential participants were initially identified by health care staff predominantly from the Pain Management Centre at the National Hospital for Neurology and Neurosurgery, UK as well as through the pain charity 'Backcare', once identified they were informed about this study and referred to our team upon further interest.

Informed consent was sought from participants for taking part in the study and dissemination of the data including the sharing of data with other researchers. All identifiable information was anonymised (e.g., names and dates of birth). An exception to the anonymisation is the attributes within the video data. Only videos or images of those participants who provided written consent to disseminate and share video data is made available to the research community. Ethics approval was obtained through the NHS Ethics committee (11/LO/007) for people with chronic pain recruited through the hospital and through the UCL ethics committee (10/9456) for people recruited through pain groups and for healthy participants.

For each potential participant a brief structured initial interview was carried out by a clinical psychologist trained in pain management. During this process eligibility was determined based on the Mini International Neuropsychiatric Interview (MINI) [98] to ascertain major psychiatric co-morbidities other than depression and anxiety that may alter emotional expressivity (e.g., psychosis or substance abuse). Having CLBP for more than 6 months was a further inclusion criterion.

From this superset, patients were excluded if: the principal pain was not located in the back, they had need of mobility aids, had joint replacement, arthrodesis or limb amputation, neuropathic pain, spinal stenosis, cardiovascular or respiratory disease, learning disability, poor understanding of English or were pregnant. A final set of 21 CLBP patients was determined (7 male, 15 female, mean age 50.5, 17 Caucasian, 3 black and 1 south-Asian). Though small, this group is typical of people with chronic pain seeking treatment: two thirds were female [195], they were mostly middle aged, and substantially disabled by their pain.

Furthermore, 28 healthy control subjects with no history of CLBP (14 male, 14 female, mean age 37.1, 26 Caucasian and 2 Asian) were also recruited from random volunteers from the local community as well as people known to the research team. The control participants were recruited to provide a variety of ways the recorded physical exercises would be executed in the absence of pain. Two main reasons have led to their inclusion. First, we assume that

Table 10.1: **CLBP Participants' Profile Summary.** Shown are the patient identifier (ID), the age in years, female (F) or male (M) gender, sum of Hospital Anxiety and Depression Scores (HADS) [208] with scale 0-42, sum of the Pain Catastrophizing Scores (PCS) [172] with scale 0-52 and mean levels of Self Reported Pain and Anxiety for all exercises in the normal (N) and difficult (D) trials with scale 0-10.

ID	Age	Gender	HADS	PCS	Pain		Anxiety	
					N	D	N	D
3	63	M	4	2	0.0	0.0	0.0	0.0
4	53	F	25	14	0.0	0.2	0.0	0.0
5	65	F	16	13	5.5	5.8	0.9	0.9
6	27	F	25	18	5.1	5.7	1.9	3.5
7	31	F	8	2	2.8	2.7	0.0	0.0
8	64	M	20	17	5.0	5.6	1.9	1.7
9	62	M	25	30	5.8	6.7	0.0	0.0
10	56	M	11	12	3.9	4.7	0.0	0.0
11	36	M	19	15	1.4	1.8	0.0	0.0
12	58	F	17	13	0.4	0.8	0.0	0.0
13	-	F	8	6	6.1	3.9	0.0	0.0
14	55	F	11	15	1.1	1.7	1.3	2.1
15	33	F	11	8	4.1	3.9	2.9	2.3
16	19	M	30	42	7.1	7.6	2.9	2.7
17	38	F	5	0	0.0	0.0	0.0	0.0
18	-	F	21	37	2.6	3.0	0.0	0.0
19	51	F	15	5	0.5	0.1	0.1	0.0
20	67	M	24	33	6.6	8.7	6.3	8.0
21	62	F	8	11	1.1	1.4	0.1	0.3
22	56	F	32	44	4.7	5.6	4.0	2.3
23	65	F	11	17	0.0	0.0	0.0	0.0
24	50	F	34	42	6.1	7.7	0.0	0.0
mean	50.5		17.3	18.0	3.2	3.5	1.0	1.1

there is not a perfect way of executing an exercise, especially when not instructed, that can be taken as a model from which people with chronic pain may deviate [168]. Second, people are idiosyncratic and hence the data should account for this to improve the effectiveness of the automatic recognition model. Hence, even if in this work the control data will not be analysed, they are included in the EmoPain dataset to allow for benchmarking in subsequent further studies after public release.

10.2.2 Trial Procedure

Before recording, the CLBP initially completed a questionnaire to ascertain pain experience, affective state and daily activity with questions based on established pain questionnaires: the Hospital Anxiety and Depression Scale (HADS) [208] and the Pain Catastrophizing Scale (PCS) [172]. The HADS score is a measure of anxiety and depression, together scored as

distress, developed for use in populations with illness and disability, and widely used in chronic pain. The PCS score assesses one of the pivotal cognitive-emotional variables in chronic pain, with substantial predictive power in behaviour [32, 62, 191, 195]. These scores are provided in columns 4 and 5 in Tab. 10.1 with ranges: HADS (range 4 to 34) and PCS (range 0 to 44). The profiles (Tab. 10.1) were gathered to provide an understanding of the representativity of the dataset. At the same time, as the dataset will continue to grow, the profiles may be useful to improve the automatic recognition systems by considering person factors (e.g., gender, level of depression).

Anthropometric measurements were then manually taken using calipers: height, upper arm lengths, forearm lengths, thigh lengths, shank lengths, waist width and shoulder width. The subject's weight was also measured. Full body frontal and sagittal photographs were taken of each participant while standing inside a cube framework of a known size. These images were annotated to determine the skeletal proportions at later stage to inform the motion capture data. This data were necessary for the calibration of the movement recording sensors.

Three sensory systems (detailed in Sec. 10.2.3) were then attached to the participant: four wireless surface electromyographic (sEMG) probes (Fig. 10.4b), a motion capture suit consisting of eighteen microelectromechanical (MEMS) based Inertial Measuring Units (IMU) (Fig. 10.4a) and a head mounted microphone. System initialization also included the adjustment of a camera rig supporting five face level cameras to the correct height (detailed in Sec. 10.2.3) and the calibration of the motion capture suit.

The exercises undertaken by the participants were a set of basic actions agreed by physiotherapists with expertise in treating CLBP. The exercises were varied yet consistent with known movements that generally place demands on the lower back. They are also functional activities that represent everyday tasks that those with CLBP may perceive as difficult and thus avoid for fear of pain [113].

For each exercise, two levels of difficulty were used and performed separately to elicit a wider range of pain-related behaviour. A minimum of two trials (one at each level of difficulty) were then conducted for each participant. The easier trial (normal) consisted of the following seven exercises: 1) standing on the preferred leg for five seconds initiated at the time of the subject's own choosing, repeated three times, 2) sitting still on a bench for thirty seconds, 3) reaching forwards with both hands as far as possible while standing, 4) standing still for thirty seconds, 5) sitting to standing initiated at the time of the subject's own choosing, repeated three times, 6) bending down to touch toes and 7) walking approximately 10 metres with one 180 degree

turn.

In the difficult trial, four of the exercises were modified to increase the level of physical demand and possibly of anxiety: 1) standing on the preferred leg for five seconds initiated upon instruction repeated three times and then on the non-preferred leg in the same manner, 3) reaching forwards with both hands as far as possible while standing holding a 2 kg dumbbell, 5) sitting to standing repeated three times initiated upon instruction, and 6) walking as before while carrying one 2 kg weight in each hand, starting with bending down to pick up the weights.

After each exercise instance the CLBP group also self reported the level of pain and anxiety from a 0-10 scale, the mean value of these scores are shown in columns 6-9 in Tab. 10.1, the N and D descriptor indicates the normal and difficult exercise set respectively.

10.2.3 Recording Apparatus

As rehabilitation technology moves into non-clinical settings, an understanding of system requirements in terms of sensing modality, configuration and data granularity for affect aware systems is needed. We use apparatus to that maximises fidelity and resolution. This will allow the research community to determine the minimum levels of data dimensionality, granularity and accuracy needed for robust recognition and further facilitate the design of wearable technology or cheaper and less invasive motion capture technology if the feature requirements are within the sensing limitations of the simpler devices. For example, with the advent of more accurate marker-less sensors (e.g Kinect 2) there is a greater potential for these sensors to be used.

Cameras and Audio

We configured 8 monochrome video cameras as shown in the plan view (Fig. 10.1) and the photo (Fig. 10.2). All cameras had a resolution of 1024×1024 pixels and a frame rate of 58 fps. 5 of the cameras covered the frontal 90 degrees of a circle around the main exercise spot at ca. 1.5m height, and were mounted together on an aluminium rig. Camera 8 pointed up from the floor so that the subject's face is captured when leaning forward. A long range camera was placed at the front right corner to capture a general overview of the scene. Another long range camera was placed at the front centre to capture facial expression during the walk exercise. The use of this multiple view camera set up allows for more unconstrained instruction during the exercises and therefore capturing natural movements. The main exercise area was walled by a series of 2m whiteboards to improve the passive lighting conditions. In total, 8 active lights were used: 2 pointed to the whiteboards behind the camera rig, 4 pointed from above

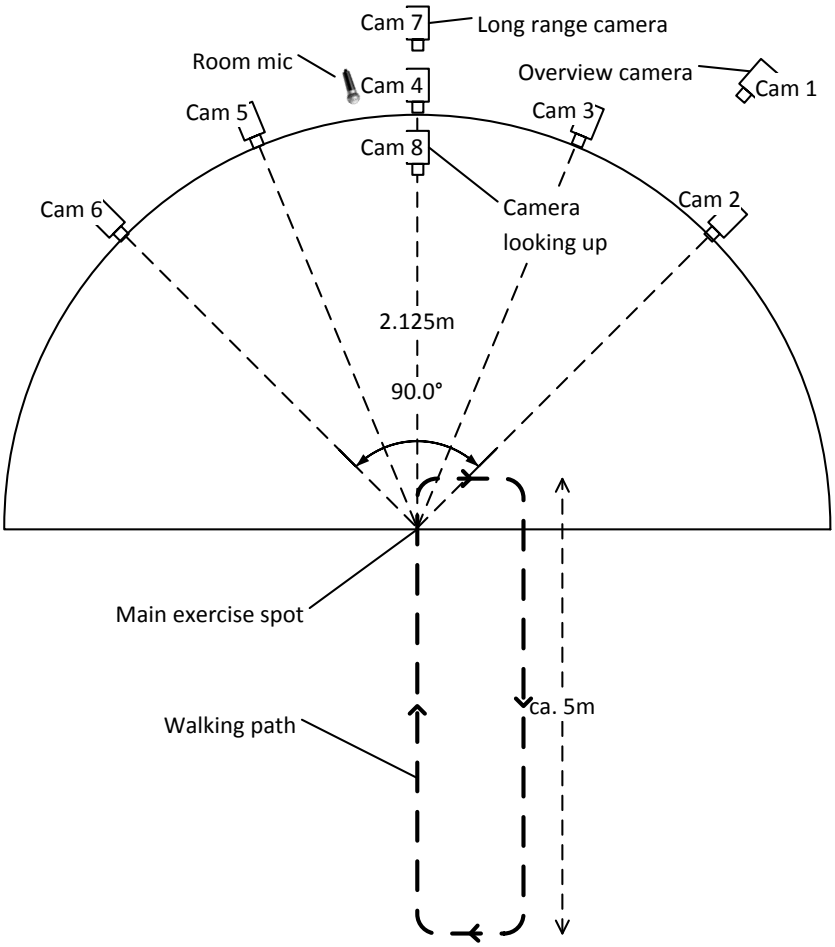


Figure 10.1: Plan view of the configuration of eight high resolution cameras, which includes one overview camera (Cam 1), five cameras mounted on a common rig to cover the frontal 90 degrees of a circle around the subject to allow for unconstrained natural movement (Cam 2-6), one long range camera for distance exercises (Cam 7) and one floor camera to capture the face during forward flexion (Cam 8).

the camera rig to the main exercise point, and 2 pointing from below the camera rig to the main exercise point. The tall whiteboard panels also created a more private space for the participants, only the participant and the physiotherapist or psychologist were allowed in this area. An example frame from all synchronized cameras is shown in Fig. 10.3.

The audio signal was captured with two microphone channels, recorded at a rate of 48 kHz with 24 bit Pulse Code Modulation. The first channel was provided by an AKG C-1000S MKIII condenser microphone that was placed next to the centre camera on the rig and pointed towards the main exercise point. The second channel was recorded from a wireless AKG HC 577 L condenser headset microphone that was worn by the subject.

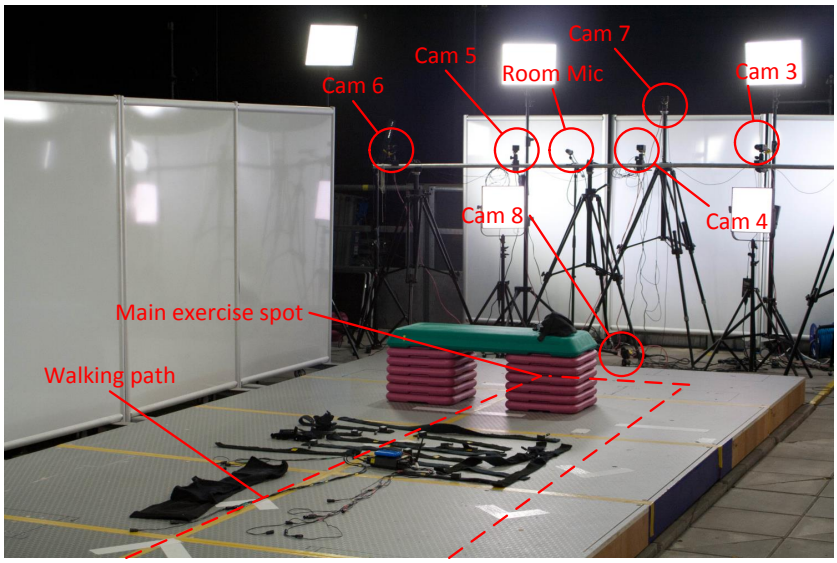


Figure 10.2: Photo of the EmoPain recording setup. The main exercise spot is just behind the bench and the walking path ends in front of the bench. Cam 1 and 2 are not visible.

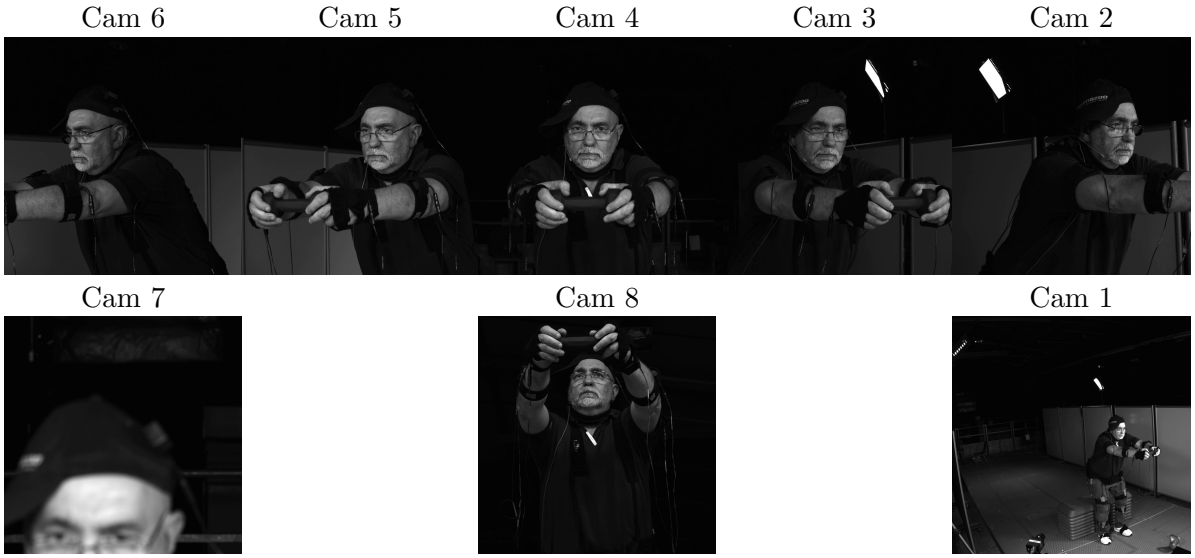
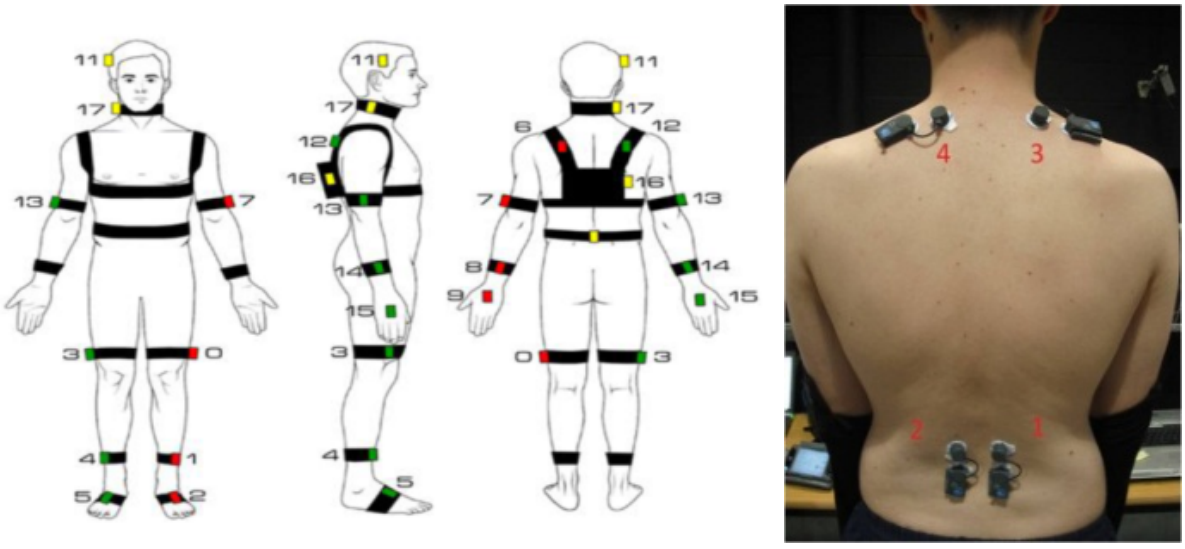


Figure 10.3: Synchronized example frame from all eight camera views, showing a CLBP patient during difficult trial of the reaching forward exercise. The long range camera (Cam 7) is only in focus during the walking exercise, when the subject is further away from the camera.

Motion Capture and Electromyography

A customized motion capture suit that specifically addresses the comfort requirements of CLBP patients based on the Animazoo IGS-190 system was used. Each sensor was a microelectromechanical systems (MEMS) based inertial measurement unit (IMU) with Velcro attachment straps; this was done to minimize the amount of tight fitting material worn by the



(a) (b)

Figure 10.4: IMU and EMG attachments with annotated sensor IDs: (a) customized motion capture suit (Animzaoo IGS-190), which includes eighteen inertial measuring units attached with Velcro strapping on all main rigid body segments. The use of minimal attachment material reduces the sense of restrictiveness and to encourage naturalistic motion. (b) Four wireless surface electromyographic sensors (BTS FREEEMG 300). Probes 3 and 4 are placed on the upper fibres of the trapezius muscles. Probes 1 and 2 are placed bilaterally on the lumbar paraspinal muscles approximately at the 4th/5th lumbar vertebra.

participants to enhance comfort, reduce the sense of restrictiveness and maximize naturalistic motion. Twelve sensors were placed on rigid limb segments (4 limbs $\times$3 segments); one on the hip, centre of the torso, and one on each shoulder, neck and on the head totalling eighteen sensors (see Fig. 10.4a). The IMUs were connected in parallel and each returned 3-D Euler angles sampled at 60 Hz.

The whole body skeletal proportions of each subject (gathered as described in 4.2) combined with the rotational information from the Euler angle data were used to calculate the positional triplets of 26 anatomical points in 3-D Cartesian space. This was done using the MoCap toolbox for Matlab [96]. Four wireless sEMG adhesive probes (BTS FREEEMG 300) were attached to the skin (Fig 2b). Two probes were placed on the upper fibres of the trapezius muscles orientated along the alignment of the fibres of the muscle bilaterally. Two further probes were placed on the lumbar paraspinal muscles approximately at the lumbar 4/5 level bilaterally. The skin con-tact area was initially cleaned using isopropyl alcohol prior to attachment. Two disposable 24 mm silver/silver chloride electrodes containing integrated adhesive and conductive gel were snapped onto each sensor. The data was recorded at 1 kHz.

Synchronisation

The four recording systems (cameras, audio, motion capture and EMG) were controlled by a single triggering script which starts the four systems in sequence. The start and end timestamp of each modality were recorded based on a common clock.

The cameras were synchronized between each other by a trigger signal that was sent by a master camera. This trigger signal was recorded as an additional audio channel and it provided further synchronisation between the cameras and the audio. Moreover, the motion capture system provides an external trigger signal which was also recorded as an audio channel. The sEMG system started with the first camera trigger; hence the synchronization between video and sEMG is given.

This information is sufficient to align all modalities post recording with an extremely low error margin; the resulting audio-visual synchronization error is bounded to 25μs. Specific details about this synchronization procedure can be found in [104].

10.3 Labelling of Pain Expression in the Face

Labeling naturalistic data is a complex and challenging process, especially when current coding systems are not well established. In this section we describe the labeling process used for this dataset, the rationale behind it and through its analysis we discuss issues that this raises. We describe the rater based labelling procedures for pain expressions from face videos. See [7] for pain related movement behaviours from videos of a full body perspective.

The facial expression of pain (*grimacing*) of 17 patients was continuously labelled by eight independent naïve raters. The videos of 4 patients were not included in this procedure due to non consent for video release or synchronization error. The raters (5 female and 3 male) were 22 to 30 years old and have no particular experience in rating pain. Naïve raters were used for facial expressions of pain to maximize the number of ratings (as FACS was not used) by relying on general human recognition levels in reading pain from face. However, in order to familiarize the raters with pain expressions and the rating procedure, they were instructed to rate the ShoulderPain database [109] as preliminary step.

Once the training had been completed, the raters visually inspected the EmoPain videos showing a simultaneous dual view from two cameras: the central camera 4 and the camera pointing up from below camera 8 (see Fig. 10.1). The camera 8 footage was included as some of the exercise involved a forward flexion motion where only a camera point up form below would

Figure 10.5: Example screen as presented to the raters. It shows the parallel view of Cam 4 and 8, as well as a pain rating slider on the right. Pain is annotated by a joystick and direct visual feedback is provided by the slider position and color. The color changes continuously from black for no pain to bright red for maximum pain.

capture the face during these motions. Each video contained the entirety of one unsegmented trial (described in Sec. 10.2.2), the durations of which are the trial lengths in actual time, ranging from 3 to 6.5 minutes with an average of 4.6 minutes.

Each video was loaded into our self-developed annotation tool that uses a gaming joystick as an input device. To provide an as natural setting as possible, play back was done at real-time with 29 fps. The annotators were instructed to move the joystick according to their personal perception of pain, while the neutral joystick position describes no pain and the maximum forward displacement represents the highest possible pain level. The currently annotated pain level is visually reported as a bar on the side of the video in order to give the annotators immediate feedback and thus locate the current pain level between no pain and the maximum possible level. An example screen-shot from the annotation tool is shown in Fig. 10.5. The procedure provides multiple ratings per trial from each rater. Each sequence contains continuous values between 0 and 1, where 0 represents the neutral position and 1 the maximum position of the joystick. An example pain sequence with annotations from all raters is shown in Fig. 10.6.

The rating procedure differs from [109], where pain is labeled by determining the discrete intensity labels of a predefined pain-related set of action units (as defined by the FACS [43]), and then calculating pain according to these labels as indicated by [144] and resulting in a 16

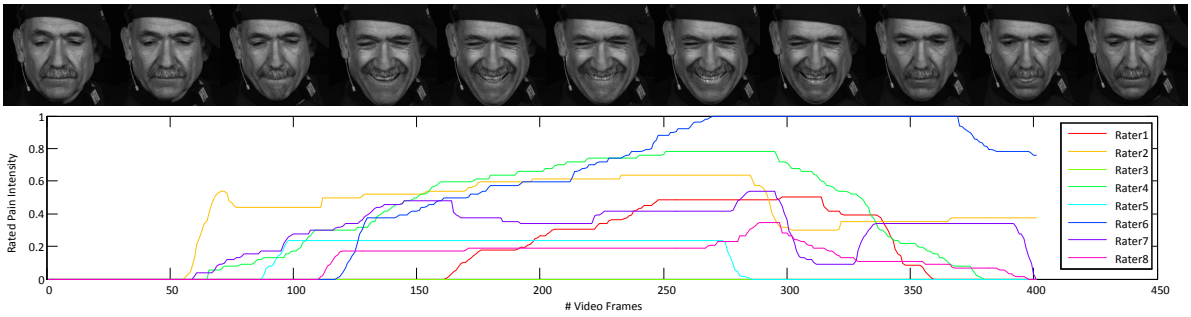


Figure 10.6: Cropped video frames from Camera 4 showing an example grimace (above) with all eight temporally concurrent observer's ratings for pain (below).

level discrete pain scale. In contrast to that, we directly measured pain by observer ratings which lead to a truly continuous pain scale.

The Intra-Class Correlation Coefficient ICC(C,1) [125] on the continuous ratings of all raters is low at 0.21, which can be due to varying reaction times, different individual pain perception or misinterpretation of non-pain related facial expressions. In order to get a more reliable continuous ground-truth, we identify a subset of raters that interpret facial expressions of pain in a similar way and combine their ratings by taking the average per frame. The exploration of more sophisticated combination methods (e.g., Dynamic Probabilistic CCA [133]) is out of the scope of this thesis and is thus left for future work. The ICC between each pair of the raters is shown in Fig. 10.7. It shows that rater 2 barely correlates with any other rater, its ICC is at most 0.2. Raters 4, 6 and 7 fairly agree with the other raters, their ICCs are between 0.18 and 0.38. However, the subset that highly correlates with each other are raters 1, 3, 5 and 8. Their pairwise ICCs range from 0.56 to 0.74, and thus we only use these raters to obtain the pain ground-truth. The common ICC of these 4 raters is 0.65, which is considerably higher than the 0.21 between all raters. This does not mean that the other ratings are 'wrong', as every observer might have a different subjective perception, see Sec. 2.2. However, for reliable automatic pain recognition, we need to focus on pain ratings that are independently reproducible with a high ICC, which is only given for raters 1, 3, 5 and 8.

It is possible that rating on a few discrete levels (e.g. 2 or 3) would lead to a better agreement, since rating continuous levels is more time-consuming, involves more steps and thus is more error prone [83]. However, given the very low agreement between certain raters and high agreement between others, the discrepancy is probably caused by the different interpretations of facial expressions, rather than the rating procedure. If it would be caused by the rating procedure, then it should affect all raters equally.

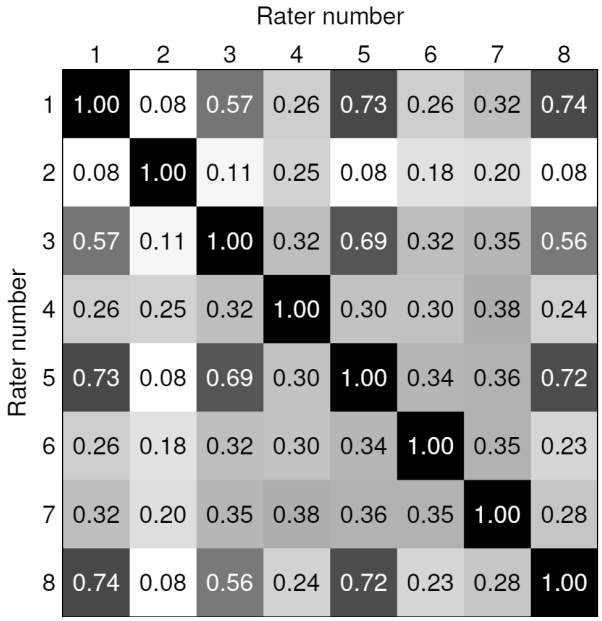


Figure 10.7: The Intra-Class Correlation Coefficient (ICC) of the pain ratings between all pairwise combinations of the 8 raters. Each row and each column corresponds to one rater, e.g., the value in row 3 and column 4 corresponds to the ICC between raters 3 and 4. All values on the diagonal are 1, since each rater fully agrees with itself. The matrix is symmetric, because the ICC is commutative.

The distribution of the obtained pain ground-truth is highly biased towards the pain level 0, which is annotated for 89% of the frames. The distribution of frames is shown in Tab. 10.2. The majority of pain ranges between intensities 0 and 0.2. Levels 0.2 - 0.7 are relatively rare, while there is again a small peak at 0.7 - 0.8. Pain is usually shown during the exercise movement, while the rest of the time the participants receive instructions or give feedback to the instructor, and thus do not show pain.

10.4 Facial Landmark Localization

In this section we describe the preprocessing applied to EmoPain as preliminary step to pain intensity estimation. To this end, we use standard front view imagery over the whole dataset and extract PTS, LBP and DCT feature sets, see Chap. 6 for details.

Facial point tracking as proposed in [196] was applied to the video sequences acquired by front view camera 4 (see Fig. 10.1) during the annotated exercises. This yielded the tracked positions of 49 inner facial landmarks. However, due to the tracking method's applicability being dependent on frontal views, failure was likely to occur when head poses exceeded approximately 30 degrees of out-of-plane rotation. We manually removed the frames where at

Table 10.2: Distribution of the continuous pain ground-truth over different histogram bins. The first bin contains only the frames which equal intensity zero. The other bins are intervals with the size 0.1.

Intensity	# frames
0	564,042
0.0 - 0.1	66,108
0.1 - 0.2	2,009
0.2 - 0.3	166
0.3 - 0.4	155
0.4 - 0.5	164
0.5 - 0.6	140
0.6 - 0.7	189
0.7 - 0.8	486
0.8 - 0.9	39
0.9 - 1.0	0
total	633,498

least half of the point locations were wrongly assigned and thus frames with minor errors remained in the data.

10.5 Results for Automatic Pain Intensity Estimation

We have evaluated the best methods from the previous chapters on the EmoPain data for continuous pain intensity estimation: DCT+LBP RVR fusion from Chap. 7, DSRVM from Chap. 8, LT and SVR from Chap. 9. In order to better balance the pain targets, we excluded all frames with pain intensity zero and thus the frames used for evaluation are a subset of the database, which fulfills the following conditions: (1) the frame has annotated pain ground-truth, (2) the frame has annotated landmarks and (3) the pain intensity is non-zero.

By omitting zero pain scores, we focus on the more difficult but also more interesting frames. If the score is zero, then all raters must have rated zero and thus there is no discrepancy between the raters, which indicates that these frames are easy to interpret. If the score is non-zero, then either the raters disagree and only a subset has rated pain, or all raters have rated pain (potentially with different intensities). Nevertheless, the set of non-zero intensities is more important for continuous estimation, since we only exclude an infinitesimal small value from the intensity range. In contrast to that, the zero-level intensity would be important for pain detection, since it contains reliable no-pain samples.

The resulting data subset comprises 34 trials (described in Sec. 10.2.2) from 17 patients with a total of 37,059 frames. Their pain distribution is shown in Tab. 10.3. For evaluation,

we randomly sample 2,000 frames from the training subjects.

Table 10.3: Pain distribution of the EmoPain subset used for recognition experiments.

Intensity	# frames
0.0 - 0.1	35,695
0.1 - 0.2	1,105
0.2 - 0.3	112
0.3 - 0.4	48
0.4 - 0.5	14
0.5 - 0.6	40
0.6 - 0.7	36
0.7 - 0.8	9
0.8 - 0.9	0
0.9 - 1.0	0
total	37,059

The performance results are shown in Tab. 10.4. The MSE is much lower than for ShoulderPain, due to the different target scale: the range is from 0 to 1 instead of 0 to 15 for PSPI. The CORR is lower in comparison with the results on ShoulderPain. However, this can be explained by the differences between nature of the datasets: our database consists of subjects who suffer from chronic pain and thus many of their expressions are subdued due to the long time exposure. Additionally, our data contains various other facial expressions (mainly smiles and speech), which further complicate the recognition tasks. In contrast to that, the results on ShoulderPain are obtained on data which solely contains acute pain expressions in a more constraint scenario (no body movement). LT results are better than RVR fusion and DSRVM is better than LT and SVR. DSRVM provides the best results by focusing on the relevant facial parts, followed by SVR. Ideally, LT could reach similar results, but due to the relatively weak dependence of chronic pain in CLBP patients to the facial points, the LT connects the pain target node to the root. Thus pain is about equally influenced by the distribution of all facial points, but the dependence is rather weak and thus leads to lower results than DSRVM and SVR.

Table 10.4: Pain intensity estimation results on the EmoPain data for the best methods from the previous chapters. The results that are statistically different to DSRVM according a t-test with $p = 0.05$ are marked with *.

Model	Features	CORR	MSE	ICC
RVR	DCT+LBP	0.204*	0.00199*	0.146*
DSRVM	LBP	0.368	0.00141	0.252
LT	PTS	0.257*	0.00147	0.176*
SVR	PTS	0.280*	0.00149	0.225

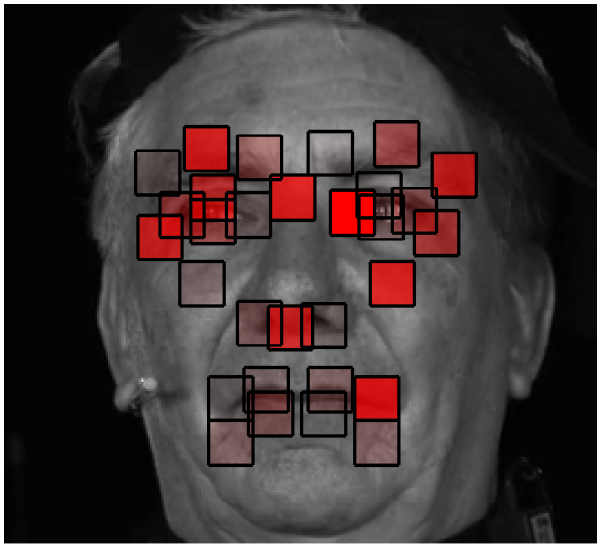


Figure 10.8: The values of kernel weights v learned by DSRVM for pain intensity estimation are indicated by the intensity of color red of the corresponding patches. Each patch corresponds to one kernel, and the larger the kernel weight the redder the patch.

Fig. 10.8 shows a visualization of the kernel weights learned by DSRVM for pain intensity estimation. The weights are mainly centered around the eyes and also one mouth corner. The weight distribution is similar as the ones learned on ShoulderPain, compare Fig. 8.6.

Fig. 10.9 depicts facial landmarks locations generated from an LT model trained on the EmoPain data. The pain face shows lowered inner brows, tightened eyelids and elevated mouth corners.

10.6 Conclusion

This chapter paves the road for the development of much needed pain sensing systems for CLBP rehabilitation. Since this research theme is little studied, we endeavoured to collect a rich dataset with specifically selected participants and sensing modalities to elicit and record naturalistic pain related behaviour based on well established behavioural psychology frameworks [89,173]. The data includes audio, video, EMG and motion capture modalities of CLBP patients while performing physical exercises. In order to provide a reliable pain labeling of facial expressions, the videos have been annotated on a continuous scale by multiple raters.

Based on these pain labels, We have evaluated several methods for continuous pain intensity estimation. These results provide a foundation for further specific investigation on the

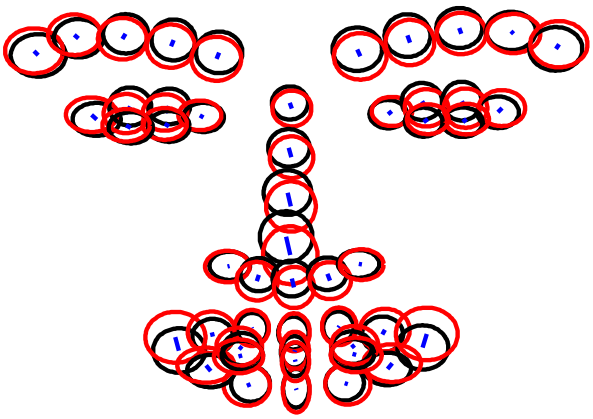


Figure 10.9: Facial landmark locations generated from an LT model trained on the EmoPain data. The face figure depicts the standard deviation of landmark locations for both the neutral face (black ellipses) and the pain face (red ellipses). Each of the ellipses is drawn with the corresponding distribution mean as center. Also, every face figure shows the difference between the mean of zero intensity and highest intensity (blue lines). If there is no difference, then only a small blue dot is visible. The ellipses cover each other, i.e., if the distribution does not change, then just a red ellipse is visible.

provided labels or for further label sets that could be generated from additional rating. The intensity estimation scores reported here are not as high as the current state of the art for the ShoulderPain dataset, part of the cause could be attributed to the unconstrained nature of the EmoPain data in terms of the various movements, presence of speaking and other non pain expressions. However, even human observers find it more difficult to recognize pain within CLBP patients (in contrast to acute pain within ShoulderPain), as can be seen from the rater discrepancy in the annotations. This confirms that the recognition of chronic pain remains a difficult task, which could be due to the insufficiency of the face as sole pain indicator or due to the low expertise level of the raters. To improve the rating accuracy, it would be beneficial to either use better trained raters (e.g. physiotherapists or other health-care professionals that have already experience with patients suffering from chronic pain) or to include different modalities (e.g., body gestures, audio or EEG).

Chapter 11

Concluding Remarks and Future Work

We have addressed estimation of continuous-valued intensities of facial expressions – the problem that has received scant attention in prior work. We did so within the regression framework. As baseline, we applied RVM regression to the target problem while using different features from the whole face. Motivated by psychological studies suggesting that it is local facial features rather than the holistic view of the face that matters for facial behavior understanding, we have formulated two regression methods – Doubly Sparse Relevance Vector Machine (DSRVM) and Latent Trees (LT). These models address the problem by weighting facial regions and thus focus on relevant information to solve the target task. LT extends the weighting capabilities to multiple target tasks and learns a tree model that describes the dependence between facial landmarks and multiple expression targets.

DSRVM generalizes RVM by jointly choosing a sparse set of relevant kernels associated with face parts, and a sparse set of relevance vectors (i.e., training data) for modeling facial expressions. This also advances related multiple-kernel learning (MKL) methods, typically specified within the max-margin framework, where enforcing joint sparsity of kernel weights and relevance vectors is difficult.

Furthermore, we formulated a novel LT model for AU intensity estimation in videos based on locations of facial landmark points in each frame. For learning LT structure, we specified an efficient algorithm that iteratively maximizes log-likelihood of training data while maintaining model complexity low. In our comparison with discriminative approaches on the benchmark datasets, LT produced superior results, especially in realistic settings of noisy detections of facial landmark points. Probabilistic sampling from LT generated meaningful facial expressions, demonstrating good generalization capabilities of LT and effectiveness of our structure learning algorithm in capturing higher-order dependencies among the high-dimensional input

features and target AU intensities.

We have also created the EmoPain dataset, which facilitates the creation of automatic affect recognition systems for chronic lower back pain rehabilitation. Since this research theme is little studied, we endeavoured to collect a multifaceted dataset with specifically selected participants and sensing modalities to elicit and record naturalistic pain related behaviour based on well established behavioural psychology frameworks [89, 173].

The recognition models have been evaluated on the challenging ShoulderPain, DISFA and EmoPain datasets. The metrics that we have used for our evaluation and comparison are the mean squared error (MSE), and Pearson correlation coefficient (CORR) and the Intra-class Correlation Coefficient (ICC). In most cases, DSRVM and LT yield better results than competing methods. In addition, DSRVM and LT can be used to provide insights in the nature of facial expressions, since it learns which face parts provide the most relevant visual cues for estimating the target facial behavior.

Our results showed that AU intensity estimation can be improved by focusing on a sparse subset of facial regions instead of the full face. Each AU has an individual discriminative region, which can be identified by our newly developed models. This finding is in line with previous work on region identification [81, 129, 205], which also reported improved results by focusing on tailored facial regions. In contrast to [81, 205], our work jointly infers the regions and target regression function, similar as [129], but with the additional flexibility of non-linear regression.

Regarding the AU targets, we found that AUs 9, 15, 17 and 20 were better recognized by appearance features and the DSRVM model, while shape features and LT were better for AUs 1, 2, 5 and 25. This indicates that additional to the weighting facial regions, optimal recognition of each AU also requires adaption of different feature types. The results of [162] also indicate varying importance between shape and appearance features per AU target. We suggest for future research to combine the weighting of facial regions for shape and appearance features, and thus automatize the feature adaption in addition to the region weighting.

The facial region related to pain intensity was mainly found to be around the eyes and mouth, which is a similar result as [167] found for pain detection. The acute pain within ShoulderPain was well recognized, while chronic pain within EmoPain posed difficulties. EmoPain had a low agreement between human raters as well, which indicates that the problem is not caused by our automatic recognition procedure, but lies within the original data: the recording setup is less restricted than for ShoulderPain, since participants can move freely and

show frequently different expressions than pain. For further research, we would recommend to improve the ground-truth annotations by recruiting pain experts as raters and apply improved pre-processing, e.g. by temporal alignment [133].

11.1 Opportunities for Future Work

The work within this thesis opens up new possibilities for further research, which includes several major directions: improving the understanding of pain in general, development of new applications and extensions of the regression models, as well as gaining further insights from the EmoPain data.

Within this work, we focused on recognizing pain from an observer perspective, specifically from observed facial expressions. A natural extension of the current work would be to broaden the focus and thus take further observations into account, like body gestures, non-verbal vocalizations or physiological measures. All of these factors are influenced by pain and thus could help to improve recognition accuracy, especially in the cases where the face alone provides not sufficient information. We observed that there are differences in the importance of facial regions for different types of pain, as indicated by the qualitative results on ShoulderPain and EmoPain. It is likely that similar differences of importance exist within other observed factors and thus the identification of most discriminative factors would be a promising direction for future research.

Going a step further by analyzing factors beyond the observer perspective could not only improve pain recognition but also contribute to the understanding of the pain experience itself. Building a model that helps to shine light on the inter-dependence between pain stimulus, intrinsic and extrinsic factors, self-report and non-verbally expressed pain would allow to identify most discriminative factors for pain and provide a more precise description of the pain process. Including this additional factors could be achieved by context-sensitive models, like [152].

From a modeling perspective, a promising improvement would be to include temporal information in the inference process. All models within this work operate only on a very small time instance, i.e., a single video frame or a window of a few frames. Each of the time instances is treated independently and thus we try to infer the intensity of facial expressions from a single time-snapshot only - without taking the past or future into account. However, intuitively, facial expressions show a characteristic development over time (e.g., we do not expect rapid jumps between pain and happiness) and thus information from the past and future would be

valuable to infer the present. Previous work confirmed this assumption by showing a recognition improvement by using temporal models for AU intensity estimation [103, 122, 152]. Thus, a temporal extension of our models that allows for facial region selection as well as temporal inference over video sequences, would be likely to improve the recognition performance.

The EmoPain data has been labeled in terms of continuous pain intensity, but further feelings like anxiety and depression are shown as well and their labeling is already planned, which could lead to the development of additional recognition methods. Furthermore, EmoPain includes multiple camera views and audio tracks that have not been explored so far and which could be used to develop more robust recognition methods that join several views or multiple modalities.

Bibliography

[1] Affectiva Inc. http://www.affectiva.com. Suite 320, 465 Waverley Oaks Road, Waltham, MA 02452, USA. 13

[2] Realeyes OU. http://www.realeyes.me. 79 Wardour Street, London W1D 6QB, UK. 13

[3] Seeing Machines Ltd. http://www.seeingmachines.com. Level 1, 11 Lonsdale St, Braddon, ACT, Australia 2612. 14

[4] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete Cosine Transform. *IEEE Trans. Comput.*, C23(1):90–93, 1974. 14, 30, 58, 61

[5] J. Alabort-i Medina, E. Antonakos, J. Booth, P. Snape, and S. Zafeiriou. Menpo: A Comprehensive Platform for Parametric Image Alignment and Visual Deformable Models. In *Proc. ACM Int. Conf. Multimed.*, MM '14, pages 679–682, New York, NY, USA, 2014. ACM. 28

[6] M. S. H. Aung, N. Bianchi-Berthouze, A. C. de C. Williams, and P. Watson. Automatic Recognition of Fear-Avoidance Behahviour in Chronic Pain Physical Rehabiliation. In *Proc. 8th Int Conf. Pervasive Comput. Technol. Healthc.*, 2014. 129

[7] M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, M. Shafizadeh, A. C. Elkins, N. Kanakam, A. de Rothschild, N. Tyler, P. J. Watson, A. C. d. C. Williams, M. Pantic, and N. Bianchi-Berthouze. The automatic detection of chronic pain-related expression: requirements, challenges and a multimodal dataset. *IEEE Trans. Affect. Comput.*, page (to appear), 2015. 43, 127, 137

[8] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Continuous Conditional Neural Fields for Structured Regression. In *Eur. Conf. Comput. Vis.*, pages 593–608. Springer, 2014. 34, 37, 56

[9] T. Bänziger and K. R. Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *Bluepr. Affect. Comput. A Sourceb.*, pages 271–294, 2010. 45

[10] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan. Automatic Recognition of Facial Actions in Spontaneous Expressions. *J. Multimed.*, 1(6):22–35, sep 2006. 34, 35

[11] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast Kernel Classifiers with Online and Active Learning. *J. Mach. Learn. Res.*, 6:1579–1619, dec 2005. 81

[12] D. Borsook, L. Becerra, and R. Hargreaves. Biomarkers for chronic pain and analgesia. Part 1: the need, reality, challenges, and solutions. *Discov Med*, 11(58):197–207, 2011. 22

[13] S. Brahnam, L. Nanni, and R. Sexton. Neonatal facial pain detection using NNSOA and LSVM. In *Int. Conf. Image Process. Comput. Vis. Pattern Recognit.*, 2008. 56

[14] S. Brahnam, C.-F. Chuang, F. Y. Shih, and M. R. Slack. Machine recognition and representation of neonatal facial displays of acute pain. *Artif. Intell. Med.*, 36(3):211–22, mar 2006. 32

[15] J. E. Brown, N. Chatterjee, J. Younger, and S. Mackey. Towards a physiology-based measure of pain: patterns of human brain activity distinguish painful from non-painful thermal stimulation. *PLoS One*, 6(9):e24124, 2011. 22

[16] M. C. Bushnell, M. Ceko, and L. A. Low. Cognitive and emotional control of pain and its disruption in chronic pain. *Nat. Neurosci. Rev.*, 14(1):502–551, 2013. 128

[17] D. Cai, X. He, W. V. Zhang, and J. Han. Regularized Locality Preserving Indexing via Spectral Regression. In *Proc. Conf. Inform. Knowl. Manag.*, pages 741–750, New York, NY, 2007. ACM. 32, 91, 93

[18] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systemes et Information, INSA de Rouen, Rouen, France, 2005. 85

[19] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. CHoG: Compressed histogram of gradients a low bit-rate feature descriptor. In *Conf. Comput. Vis. Pattern Recognit.*, pages 2504–2511. IEEE, 2009. 30

[20] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011. 91, 111

[21] M. J. Choi, V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Learning Latent Tree Graphical Models. *J. Mach. Learn. Res.*, 12:1771–1812, jul 2011. 102, 107

[22] R. Close, J. N. Wilson, and P. Gader. A Bayesian approach to localized multi-kernel learning using the relevance vector machine. In *Geosci. Remote Sens. Symp.*, pages 1103–1106, 2011. 81, 82

[23] J. F. Cohn, Z. Ambadar, and P. Ekman. Observer-based measurement of facial expression with the Facial Action Coding System. *Handb. Emot. elicitation Assess.*, pages 203–221, 2007. 19, 20

[24] J. F. Cohn and P. Ekman. Measuring facial action. *Handb. methods nonverbal Behav. Res.*, pages 9–64, 2005. 19

[25] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, jun 2001. 27

[26] T. F. Cootes and C. J. Taylor. Active shape models—'smart snakes'. In *BMVC92*, pages 266–275. Springer, 1992. 27

[27] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *BMVC92*, pages 9–18. Springer, 1992. 27

[28] D. Cosker, E. Krumhuber, and A. Hilton. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *Int. Conf. Comput. Vis.*, pages 2296–2303. IEEE, 2011. 43, 45

[29] T. Cover and P. Hart. Nearest neighbor pattern classification. *Inf. Theory, IEEE Trans.*, 13(1):21–27, 1967. 35

[30] K. D. Craig. The facial expression of pain Better than a thousand words? *APS J.*, 1(3):153–162, 1992. 10, 23

[31] K. D. Craig, K. M. Prkachin, and R. E. Grunau. The Facial Expression of Pain. In D. C. Turk and R. Melzack, editors, *Handb. Pain Assess.*, pages 117–133. 2011. 10, 89

[32] G. Crombez, C. Eccleston, S. V. Damme, J. W. S. Vlaeyen, and P. Karoly. The fear avoidance model of chronic pain: the next generation. *Clin J Pain*, 28:475–483, 2012. 128, 132

[33] T. Damoulas, Y. Ying, M. A. Girolami, and C. Campbell. Inferring Sparse Kernel Combinations and Relevance Vectors: An Application to Subcellular Localization of Proteins. In *Int. Conf. Mach. Learn. Appl.*, pages 577–582, 2008. 81, 84, 85, 97

[34] C. Darwin. *The Expression of the Emotions in Man and Animals*. Oxford University Press, London, England, 1872. 9

[35] C. P. de Campos and Q. Ji. Efficient Structure Learning of Bayesian Networks Using Constraints. *J. Mach. Learn. Res.*, 12:663–689, jul 2011. 102

[36] R. della Volpe, T. Popa, F. Ginanneschi, R. Spidalieri, R. Mazzocchio, and A. Rossi. Changes in coordination of postural control during dynamic stance in chronic low back pain patients. *Gait Posture*, 24(3):349–355, 2006. 128

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. Ser. B*, 39(1):1–38, 1977. 109

[38] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006. 52

[39] F. Dornaika and J. Orozco. Real time 3D face and facial feature tracking. *J. Real-Time Image Process.*, 2(1):35–44, 2007. 47

[40] E. Douglas-Cowie, R. Cowie, C. Cox, N. Amier, and D. K. J. Heylen. The sensitive artificial listner: an induction technique for generating emotionally coloured conversation. *Lr. Work. Corpora Res. Emot. Affect*, 2008. 43

[41] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Neural Inf. Process. Syst.*, 9:155–161, 1996. 35, 84

[42] H. K. Ekenel and R. Stiefelhagen. Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization. In *Conf. Comput. Vis. Pattern Recognit. Work.*, page 34, jun 2006. 56

[43] P. Ekman and W. V. Friesen. Facial action coding system: A technique for the measurement of facial movement, 1978. 20, 32, 90, 94, 98, 138

[44] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial action coding system.* A Human Face, Salt Lake City, UT, 2002. 9, 11, 12, 20, 22, 40

[45] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System.* Oxford Univ. Press, USA, 2005. 9, 10

[46] P. Ekman and W. V. Friesen. Nonverbal leakage and clues to deception. *Psychiatry*, 32(1):88–106, 1969. 44

[47] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *J. Pers. Soc. Psychol.*, 17(2):124, 1971. 19

[48] P. Ekman and W. V. Friesen. Felt, false, and miserable smiles. *J. Nonverbal Behav.*, 6(4):238–252, 1982. 10

[49] M. El Aroussi, A. Amine, S. Ghouzali, M. Rziza, and D. Aboutajdine. Combining DCT and LBP Feature Sets For Efficient Face Recognition. In *Int. Conf. Inf. Commun. Technol. From Theory to Appl.*, pages 1–6, apr 2008. 56

[50] A. C. Elkins, S. Zafeiriou, J. Burgoon, and M. Pantic. *Unobtrusive Deception Detection*, pages 503–515. Springer, 2014. 14

[51] B. Fasel and J. Luettin. Recognition of asymmetric facial action unit activities and intensities. In *Int. Conf. Pattern Recognit.*, volume 1, pages 1100–1103. IEEE, 2000. 34, 35

[52] I. Fasel, M. Bartlett, and J. Movellan. A comparison of Gabor filter methods for automatic detection of facial landmarks. In *Int. Conf. Autom. Face Gesture Recognit.*, pages 242–246. IEEE, 2002. 30

[53] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, Inc., 2004. 102

[54] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Anal. Mach. Intell. IEEE Trans.*, 32(9):1627–1645, 2010. 26

[55] R. A. Fisher. The statistical utilization of multiple measurements. *Ann. Eugen.*, 8(4):376–386, 1938. 31

[56] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In P. Vitányi, editor, *Comput. Learn. Theory*, volume 904, pages 23–37. Springer, 1995. 32, 36

[57] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Ann. Stat.*, 28(2):337–407, 2000. 32, 36

[58] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.*, 32(200):675–701, 1937. 52, 53, 67, 96, 97, 114, 115

[59] N. Friedman. The Bayesian Structural EM Algorithm. In *Proc. 14th Conf. Uncertain. Artif. Intell.*, pages 129–138, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 102, 103

[60] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *J. Comput. Biol.*, 9(2):331–353, 2002. 103

[61] D. Gabor. Theory of communication. *J. Inst. Electr. Eng. III Radio Commun. Eng.*, 93(26):429–441, 1946. 30

[62] R. J. Gatchel, Y. B. Peng, M. L. Peters, P. N. Fuchs, and D. C. Turk. The biopsychosocial approach to chronic pain: scientific advances and future directions. *Psychol Bull.*, 133(4):581–624, 2007. 128, 132

[63] B. Gholami, W. M. Haddad, and A. R. Tannenbaum. Agitation and pain assessment using digital imaging. In *Int'l Conf. Eng. Med. Biol. Soc.*, pages 2176–2179. IEEE, jan 2009. 32

[64] M. Girard, F. Cohn, and F. De la Torre. Estimating smile intensity : A better way. *Pattern Recognit. Lett.*, (in press), 2014. 34, 36

[65] D. S. Goldberg and S. J. McGee. Pain as a global public health priority. *BMC Public Health*, 11:770, 2011. 128

[66] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, 12:2211–2268, 2011. 15, 72

[67] D. Gonzalez-Ortega, F. J. Diaz-Pernaz, M. Martinez-Zarzuela, and M. Anton-Rodriguez. A Kinect based system for cognitive rehabilitation exercises monitoring. *Comput. Methods Programs Biomed.*, 113:620–631, 2014. 127

[68] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 29

[69] H. Gunes and B. Schuller. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image Vis. Comput.*, 31(2):120–136, 2013. 49

[70] S. D. Gunnery, J. A. Hall, and M. A. Ruben. The Deliberate Duchenne Smile: Individual Differences in Expressive Control. *J. Nonverbal Behav.*, 37(1):29–41, 2013. 10

[71] Y. Guo, G. Zhao, and M. Pietikäinen. Dynamic Facial Expression Recognition Using Longitudinal Facial Expression Atlases. In *12th Eur. Conf. Comput. Vis.*, pages 631–644. Springer, Firenze, Italy, 2012. 72

[72] M. T. Hagan, H. B. Demuth, and M. H. Beale. *Neural network design*. Pws Pub. Boston, 1996. 36

[73] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma. Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders. *J. Neurosci. Methods*, 200(2):237–256, 2011. 34, 36, 56

[74] Z. Hammal and J. F. Cohn. Automatic detection of pain intensity. In *Proc. 14th Int. Conf. Multimodal Interact.*, pages 47–52, Santa Monica, CA, USA, oct 2012. ACM. 34, 35, 40, 49, 56

[75] S. Harmeling and C. K. I. Williams. Greedy Learning of Binary Latent Trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6):1087–1097, jun 2011. 102, 103, 107, 108, 111

[76] U. Hess, R. Banse, and A. Kappas. The intensity of facial expression is determined by underlying affective state and social situation. *J. Personal. Soc. Psychol.*, 69(2):280–288, 1995. 10

[77] U. Hess, S. Blairy, and R. E. Kleck. The intensity of emotional facial expressions and decoding accuracy. *J. Nonverbal Behav.*, 21(4):241–257, 1997. 10

[78] G. Hommel. A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika*, 75(2):383–386, 1988. 52, 53, 67, 96, 97, 114, 115

[79] W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan. Driver fatigue detection based on eye tracking and dynamic template matching. In *Int. Conf. Networking, Sens. Control*, volume 1, pages 7–12. IEEE, 2004. 14

[80] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. John Wiley & Sons, 2001. 31

[81] L. A. Jeni, J. M. Girard, J. F. Cohn, and F. De La Torre. Continuous AU Intensity Estimation using Localized, Sparse Facial Feature Space. *10th Int. Conf. Autom. Face Gesture Recognit.*, 2013. 34, 35, 36, 146

[82] L. A. Jeni, A. Lőrincz, T. Nagy, Z. Palotai, J. Sebők, Z. Szabó, and D. Takács. 3D shape estimation in video sequences provides high precision evaluation of facial expressions. *Image Vis. Comput.*, 30(10):785–795, 2012. 34, 35, 56

[83] M. P. Jensen and P. Karoly. Self-report scales and procedures for assessing pain in adults. In *Handb. Pain Assess.*, pages 19–41. 1992. 139

[84] B. Jiang, M. F. Valstar, B. Martinez, and M. Pantic. Dynamic Appearance Descriptor Approach to Facial Actions Temporal Modelling. *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 44(2):161–174, 2014. 58

[85] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. 31

[86] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous Pain Intensity Estimation from Facial Expressions. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, editors, *Adv. Vis. Comput.*, volume 7432 of *Lecture Notes in Computer Science*, pages 368–377, Heidelberg, jul 2012. Springer. 16, 34, 72, 90, 100, 122

[87] S. Kaltwang, S. Todorovic, and M. Pantic. Latent Trees for Estimating Intensity of Facial Action Units. In *IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2015. 34

[88] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Fourth IEEE Int. Conf. Autom. Face Gesture Recognit.*, pages 46–53. IEEE, 2000. 43, 45

[89] F. J. Keefe and A. R. Block. Development of an observation method for assessing pain behaviour in chronic low back pain patients. *Behav. Ther.*, 13:4, 1982. 128, 143, 146

[90] M. Kim and V. Pavlovic. Structured output ordinal regression for dynamic facial emotion intensity prediction. In *Eur. Conf. Comput. Vis.*, pages 649–662. Springer, 2010. 37

[91] R. Kindermann and J. L. Snell. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI, 1980. 36

[92] S. Koelstra, M. Pantic, and I. Patras. A Dynamic Texture-Based Approach to Recognition of Facial Actions and Their Temporal Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(11):1940–1954, nov 2010. 72

[93] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 102, 104

[94] M. V. Korff, J. Gruman, J. Schaefer, S. Curry, and E. H. Wagner. Collaborative management of chronic illness. *Ann. Intern. Med.*, 127:1097–1102, 1997. 128

[95] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th Int. Conf. Mach. Learn.*, pages 282–289. Citeseer, 2001. 37

[96] N. D. Lawrence. Mocap toolbox for matlab. *Available on-line at http://www.cs.man.ac.uk/neill/mocap*, 2011. 136

[97] P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Houghton Mifflin, 1968. 109

[98] Y. Lecrubier, D. V. Sheehan, E. Weiller, P. Amorim, I. Bonora, K. H. Sheehan, J. Janavs, and G. C. Dunbar. The Mini Neuropsychiatric Interview (MINI). A short diagnostic structured interview: reliability and validity according to the CIDI. *Eur. Psychiatry*, 12(5):224–231, 1997. 130

[99] D. D. Lee, H. S. Seung, and Others. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, oct 1999. 31

[100] M. Leeuw, M. E. J. B. Goossens, S. J. Linton, G. Crombez, K. Boersma, and J. W. S. Vlaeyen. The fear-avoidance model of musculoskeletal pain: current state of scientific evidence. *J Behav. Med.*, 30(1):77–94, 2007. 128

[101] S. Z. Li. *Markov random field modeling in computer vision*. Springer Science & Business Media, 2012. 36

[102] Y. Li, S. M. Mavadati, M. H. Mahoor, and Q. Ji. A unified probabilistic framework for measuring the intensity of spontaneous facial action units. In *IEEE Int.Conf. Autom. Face Gesture Recognit.*, pages 1–7, apr 2013. 34, 35, 37, 102

[103] Y. Li, S. M. Mavadati, M. H. Mahoor, Y. Zhao, and Q. Ji. Measuring the Intensity of Spontaneous Facial Action Units with Dynamic Bayesian Network. *Pattern Recognit.*, 48(11):3417—-3427, 2015. 148

[104] J. Lichtenauer, J. Shen, M. Valstar, and M. Pantic. Cost-effective solution to synchronised audio-visual data capture using multiple sensors. *Image Vis. Comput.*, 29(10):666–680, sep 2011. 137

[105] G. C. Littlewort, M. S. Bartlett, and K. Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image Vis. Comput.*, 27(12):1797–1803, 2009. 32, 72

[106] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, nov 2004. 30

[107] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pages 94–101, jun 2010. 21, 43, 45

[108] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews. Painful monitoring: Automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database. *Image Vis. Comput.*, 30(3):197–205, 2012. 32, 61, 72

[109] P. Lucey, J. Cohn, K. Prkachin, P. Solomon, and I. Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *Int. Conf. Autom. Face Gesture Recognit.*, pages 57–64. IEEE, 2011. 22, 32, 39, 43, 57, 109, 111, 137, 138

[110] P. Lucey, J. Howlett, J. Cohn, S. Lucey, S. Sridharan, and Z. Ambadar. Improving Pain Recognition Through Better Utilisation of Temporal Information. In *Proc. Int. Conf. Audit. Speech Process.*, number September, pages 167–172, 2008. 56

[111] K. M. and Prkachin. The consistency of facial expressions of pain: a comparison across modalities. *Pain*, 51(3):297–306, 1992. 23

[112] M. H. Mahoor, S. Cadavid, D. S. Messinger, and J. F. Cohn. A framework for automated measurement of the intensity of non-posed Facial Action Units. In *Comput. Vis. Pattern Recognit. Work.*, pages 74–80, 2009. 34, 35, 56

[113] C. J. Main and P. J. Watson. Guarded movements: development of chronicity. *J. Musculoskelet. Pain*, 4(4):163–170, 1996. 132

[114] L. Marquié, E. Raufaste, D. Lauque, C. Mariné, M. Ecoiffier, and P. Sorum. Pain rating by patients and physicians: evidence of systematic pain miscalibration. *Pain*, 102(3):289–96, apr 2003. 13

[115] M. O. Martel, T. H. Wideman, and M. J. L. Sullivan. Patients who display protective pain behaviors are viewed as less likable, less dependable, and less likely to return to work. *Pain*, 153(4):843–849, 2012. 128

[116] C. Massot and J. Hérault. Model of frequency analysis in the visual cortex and the shape from texture problem. *Int. J. Comput. Vis.*, 76(2):165–182, 2008. 30

[117] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *Eur. Conf. Comput. Vis.*, pages 720–735. Springer, 2014. 27

[118] D. Matsumoto and B. Willingham. Spontaneous facial expressions of emotion of congenitally and noncongenitally blind individuals. *J. Personal. Soc. Psychol.*, 96(1):1–10, 2009. 10, 58

[119] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Comput. Vis.*, 60(2):135–164, 2004. 27

[120] B. J. Matuszewski, W. Quan, L.-K. Shark, A. S. McLoughlin, C. E. Lightbody, H. C. A. Emsley, and C. L. Watkins. Hi4D-ADSIP 3-D dynamic facial articulation database. *Image Vis. Comput.*, 30(10):713–727, 2012. 42, 43

[121] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. DISFA: A Spontaneous Facial Action Intensity Database. *IEEE Trans. Affect. Comput.*, 4(2):151–160, 2013. 16, 34, 35, 37, 41, 43, 49, 50, 56, 91, 93, 100, 102, 109, 111, 122

[122] S. M. Mavadati and M. H. Mahoor. Temporal Facial Expression Modeling for Automated Action Unit Intensity Measurement. In *Int. Conf. Pattern Recognit.*, pages 4648–4653. IEEE, 2014. 34, 37, 50, 148

[123] D. McDuff, R. El Kaliouby, T. Senechal, M. Amr, J. F. Cohn, and R. Picard. Affectiva-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected" In-the-Wild". In *Conf. Comput. Vis. Pattern Recognit. Work.*, pages 881–888. IEEE, jun 2013. 43, 44

[124] D. McDuff, R. El Kaliouby, T. Senechal, D. Demirdjian, and R. Picard. Automatic measurement of ad preferences from facial responses gathered over the Internet. *Image Vis. Comput.*, 32(10):630–640, 2014. 13

[125] K. O. McGraw and S. P. Wong. Forming inferences about some intraclass correlation coefficients. *Psychol. Methods*, 1(1):30–46, 1996. 50, 51, 139

[126] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder. The SEMAINE Database: Annotated Multimodal Records of Emotionally Colored Conversations between a Person and a Limited Agent. *IEEE Trans. Affect. Comput.*, 3(1):5–17, 2012. 43, 46

[127] A. Metallinou, A. Katsamanis, and S. Narayanan. Tracking continuous emotional trends of participants during affective dyadic interactions using body language and speech information. *Image Vis. Comput.*, 31(2):137–152, 2013. 32

[128] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Anal. Mach. Intell. IEEE Trans.*, 27(10):1615–1630, 2005. 30

[129] M. R. Mohammadi, E. Fatemizadeh, and M. H. Mahoor. Intensity Estimation of Spontaneous Facial Action Units Based on Their Sparsity Properties. *IEEE Trans Sys., Man Cybern. Part B*, PP(99):1–10, 2015. 146

[130] V. Molony and J. E. Kent. Assessment of Acute Pain in Farm Animals Using Behavioral and Physiological Measurements. *J. Anim. Sci.*, 75:266–272, 1997. 22

[131] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, University of California, Berkeley, 2002. 37

[132] L. Nanni, S. Brahnam, and A. Lumini. A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states. *Expert Syst. Appl.*, 37(12):7888–7894, 2010. 72

[133] M. Nicolaou, V. Pavlovic, and M. Pantic. Dynamic Probabilistic CCA for Analysis of Affective Behaviour and Fusion of Continuous Annotations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1299–1311, 2014. 139, 147

[134] M. A. Nicolaou, H. Gunes, and M. Pantic. Output-associative rvm regression for dimensional and continuous emotion prediction. *Image Vis. Comput.*, 30(3):186–196, 2012. 95, 96

[135] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002. 14, 30, 57, 61

[136] J. Orozco, B. Martinez, and M. Pantic. Empirical analysis of cascade deformable models for multi-view face detection. In *IEEE Proc. Int'l Conf. Image Process.*, 2013. 27

[137] A. Ortony and T. J. Turner. What's basic about basic emotions? *Psychol. Rev.*, 97(3):315–331, 1990. 11, 58, 71

[138] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *IEEE Int. Conf. Multimed. Expo*, pages 317—-321. IEEE, 2005. 43, 46

[139] M. Papastergiou. Exploring the potential of computer and video games for health and physical education: A literature review. *Comput. Educ.*, 53(3):603–622, 2009. 127

[140] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rogers. A review of wearable senors and systems with application in rehabilitation. *J. Neuroengineering Rehab.*, 9:12, 2012. 127, 128

[141] I. Patras and M. Pantic. Particle filtering with factorized likelihoods for tracking facial features. In *Int. Conf. Autom. Face Gesture Recognit.*, pages 97–102. IEEE, 2004. 28

[142] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Anal. Mach. Intell. IEEE Trans.*, 27(8):1226–1238, 2005. 32

[143] K. Prkachin and K. Craig. Expressing pain: The communication and interpretation of facial pain signals. *J. Nonverbal Behav.*, 19(4):191–205, 1995. 22

[144] K. Prkachin and P. Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008. 23, 40, 138

[145] S. Qiu and T. Lane. Multiple Kernel Learning for Support Vector Regression. Technical report, Univ. New Mexico, 2005. 15, 72

[146] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989. 37

[147] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *J. Mach. Learn. Res.*, 9:2491–2521, 2008. 15, 49, 72, 81, 84, 85

[148] T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *Pattern Anal. Mach. Intell. IEEE Trans.*, 21(4):291–310, 1999. 30

[149] P. Reicherts, A. B. M. Gerdes, P. Pauli, and M. J. Wieser. On the mutual effects of pain and emotion. *Pain*, 154(6):793–800, 2013. 90

[150] R. Rosenthal. Conducting judgment studies. In K. Scherer and R. Ekman, editors, *Hand- B. methods nonverbal Behav. Res.*, pages 287–361. Cambridge University Press, New York, 1982. 19, 22

[151] O. Rudovic, V. Pavlovic, and M. Pantic. Automatic Pain Intensity Estimation using Heteroscedastic Conditional Ordinal Random Fields. In *Int. Symp. Vis. Comput.*, Crete, Greece, 2013. 34, 37, 40, 56

[152] O. Rudovic, V. Pavlovic, and M. Pantic. Context-sensitive Dynamic Ordinal Regression for Intensity Estimation of Facial Action Units. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):944–958, 2015. 16, 34, 37, 40, 50, 56, 100, 122, 147, 148

[153] O. Rudovic and M. Pantic. Shape-constrained Gaussian Process Regression for Facial-point-based Head-pose Normalization. In *Int'l Conf. Comput. Vis.*, pages 1495 – 1502. IEEE, 2011. 68

[154] J. A. Russell. Core affect and the psychological construction of emotion. *Psychol. Rev.*, 110(1):145–172, 2003. 58

[155] J. Russell. A circumplex model of affect. *J. Personal. Soc. Psychol.*, 39(6):1161–1178, 1980. 47

[156] M. V. Saarela, Y. Hlushchuk, A. C. d. C. Williams, M. Schürmann, E. Kalso, and R. Hari. The compassionate brain: humans detect intensity of pain from another's face. *Cereb. cortex*, 17(1):230–237, 2007. 22

[157] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987. 102

[158] G. Sandbach, S. Zafeiriou, and M. Pantic. Markov Random Field Structures for Facial Action Unit Intensity Estimation. In *Int. Conf. Comput. Vis. Work.*, pages 738–745. IEEE, dec 2013. 16, 34, 35, 36, 50, 56, 100, 122

[159] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert. Recognition of 3D facial expression dynamics. *Image Vis. Comput.*, 30(10):762–773, 2012. 72

[160] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. Comput. Vis.*, 91(2):200–215, sep 2011. 28

[161] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *COST Work. Biometrics Identity Manag.*, pages 47–56. Springer, 2008. 43, 46

[162] A. Savran, B. Sankur, and M. T. Bilge. Regression-based intensity estimation of facial action units. *Image Vis. Comput.*, 30(10):774–784, 2012. 34, 35, 36, 146

[163] K. R. Scherer and H. Ellgring. Are facial expressions of emotion produced by categorical affect programs or dynamically driven by appraisal? *Emotion*, 7(1):113–130, 2007. 11, 71

[164] B. Schuller, M. Valster, F. Eyben, R. Cowie, and M. Pantic. AVEC 2012: the continuous audio/visual emotion challenge. In *Proc. 14th Int. Conf. Multimodal Interact.*, pages 449–456, Santa Monica, CA, USA, 2012. ACM. 95, 96

[165] K. Shiarlis, J. Messias, M. van Someren, S. Whiteson, J. Kim, J. Vroon, G. Englebienne, K. Truong, V. Evers, N. Pérez-Higueras, Ignacio Perez-Hurtado, R. Ramon-Vigo, F. Caballero, L. Merino, J. Shen, S. Petridis, M. Pantic, L. Hedman, M. Scherlund,

R. Koster, and M. Herve. TERESA: A Socially Intelligent Semi-autonomous Telepresence System. *ICRA-2015, Work. Mach. Learn. Soc. Robot.*, 2015. 14

[166] P. E. Shrout and J. L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychol. Bull.*, 86(2):420–428, 1979. 49, 50

[167] K. Sikka, A. Dhall, and M. S. Bartlett. Classification and weakly supervised pain localization using multiple segment representation. *Image Vis. Comput.*, 32(10):659—-670, 2014. 146

[168] A. Singh, A. Klapper, J. Jia, A. Fidalgo, A. T. Jimenez, N. Kanakam, N. Bianchi-Berthouze, and A. Williams. Motivating People with Chronic Pain to do Physical Activity: Opportunities for Technology Design. In *Proc. SIGCHI Conf. Hum. factors Comput. Syst.*, pages 2803–2812, 2014. 13, 128, 131

[169] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Stat. Comput.*, 14(3):199–222, 2004. 73, 85

[170] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *J. Mach. Learn. Res.*, 7:1531–1565, 2006. 15, 72, 84, 85, 86

[171] G. Stratou, A. Ghosh, P. Debevec, and L. P. Morency. Effect of illumination on automatic expression recognition: a novel 3D relightable facial database. In *Int. Conf. Autom. Face Gesture Recognit. Work.*, pages 611–618. IEEE, 2011. 43, 46

[172] M. J. L. Sullivan, S. R. Bishop, and J. Pivik. The Pain Catastrophizing Scale: Development and validation. *Psychol. Assess.*, 7:524–532, 1995. 131

[173] M. J. L. Sullivan, P. Thibault, A. Savard, R. Catchlove, J. Kozey, and W. D. Stanish. The influence of communication goals and physical demands on different dimensions of pain behaviour. *Pain*, 125:270–277, 2006. 143, 146

[174] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001. 14, 49, 62, 63, 72, 74, 76, 79, 84, 85

[175] M. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In C. Bishop and B. J. Frey, editors, *Proc. 9th Int. Work. Artif. Intell. Stat.*, pages 1–13, Key West, FL, 2003. 76, 79

[176] Y. Tong, J. Chen, and Q. Ji. A unified probabilistic framework for spontaneous facial action modeling and understanding. *Pattern Anal. Mach. Intell. IEEE Trans.*, 32(2):258–273, 2010. 43, 45

[177] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1683–1699, 2007. 43, 45, 46, 100

[178] I. Tracey and M. C. Bushnell. How neuroimaging studies have challenged us to rethink: is chronic pain a disease? *J Pain*, 10(11):1113–1120, 2009. 128

[179] D. C. Turk and A. Okifuji. Psychological factors in chronic pain: evolution and revolution. *J Consult Clin Psychol.*, 70(3):678–690, 2002. 129

[180] D. C. Turk and T. E. Rudy. IASP taxonomy of chronic pain syndromes: preliminary assessment of reliability. *Pain*, 30(2):177–189, 1987. 128

[181] D. C. Turk and R. Melzack. The measurement of pain and the assessment of people experiencing pain. *Handb. Pain Assess.*, pages 3–18, 2011. 10

[182] D. Tzikas, A. Likas, and N. Galatsanos. Large scale multikernel RVM for object detection. In *Adv. Artif. Intell.*, pages 389–399. Springer, 2006. 81, 82

[183] G. Tzimiropoulos and M. Pantic. Gauss-Newton Deformable Part Models for Face Alignment in-the-Wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1851–1858. IEEE, 2014. 10

[184] M. F. Valstar and M. Pantic. Fully Automatic Recognition of the Temporal Phases of Facial Actions. *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 42(1):28–43, 2012. 28, 72

[185] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In *Proc. Int'l Conf. Lang. Resour. Eval. W'shop Emot.*, pages 65–70, 2010. 43, 46

[186] M. F. Valstar, T. Almaev, J. M. Girard, G. Mckeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. FERA 2015 - Second Facial Expression Recognition and Analysis Challenge. In *IEEE Conf. Autom. Face Gesture Recognit. Work.* IEEE, 2015. 34, 37, 56

[187] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer. Meta-analysis of the first facial expression recognition challenge. *Syst. Man, Cybern. Part B Cybern. IEEE Trans.*, 42(4):966–979, 2012. 43, 45

[188] L. Van der Maaten, E. Postma, and H. Van Den Herik. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.*, 10(February):1–41, 2009. 32

[189] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995. 35

[190] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, 57(2):137–154, 2004. 26

[191] J. W. S. Vlaeyen and S. J. Linton. Fear-avoidance and its consequences in chronic musculoskeletal pain: a state of the art. *Pain*, 85:3, 2000. 128, 132

[192] M. Von Korff. Assessment of chronic pain in epidemiological and health services research: Empirical bases and new directions. *Handb. Pain Assess.*, pages 455–473, 2011. 10

[193] G. K. Wallace. The JPEG still picture compression standard. *Commun. ACM*, 34(4):30–44, apr 1991. 59

[194] A. C. d. C. Williams. Facial expression of pain: An evolutionary account. *Behav. Brain Sci.*, 25(04):439–455, 2002. 10, 11, 20, 22, 23, 90, 116

[195] A. C. Williams, C. Eccleston, and S. Morley. Psychological therapies for the management of chronic pain (excluding headache) in adults. *Cochrane Database Syst. Rev.*, 11, 2012. 128, 130, 132

[196] X. Xiong and F. De La Torre. Supervised descent method and its applications to face alignment. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 532–539, 2013. 28, 140

[197] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. *PLoS One*, 9(1):e86041, 2014. 43, 44

[198] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu. Casme database: a dataset of spontaneous micro-expressions collected from neutralized faces. In *Int. Conf. Work. Autom. Face Gesture Recognit.* IEEE, 2013. 43, 44

[199] Z. Zafar and N. A. Khan. Pain Intensity Evaluation through Facial Action Units. In *Int. Conf. Pattern Recognit.*, pages 4696–4701. IEEE, 2014. 34, 35, 56

[200] S. Zafeiriou and I. Pitas. Discriminant Graph Structures for Facial Expression Recognition. *IEEE Trans. Multimed.*, 10(8):1528–1540, 2008. 72

[201] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE Trans. Patt. Anal. Mach. Intel.*, 31(1):39–58, 2009. 9, 11, 71, 72

[202] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3D dynamic facial expression database. In *IEEE Int. Conf. Work. Autom. Face Gesture Recognit.*, pages 1–6, apr 2013. 43, 44

[203] Y. Zhang, L. Zhang, and M. A. Hossain. Adaptive 3D facial action intensity estimation and emotion recognition. *Expert Syst. Appl.*, 42(3):1446–1464, 2015. 34, 35, 36, 56

[204] G. Zhao and M. Pietikainen. Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):915–928, 2007. 58

[205] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning active facial patches for expression analysis. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2562–2569, 2012. 72, 146

[206] J. Zhu and E. P. Xing. On Primal and Dual Sparsity of Markov Networks. In *Int. Conf. Mach. Learn.*, pages 1265–1272, Montreal, QC, Canada, 2009. 15

[207] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Conf. Comput. Vis. Pattern Recognit.*, pages 2879–2886. IEEE, 2012. 26

[208] A. S. Zigmond and R. P. Snaith. The Hospital Anxiety and Depression Scale. *Acta Psychiatr. Scand.*, 67:361–370, 1983. 131