Utrecht University

Individualized Smile Detection Using Transfer Learning

Inge Becht
ICA-4157281

Supervisor
dr. ir. R.W. (Ronald) Poppe

Second Examiner
prof. dr. R.C. (Remco) Veltkamp

Game & Media Technology
Utrecht University
January 13, 2017

ABSTRACT

In current machine learning approaches to visual smile detection the training data generally consists of a variety of different faces displaying a smile or neutral expression, and the target data introduces a new face to classify. Although this approach has shown promising results towards correctly classifying the newly introduced target face, we believe that performance can be improved by acknowledging that there are differences in smile-styles and facial differences throughout the training process. In this research we aim to do so by applying transfer learning. This entails first training a classifier on a data set showing a wide variety of smiling and non-smiling faces, as is the general approach. Afterwards we apply transfer learning, by means of an Adaptive SVM, where a small number of labeled instances from the target face is provided as to make the classifier more specific to the target face. To make transfer learning effective we use low-level geometric features which we expect to capture the difference between smiles and no-smile in a variety of different faces and smile styles.

We evaluate our approach by comparing performance against alternative strategies. We find that using a traditional classifier trained on an aggregated data set containing the general and target data outperforms our baseline and our suggested transfer learning approach for each of the test videos. A big problem for the transfer learning approach seems to be the quality of the labeled data of the target face used during training. The aggregated approach seems less effected by it, making it a preferred approach for real-life applications where labeled target data will be sparse and difficult to select. However, its performance improvements does not outweigh its efforts in current experiments, and further research should focus on the choice of target data to use during the training process.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Ronald Poppe for his guidance and support throughout my thesis. Additionally I would like to thank Daniel Mcduff for providing me access to the AM-FED data set and answering my questions, and to the Chehra Team for providing me with the Chehra tracking code.

CONTENTS

1

INTRODUCTION

1.1 SMILE DETECTION

Smile detection is part of the research field of affect recognition, in which a person's facial features, pose, and speech data are studied to determine what underlying emotions the person is experiencing. In this context, emotions are mostly treated as a discrete category, where a person can be experiencing one of few distinct emotions, for example, sadness, happiness, angry, etc. This idea of discrete emotion categories stems from psychological research in the field of emotion expression.

In the typical machine learning approach to affect recognition, a classifier is trained on a training set of facial features (extracted from a single picture or over a couple of video frames) from different faces showing a variety of pre-determined prototypical emotion categories. Then, the classifier is applied on test data, often consisting of feature data of a face not used during training, and this outputs the most likely emotion(s) that the face is displaying. In early affect recognition studies, the training data consisted of emotions acted out by actors in a controlled setting. An example of such a data set, that has been widely used in different researches, is the Cohn-Kanade data set, where the labels consists out of 6 basic emotions: happiness, surprise, sadness, disgust, anger, and fear[20].

Although lots of studies reported high accuracy when detecting these different types of emotions using such highly controlled data sets[22][4][42], the contribution to real world scenarios is rather limiting, as emotions are often more subtle to detect, and don't always belong to one of the basic categories in particular. What is more, the interpretation of a facial expression can mean different things depending on its context. In recent years a shift towards more practical research can be witnessed, with studies focusing now on spontaneous emotion data, specialising research into detecting specific emotions (e.g. smile detection or pain detection[23]) and abandoning the holistic emotion categories by recognizing emotions inside a specific context (e.g. spotting the difference in genuine or posed emotions[14][43], determining the meaning of a smile[18][33]).

1.2 GOAL

In this thesis, we aim to improve upon the existing body of research on spontaneous smile detection, by acknowledging the difference in smile-styles and facial differences throughout the training process. Although most smile detection approaches use spontaneous facial data during the training process, these approaches often assume that a single generic decision model is sufficient to distinguish between smiles in a variety of different faces. A single generic model means a single classifier is fed training data from a wide variety of faces showing smiles and non-smiles, with disregard of how these faces might differ. Figure 1 depicts the diversity that can occur in such smile data; the smiling faces all differ in mouth area, mouth shape, the direction of the mouth corners in comparison to the center of the

upper and lower lip, eye area, etc. but in all cases it is apparent that the face depicts a smile. These differences seem to stem from difference in smile-styles (e.g. clenching the lower jaw, smirking), but similar smile-styles also look different on different faces due to geometrical facial differences.

Figure 1.: Diversity in smiles of different people (taken from the GENKI data set).

We argue that disregarding this diversity during the training process will cause suboptimal results. To this end, our objective in this research is *to improve the detection rate for smiles over a big variety of faces by explicitly focusing on the aforementioned geometric differences seen in spontaneous smiles*. We approach this by constructing a person-specific classification, that uses information about the person to be classified. We coin the term *individualized smile detection* for our research effort, as it conveys the approach of our proposed solution to smile detection.

As a side note; although recent research on spontaneous smile detection has combined this challenge with improving performance on "in the wild" difficulties (like illumination, occlusion, and diversity in appearance like facial hair and glasses), overcoming these challenges is not the aim of this research. We assume that we deal with perfect geometrically acquired data from our data sets and prune data to acquire this, if necessary.

The importance of researching individualized smile detection is apparent when looking at the use cases of affect recognition in current research applications. The main application fields are the improvement of Human-Computer/Robot Interaction(HCI/HRI), and for assisted diagnosing and self help applications in the field of psychiatry[42]. Studies integrating visual affect recognition of the former category have applied it to detect student engagement in a computer-aided classroom[5], to create emotional aware virtual conversation partners[32][17] and to create a affect-sensitive robotic game companion[7]. An example of the latter, is diagnosing depression by recognizing subtle changes in facial expression[26].

Such applications would benefit from individualized affect recognition, as in all cases these applications deal with a wide variety of users, and should perform well for each of them. Often these use cases do not easily allow for the user to inform the system that the detected emotion is wrong, and even if such a feedback loop is in place, repeating mistakes can lead to frustration on the user's part or even distrust in the system, ceasing interaction

altogether. In the assisted diagnosis and self-help tool case, it is even more crucial for the system to be an accurate aid without outside interference, or its reliability can be taken into questioning.

1.3 METHOD

There are several ways to go about making a more individualized approach towards smile detection. A naive solution consists of training a single classifier for each new user by using their face data exclusively in the training process. This is, however, not a feasible approach: it requires copious amounts of data that show the diversity of face configurations possible during smiles and non-smiles for a single person, which is not easily attainable for the real life use cases mentioned in the previous section. Moreover, such an approach would not be able to exploit the fact that smiles from different people do contain similarities up to a point. Previous work has shown that there are some fundamental elements present in what makes a smile, for instance the raising of the cheeks[15], so we could still benefit from a large, generic data set with versatile smiling data.

Hence, we take an alternative approach: First, we train a single model on a pre-existing data set of smiling and non-smiling faces, as is the general approach for most research studies creating emotion detection methods. Afterwards, a technique called transfer learning is applied to adapt this general model to novel persons in the test data. Transfer learning is a class of algorithms that is used for adapting existing models to a new task that is in some sense similar to the original task (by for example having the same feature space), but differs in the distribution of feature values. This essentially is the case for our objective, and comes with several benefits. Firstly, it draws knowledge from existing information about what comprises a smile. Secondly, transfer learning can reduce training times, as the retraining process does not involve an enormous amount of data.

In this thesis, we will explore the appropriateness of transfer learning as a solution for individualized smile detection. To this end, we take a practical approach throughout the research; we will create our own generic smile classifier, finding features we deem necessary for the purpose of the research, and then apply some form of transfer learning on the acquired classifier. To guide the research and to clarify how we intent to evaluate performance, we have proposed fundamental research questions which will be clarified in the following section.

1.4 RESEARCH QUESTIONS

The first research question is an all encapsulating question that summarizes the goal and method of our research:

Research Question. *Can transfer learning improve the detection rate of spontaneous smiles when applied on a generically trained classifier trained with geometric features?*

To thoroughly answer this research question, we distinguish between two different sub questions to be answered separately. Our first sub question reflects the ambition to resolve what features work best for our research goals:

Sub-Question 1. *What geometric features prove effective in personalized smile detection?*

There are a wide variety of geometrical features applicable to detecting if a face is smiling or not. Previous work has shown approaches using distances between facial landmarks (e.g.

inner eye points, mouth corners) and temporal features in video data, by extracting such geometric features over the course of a few consecutive frames. The geometric features that we will consider should be able to capture differences not only between the smile/no-smile binary, but should be sensitive enough to capture the differences in smiling faces. For our approach of transfer learning, this means that the features we use must have a measurable difference in value distribution. For this reason, our choice of features will divert slightly from earlier research, as this was not a criteria necessary for these earlier researches.

The second research question reflects the desire to evaluate the choice of transfer learning for individualized smile detection:

Sub-Question 2. *How does transfer learning compare to alternative approaches of individualized smile detection?*

To determine the effectiveness of transfer learning, its performance has to be evaluated against a baseline. This baseline will be a generic classifier that does not use the added step of transfer learning, and performance evaluation consists of comparing scores over a series of different faces. This approach however, does not give insight in how well transfer learning performs in comparison to other individualization methods. For this reason we use an evaluation structure as is used in [40] where their transfer learning approach is pitted against alternative techniques.

1.5 CONTRIBUTION

The contribution of this thesis is to provide insight into what improvements the addition of transfer learning can make on regular machine learning in the field of affect recognition. The implementation used throughout this research is not to be considered a state-of-the-art approach, but it should rather be valued for the insight it gives towards answering the posed research questions. As such, the answers to the sub questions are limited in scope; sub-question 1 will not be exhaustively tested, but on a limited amount of feature sets which we expect will perform decently for our objective. Similarly, sub-question 2 is only answered specifically for the transfer learning approach that we decide to implement for our problem. We do not guarantee using an optimal feature set or classifier, but try to set the first steps in the right direction for achieving individualized smile detection.

1.6 OVERVIEW

This thesis is structured as follows. First, Chapter 2 explores the existing body of work in the fields of smile detection and transfer learning. Chapter 3 gives an overview of the data used throughout this research for training and evaluation purposes and explains the rationale behind choosing these data sets. Chapter 4 describes the approach taken towards answering the research questions. Chapter 5 describes the evaluation conditions, depicts the resulting outcome of the performed tests and discusses the results by relating them to the proposed research question. Finally, Chapter 6 concludes our research.

2

BACKGROUND

2.1 VISUAL SMILE DETECTION

The general approach to visual smile detection is applying some form of machine learning that is able to distinguish between the different classes, usually smile or no-smile. The quality of a trained classifier depends on the selection of training data, features and the type of classifier chosen. Figure 2 shows the general pipeline for existing research in smile detection. In this section we will discuss the specific steps of this pipeline, for instance the different feature choices, the choice of classifiers, the detection goal and how successful these approaches are.

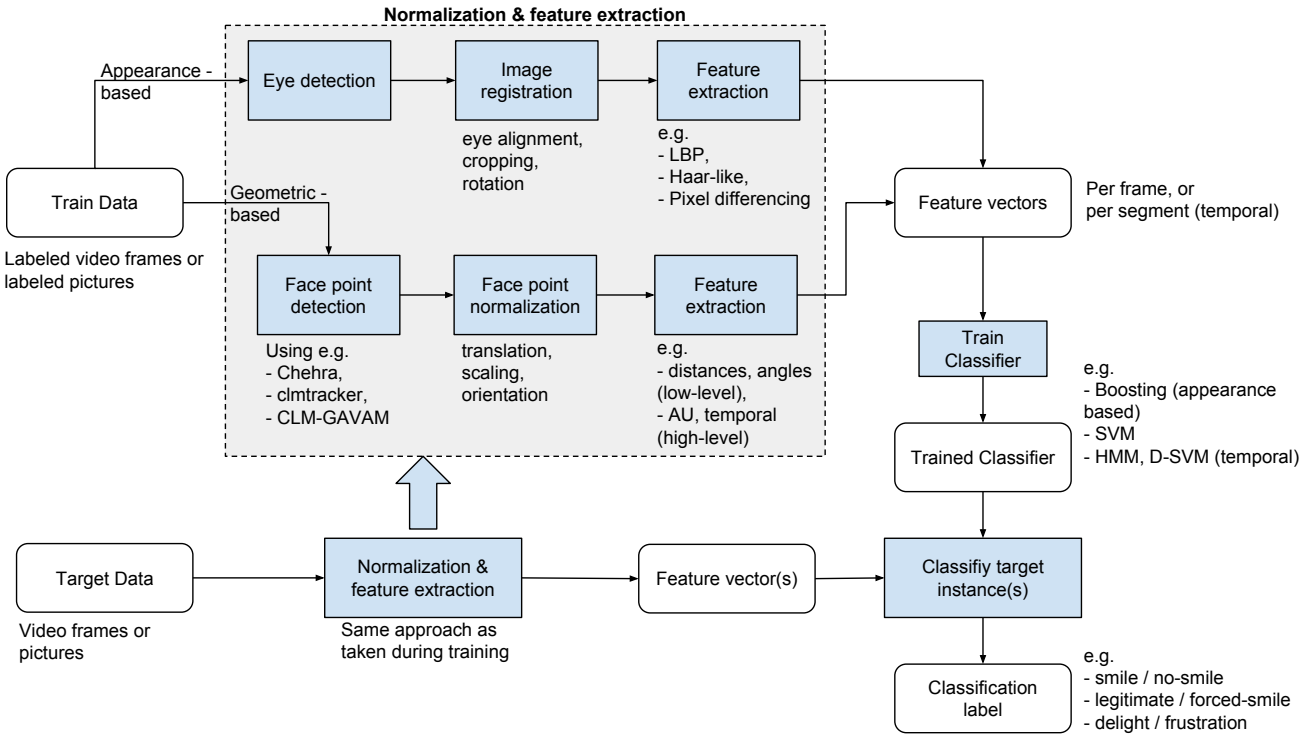


Figure 2.: The general pipeline of training and testing for visual smile detection.

2.1.1 *Visual Features*

Visual features used for smile detection can basically be divided in two different categories: *appearance*-based and *geometric*-based.

2.1.1.1 *Appearance Based Features*

Appearance based features describe texture information of an image. In case of facial data, these features encode textural data like wrinkles, bulges and furrows[42]. The following appearance based approaches have been used specifically in smile detection research:

Local Binary Patterns(LBP)[25][38]: Provide a way of encoding textural information by expressing the difference in gray-scale value between a center pixel and its neighbouring pixels along a radius and sampling interval chosen by the user. It encodes this difference with the use of a single binary digit; writing down a 1 if a neighbours gray-scale value is higher than that of the center pixel, and otherwise a 0. By encoding multiple pixels in a region in this way, a histogram can be constructed, which becomes an LBP feature.[27]

Gabor Energy Filters[38]: A filter for edge detection that models the cells of a primate's visual cortex. The filter consists of a combination of a Gaussian filter and a sinusoid function with configurable orientation, variance, and frequency. When applying Gabor Energy Filters for machine learning, a bank of filters has to be created with different frequencies and orientation, so that edges in each direction and with correct sensitivity can be successfully detected throughout the image.

Haar-like features[38]: Takes the sum of differences between adjacent, same-sized rectangular regions in a gray-scale image as a single feature. Different types of rectangular configurations are possible, which are able to detect light-to-dark transition boundaries, in different directions of the image (horizontal, diagonal, vertical). Haar-like features are known for their application in face detection[37], where the most important rectangular features use intensity difference between eye and cheek region and intensity between eyes and nose bridge to determine if a face is depicted in a picture.

Edge Orientation Histograms[38]: After performing edge detection on an image, the resulting edge representation is divided in sub-windows, and each pixel of the sub-window is binned in a histogram (one per sub-window) corresponding to its edge orientation and intensity. As different edge orientations and intensity are expected for different facial expressions, e.g. diagonal oriented lower lip line while smiling, such a feature could be able to distinguish between different smile styles.

Pixel Differencing[34]: A single feature consists of taking the difference between any two pixels in an image. Compared to for example Haar-features a single pixel differencing feature does not tell us much about regional structures or edges, but by taking the difference between a large amount of pairs of pixels, expected values can be extracted that correspond to intensity found for facial topologies; for instance differencing pixels in the cheek area (mostly lighter) with those around the mouth and eye area (mostly darker). For pixel differencing to work consistently, some pre-processing has to be applied that normalized illumination conditions.

Generally speaking, an appearance-based approach extract an extensive amount of features. For instance, in the pixel based difference approach of [34] over $300,000$ features are created for just a single image size of 24x24 pixels. Not all of these features will be of importance, and each of them will only contribute a small bit to making the final classification, at best. Consequentially, appearance based features are often used with Boosting algorithms[38][37][34][39]. Boosting works by iteratively selecting features that perform

better than random and combining them into a single classifier that performs well in separating the different labels. Using boosting, it is also possible to get a sense of which weak classifiers are considered important for classification, as those end up prominently in the final classifier.

Appearance based features are not without limitation. All images used in the train and test process have to be proficiently normalized (i.e. normalizing scale, position and orientation), as it is expected that local information present the same regions of the face in all of the data. This is done by first detecting the eye positions in each image, then rotating and cropping the images so that all eyes are aligned. Whitehall et al. compare different appearance based features in their research, and found that successful image registration is an important part of successful classification[38]; automatic registration (by locating eye position automatically) caused a drop in performance to manual registration in their evaluation with a data set varying strongly in image conditions. Additionally, Shan et al. applied pixel differencing and found that pose variation influenced the classification performance, with less mistakes being made in the frontal case [34]. Difference in orientation can be hard to recover from with appearance based features, as there is no in-frame normalization of rotation possible from the image data. A possible solution that comes to mind is detecting strong in-frame rotations and using a classifier trained on similarly orientated data, but this will require a lot of additional data variety.

2.1.1.2 *Geometric Features*

Geometric features consider the shape and relation between specific landmarks in the face or body. A first step in creating such features is reliably detecting landmarks, and accurately tracking them over multiple frames in the case of video data. This differs from face registration as mentioned in the appearance based feature section, as this data needs to be of higher resolution so that there is great certainty about the precise places of the jaw line, mouth corners, Cupid's bow, etc. Some examples of existing face tracking software are Chehra[2], CLM-GAVAM tracker[3] and clmtracker[1][31].

Geometric features can be constructed on various abstraction levels. On the lowest level, the features are extracted on a frame-to-frame basis, and consist of elementary calculations, e.g. distance calculations between the facial landmarks [18][35], the angle between two landmarks (with regards to the x-axis)[14], and extracted rotational values (pitch, yaw roll) of the head itself [35]. The per-frame feature vectors can be used for training a variety of classifiers, of which Support Vector Machines are often used[35][18].

Low-level features are straightforward to extract, but do not contain a lot of information on their own. As such it can be difficult to understand how a classifier uses this information to learn. As an extra step, low-level features can be combined to create higher level, semantically interpretable features. In the works of Valstar et al., face point displacement and the distances between points over multiple frames are used to detect the presence of Action Units[36]. Action Units are part of the Facial Action Coding System and represent muscle movement in the face[20]. The action units detected by Valstar et al. are directly related to the detection of a smile[15], and represent lip corner pull and cheek puffing respectively[9][35][36]. These AUs are then used as features for the final smile detection classification.

Another way of combining these low-level features is by combining them temporally. The most straightforward way of doing this would be by considering a sequence of per-frame extracted features during training. Hoque et al. developed an alternative technique in

1 https://github.com/auduno/clmtrackr

which distance features from 30 consecutive smiling frames (a segment) were combined into new features that specified information about these segments[18]. Such features included the percentage of frames in the segment having a higher smile intensity than the average intensity of the whole smile sequence, mean smile intensity in the segment, the gradient over the whole segment and the max intensity ratio of the segment compared to the whole smile duration.

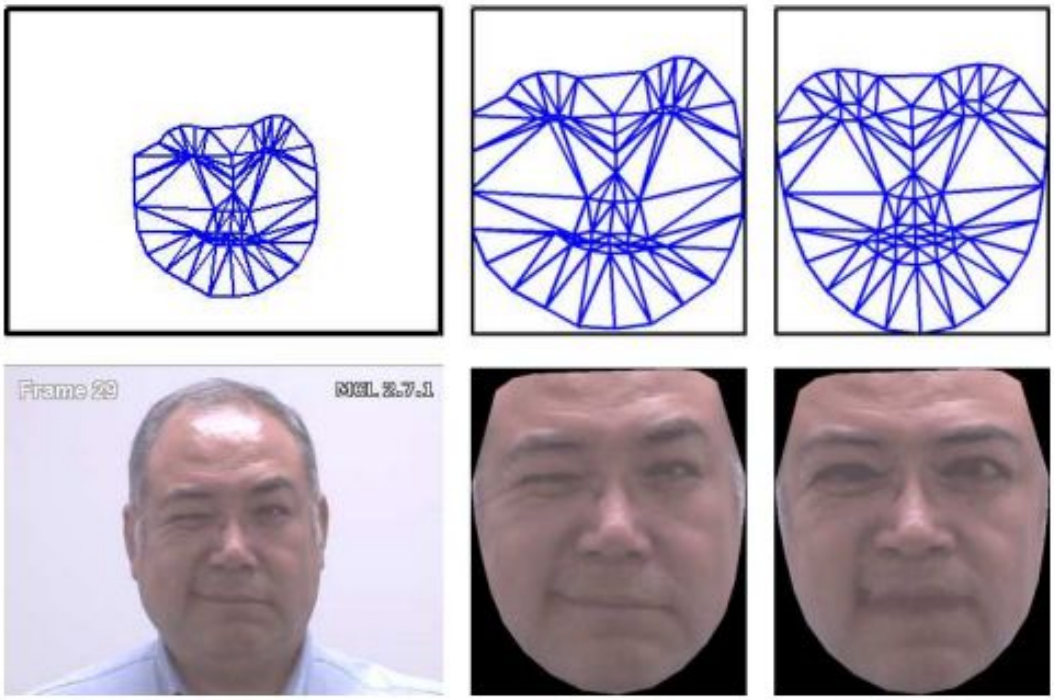
Temporal features require a classifier that is able to model the temporal nature of the data. One such a model used in smile detection research is the Hidden Markov Model (HMM). HMMs are finite state machines where the states are hidden and we only have direct access to a sequence of observations [13]. When classifying a temporal sequence using a HMM, each class results in a single HMM of which the transition probability and observation probability for all states and observations are calculated using the instances of that class. During classification the observed sequence in the target data is assigned the class of the HMM that results in the highest probable state sequence.

Using geometric features also has its downside. Their performance depend on the accuracy of the facial tracker, and it is not unlikely for tracking to be negatively influenced by lighting, occlusion, quick facial movements, facial features (like facial hair and glasses), and extreme head rotations[2]. When the accuracy drops, the face points that the geometric features are based on can cause errors in the final classification, so a robust technique that can automatically recover from errors is of great importance for successful tracking. Another downside of geometric features is that they disregard textural information. Smiling consists of more than just translation and rotation of points; dimples, exposed teeth, crow's feet, and forehead wrinkles can all help uncover if someone is showing a smile and this data can not easily, or not at all, be determined by tracking facial points.

2.1.1.3 *Combined features*

Given the complementary nature of geometric and appearance based features, combining these features would seem to improve performance. Zhang et al. have compared performance of 36 facial coordinates to gabor wavelet features, to a combined approach for facial expression recognition, and found that the combined approach resulted in about the same performance as just using the gabor wavelet features (93%), with only the coordinates performing significantly worse (73.3%) [44]. The results from this research are by no means conclusive about the importance of geometric features, as the most basic type of geometric features were extracted for experimenting.

Ashraf et al. employ an *active appearance model*(AAM) to decouple shape and appearance of a face for pain detection[1]. The active appearance model is a triangulated mesh that can be linearly deformed to fit a face in an image, see Figure 3a. By adjusting the linearly deforming shape parameters of the mask, a sense of geometrical deformation is gathered from the face. This appearance base model results in three feature sets; the models vertex coordinates of the AAM fitted on a face with head rotation, scale and translation removed(Figure 3b(top)), the raw pixel values of the face image warped so as to remove these rotations(Figure 3b(bottom)), and the raw pixel values of the isolated fame image warped in a way so it presents a neutral representation of the face(Figure 3c(bottom)). The first two have been evaluated separately, the latter only in combination with the geometric feature points obtained in the first feature set. It was found that this last combination of geometric features and the raw pixel values performed the best. This latter approach circumvents the limitation of using appearance based features when faces are heavily rotated; by warping



a) AAM fitted to frame (top), the frame used(bottom)

b) AAM normalized (top) and face deformed to fit normalized AAM (bottom)

c) neutral AAM (top), face deformed according to neutral AAM (bottom)

Figure 3.: Example of AAM derived representations, taken from [1].

the face back to a full-frontal position, the facial features can be aligned, making classification more straightforward.

2.1.2 *Smile Detection Research Topics*

The research field of smile detection has been less focusing on distinguishing between smiling faces from neutral faces in controlled environments, and more on detailed analysis; like detecting smiles in the wild, differentiating between legitimated and posed smiles and detecting meaning behind smiles. Here we discuss related works based on these objectives, their approach and the results from their experiments.

2.1.2.1 *Improving* In-The-Wild *Detection*

Whitehill et al. set the first steps towards *in-the-wild* smile detection by creating a data set (GENKI) of 25,000 pictures of people acquired from online repositories, manually labeled for the occurrence of a smile[38] and evaluated different variables, e.g. different appearance based feature types, amount of training data used and type of classifier. In their research, Whitehill et al. found that when trained using GentleBoost, Edge Orientation Histograms, Haar-like features and a combination of the two performed equally well with around 97% accuracy when using at least 2000 instances of training data. When using a Linear SVM, the same accuracy could be detected using Gabor Energy Filters. Additionally, they test the generalizing performance of a classifier trained on GENKI by testing it on the Cohn-Kanade[20] data set and, vice versa, by training on Cohn-Kanade and testing on GENKI,

finding that 98.4% of the Cohn-Kanade data set can be classified correctly using GENKI, and only 84.9% the other way around. This indicates that the GENKI data set is more universally applicable for smile detection than the heavily controlled Cohn-Kanade data set. Figure 4 shows the diversity in smile style and environmental conditions in the GENKI data set over the Cohn-Kanade data set. Although the GENKI data set is a step in the right direction with regards to *in-the-wild* detection, it can not be called completely spontaneous, as most smiles gathered consists of a rather forced picture-style smile instead of complete spontaneity.

Figure 4.: Top: Frames depicting a smile from the Cohn-Kanade dataset[20] (©Jeffrey Cohn).
Bottom: Frames depicting a smile from the GENKI data set.

In the same vain, the research of Shan et al. [34] continues the improvement of *in-the-wild* smile detection by testing pixel differencing with AdaBoost on a public subset of the GENKI data set. The evaluation was done by comparing their approach to the Local Binary Pattern and Gabor Energy Filters as used in [38]. The accuracy of the raw pixel value was around 80% against 87.10% and 89% for LBP and Gabor respectively when trained using a Linear SVM. Using the top 500 features, pixel differences perform with 89.7% accuracy, but already 85% with only 20 pairs of pixels, making the classification process a lot faster. A mentioned limitation of their approach is difficulty in correctly handling pose variation. Another limitation is that pixel differencing is not easily expendable for improvements, like for example tracking pixel differences over multiple frames to develop a temporal approach.

2.1.2.2 *Legitimate Smile Detection*

A smile is usually an indication of an underlying emotional state, i.e. being in a happy mood or amused. When this underlying state is the cause of the smile it can be considered legitimate. When a person forces a smile, for example out of social compliance, we speak of an illegitimate smile that misses the expected emotional state. Research has been conducted to differ between the two types of smiles.

In the works of Valstar et al.[35] geometric features are used to distinguish between acted and legitimate smile video data. Valstar et al track 12 points on the face (4 on each eye and around the mouth), together with the head orientation, scale and position and 5 landmark points around the shoulder. The authors create low- and high-level abstraction techniques. The low-level abstraction techniques contain the position of the tracked data points and the distances between face and shoulder points. In the high abstraction strategy, the authors transfer the low-level features into Action Units, and train a specific classifier per segment

of a smile; the onset, offset and apex. The strategies were evaluated on videos from the MMI-facial expression database, and it was found that the abstract strategy performed best in distinguishing between spontaneous and posed smile, with a classification rate of 93%. The authors also experimented with different combinations of the modalities and found that the head motion data aligned is the most important, but combining all three together delivers the best result, regardless of the abstraction level used.

Dibeklioğlu et al. research the same objective of distinguishing between posed and spontaneous smiles, but focus on eyelid movement, as it was observed by Duchenne that specific contractions in the eyelid only occur in spontaneous smiles[14]. The authors limit themselves to geometric features, used both on high- and low-level abstractions. Firstly, the performance of eyelid movement for classification was compared to that of movement in the mouth, eyebrows and cheeks, using Continuous Hidden Markov Models and face point position as the features. It was found that eyelid movement features performed best with around 86% accuracy with 6 hidden states when testing on the Cohn-Kanade and BBC smile data set. Additionally, temporal eyelid features were extracted for both eyes; the distance of the upper-eyelid center to the center line between upper and inner eye corner, and an angle that indicates how far each eye is opened. Using a naive Bayes approach a classification accuracy of 91.3% was achieved using the standard deviation, minimum, maximum and mean value over a sequence of these features.

2.1.2.3 *Different Smile Categories*

Hoque et al. try to distinguish between frustrated and delighted smiles[18]. To do so, the researchers gather their own data, recording people while exposing them to a task inciting frustration and delight and labeling the data accordingly. The authors then extracted a smile intensity using the SHORE API[16], and distance features between 22 face points. From these features per frame the average was calculated over a whole video showing frustrated/delighted smiles for the final feature vector. Testing a variety of classifiers, the highest accuracy achieved was 48.1% using an SVM.

Additionally, the authors tested temporal patterns by cutting the video up in sequences that differentiate between smiles and non-smiles, and split the smile sequences up in segments of 1 second long. For each of these 1-second segment four features are extracted: percentage of frames in 1-second segment above mean of the whole smile sequence, mean smile intensity during the 1-second segment, gradient across segment, and max smile intensity comparison between segment and complete sequence. The best accuracy of this temporal approach was 92.30% and obtained by means of a D-SVM and outperformed human judgement. Instead of averaging out the feature values for each smile-sequence like for the SVM, the D-SVM appended all the temporary features encoded for each of the sequences and used Principal Component Analysis to reduce the dimensionality to the first 4 principal components. The resulting classifier is considered pseudo-dynamic as the D-SVM does not use the features values of the 1-second segments temporal like when using an HMM for classifying, but it is able to encode the shape of the data-variance better than the mean-approach of the SVM, resulting in about 7% improvement in accuracy.

2.2 TRANSFER LEARNING

The basic premise of transfer learning is that a classifier is learned on one set of data, which we will call the *auxiliary* data ($\mathcal{D}^a$), while we want it to perform well on a different data set of never before seen data, called the *primary* data ($\mathcal{D}^p$), that differs in some way from $\mathcal{D}^a$. Dif-

ference in different data sets means that the feature space might differ, or the distributions of the data sets[28], which the next section will expand upon.

A reason to apply transfer learning, instead of learning a new classifier from $\mathcal{D}^p$ only, is that no or few labeled instances might be available in this data set. This is a common problem in machine learning, as acquiring and labeling data is time consuming and not always a viable solution. In this case, transferring knowledge from existing data that is related will be more meaningful than training with just the limited data in $\mathcal{D}^p$, which might cause overfitting. The meaning of the term *related* is not uni-vocally defined within transfer learning. According to Yang et al. a good heuristics for relatedness is the performance on $\mathcal{D}^p$ of a classifier trained with $\mathcal{D}^a$, with a high performance indicating a good basis for transfer learning as apparently feature distribution and conditional probabilities are similar [40]. This definition is however difficult to apply, as "high performance" is not well-defined and not something that can be straightforwardly tested for $\mathcal{D}^p$ when there is no labeling available (as is the case for practical applications). Additionally, there are cases of transfer learning where the feature spaces differ, and another measure of similarity needs to be used.

Regardless, finding related auxiliary data for training is not an impossible task. For instance, in the field of object recognition, $\mathcal{D}^a$ might consist of images of objects captured under perfect (i.e. uniform lighting, non-occluded, isolated) conditions gathered from an online merchant, while $\mathcal{D}^p$ might consist of in-the-wild picture data, shot with a different camera in a home environment with varying perspectives[21]. In such condition, the classifier trained on $\mathcal{D}^a$ will perform sub-optimal out-of-the-box, but can be readjusted with transfer learning to better handle the variances in feature representation of the the new environment, reducing the original false classifications. Note that transfer learning will simultaneously likely reduce the results on $\mathcal{D}^a$, but we are only interested in the results on the primary data set. There are several ways of using the primary data when transfer learning. We will limit our description to the *inductive* case, as it is the approach we will take throughout the research. In the *inductive* case, a small (and on its own insignificant) portion of $\mathcal{D}^p$ is labeled ($\mathcal{D}^p_l$) so it can be utilised throughout the transfer learning process, and will improve performance on the unlabeled part ($\mathcal{D}^p_u$).

2.2.1 *Conditions*

When considering how data sets can be different, Pan et al. distinguish between differences in the domain and in the task[28]. They define a domain as a tuple $D = \{\mathcal{X}, P(X)\}$ where $\mathcal{X}$ is the feature space and $P(X)$ is the marginal probability distribution of the feature data. The task is defined as $T = \{\mathcal{Y}, P(y|x)\}$, where $\mathcal{Y}$ is the set of labels assigned to the instances, and $P(y|x)$ is the conditional probability of label $y \in \mathcal{Y}$ belonging to instance $x \in X$. During feature extraction and labeling of a data set, the $\mathcal{X}, P(X)$, and $\mathcal{Y}$ terms are assigned. The $P(y|x)$ is learned by a classifier applied on the data set. Consequentially, both $\mathcal{D}^a$ and $\mathcal{D}^p$ have their own domain and task, which will be indicated by D^a, T^a and D^p, T^p respectively.

A difference in domain between D^a and D^p can mean one of two things. It could mean that $\mathcal{X}^a \neq \mathcal{X}^p$, i.e. the feature spaces are not the same. This occurs in the works of Dai et al.[11] where transfer learning is applied on cross-language text classification. Here, $\mathcal{D}^a$ consists of texts in the English language and $\mathcal{D}^p$ consists of German texts. The feature spaces in this example differ, while $\mathcal{X}^a$ consists solely of English terms, and $\mathcal{X}^p$ of German terms. When learning a traditional classifier on $\mathcal{D}^a$, it is not able to classify instances with a different set of features, and transfer learning can remedy this problem.

A difference in domain could also happen when feature spaces match, but instead $P^a(X) \neq P^p(X)$. An example of this can be found in the research of Yang et al. where transfer learning

is applied to determine the presence of a specific concept (like 'studio-segment', 'outside') in fragments of different television programs[40]. The same set of gabor wavelet features and color moments are extracted from the video frames (i.e. $\mathcal{X}^a = \mathcal{X}^p$), but because training and classifying is done on different television shows $P^a(X) \neq P^p(X)$. For example, if the program in $\mathcal{D}^a$ has a blue studio backdrop, while the one in $\mathcal{D}^p$ contains shades of red, the feature distribution of the color moments will differ, and possibly not even overlap. In such a case, a traditional classifier might make mistakes as its decision function is making decisions based on data that is not representative of the new instances.

A difference in task means that the classifier that was trained on $\mathcal{D}^a$ does not directly map to the classification task we want to perform on $\mathcal{D}^p$. This can manifest itself in two ways. Firstly, this could mean that $\mathcal{Y}^a \neq \mathcal{Y}^p$. An example can be found in the works of Raina et al. where $\mathcal{D}^a$ consists of handwritten letters and $\mathcal{D}^p$ consists of handwritten numbers[30]. In their research $\mathcal{X}^a = \mathcal{X}^p$, as features in both data sets consist of pixel intensities from 28x28 gray scale images. However, the classes differ; $\mathcal{Y}^a$ consists of 24 classes, indicating which letter is shown, and $\mathcal{Y}^p$ consists of 10 classes, indicating which number is shown. From this it can also be assumed that $P^a(X) \neq P^p(X)$.

Secondly, a task difference could instead mean that the probability of a certain label belonging to a certain instance not equal in both data sets, i.e. $P^a(y|x) \neq P^p(y|x)$. This entails that labels $\mathcal{Y}^p$ have been assigned under different conditions than $\mathcal{Y}^a$.

In the case $\mathcal{D}^a$ and $\mathcal{D}^p$ differs in neither task nor domain, the classification problem can be regarded as a traditional machine learning problem where no transfer learning is necessary.

2.2.2 *Limitations*

Transfer learning is not without limitations. Although we have described some guidelines above on when transfer learning can be applied, this does not mean that applying transfer learning will improve results directly. It is difficult to determine how much two data sets differ from each other; even though we might be able to show that marginal distributions differ, there is not a unified metric that indicates if transfer learning is appropriate to apply[10]. In fact, it might even be that an effect called negative transfer might occur, which means that the addition of transfer learning can cause a worse performance than not using it at all. This can happen due to the auxiliary and target data not being similar enough. Another risk in that in the case of inductive transfer learning, $\mathcal{D}_l^p$ might not be representative of $\mathcal{D}^p$. If this is the case, than $P_l^p(X) \neq P_u^p(X)$, and transfer learning is not applied effectively. Currently, the exact limitations of transfer learning are still being researched[10].

2.2.3 *Approaches*

Transfer learning can be applied in various ways. Which one to use depends both on how auxiliary and target data differ. We here discuss four categories of transfer learning that use different strategies for transfer learning; *instance transfer*, *feature representation* and *parameter* transfer which have been suggested by Pan et al, and a fourth category called *model transfer*.

2.2.3.1 *Instance Transfer*

Instance transfer makes use of the assumption that some instances in $\mathcal{D}^a$ are more important for correct classifications in $\mathcal{D}^p$ than other instances. By determining which of the instances, a weighting of importance can take place in the auxiliary data set, improving the performance of the final classifier.

An example of an approach that works on this principal is created by Dai et al, where a classifier called *TrAdaBoost*[12] is trained on $\mathcal{D}^a \cup \mathcal{D}_l^p$. Their approach is based on AdaBoost, which assigns weights to different instances so that the final classifier performs well on the test data (in this case $\mathcal{D}_u^p$). AdaBoost assumes that the feature distribution in the train and test data are similar (as in traditional machine learning), which is not the case when using transfer learning. For this reason, the algorithm is slightly modified.

Normally in AdaBoost, after training the classifier each iteration, the instances that were wrongly classified get their weights increased to boost their importance in the next iteration. This still happens for the instances in $\mathcal{D}_l^p$, but the instances wrongly classified in $\mathcal{D}^a$ have their weight reduced. This way, instances from the auxiliary data that seem to clash with the primary data, will influence the classifier less each iteration. Dai et al. test the performance of TrAdaBoost on a variety of textual machine learning problems, and show that in case there is a lot less labeled primary data to auxiliary data, the TrAdaBoost outperforms a SVM baseline considerably, which shows that TrAdaBoost can be successful in distinguishing between helpful and unhelpful data in the auxiliary data set. The authors mention two downsides of their technique. First, they report that the performance of the classifier learned with TrAdaBoost is sensitive to the quality of the auxiliary data, meaning that if $P^a(X)$ differs too much from $P^p(X)$, TrAdaBoost might not increase performance, and it is not clear what the conditions for this are. Secondly, the convergence can be slow.

Yao et al. extend the research of Dai et al. [41], trying to combat the possibility of negative transfer by using multiple auxiliary data sets in the transfer learning process and calling this addition *MultiSourceTrAdaBoost*. Where the original TradaBoost only uses the combination of the single auxiliary data set and the labeled primary data when learning the weak classifiers, the approach of Yao et al. learns the weak classifier over the combination auxiliary data that fits (i.e. reduces the error to) the target data best in the current iteration. The idea here is to reduce the chance of negative transfer, because at each iteration the best auxiliary candidate data is applied. Yao et al. evaluate their approach using an object category recognition problem; the primary data consists of an image category to recognize, and the auxiliary data consists of other object categories. From these, the bag-of-words method is used for feature extraction. The authors use AdaBoost (which only uses the target data to learn) and TrAdaboost (which combines all auxiliary data into one set) as their baseline, with an linear SVM as the used classifier in all cases, and vary the amount of auxiliary data sets (1 to 10) and number of positive instances (1 to 50) in the labeled target training data. Their results show that MultiSourceTrAdaBoost performs the same as TrAdaBoost when only a single auxiliary data set is used (as to be expected), and when more are available, it performs significantly better than TrAdaBoost when few positive training samples are available in the target labeled data. Moreover, its accuracy and consistency grows when the number of auxiliary sets grows and only a single positive instance is available in the target data.

2.2.3.2 *Feature Representation Transfer*

When transferring feature representations, the primary and auxiliary data are both mapped to the same feature space, which levels out the features used and the distribution. Zhong et al. try to find a kernel mapping of both data sets to a new, third, feature space where the marginal distribution is similar of both data sets [45]. This is done using Kernel Discriminant Analysis. Afterwards, two different strategies are applied. In the first approach, called *KMapEnsemble*, the auxiliary instances that have a dissimilar conditional probability to the target data are determined and removed before training a classifier. These instances

are found using Bisecting K-Means clustering; all labeled data starts off as one big cluster, and then gets split into two clusters using K-Means in case the sum of squared error is reduced for the new clusters or if the current purity is below a certain threshold (i.e. the cluster contains too many instances of different classes). After the algorithm terminates, the auxiliary instances in a cluster that match the prevailing target instance in that cluster are used during training. A second approach, called *KMapWeighted* does not use the Bisecting K-Means, but uses the TrAdaBoost approach on the new feature space by lowering auxiliary instances that are wrongly classified. Zhong et al. evaluate their method with a text categorization task, and its outcome is compared to a variety of traditional machine learning techniques and TrAdaBoost. The results show that KMapEnsemble is in most cases (24 out of 27 different data set and base classifier test) the best performing classifier with up to a 25% increase in accuracy. The comparison of the KMapWeighted approach and TrAdaBoost show the difference it makes to map both data sets to the new feature space. In most cases, KMapWeighted performs better than TrAdaBoost, but often with a smaller margin than KMapEnsemble. The authors also compare the KMapEnsemble case to a case without the clustering procedure and a case that does not use the feature mapping and evaluate at least 5% increase in accuracy using the KMapEnsemble technique.

2.2.3.3 *Parameter Transfer*

Parameter Transfer works on the assumption that the classifier trained on the auxiliary data will share similar parameters with a classifier trained on the primary data. What exactly this entails, depends a great deal on the model applied. Yao et al. create an approach based on TrAdaBoost, and call it *TaskTrAdaBoost*. Similar to their MultiSourceTrAdaBoost, this approach consists of multiple auxiliary data sets. In a first phase, all possible weak classifiers are constructed individually over each auxiliary data set (by essentially using AdaBoost), keeping only those that perform well enough given a specific threshold. In the second phase, TrAdaBoost is applied, but with a different weak classifier selection criterion: in each iteration the best performing weak classifier from phase 1 on the labeled target data is chosen to incorporate into the classifier. In this second phase, the auxiliary data is not used in the instance weighting process. The parameter that is assumed to be shared in this case between source and target, are the weak classifiers learned for each of the source data sets; The second phase only needs to determine which of the parameters are shared between source and target. This approach is pitted against the MultiSourceTrAdaBoost in the object category detection challenge mentioned earlier, and shows about the same results, with the addition that that it is less prone to over-fitting than MultiSourceTrAdaBoost when few positive instances are available in the target data.

2.2.3.4 *Model Transfer*

In model transfer, the auxiliary data is not used directly in the transfer learning process, but rather the classifier trained on the auxiliary data. The decision function of this classifier is in some way adapted to increase performance on the primary data.

Yang et al. [40] developed Adaptive Support Vector Machines (A-SVM) that use a regular SVM on the auxiliary data. Then, this classifier is fed to the Adaptive SVM, which adds another term to the decision function learned by the regular SVM. This extra term is in essence the same decision function, but only learned on $\mathcal{D}_l^p$, and its influence is minimized as much as possible, so that the final decision boundary will be similar to the one learned on $\mathcal{D}^a$. The user can fine-tune the influence by choosing the value C of the decision function learned on $\mathcal{D}_l^p$. Additionally, multiple auxiliary data sets can be utilised in this approach,

by making the base classifier a weighted sum of all auxiliary classifiers learned, and tagging along the extra decision function for $\mathcal{D}^p$ in the same manner.

Yang et al. evaluate the Adaptive SVM approach using the domain of video semantics analysis, where a different news program is trained with the auxiliary data than in the primary data. $\mathcal{X}$ here consists of gabor wavelet features and color moments extracted from key frames, and $\mathcal{Y}$ is a binary label that denotes the presence of a concept, for example presence of environmental ques, genre, and presence of objects. The prediction process consists of deciding for each frame if a chosen concept is shown in that frame. Yang et al found that the Adaptive SVM approach improves performance compared to training a regular SVM with $\mathcal{D}_l^p$ or with $\mathcal{D}^a$. The Adaptive SVM performs similar when training a regular SVM on $\mathcal{D}^a \cup \mathcal{D}_l^p$. Due to the reduced training time (about ten times faster), the re-usability of the auxiliary classifiers and reduced influence of SVM's C parameter, they consider the Adaptive SVM a better choice than the single SVM classifier that combines $\mathcal{D}^a$ and $\mathcal{D}_l^p$.

3

DATA

For this research a combination of two data sets containing smile and non-smile data has been utilised for both train and evaluation purposes.

3.1 GENKI DATA SET

The GENKI data set has been created by Whitehill et al. with the intention of collecting smile data in a variety of different environments, i.e. indoors and outdoors, as well as of people of varying age, gender, ethnicity and appearances, like facial hair and glasses [38]. The original GENKI data set used in their research consists of 63.000 images in total collected from public online repositories. An online version is offered for external use, containing a subset of 4000 pictures in total. These images have been hand labeled by the researchers as either showing a *smile* or *no-smile*. In this research the labels are used as provided by the data set.

The GENKI data set provides this research with an extensive data set of different faces, making it possible to easily train a general classifier. Because the GENKI data set consists mostly of smiles that are posed for pictures, see Figure 1, it can not be called completely spontaneous, but it is still diverse, i.e. the portrayed smiles differ in intensity and display unique smiling styles per person. A limitation of this data set is that it inhibits us from using temporal feature data during the training process, making the geometric feature data quite limited when used for a classifier.

3.2 AM-FED DATA SET

The Affectiva-MIT Facial Expression Dataset (AM-FED)[24] was gathered as part of the research by McDuff et al. in which they created a framework for crowd sourcing and analyzing smile data[25]. The data set shows people recorded in front of their web cam while watching one of three humorous Super Bowl commercials provided throughout the experiment. The data set contains 242 different videos, containing in total 168,359 frames. The AM-FED data set provides the video files, various hand-labeled FACS features, self-report data from the subjects and two sets of smile labels. One of the sets of labels was manually generated and the other by a classifier trained on LBP features around the mouth area. Although their evaluation indicates an AUC performance of 89.9% on the AM-FED data set with the automatically generated smile data, we opted to use the manually provided labeling instead, as the automatically generated labeling seems inconsistent for some of the faces, and the manual labels also were used as ground truth in their own research.

The manually labeled smiles have a value ranging between 0% and 100%, with 0 meaning none of the labelers believe the subject was smiling at that point in time, and 100 indicating all of them believe the subject was smiling. For our purposes, we chose to categorize all

labels in the range 75% − 100% as a smile, and everything below that as a no smile. Mostly, all preceding and subsequent frames gradually drop in a lower score, meaning the majority is centered around the moment of apex, which is good enough for our purposes. Because the data was only labeled at a time instance where the value changed (so when one of the labelers decided the subject started or stopped smiling) we generate frame-by-frame labels from this data.

Because not all videos in the data set contain a lot of smile data, and even less so after setting the aforementioned smile threshold, only a small set of videos ended up useful for our purposes. To determine which videos are suitable, we take the following criteria into consideration. Firstly, the data should contain a lot of smile "patches" (i.e. number of uninterrupted frames classified as a smile). A lot of the data consists of two apex moments at the end of the video (presumably where the video climax takes place), and this leaves little data to evaluate with cross-validation, as each fold should contain enough smile data during transfer learning, and we do not want to shuffle the data. Secondly, we would like the final selected data to be diverse, so that transfer learning will make sense. Based on these criteria, we removed videos of which less than 40% and more than 70% of the frames were labeled as a smile. From the remaining videos, the amount of smile patches were calculated and a manual selection of videos was made that ensured a diverse set of smiles and a large number of smile segments to work with. The videos shown in Figure 5 have been selected. Table 18 in Appendix A provides some extra information about the videos.

Figure 5.: Frames labeled as a smile from all selected AM-FED data.

4

APPROACH

Chapter 2 discussed the general flow of current approaches towards smile detection; the acquisition of train data with smiling and non smiling faces, the feature choice i.e. appearance based or geometric based, the classification process and their performance measures. In our research, we will take novel decisions for these steps, which we will discuss and motivate in this chapter.

4.1 CHOICE OVERVIEW AND MOTIVATION

4.1.1 *Transfer Learning*

The biggest change with regards to earlier research in smile detection is the choice of applying transfer learning, rather than traditional machine learning. Our motivation for using transfer learning is based on the idea that no two faces are exactly alike during smiling. When we train on different faces than that the classifier will be used on, we are expecting that the feature distributions will not match up perfectly, i.e. $P^a(X) \neq P^p(X)$ where $P^a(X)$ is the marginal probability distribution of the train data instances, and $P^p(X)$ that of the target data. Additionally, even in the unlikely case that feature distribution is similar in different faces, this might mean that $P^a(y|x) \neq P^p(y|x)$, i.e. the same facial features do not indicate a smile in different faces due to distinct smile styles (as displayed in Figure 1). Both of these are conditions under which transfer learning can offer a better solution than traditional machine learning, which makes it a viable approach to investigate.

We regard the challenge of individualized smile detection as an *inductive* transfer learning problem, meaning that there will be a small amount of $\mathcal{D}_l^p$ available during the transfer learning process. Using no labeled data for the primary set would make the transfer task *transductive*, which has the use case restriction $T^a = T^p$[28], meaning that the probability of assigning classification label y is as likely in both domains (i.e. $P^a(y|x) = P^p(y|x)$). This is an assumption we cannot make, as mentioned in the previous paragraph. Moreover, when considering the applications that might benefit from individualized smile detection, specifically when using it for diagnostics in a self-therapy tool, we believe adding a short, one-time calibration phase to determine $\mathcal{D}_l^p$ is a reasonable trade-off between user involvement and reliability of the final smile detection.

Chapter 2 provides an overview of different strategies available for performing inductive transfer learning. From these, we choose to apply a model-based transfer approach by means of the Adaptive SVM, developed by Yang et al.[40]. We prefer using a model-based approach over instance-based approaches, like TrAdaBoost, as the latter views all data in $\mathcal{D}_l^p$ equally during training. As $\mathcal{D}_l^p$ will consists only of a small amount of data, outliers that do not represent $\mathcal{D}_u^p$ might have a big effect when TradaBoost increases the weight of instances for multiple iterations. Although the authors show that TradaBoost trained with

an SVM performs better using a small amount of $\mathcal{D}_l^p$ than a regular SVM trained on $\mathcal{D}^a \cup \mathcal{D}_l^p$, we believe that setting the parameters of an Adaptive SVM to regulate over-fitting a more intuitive and possible better performing approach.

Applying the KmapEnsemble approach for feature representation transfer would seem like a good alternative to consider for individualized smile detection; the approach of transferring $\mathcal{D}_l^p$ and $\mathcal{D}^a$ to a feature space and selecting only those instances in $\mathcal{D}^a$ that match with $P^p(y|x)$ would seem to solve both our problems of differences in the domain as well as in the task. However, it is less clear to foresee its performance on the domain of smile detection rather than text classification. Additionally, Support Vector Machines have been used in various previous researches in smile detection, and although that does not mean it is appropriate for our specific research, it seems a safer approach to take.

A last benefit of using A-SVMs is that it provides a lot of re-usability and accordingly saves us a lot of computation time, as the SVM trained on $\mathcal{D}^a$ data can be reused each time we change parameters or primary data sets in the adaptive transfer trials for evaluation purposes.

Section 4.3 will explain how the transfer learning pipeline using the Adaptive SVM is performed for classification purposes.

4.1.2 *Feature Choice*

Chapter 2 has discussed both geometrical and appearance-based features as candidates for visual smile detection. In this research we limit ourselves to low-level geometrical features. Although appearance-based features have shown promising performance in past research[38][25][34], these features were extracted in a black box fashion, where a boosting algorithm takes care of determining the underlying value of each of the features. We, however, want to test our hypothesis that a specific set of low-level geometrical features results in unique feature distributions between different people, so that $P^a(X) \neq P^p(X)$ in case the auxiliary data set and primary data set consists of different faces. By selecting low level features which we expect cause a difference in probability estimates, like mouth corner orientation and distance, we can apply transfer learning to learn this difference in probability estimates. Combining these low level features into temporal or AU features possibly causes a same discrepancy in feature value distribution, but doing so adds an extra abstraction level to be analyzed when results are not as expected. Given our first steps towards individualized smile detection, we limit ourselves first to testing the basic premise. In case the expected behaviour is found, we can continue from there.

We create a few different feature sets to gain understanding of the impact of different feature types, like face rotations and face point distances when trying to classify smiles. In Section 4.2 the process of extracting features from the data is explained, as well as an explanation of the the specific low-level geometric features that are chosen.

4.1.3 *Label Choice*

Although related research has become more specific in detecting the meaning behind the smile data, i.e. focusing on different types of smiles or the legitimacy of a smile, we will regard all images in each of the data sets using the classes $Y = \{smile, no\text{-}smile\}$. Although this might seem like a step backwards compared to the more specific research interest of recent smile detection research, it is not important for us to know the intention behind a smile to learn the impact of our research for the improvement of individualized smile

detection. Our choice for labeling is straightforward to implement as both the GENKI data set already utilises the same binary labels, and the smile intensity provided by AM-FED was thresholded as stated in the previous chapter.

4.2 FEATURE EXTRACTION PIPELINE

When training a classifier with geometric feature data, all frames of the data set have to undergo a process of face point extraction, data normalization and feature extraction. Figure 6 visualizes this process on a single picture from the GENKI data set. An in depth explanation of the different steps is provided in this following section.

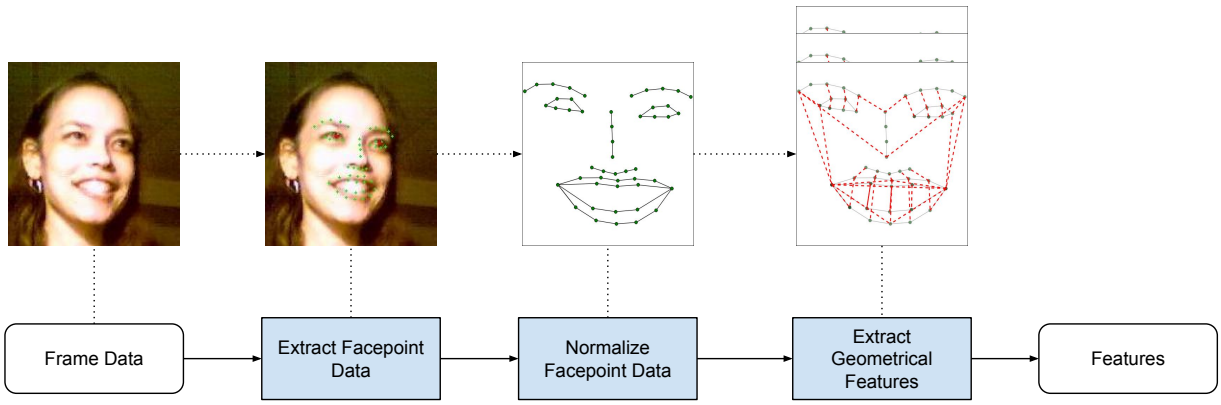


Figure 6.: The feature normalization pipeline demonstrated on a single picture of the GENKI data set.

4.2.1 *Point Data Extraction*

A geometrical approach towards smile detection requires information about important facial landmarks in each frame; like eye and mouth corner position. There are several methods for point data extraction mentioned in Chapter 2, from which this thesis utilises the Chehra face tracker developed by Asthana et al.[2]. The Chehra tracker was chosen for its high dimensional face point data, and its ability to converge quickly and accurately while being able to recover from short occlusions during informal tests with webcam data. The resulting detection when using Chehra consists of 49 face points, of which 5 points are tracked per eyebrow, 6 points per eye, 9 points for the nose, 6 for the inner lip and 12 for the outer lip line. Additionally, Chehra tracks the 3D position of the face each frame and thus provides us with yaw, pitch and roll data as well. Figure 7 shows the face points extracted by Chehra.

The Chehra face tracker has been applied on all picture data provided by the GENKI and AM-FED data set. Because Chehra utilises consecutive frames for improved face point placement accuracy, correct placement of face points was impossible on a sizable portion of the GENKI images. All images were manually inspected and instances with incorrectly placed face points or no detected face points were discarded from further use. This left us with 2963 images from the initial 4000 in the data set that are used throughout this research; 1610 images showing smiles and 1353 images showing no smile.

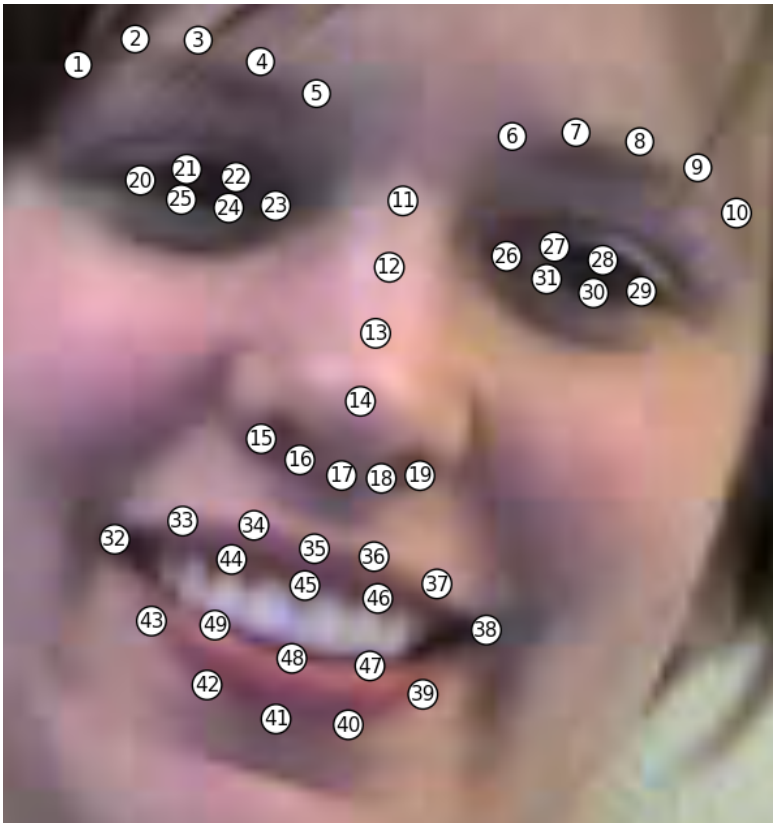


Figure 7.: Facepoint data on a frame of AM-FED Video 1.

Performing face tracking with Chehra on the AM-FED data also failed sometimes, often because of overexposure, facial hair, multiple people entering the screen and facial occlusions. Prior to selecting the 10 data sets mentioned in Chapter 3, we ran the face tracker on a sizable set of AM-FED videos, and kept the ones where tracking performed well enough in most frames. From these we pruned the few frames that badly presented the actual face points of the frames. Then from this data we made the video selection with the criteria explained in Chapter 3. See Table 18 in Appendix A for information about the amount of frames pruned for each video. Removing these frames does not cause a lot of problems as we don't extract any temporal information during the training process.

4.2.2 *Point Data Normalization*

The extracted face point data by Chehra has to undergo normalization before we are able to extract meaningful geometric feature data. If normalization is not applied, the danger exists that features extracted from the face point data are obscured by undesirable deformation that can have effect on the final classification process. For example, features using distances between points can encode variance with regards to how close a face is towards the camera, making it unsuitable for smile detection. The point normalization consists of 3 steps; normalizing translational variance, scale variance and rotational variance.

Translational variance is removed by making the center nose tip (point 14 in Figure 7) the origin of all the face point frames. Afterwards, the scale variance is removed by scaling the size of the nose bridge to be 30 pixels for each of the face point sets (the length between point 11 and point 14 in Figure 7). The rationale behind this is that this distance is not subjected to change throughout a video, and thus scaling this to be uniform does not affect

the geometrical makeup of a smiling face. Only in frames with strong pitch rotation might this cause a problem, as the pitch rotation might optically shorten the nose. Unfortunately, no set of points in the face can be used without having some kind of distance change in case of extreme head rotations. The effect of these two normalizations for Video 7 is shown in Figure 8 with the results scaled up by a factor of 10, so as to clearly show the resulting face point distributions.

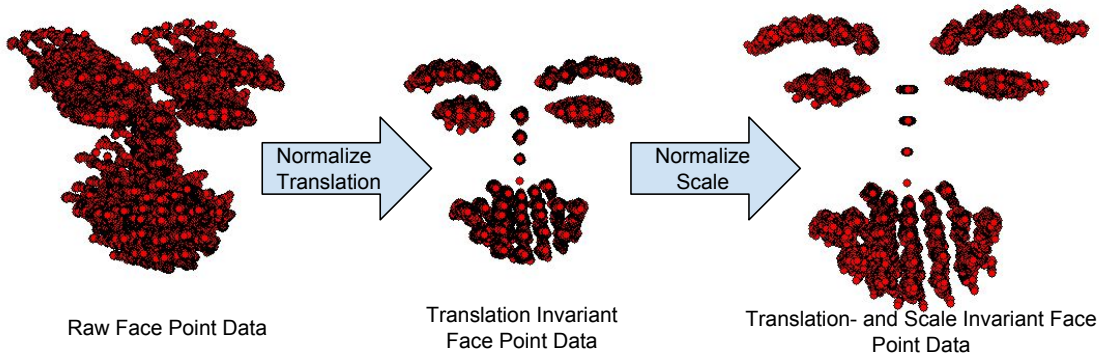
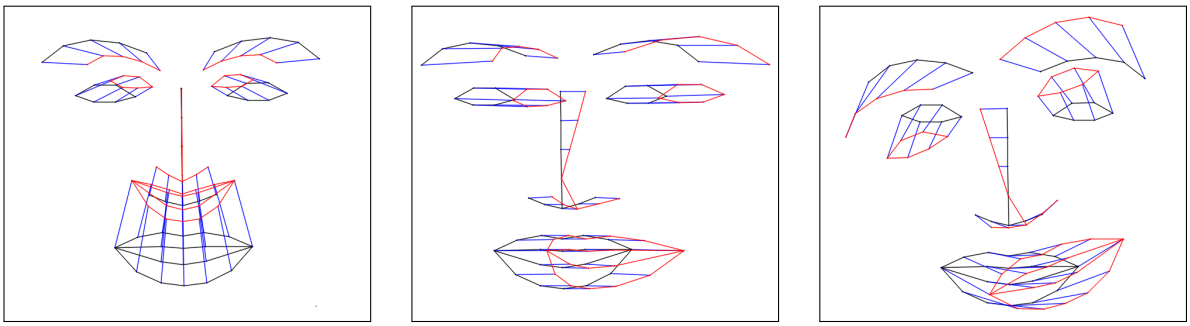


Figure 8.: Translation and scale normalization for Video 7.

Lastly, the rotation variance needs to be removed from the face point data. By applying Principal Component Analysis (PCA) one can restructure data in such a way that the axis of the data align with the directions in which the variance is highest. By considering our translation and scale invariant face points as a single point in an 98 dimensional space (where the x and y value of each face point are considered their own dimension) we are able to show these variances, and remove the principal components that contain the variances that need to be removed; the yaw, pitch and roll. Figure 9 shows the first three principal components of the GENKI data set which represent the pitch, yaw, and roll respectively. The black outline shows the average face, the blue lines the variance contained by the principal component and the red outline shows the face with the variance applied. A standard deviation of 100 was used, to exaggerate the information contained in the the principal component applied.



pitch **yaw** **roll**

Figure 9.: Visualisation of the first three Principal Components of the GENKI data, showing the pitch, yaw and roll variance respectively.

Once the principal components containing the rotation variance are established, removal of these variances is as follows. During PCA the average face is calculated, and for each of the face points the weight for each principal component is revealed. The data is then reconstructed and the weights of the undesired principal components are set to zero. This

results in the data as seen in Figure 10. The removal of these principal components can also remove non-rotational variances when the rotational variances are not cleanly separated in the principal components. The pitch visualisation in Figure 10 seems to show this, as the mouth corners seem to move upwards more than to be expected of only a pitch-rotation. There also seems to be a little bit of scale variance left in this principal components, which might be caused by frames with big pitch rotations (of which the difficulty with regards to scale rotation normalization has been clarified previously). Looking at the resulting face point distribution helps determine the impact of removing the variance and if enough diversity is kept in the data set to work with. Figure 10 shows the end result of the rotation normalization on GENKI. Using the representation of these final face points it can be determined if the rotation was successfully removed, and if the smile variance is still contained in the reconstructed data. In the case of GENKI it is a bit more difficult to see due to the different faces that are contained in the data, causing a diverse end result.

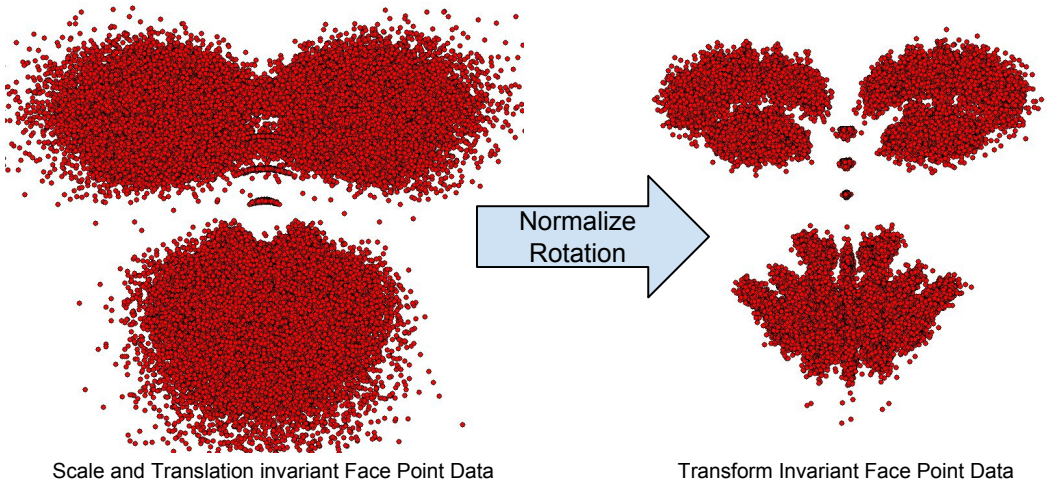


Figure 10.: Visualisation of the GENKI face point data before and after rotational normalization

Normalization has to be performed on both the train and test data using the same approach in both cases. For the translation and scale normalization this is done by using the same reference points in both cases; the ones that we have described above. In the rotational variance normalization we have to remove the same eigenvectors in both train and test data, else the resulting data becomes incomparable. We approach this as follows. We determine the eigenvectors containing the rotational variance in the data we will be using for training, and remove the same eigenvectors in the test set. In practice this usually consists of removing the first 3 eigenvectors.

4.2.3 *Feature Extraction*

We believe that low-level geometric features extracted from the normalized face point data will be appropriate for transfer learning, as there exists geometrical data in the face that will create different marginal distributions when different faces and different smile styles are presented. Such features would be difficult to learn under traditional machine learning conditions, but are very discriminative in nature, otherwise. For example one might expect that the distance between the lower lip center (point 41 extracted with the Chehra face tracker) and mouth corner (point 31 or 38) is discriminative during smiles and non smiles, but the boundary might be different for two people with distinct smile styles. Figure 11

shows that this is indeed the case for AM-FED Video 9 and Video 10. In both cases the feature is discriminative, but its marginal distribution (in blue) and conditional probability (red and green) differ. During feature selection we hand-pick features which we expect behave similar in a similar fashion.

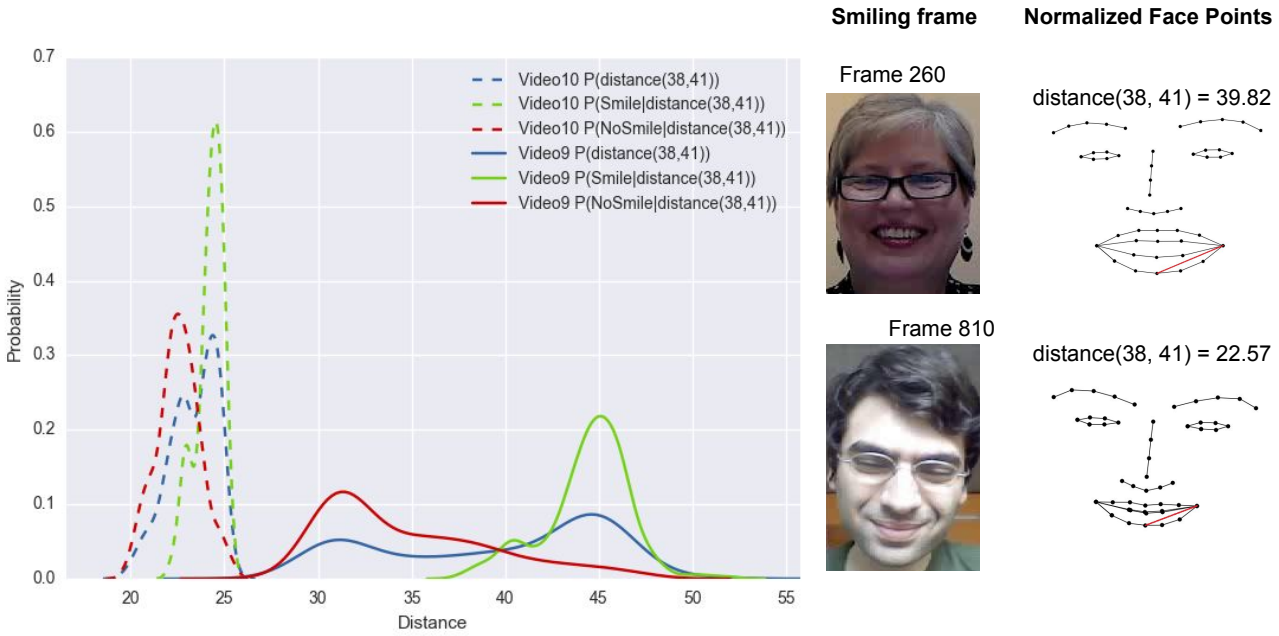


Figure 11.: Conditional and feature distribution for the distance between points 38 and 41 of Video 9(top) and Video 10(bottom).

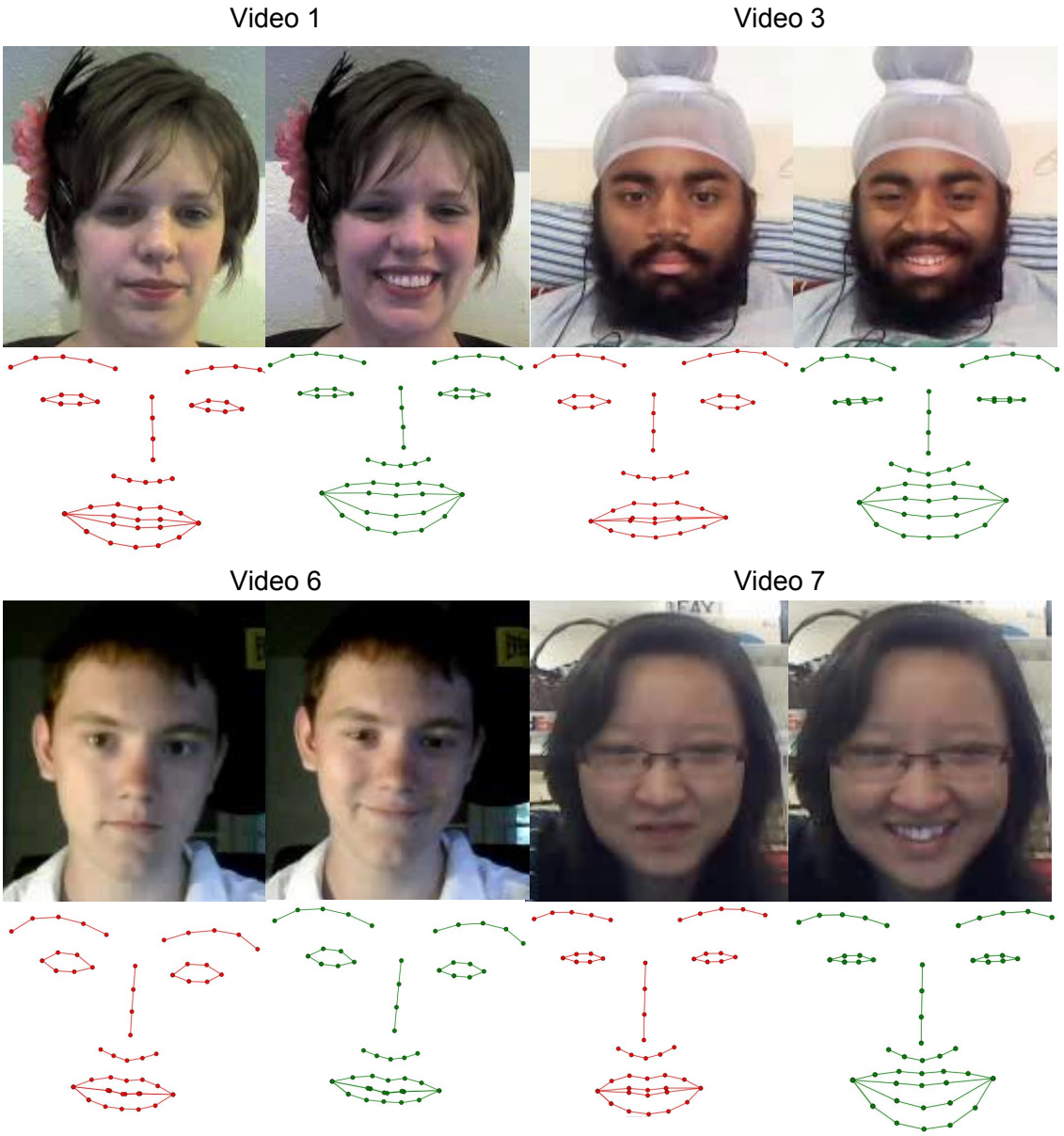
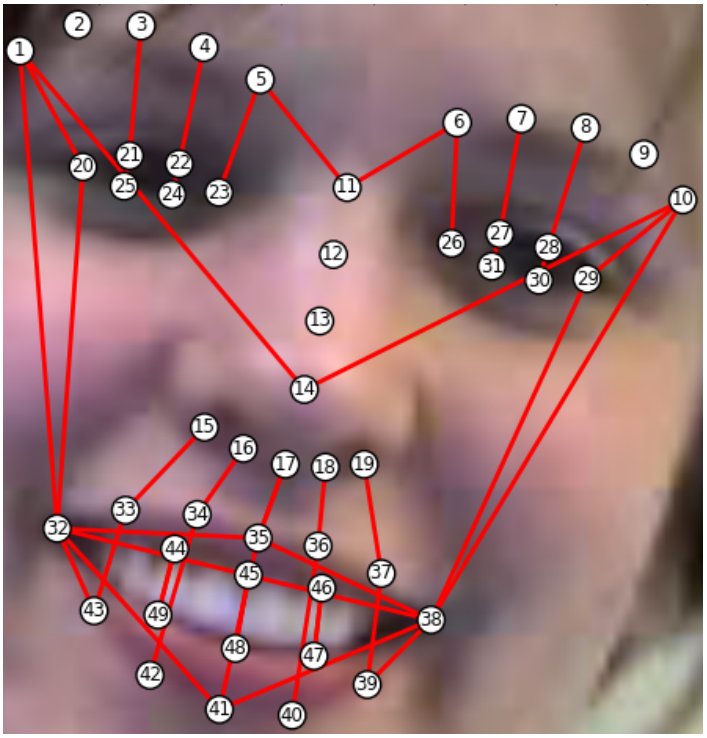


Figure 12.: No-smile(red) and smile(green) face comparison for Video 1, 3, 6 and 7.

We will use the following 5 feature sets:

- *Distances*: This set is based on distances between facial points. By looking at a few examples of smile vs neutral facial deformations in our AM-FED data set, see Figure 12, we created a small set of distance features which seem to capture the essence of distance changes during smiles, and of which the exact value seems to vary from face to face. If we compare the smiling frames with the neutral frames, in most cases the eyes contract, the upper lip moves closer to the nose, the mouth opens, the mouth shape elongates, and in some cases the mouth corners pull upwards relative to the rest of the mouth (Video 1 and Video 7). From this, we gathered the candidate distance features, presented in Figure 13 accompanied with a table of the face point numbers. The eyebrow distance features were added for good measure.

- *Angles*: A feature set that calculates the angle between a line segments of face points and the x-axis. Figure 14 shows the features used in this data set. In this feature set We

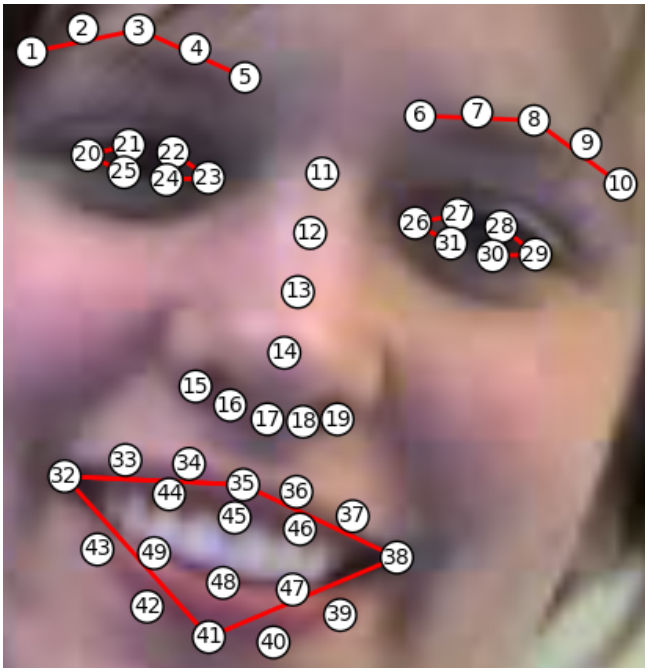


Distance		Distance		Distance	
p_1	p_2	p_1	p_2	p_1	p_2
1	32	35	41	34	16
10	38	36	40	35	17
32	38	37	39	36	18
35	17	21	25	37	19
48	45	22	24	32	35
49	44	27	31	38	35
32	43	28	30	32	41
38	39	20	1	38	41
45	48	21	3	32	20
46	47	22	4	38	29
5	11	23	5	34	42
6	11	26	6	33	15
1	14	27	7	33	43
10	14	28	8	29	10

Figure 13 & Table 1: Distance features used for feature set *Distances*

are mostly interested in the angular orientation of the mouth corners, the eyes and eyebrows. The orientation between line segments in the mouth and the x-axis provides us with the orientation of the mouth corners with respect to the upper and lower lip, which is more difficult to model with only distances, but can be a telltale sign of someone smiling due to cheek raising[15]. The angules that we calculate for the eyes are inspired by [14], where a similar approach is used to determine the openness of the eyes during fake and real smiles. By checking the angle with the x-axis for both the upper and lower eyelid, we can distinguish between blinking, which is caused mostly by the upper eyelid, and the contraction of only the lower lid that actually indicates smiling [15]. We use the x-axis as a frame of reference rather than the head orientation, as we believe our image normalization aligns eyes and mouth about level with the x-axis. Again, the addition of angular data on the eyebrows is not something we expect specific values of, but might help distinguish between faces that do not contain smiles, but rather other emotions that might be confused for smiling behaviour that occurs in our data.

- *Orientation*: A smile can be paired to a specific head orientation, especially when the smile originates from intense emotions; for example throwing back your head in a fit of laughter. We create an orientation feature set that only consists of the head pitch, roll and yaw, as extracted by Chehra to determine the role of orientation for individualized smile detection. We expect, however, to not get much performance out of this data alone, especially since in both our AM-FED and GENKI data set each of the participant have their focus set on a fixed target, the camera in GENKI, and the video fragment in AM-FED.



Angle w.r.t x-axis		Angle w.r.t x-axis	
p_1	p_2	p_1	p_2
32	41	26	27
32	35	28	29
38	41	26	31
38	35	30	29
20	25	1	3
23	24	3	5
20	21	6	8
22	23	8	10

Figure 14 & Table 2: Angle features used for feature set *Angles*

- *Combined*: Contains the above three feature sets, merged together in a single feature set. We expect an improved result with regards to the above three sets, given the novel information that each of the sets contains.
- *All*: The distance and angle extracted between all pairs of face points, together with head orientation. This will contain a lot of redundant information, like the distance between different nose points, which might cause a negative effect of the transfer learning. We keep it around to check the performance of our manually chosen feature set in comparison to one that does not explicitly consist of features that seem to be useful for transfer learning.

4.3 TRAINING PROCESS

Now that we have established what features to use, and how we extract these from the initial data, we can talk about the training process. To explain the application of transfer learning for our use cases, we first talk about the traditional training process using a regular SVM, as transfer learning adds an extra step to this process.

4.3.1 *Traditional Machine Learning Using SVM*

An SVM is a supervised learning method that performs binary classification. Figure 15 shows the general process of training and classifying using an SVM. We start with auxiliary data set $\mathcal{D}^a$ which in this example consists of images from the GENKI data set. The figure visualises a few instances of GENKI, and their corresponding ground truth label, either *smile* ($y = 1$) or *no-smile* ($y = 0$). We perform the feature pipeline discussed in the previous chapter for one of the chosen feature sets, and this returns the feature data per instance in the GENKI set (X^a). This becomes the input for the SVM, together with the corresponding

ground truth labels Y^a, which will learn a decision boundary between the two different classes of labels. In this stage, the classifier is ready to classify unseen data. When using the classifier for classification purposes, we expect to receive unlabeled data from a different source $\mathcal{D}_u^p$ than was used for training, of which we want to know the correct classification of. In Figure 15, $\mathcal{D}_u^p$ consists of a single image from Video 1 as an example target set. The same feature pipeline is then applied to $\mathcal{D}_u^p$, as was done for $\mathcal{D}^a$, with the same set of features, and with the same principal components removed during the rotation normalization of the face. The SVM receives the feature data, and outputs the most likely label according to the SVM for each instance. Figure 15 shows the output $y = 0$, in this case the SVM incorrectly predicted that no smile was depicted, while the ground truth indicated a smile was present in the frame. How the SVM makes this classification, will be discussed next.

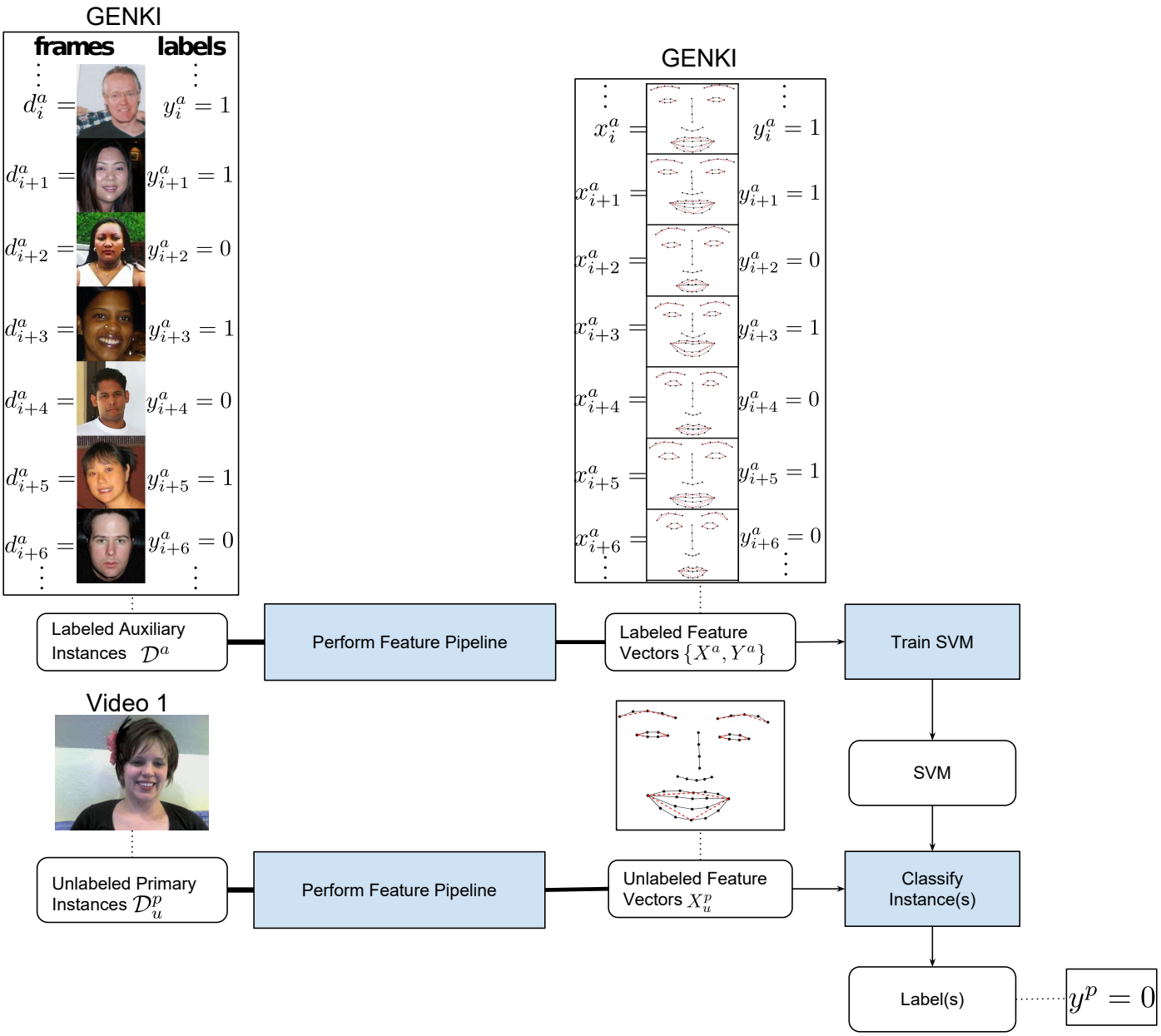


Figure 15.: Steps for performing supervised machine learning by means of an SVM, resulting in an unesired labeling.

4.3.2 *Support Vector Machine*

An SVM[6] is a binary classifier that sets out to find a decision boundary that maximizes the margin between two classes, i.e. the closest instances of each class should be as far away to the decision boundary as possible. The classification is made by the decision function $f(\vec{x}) = \vec{w}^T\vec{x} - b$, where $\vec{x}$ is the feature vector to be classified, $\vec{w}$ is the vector normal to the separating hyperplane, $\frac{b}{||\vec{w}||}$ is the offset to the origin from the hyperplane along $\vec{w}$, and the sign of $f(\vec{x})$ indicates the class to which the feature vector belongs according to the SVM.

In the case that the two classes are linearly separable, we can choose the parameters of the hyperplane $\vec{w}^T\vec{x} - b = 0$ by setting the constraint that all instances should be correctly classified, i.e. $\forall y_i y_i(\vec{w}^T\vec{x}_i - b) \geq 1$ where $y_i \in \{-1, 1\}$. Additionally, we want the distance between the decision boundary and the closest instances of each class, i.e. the support vectors, to the boundary to be maximal. By solving

$$\min_{\vec{w}} \frac{1}{2}||\vec{w}||^2$$

with the above mentioned constraint, we can find the decision function with the largest minimal margin using, for instance, quadratic programming techniques.

For real world problems maximizing the margins between the support vectors and assuming perfect separability of the two classes is a naive approach to take; it is likely that both classes contain outliers that do not represent the overall shape of the class distribution. In this case, using $\forall y_i y_i(\vec{w}^T\vec{x} - b) \geq 1$ as a hard constraint will cause an overfit of the data, or might even be impossible to adhere to in case the data is not linearly separable at all. To cope with this, the equation to solve for is changed to:

$$\min_{\vec{w}} \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{N} \varepsilon_i \tag{1}$$

The first term is the same as before, and tries to maximize the distance between the decision boundary and the support vectors. The second term is added to minimize the errors made by the SVM. The value of ε_i is the classification error made on instance $\vec{x}_i$ of the training data set of size N, calculated by $\varepsilon_i = max(0, 1 - y_i(\vec{w}^T\vec{x}_i - b))$. The value of ε_i does not go below 0, so as not to penalize instances that are classified on the right side of the decision boundary. The term C is added to regulate between the maximizing of the margin to the support vectors, and the resulting classification errors from that decision function. When a large value is chosen for C, it will minimize the error but might overfit the decision function by valuing outliers too much.

When the data is not linearly separable, insisting on a linear decision boundary can give bad results. A possible solution is to map the feature data using a function ϕ, both during training and classification, into a higher dimension in which the data is linearly separable, or improves linear separability. Because mapping each instance to a higher dimension can become computationally expensive, SVMs apply a kernel 'trick', rather than applying ϕ directly on all instances. Applying a kernel $k(\vec{x}_i, \vec{x}_j)$ does not explicitly map each of the instances to the higher feature space, but when applied to two feature vectors, it returns the same result as taking the dot product in higher order, i.e $k(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$. Part of solving Equation 1 using quadratic programming consists of taking the summation of the dot products between all combination of instances. By substituting this dot product for the kernel instead, the classifier considers the data in the desired elevated dimension, without explicitly using ϕ.

In our research we use a Radial Basis Function (RBF) as the kernel, which maps the data using a Gaussian model. It is considered a good first kernel choice when the data contains more instances than features [19], which is the case for most of the classifiers that we will train. When using an RBF Kernel, we have two parameters to decide on, namely C, the value that is used to trade-off between classification error and possible overfitting of the hyperplane, and a value for γ that is specific to the RBF and determines the influence of instances. For a high value of γ, training instances having a small reach, possibly causing the decision function to overfit to the points nearest to the decision boundary. For a low value of γ the training instances will have a large reach, which causes the decision boundary to consider the general shape of the data sets more[1]. Choosing the value for C and γ is often done using a grid-search, where various combinations of the two parameters are trained using cross-validation and the best performing values are chosen for the final classifier. This is the route that we will take as well, as will be explained in Section 4.4. To implement our SVM we use *LIBSVM* by [8].

Lastly, when the data the SVM is trained with is not well balanced, it might cause the classifier to return the most likely class, no matter the feature value of the to-be-classified instance. Even though our data sets are not extremely unbalanced we take preventive measures by setting the C value separately for a class in relation to how many instances there are. For instance, if there are 5 times less amount of instances for label $y = 0$, its C value becomes 5 times as high respective to the $y = 1$ instances. This means the classifier will penalize mistakes 5 times as much for negative labels. LIBSVM has a built-in method[2] for this operation which we use to balance the data.

4.3.3 *Transfer Learning using A-SVM*

Figure 16 shows the process of applying inductive transfer learning by means of an A-SVM. Most of the process is identical to the one shown in Figure 15; the newly added steps are located in the colored block. The traditional case uses two different sets of data; the train data $\mathcal{D}^a$ and the target instance(s) $\mathcal{D}_u^p$ for which classification is desired. The inductive transfer learning step adds a second training set $\mathcal{D}_l^p$ that contains manually labeled instances of the same face as the target face that we want to classify (Video 1 in this example). The features are computed from these instances, and become the input of the A-SVM classifier, together with the SVM trained on $\mathcal{D}^a$. The A-SVM then trains a new classifier; adapting the decision function in the SVM to take into account the instances in $\mathcal{D}_l^p$ as to improve the performance on $\mathcal{D}_u^p$.

4.3.4 *A-SVM*

In Section 4.3.2 we have discussed the general idea of how SVMs are trained. Let us refer to the decision function trained in the SVM step as $f^a(x)$. Now, the decision function in the adaptive case adds an extra term to $f^a(x)$ [40]:

$$\begin{aligned} f(\vec{x}) &= f^a(\vec{x}) + \Delta f(\vec{x}) \\ &= f^a(\vec{x}) + \vec{w}^T\vec{x} - b \end{aligned} \tag{2}$$

1 http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html
2 https://www.csie.ntu.edu.tw/ cjlin/libsvm/faq.html#f410

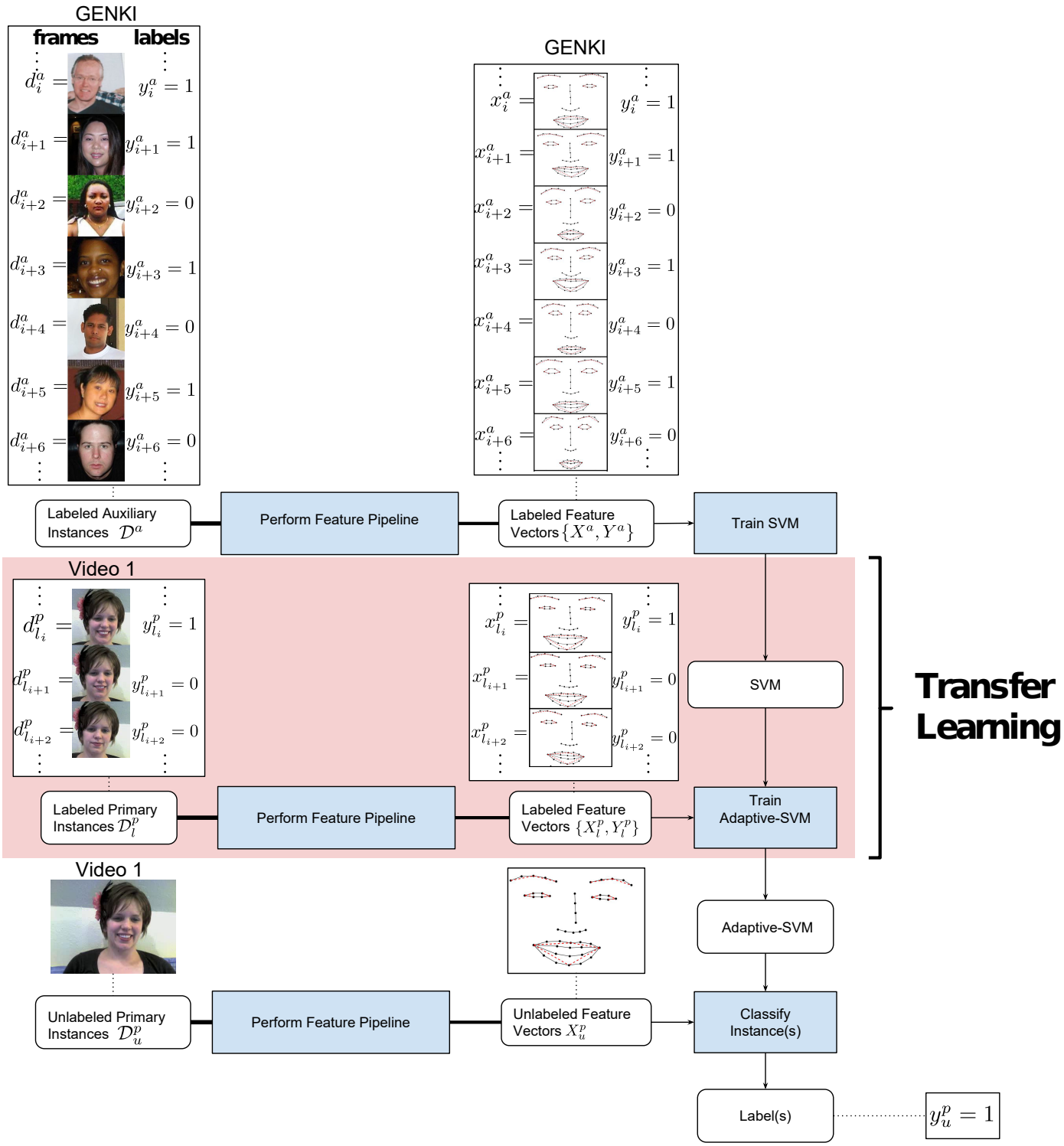


Figure 16.: Steps for performing inductive transfer learning by means of an A-SVM.

Here, the values for $\vec{w}$ and b are different from the parameter values learned in $f^a(\vec{x})$. In essence, $f(\vec{x})$ consists of a single decision boundary constructed from two separate decision boundaries, of which $f^a(\vec{x})$ only trained on $\mathcal{D}^a$, and $\Delta f(\vec{x})$ only trained on $\mathcal{D}_l^p$.

To find the values $\vec{w}$ and b for $\Delta f(x)$, the same equation is solved as before (Eq. 1), but with a new unrelated value C, that now determines the influence of the newly introduced data $\mathcal{D}_l^p$,and with a different meaning for ε_i. Previously we saw that the error consisted

of the distance to the decision boundary for wrongly classified instances. In the adaptive SVM, this changes to include the boundary error made by $f^a(x)$ as well, i.e:

$$\begin{aligned} \varepsilon_i &= max(0, 1 - y_i f^a(\vec{x}_i) - y_i \Delta f(\vec{x}_i)) \\ &= max(0, 1 - y_i f(\vec{x}_i)) \end{aligned} \tag{3}$$

Here, $y_i f^a(\vec{x}_i)$ does not change throughout the learning process, as it is not influenced by the values learned for $\vec{w}$. As a result, when the value of C is high, the final classification of $f(\vec{x})$ will be influenced more by the decisions made by $\Delta f(\vec{x})$, as it tries to compensate each individual error in $\mathcal{D}_l^p$ made by $f^a(\vec{x})$, rather than adhering to the decision boundary learned by $f^a(\vec{x})$. When the value of C is low, the decision boundary of $f(\vec{x})$ will be similar to $f^a(\vec{x})$ [40].

The A-SVM has the ability to use multiple auxiliary data set. The final decision function is similar to Equation 3, but instead of having one auxiliary decision function $f^a(x)$, there are multiple of which the values are summed. In our research we only apply the single auxiliary case, as our research does not focus on experimenting with the contents of different auxiliary data.

4.3.5 *Choosing $\mathcal{D}_l^p$*

The data to consider for $\mathcal{D}_l^p$ depends on a few factors. First, we would like to reduce the number of instances as much as possible, as this requires manual effort from the user of the smile detector. Moreover, the assumption that not a lot of data is available is one of the main reasons to consider the transfer learning approach in the first place. Secondly, the data contained in $\mathcal{D}_l^p$ should be a good representation of the data that the classifier will be used for, i.e. $P_l^p(X) = P_u^p(X)$ and $P_l^p(y|x) = P_u^p(y|x)$. However, choosing $\mathcal{D}_l^p$ in this way is not trivial. Although in our case $\mathcal{D}^p$ is labeled and we thus can easily verify if $P_l^p(X) = P_u^p(X)$ for each proposed $\mathcal{D}_l^p$, when using the classifier in a real world setting no labels are available to test $P_l^p(y|x)$ against. For this reason we steer clear from matching the probability estimates of the labeled and unlabeled portions and test the performance of a more general approach.

An alternative approach that we could consider, is to use instances that the classifier was not very sure about while classifying, i.e. are close to the decision hyperplane. These instances have a higher probability of being wrongly classified. Although there is merit to this approach, we can not easily evaluate this in our research. If we purposely add instances from $\mathcal{D}^p$ to $\mathcal{D}_l^p$ of which the probability is high that these are wrongly classified, then the instances in $\mathcal{D}_u^p$ will most likely perform better, as most of the problematic data is not part of it anymore. Additionally, the data in AM-FED is not abundant enough to divide these on-the-edge label assignment in a unlabeled and labeled data set.

Just splitting up $\mathcal{D}^p$ into 10 folds and using cross-validation to check performance, ignores our assumption of having little data available, so the approach we take is a little different. We decide upon a number of smiling frames beforehand and the total number of instances to include into $\mathcal{D}_l^p$. A same approach was taken by Yang et al, who tested 1, 3, 5, and 10 positive instances for different trials, and 16 times as much negative instances for each of the trials. In our research, we use 75 instances in total for $\mathcal{D}_l^p$, of which 15 are labeled as smiles. We chose the data a bit more balanced compared to Yang et al., as the occurrence of our classes is less exclusive than in their case. Although Yang et al. construct $\mathcal{D}_l^p$ by randomly selecting instances from their data set until $\mathcal{D}_l^p$ is filled with the desired ratio of data, we decide against this due to the temporal nature of our AM-FED data. When picking

instances at random, $\mathcal{D}_u^p$ and $\mathcal{D}_l^p$ can consist of multiple consecutive frames of which the features are a lot a like, making it more likely that the classifier will over-fit the data, and seem to perform better than is actually the case. Instead, we use the following approach. Given $\mathcal{D}^p$, we apply a sliding window of size 75 on top of the temporally ordered frames. When the 75 frames do not contain precisely 15 smiling instances the window is incremented by a single frame. When it contains 15 instances, the instances inside the window are registered as a candidate $\mathcal{D}_l^p$, and the window jumps to the first subsequent frame that was not registered. Because not all consecutive smiling segments in the data will have 15 frames or more of smile data, we also register the frames in the window in case it contains one complete smiling patch. Algorithm 1 shows the algorithm with which candidates are found for $\mathcal{D}_l^p$. Figure 17 visualizes 3 consecutive steps of the algorithm in an example scenario.

Algorithm 1: Determining candidates for D_l^p

Data: y^p ; `// vector containing ground truth labels of` $\mathcal{D}^p$
Result: One or more candidates for D_l^p

1 $i \leftarrow 0$; `// start index sliding window`
2 **while** $i <$ `length`$(y^p) - 75$ **do**
3 $\quad y^p_{window} \leftarrow y^p[i \ldots i+75]$;
4 $\quad$ **if** `containsSmilePatch`(y^p_{window}) **then**
5 $\quad\quad$ Register $D^p[i \ldots i+75]$ as candidate D_l^p;
6 $\quad\quad$ $i \leftarrow i+75$;
7 $\quad$ **else if** `smileCount`$(y^p_{window}) = 15$ **then**
8 $\quad\quad$ Register $D^p[i \ldots i+75]$ as candidate D_l^p;
9 $\quad\quad$ $i \leftarrow i+75$;
10 $\quad$ **else**
11 $\quad\quad$ $i \leftarrow i+1$;
12 $\quad$ **end**
13 **end**

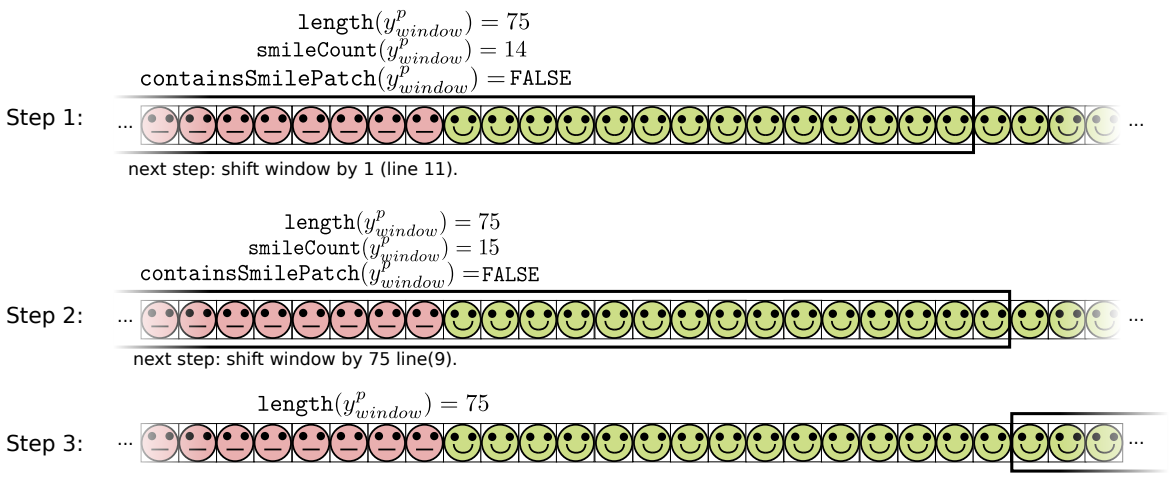


Figure 17.: Visualisation of sliding window on consecutive frame data . Step 1 corresponds to line 10 of Algorithm 1, Step 2 corresponds to line 7, and Step 3 corresponds to line 9

As Algorithm 1 shows, there are multiple candidates for $\mathcal{D}_l^p$; our assumption is that similar performance can be expected from any of these. The next section will describe how this will be verified during our tests.

A risk to mention of our approach is that the rather limited number of smiling instances that are extracted in a consecutive fashion might mean that $\mathcal{D}_l^p$ only consists of transitional data; it does not grab the full apex of a smile, rather only the few frames before or after the apex has set in, when transitioning to or from the smile. We assume that this is not a big problem, since the AM-FED smile labeling has been rather conservatively thresholded, only labeling a frame as a smile when at least 75 % of researchers agreed it contains a smile; making the frames labeled smile contain mostly the apexes.

4.4 TESTING

To determine the appropriateness of transfer learning for our objective of individualizing smile detection, we have to have some measure of performance. We recall research Sub-Question 2:

Sub-Question 2. *How does transfer learning compare to alternative approaches of individualized smile detection?*

Only comparing the performance of the A-SVM trained on $\mathcal{D}^a \cup \mathcal{D}_l^p$ with an SVM trained on only $\mathcal{D}^a$ does not give us a lot of insight. The A-SVM is at an advantage of having more data available, so we can expect performance is better (as long as no negative transfer occurs). For this reason we propose a few different evaluation trials, as have been proposed by Yang et al. In short, these are: (i) training on $\mathcal{D}^a$, testing on $\mathcal{D}^p$ (**aux**) (ii) training on $\mathcal{D}_l^p$, testing on $\mathcal{D}_u^p$ (**prim**) (iii) training on $\mathcal{D}^a \cup \mathcal{D}_l^p$, testing on $\mathcal{D}_u^p$ (**aggr**) (iv) training on $\mathcal{D}^a$, transfer step using $\mathcal{D}_l^p$, testing on $\mathcal{D}_u^p$ (**adapt**). The first three evaluations are traditional machine learning, where just the data sets for training differ. Only the last one consists of the extra transfer learning step. Each of these strategies will be used with each of the five feature sets covered in Section 4.2.3 (*Distances*, *Angles*, *Orientation*, *Combined*, *All*).

4.4.1 *Data Division*

For each of the different strategies, we will need to divide our data into train and test sets. We will refer to these sets as auxiliary and primary sets respectively, to keep it similar to the transfer learning literature.

We construct 10 different primary sets; each being one of the 10 selected and processed AM-FED videos as discussed in Chapter 3. We use them as separate data sets so that $\mathcal{D}_l^p$, which will be used during transfer learning, consists only of a single new face, as to optimally test our improvement to individualized smile detection. Utilising 10 different sets also helps us generalize our findings over multiple distinct faces.

For the auxiliary data set, we use the GENKI data. The GENKI data set can provide us insight how transfer learning influences performance when the auxiliary data consists of a big amount of varied data. We expect a rather good baseline performance from a classifier trained on GENKI alone. In case transfer learning can improve performance on a never before seen face when GENKI fulfils the function of $\mathcal{D}^a$, than there is evidence that transfer learning is successful in making the classifier 'adapt' to a new face.

4.4.2 *Strategies*

4.4.2.1 *Aux*

In the *aux* strategy we learn an SVM using auxiliary data only, and test its performance using the primary data. This boils down to a traditional machine learning approach as explained in section 4.3.1. The *aux* strategy establishes a baseline of how much performance can be gained when considering an individualized approach.

Figure 18 shows what the *aux* process is like during evaluation. This strategy will be applied using GENKI as the auxiliary data and each of the 10 AM-FED videos as separate primary test sets. This will result in 50 different scores, as each combination is tested with each of the 5 different feature sets. The resulting F1-score calculated from our predicted classification and ground truth labels will indicate performance in each case, giving us a baseline in performance for each of the different feature sets and videos.

4.4.2.2 *Prim*

The *prim* strategy consists of dividing up the primary data in a set $\mathcal{D}_l^p$ for training and $\mathcal{D}_u^p$ for testing. The auxiliary data is left out, to see how well performance is without any additional information available to support the limited data in $\mathcal{D}_l^p$. In *prim* we should split up the data in the exact same way as will be done during transfer learning, to keep conditions the same. Section 4.3.5 explained our strategy for dividing up the data, and is what we will use to divide each of our primary data sets. This approach generates multiple candidates for $\mathcal{D}_l^p$, which we will all test. The *prim* diagram shown in Figure 18 will thus be applied during evaluation for each of the 10 primary data sets, and for the each set of $\mathcal{D}_l^p$ and $\mathcal{D}_u^p$ that was generated for each of these primary data sets. For instance, when we want to evaluate the *prim* strategy for Video 1, we run Algorithm 1 to get all possible candidates for $\mathcal{D}_l^p$. For Video 1 we find 6 of such candidates for the total video. Each of these candidates are then used to apply the steps in Figure 18, where $\mathcal{D}_u^p = \mathcal{D}^p - \mathcal{D}_l^p$. We thus receive an F1-score for each $\mathcal{D}_l^p$, so for Video 1 6 scores in total, which we report the average and standard deviation of. Our expectations for *prim* are that this amount of data is not enough, and that the classifier benefits from a larger auxiliary data set to improve its performance, even though this auxiliary data does not originate from the same face.

4.4.2.3 *Aggr*

The *aggr* strategy data set combines an auxiliary data set and a candidate for $\mathcal{D}_l^p$ from which a single traditional classifier is learned, and evaluated on the corresponding $\mathcal{D}_u^p$. The approach of Yang et al. utilises a weight for the auxiliary and primary data during learning, so that classification errors in $\mathcal{D}_l^p$ are considered more significant. This way, the performance is improved on the data that actually matters. We mimic this idea by up sampling $\mathcal{D}_l^p$ to be a more prominent size of the training set. In our evaluation we will use a ratio of $2:1$ for auxiliary and primary data respectively.

4.4.2.4 *Adapt*

The *adapt* strategy is the only configuration that uses transfer learning, and its performance will indicate the value of transfer learning for individualized smile detection. Each primary video will be split up between $\mathcal{D}_l^p$ and $\mathcal{D}_u^p$ as in the *prim* strategy. Then, each of the $\mathcal{D}_l^p$ candidates is paired up with the GENKI auxiliary set. The F1-score is again calculated over

all trials where different $\mathcal{D}_l^p$ candidates are used. Our expectation is that the *adapt* strategy will perform better than just using *prim* or *aux*, and possibly similar to the *aggr*.

4.4.3 *Parameters Across Training*

Each of the different strategies trains a new classifier. To make sure that results are comparable, the values for C and γ should be chosen consistently. We do that as follows: All (Adaptive) Support Vectors are trained with an RBF Kernel. We determine a single optimal value for γ for each of the 5 different feature sets; so 5 different values for γ. These values are kept consistent over the different evaluation strategies. The value of C for the SVMs is learned specific for each combination of auxiliary data set and feature data, so again 5 different values for C. These learned values for C are used consistently in the modes where the SVMs are used, thus strategy *aux*, *aggr* and *adapt*. The value of C for the primary data, so each combination of AM-FED and feature set, is also learned during cross validating, giving 50 different values for C, and the same C values are applied during the *prim* and *adapt* settings.

The values for γ and C are learned by performing a grid search, that tries all combinations of values for the two parameters during training, and selects the one with highest accuracy over 5 folds. In our research we set the bounds as follows:

$$C \in \{2^{-5}, \ldots 2^{15}\} \times \gamma \in \{2^{-15} \ldots 2^{3}\}, stepsize = 2 \tag{4}$$

To learn the values for γ, we evaluate the accuracy using 5-fold cross-validation on each combination of feature set and GENKI data set only, as we believe other data sets, like the individual AM-FED video will have the tendency to overfit the data more and decrease performance on other data sets. As the GENKI performance for smile detection is our baseline, being able to surpass its performance even when parameters are optimized for it would only help confirming our hypothesis. After the optimal γ parameters are selected, the value of γ becomes fixed in other grid searches for determining the optimal value of C. The grid search is performed using the grid-search tool provided as part of LIBSVM.

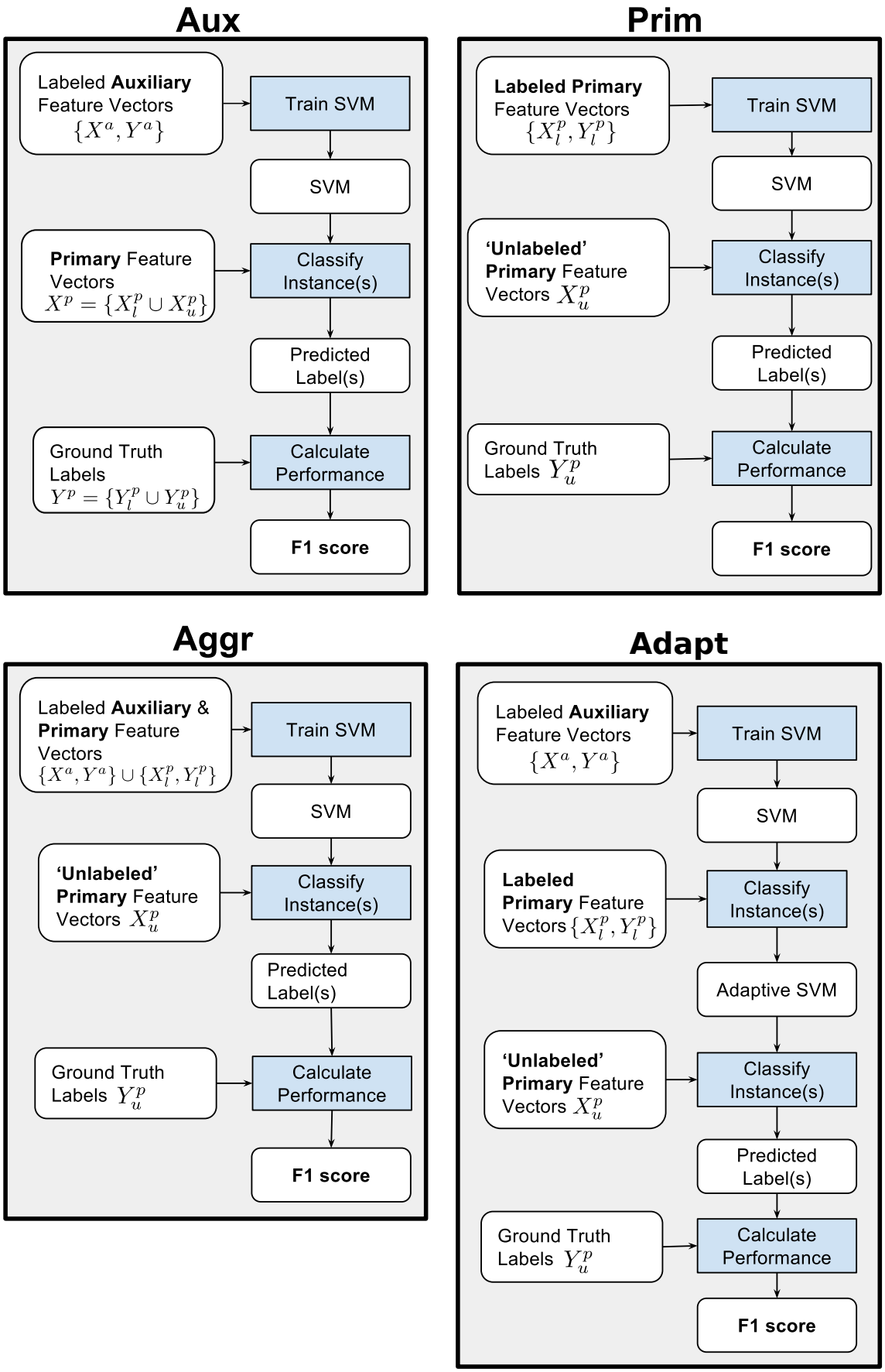


Figure 18.: The 4 different test strategies used for evaluation.

5

EVALUATION & RESULTS

Recall that the goal of our research consists of answering the following research question:

Research Question. *Can transfer learning improve the detection rate of spontaneous smiles when applied on a generically trained classifier trained with geometric features?*

We aim to find the answer with the support of the following two research questions:

Sub-Question 1. *What geometric features prove effective in personalized smile detection?*

Sub-Question 2. *How does transfer learning compare to alternative approaches of individualized smile detection?*

Research SQ 1 has led us to explore different low-level features, of which we have ended up with 5 different feature sets to evaluate: *Distances*, *Angles*, *Orientation*, *Combined* and *All*.

Research SQ 2 has resulted in four different evaluation strategies that are deemed to give a good perspective on the actual benefit of transfer learning: *aux*, *prim*, *aggr*, and *adapt*. In this section each evaluation strategy and feature set is combined to aid in answering the overarching research question. First we discuss the results of these strategy and feature sets. Afterwards we discuss possible improvements for selecting $\mathcal{D}_l^p$ and lastly we use the discussed findings to recap and answer the research questions.

5.1 PARAMETER CHOICE

Before the evaluation can take place, we perform trials to find the optimal parameters to use during training and testing. This commences with determining the best performing γ for each feature set using the GENKI data. Figure 19 shows the mapped out search space when performing a grid search using the combination of values from Equation 4.

For the *Distances*, *Combined* and *All* feature sets a (local) maximal accuracy is not completely contained in the search grid, and in all cases could have been found by lowering the minimum value of γ and increasing the maximum value of C. We refrain from doing this, as the accuracy currently found is deemed satisfactory and the pay off would be rather limited for even more extreme values of γ.

The optimal parameters and resulting accuracy of these trials are shown in Table 3. We report the accuracy rather than the F1-score as the grid-search tool bundled with LIBSVM uses accuracy for determining the best performing parameters. These reported parameters are used for all subsequent evaluations in *aux*, *aggr*, and *adapt*.

Now that an optimal value of γ is determined for each of the different feature sets with the GENKI data set, we fix this value to the optimal γ in further grid searches. Table 4 shows the grid search for C for each of the primary data sets, that will be applied during the *prim* and *adapt* strategy. The grid search for the primary set has in most cases resulted in a maxed out value $C = 2^{15}$. Although this might indicate a tendency to overfit all instances of the

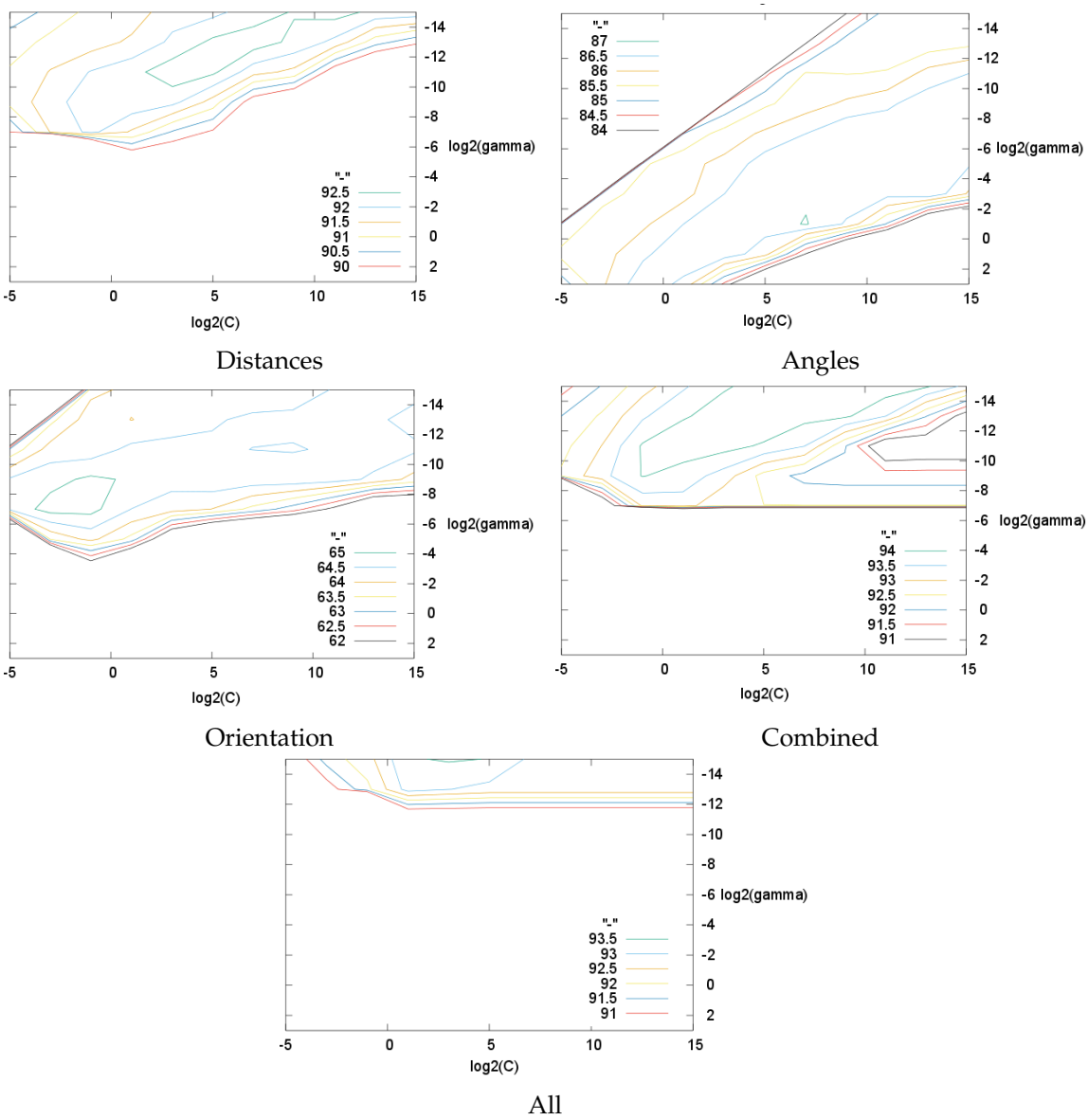


Figure 19.: Grid search results for the different feature sets on Genki Data.

Table 3.: Result of grid-search with k-fold cross-validation with $k = 5$ on the GENKI data set.

	Distances	*Angles*	*Orientation*	*Combined*	*All*
C	2^3	2^7	2^{-3}	2^5	2^3
γ	2^{-11}	2^{-1}	2^{-7}	2^{-13}	2^{-15}
Accuracy	92.811%	87.0402%	65.2042%	94.465%	93.554%

data, it might actually prove beneficial during the *adapt* strategy, by having the decision function of f^a be more heavily influenced by D_l^p. Overall, the performance of the different feature sets are promising. For each of the data sets *All* and *Combined* perform similarly as expected; with a maximum of 2% difference. The *Combined* feature set performs consistently better than *Distances*, *Angles* and *Orientation*, which seems to indicate that these data sets act complementary when combined. However, to answer research SQ 1 we will have to check feature performance over the different strategies.

As a last remark, seeing that the performance per primary data set is rather high when there is a large amount of data (as we see now during cross-validation) it provides us with a good base for testing the *prim* strategy, to see how lower amounts of data influence performance, and how precisely transfer learning can help here.

Table 4.: Result of k-fold cross-validation with $k = 5$ on Primary data sets.

Data set		Feature Sets					Average
		Distances	*Angles*	*Orientation*	*Combined*	*All*	
Video1	C	2^{13}	2^{15}	2^{11}	2^{15}	2^9	
	Accuracy	94.46%	92.838%	89.054%	96.216%	95.946%	93.703%
Video2	C	2^{15}	2^{11}	2^{15}	2^{15}	2^{11}	
	Accuracy	87.147%	81.975%	76.175%	87.931%	89.969%	84.639%
Video3	C	2^{15}	2^{15}	2^{13}	2^{15}	2^9	
	Accuracy	83.155%	74.198%	88.235%	95.187%	95.989%	87.353%
Video4	C	2^{15}	2^7	2^{15}	2^{15}	2^9	
	Accuracy	94.927%	89.853%	85.682%	95.378%	96.505%	92.469%
Video5	C	2^{15}	2^{15}	2^9	2^{15}	2^{11}	
	Accuracy	90.81%	86.82%	81.741%	92.261%	92.745%	88.875%
Video6	C	2^{15}	2^{15}	2^{11}	2^{15}	2^9	
	Accuracy	79.68%	74.772%	80.48%	91.324%	93.151%	83.881%
Video7	C	2^{15}	2^{15}	2^3	2^{15}	2^9	
	Accuracy	86.996%	81.128%	85.27%	92.98%	91.715%	87.618%
Video8	C	2^9	2^{15}	2^{15}	2^{11}	2^7	
	Accuracy	98.396%	97.594%	94.96%	98.74%	98.625%	97.663%
Video9	C	2^{15}	2^{15}	2^{15}	2^{15}	2^7	
	Accuracy	89.952%	83.852%	79.785%	95.454%	94.019%	88.612%
Video10	C	2^{13}	2^{15}	2^3	2^{15}	2^{11}	
	Accuracy	94.851%	96.453%	94.05%	96.682%	98.627%	96.133%
Average		90.038%	85.948%	85.543%	94.215%	94.729%	

5.2 AUX STRATEGY EVALUATION

Recall that the *aux* strategy builds a classifier from the auxiliary data only, and tests this on the primary data (Figure 18). Table 5 shows the result of the auxiliary data trained using the choice of parameters set out in Table 3, and evaluated on each of the AM-FED videos (i.e. our primary data). A few things come to attention immediately. Firstly, for each video, all of the feature sets perform worse than in the GENKI cross-validation results (Table 3) and in the primary data cross-validation results (Table 4). This is expected to some degree; although in both the cross-validation and the *aux* case the evaluation consists of unseen data, the aux data consists of only a singular type of smile; in case this does not suit the classifier trained on GENKI, the hit-rate will be consistently lower than when evaluating on multiple random faces displaying multiple smile-styles. This in turn also explains the higher fluctuation rate between different feature sets and primary videos in the *aux* strategies. Video 7 contains a open-mouth style smile (refer to Figure 5) that is rather typical for the GENKI data set as well, while Video 6 only shows a slight smirk on the apex, which is not caught by GENKI as it does not seem to capture subtle different between the two classes.

Although the performance per face fluctuates a lot, the performance of each feature set stays relatively similar with respect to the performance of the other features set: *Combined* and *All* perform similar and better than *Distances* and *Angles*, indicating that smile traits learned in GENKI in some way relate to each of the AM-FED videos. The odd one out seems to be the *Orientation* feature set, which fluctuates between being the worst and best performing feature set between each of the video. On closer inspection the F1-score of this feature set is rather misleading. The classifier always returns the label *smile* for Video 4, Video 5 and Video 10, which makes the classifier useless and the scores have been set to zero. For Video 2 and Video 7 the biggest majority was also labeled as a *smile*, which makes the F1-score assigned for those videos less meaningful as well. The unreliability of this feature set was as expected; having only head-orientation data is too limiting of a factor on its own, although it might very well help in the *Combined* set. For this reason we omit reporting on this feature set in the next strategy evaluations.

Table 5.: *aux*: F1-score of GENKI classifiers on AM-FED videos.

	Feature Sets					
Primary Data	*Distances*	*Angles*	*Orientation*	*Combined*	*All*	Average F1-score
Video1	0.278	0.251	0.363	0.356	0.323	0.314
Video2	0.573	0.719	0.685	0.678	0.728	0.677
Video3	0.625	0.513	0.797	0.69	0.677	0.66
Video4	0.588	0.25	0.0	0.396	0.517	0.35
Video5	0.667	0.67	0.0	0.69	0.692	0.544
Video6	0.0	0.0	0.273	0.047	0.297	0.123
Video7	0.871	0.868	0.739	0.866	0.829	0.835
Video8	0.367	0.398	0.09	0.372	0.368	0.319
Video9	0.745	0.812	0.364	0.804	0.778	0.7
Video10	0.613	0.3	0.0	0.471	0.439	0.365
Average F1-score	0.533	0.478	0.331	0.537	0.565	

To get a better insight into the classification errors, we look at the generated probability estimate for each feature set. The probability estimate is a score from 0 to 1 that indicates

how likely an instance belongs to one of the two classes. In normal circumstances, an SVM outputs a discrete value that indicates the side of the margin the instance lies on. It is possible to output a probability estimate for each instance instead. To do so, one can consider the distance of the object to the margin instead of only the side of the hyperplane it is on. Platt uses this value as the input for a parametrized sigmoid function to retrieve a probability estimate[29] between 0 (lowest probability the instance shows a smile) and 1 (highest probability instance shows a smile). This is the option we use for generating the probability estimates, and is readily available in the LIBSVM library.

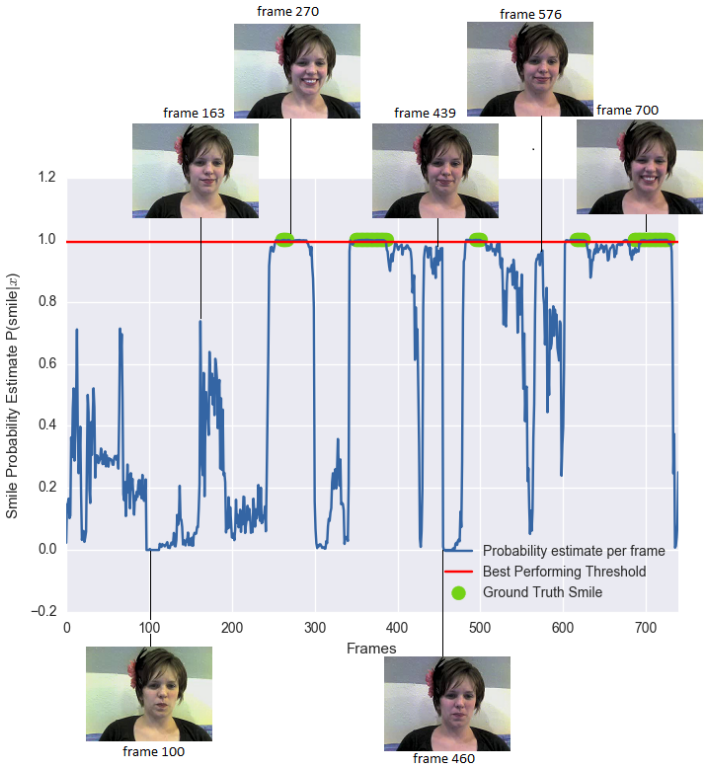
We look at the probability estimate generated for a few of these AM-FED videos. Figure 20 shows the probability estimates generated by the *Combined* Genki classifier for AM-FED Video 1, 6, 7 and 10. The plotted probability estimates are accompanied by a couple of highlighted frames which were assigned a high, low or in the middle probability estimate. The red line indicates the threshold that would result in an optimal F1-score for each of the videos. This is calculated by selecting each occurring probability estimate from each of the classified instances and using it as the discriminating threshold between smiles and non-smiles. The red line indicates the threshold that would result in the most correct classifications.

For Video 1, the probability estimates are maxed out around patches that contain ground truth smiling data, and the lowest scores show distinct non-smiling faces. This indicates that the classifier is able to distinguish between the apex smile and very light, closed mouth smiles, but the classifier is not discriminative enough. The difference in label style in in GENKI and AM-FED seems to play a role in this. For instance, frame 163, 439, and 576 arguable depict a smile, but were not labeled as such by the majority of labelers of AM-FED. If a classifier can get a better sense of the labeling-style used in the primary data using transfer learning and as a result only contain the actual apex as part of the smile, this might be positively influence the performance. The same observation can be made for Video 8; it also matches out on the ground truth smiles, but also triggers some false positives which arguably depict an apex smile.

For video 6 and 10 a main issue seems to be noise; consecutive frames jump around a lot in value. For Video 6 almost none of the frames is actually detected as a smile, as its traits are very subtle. It is expected that transfer learning might smooth out these plots, as the classifier can learn that those minor changes are not indications of smiling behaviour. It seems unlikely that this jumpiness is the result of noise in face point placement that can not be distinguished from the subtle smiles depicted, as a 91.324% accuracy was achieved during cross validation for this Video (see Table 4)

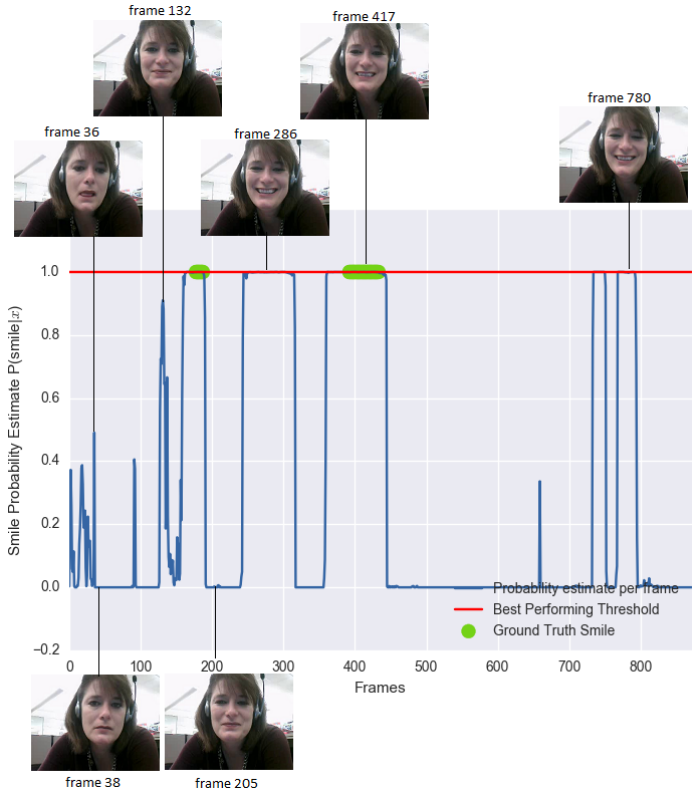
5.3 PRIM STRATEGY EVALUATION

In the *prim* case, a section of each of the AM-FED videos is selected for training using the strategy discussed in Section 4.3.5. We assume that its performance will be lacking, as it is not certain that the small number of selected instances will be a good representation of smile behaviour over the whole video and there is no additional data that provides a general decision function for smile detection. On a first glance Table 6 confirms our expectations. Although we draw data from the set we evaluate on, the performance is in most cases worse than using the auxiliary data. Table 7 highlights this difference in performance by subtracting the average F1-score from the *prim* strategy to those reported in the *aux* case. Only Video 1 and Video 6 perform better. The fact that for Video 6 using only primary data for classification improves overall classification performance is promising for the impact that transfer learning might have.

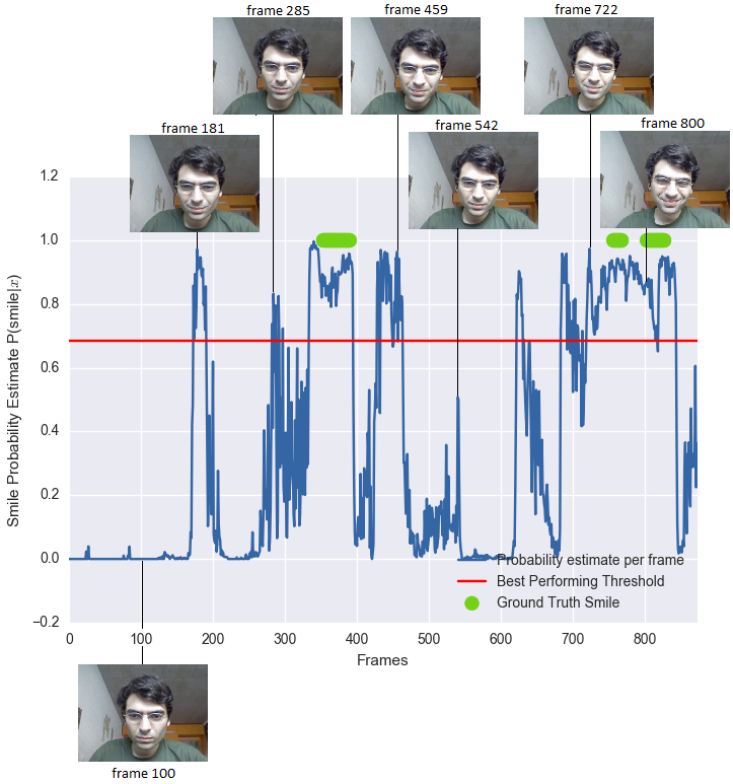


(a) Video 1

(b) Video 6



(c) Video 8



(d) Video 10

Figure 20.: Probability estimates of videos 1, 6, 8 and 10 of AM-FED data set using *aux*.

Table 6.: *prim*: Average F1-scores and their standard deviation using only primary data for training.

Primary data	Feature Sets				Average F1-score
	Distances	*Angles*	*Combined*	*All*	
Video1	0.446 ± 0.13	0.357 ± 0.21	0.504 ± 0.15	0.523 ± 0.1	0.457 ± 0.17
Video2	0.301 ± 0.24	0.421 ± 0.07	0.446 ± 0.24	0.295 ± 0.19	0.366 ± 0.21
Video3	0.288 ± 0.17	0.38 ± 0.15	0.553 ± 0.23	0.284 ± 0.15	0.376 ± 0.21
Video4	0.213 ± 0.08	0.212 ± 0.1	0.257 ± 0.04	0.27 ± 0.08	0.238 ± 0.08
Video5	0.304 ± 0.14	0.312 ± 0.11	0.457 ± 0.16	0.278 ± 0.17	0.338 ± 0.16
Video6	0.178 ± 0.06	0.199 ± 0.09	0.275 ± 0.14	0.18 ± 0.08	0.208 ± 0.11
Video7	0.603 ± 0.15	0.543 ± 0.19	0.785 ± 0.08	0.638 ± 0.15	0.642 ± 0.17
Video8	0.222 ± 0.12	0.244 ± 0.04	0.172 ± 0.11	0.358 ± 0.12	0.249 ± 0.12
Video9	0.567 ± 0.13	0.405 ± 0.2	0.561 ± 0.12	0.54 ± 0.12	0.518 ± 0.16
Video10	0.169 ± 0.15	0.172 ± 0.05	0.271 ± 0.15	0.249 ± 0.09	0.215 ± 0.13
Average F1-score	0.356 ± 0.21	0.334 ± 0.19	0.453 ± 0.23	0.389 ± 0.2	

Table 7.: *prim* performance diff with *aux*.

Primary data	Feature Sets				Average F1-Diff
	Distances	*Angles*	*Combined*	*All*	
Video1	**0.168**	**0.106**	**0.148**	**0.2**	**0.156**
Video2	-0.272	-0.297	-0.233	-0.433	-0.309
Video3	-0.337	-0.133	-0.137	-0.393	-0.25
Video4	-0.375	-0.038	-0.139	-0.246	-0.2
Video5	-0.362	-0.357	-0.232	-0.415	-0.341
Video6	**0.178**	**0.199**	**0.229**	-0.117	**0.122**
Video7	-0.268	-0.325	-0.081	-0.191	-0.216
Video8	-0.145	-0.155	-0.2	-0.01	-0.128
Video9	-0.177	-0.407	-0.243	-0.237	-0.266
Video10	-0.444	-0.128	-0.2	-0.19	-0.24
Average F1-diff	-0.203	-0.153	-0.109	-0.203	

Looking at the average performance of *prim* does not tell the whole story. By considering the performance of the individual folds, we can get some insight in the performance of our chosen candidates for $\mathcal{D}_l^p$. As the standard deviation in the results of Table 6 indicate, the F1-score fluctuates a lot between the different folds. This seems to mean that the quality for candidates of $\mathcal{D}_l^p$ is not consistent, even though our intention was to create equal performing ones. Table 11 shows the individual fold performance for the *prim* case, and shows how big of a difference different folds can make. For instance, for Video 2, the performance drops more than 0.4 between different folds.

5.4 AGGR STRATEGY EVALUATION

The aggregated case combines the *aux* and *prim* strategy to create a single classifier from the aggregated data to test the remaining primary data on. As mentioned before, we artificially expand the primary data to improve its importance during the training process. In our trials, we made the ratio of GENKI auxiliary data to training primary data 2 : 1. The results from these trials can be seen in Table 8. If we compare average performance to that of the *aux*, we get the difference shown in Table 9.

Table 8.: *aggr*: Average F1-scores and their standard deviation using an aggregation of auxiliary and primary data for training.

	Feature Sets				
Primary data	*Distances*	*Angles*	*Combined*	*All*	Average F1-score
Video1	0.584 ± 0.11	0.535 ± 0.05	0.555 ± 0.24	0.601 ± 0.06	0.569 ± 0.14
Video2	0.577 ± 0.14	0.611 ± 0.13	0.59 ± 0.12	0.441 ± 0.28	0.555 ± 0.19
Video3	0.561 ± 0.31	0.629 ± 0.12	0.569 ± 0.32	0.599 ± 0.23	0.589 ± 0.26
Video4	0.575 ± 0.08	0.567 ± 0.11	0.572 ± 0.1	0.589 ± 0.09	0.576 ± 0.1
Video5	0.541 ± 0.16	0.574 ± 0.11	0.568 ± 0.12	0.574 ± 0.11	0.564 ± 0.13
Video6	0.328 ± 0.17	0.191 ± 0.13	0.406 ± 0.12	0.328 ± 0.22	0.313 ± 0.18
Video7	0.748 ± 0.22	0.746 ± 0.15	0.786 ± 0.14	0.795 ± 0.07	0.769 ± 0.15
Video8	0.389 ± 0.08	0.349 ± 0.07	0.404 ± 0.05	0.427 ± 0.07	0.392 ± 0.07
Video9	0.729 ± 0.11	0.774 ± 0.04	0.78 ± 0.06	0.75 ± 0.09	0.758 ± 0.08
Video10	0.431 ± 0.25	0.459 ± 0.23	0.287 ± 0.29	0.46 ± 0.22	0.409 ± 0.26
Average F1-score	0.566 ± 0.22	0.564 ± 0.2	0.569 ± 0.24	0.584 ± 0.2	

The standard deviation of the F1-score in Table 8 indicates that like in the *prim* case, the actual candidate selected for $\mathcal{D}_l^p$ matters a lot for performance. When observing the outcome per fold in Table 11 and comparing it to the *prim* strategy, the fluctuation occurs similarly for the different folds. On average, the *aggr* case outperforms the *prim* data, as well as for the majority of folds. This indicates that using the GENKI data as part of the training data does have a positive effect on the primary data.

To get a better idea of how aggregated data can improve learning, we revisit the probability estimate for Video 1, 6, 8 and 10 under the aggregated condition for only the best performing folds. The best performing folds gave F1-score of 0.700, 0.492, 0.478 and 0.588 for the four videos, which outperforms the *aux* strategy in all cases by at least 0.1 and at max 0.417. Figure 21 shows these probability estimates. Video 10 seems to have had the most benefit from the *aggr* strategy; all values with high probability estimates are centered around instances with a *smile* ground truth, and the estimates jump around less. Although around the 750 frame mark there is a drop when the smile actually appears, it is clear that overall

Table 9.: *aggr* F1-score difference with *aux*.

Primary data	Feature Sets				Average F1-diff
	Distances	*Angles*	*Combined*	*All*	
Video1	**0.306**	**0.285**	**0.2**	**0.278**	**0.267**
Video2	**0.004**	−0.108	−0.089	−0.287	−0.12
Video3	−0.064	**0.116**	−0.121	−0.078	−0.037
Video4	−0.013	**0.317**	**0.176**	**0.072**	**0.138**
Video5	−0.126	−0.096	−0.122	−0.118	−0.115
Video6	**0.328**	**0.191**	**0.36**	**0.03**	**0.227**
Video7	−0.123	−0.122	−0.079	−0.034	−0.089
Video8	**0.022**	−0.049	**0.032**	**0.059**	**0.016**
Video9	−0.015	−0.038	−0.024	−0.028	−0.026
Video10	−0.182	**0.158**	−0.183	**0.02**	−0.047
Average F1-diff	0.014	0.065	0.015	−0.009	

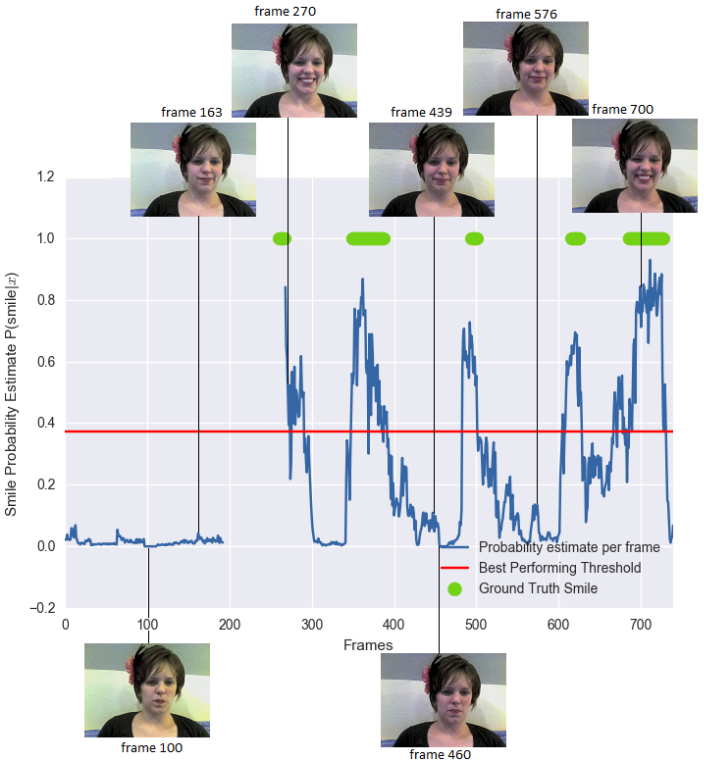
the probability estimates are better paired with actual smile data than can be observed in Figure 20. To some extent the same can be said for Video 6, but the smiling data seems still very difficult to detect due to noise in the facial data. The values still jump around a lot. Looking at the probability estimate for Video 1 and Video 8, their performances seem to have suffered a bit, even though their F1-scores have increased as well. The probability estimates no longer get maxed out around smile segments. However, false positives are reduced. For Video 1 it even seems that the after-apex motions can be detected; the slow detraction of the lips gives a slow reduction of the probability estimates over the whole smile.

5.5 ADAPT STRATEGY EVALUATION

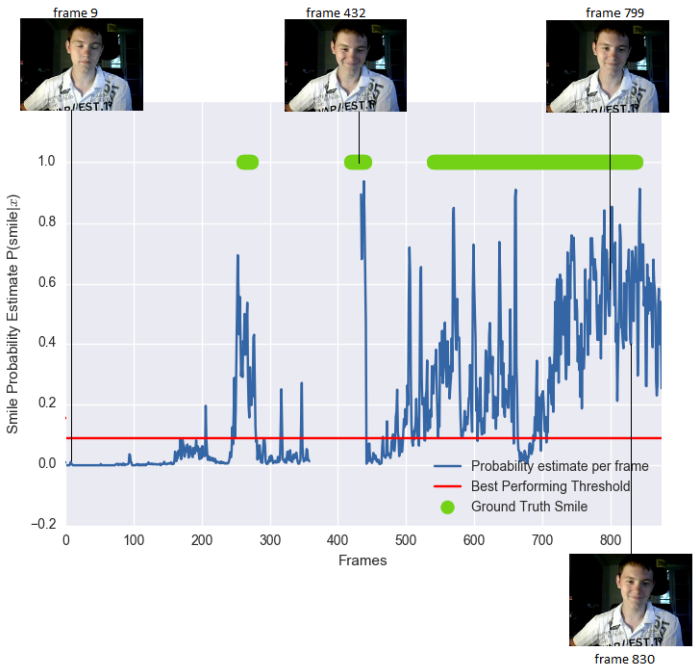
In the *adapt* strategy the classifiers trained for *aux* are combined with the data used in *prim* using an A-SVM. The results of these trials can be viewed in Table 10 and show the worst and most fluctuating performance as of yet. Table 11 shows that the transfer learning approach in most cases outputs a performance of 0, and rather underwhelming performances in most of the non-zero folds. This is rather surprising, since we expected a performance similar to *aggr*. This leads to the belief that training the A-SVM currently does not benefit from the data, and negative transfer takes place.

To research this further, we compare the *aux* and *adapt* strategy by evaluating their performances on the auxiliary (GENKI) data, rather than the primary data sets. Normally, testing on the same instances used during training is bad practice for assessing the performance of a classifier, but in this specific case it shows us how the auxiliary data is influenced by the transfer learning process. Table 12 shows the F1-scores obtained over the auxiliary data in the *aux* strategy. The difference in F1-score between the different feature sets is about similar to the difference in accuracy reported in the cross-validation results of Table 3. The fact that not a perfect F1-score of 1 was obtained tells us that no perfect separability was achieved during training, even with the applied RBF kernel.

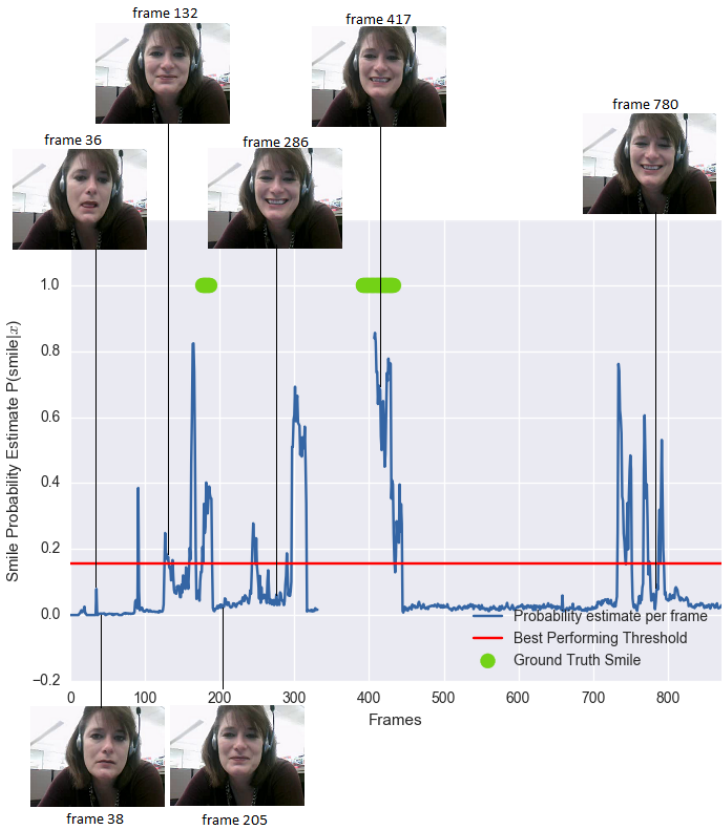
Table 13 shows the performance of the *adapt* strategy on the auxiliary data set (GENKI), with the same $\mathcal{D}_l^p$ as used in the previous evaluated strategies. The table should be read the same way as Table 10, with the difference that the reported F1-scores does not depict the average on $\mathcal{D}_u^p$ for each fold, but the average score achieved on the GENKI data instead.



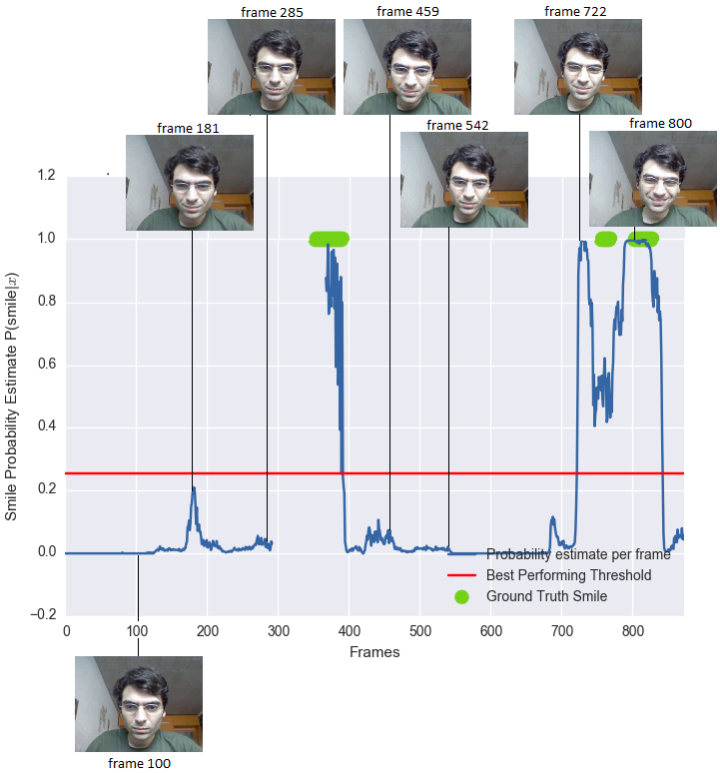
(a) Video 1



(b) Video 6



(c) Video 8



(d) Video 10

Figure 21.: Probability Estimates of videos 1, 6, 8 and 10 of AM-FED data set using *aggr*.

Table 10.: *adapt*: Average F1-scores and their standard deviation using transfer learning.

Primary data	Feature Sets				Average F1-score
	Distances	*Angles*	*Combined*	*All*	
Video1	0.105 ± 0.19	0.152 ± 0.14	0.194 ± 0.27	0.415 ± 0.3	0.216 ± 0.26
Video2	0.301 ± 0.3	0.236 ± 0.2	0.301 ± 0.3	0.308 ± 0.28	0.287 ± 0.28
Video3	0.292 ± 0.3	0.174 ± 0.15	0.28 ± 0.34	0.48 ± 0.28	0.306 ± 0.3
Video4	0.211 ± 0.2	0.146 ± 0.25	0.258 ± 0.23	0.576 ± 0.07	0.298 ± 0.26
Video5	0.002 ± 0.0	0.096 ± 0.1	0.311 ± 0.27	0.345 ± 0.27	0.188 ± 0.24
Video6	0.0 ± 0.0	0.01 ± 0.01	0.044 ± 0.02	0.122 ± 0.14	0.044 ± 0.08
Video7	0.133 ± 0.22	0.119 ± 0.12	0.237 ± 0.26	0.51 ± 0.29	0.25 ± 0.28
Video8	0.049 ± 0.07	0.239 ± 0.17	0.024 ± 0.03	0.249 ± 0.18	0.14 ± 0.17
Video9	0.144 ± 0.28	0.25 ± 0.19	0.322 ± 0.22	0.56 ± 0.29	0.319 ± 0.29
Video10	0.178 ± 0.19	0.211 ± 0.21	0.11 ± 0.19	0.265 ± 0.27	0.191 ± 0.23
Average F1-score	0.137 ± 0.23	0.162 ± 0.18	0.215 ± 0.26	0.406 ± 0.29	

The F1-scores depicted in this table are dramatically lower than in the *aux* case, suggesting that the Adapt SVM used the $\mathcal{D}_l^p$ to learn a new decision boundary that differs from the original SVM to diminish performance on the auxiliary data, but does not actually improve performance.

Table 13 shows a high standard deviation for some of the videos, indicating that the performance fluctuates per fold, similar as in Table 10. This suggests that it might be possible that F1-scores between the primary data and the auxiliary data during the *adapt* strategy might be connected in some way. For instance, a low score on the GENKI data might mean a high performance on the primary data, as the new decision boundary classifies very differently than before. The per fold F1-score for the auxiliary data is reported in Table 11) under the name $\mathcal{D}^a$. In these per-fold results there does not seem to be a strong connection present between the *adapt* $\mathcal{D}^a$ and *adapt* performance; low F1-scores on the auxiliary data can mean both low and high F1-scores on the primary data.

Table 11.: Individual fold performance for *prim*, *aggr*, *adapt* and *adapt* tested on GENKI (*adapt* $\mathcal{D}_a^p$) with the *Combined* feature set. Best performing strategy per fold is indicated in bold (*adapt* $\mathcal{D}_a^p$ not considered).

Video	Strategy	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Average
Video1	*prim*	0.561	0.412	**0.697**	**0.29**	0.382	**0.679**	0.503
	aggr	**0.701**	**0.643**	0.696	0.02	**0.644**	0.629	**0.555**
	adapt	0.0	0.0	0.682	0.0	0.46	0.022	0.194
	adapt $\mathcal{D}_a^p$	0.517	0.614	0.238	0.641	0.44	0.328	0.463
Video2	*prim*	0.688	0.204					0.446
	aggr	**0.711**	**0.468**					**0.59**
	adapt	0.603	0.0					0.301
	adapt $\mathcal{D}_a^p$	0.562	0.026					0.294
Video3	*prim*	0.741	**0.811**	**0.275**	0.384			0.553
	aggr	0.742	0.799	0.014	**0.719**			**0.569**
	adapt	**0.841**	0.272	0.0	0.007			0.28
	adapt $\mathcal{D}_a^p$	0.631	0.313	0.111	0.249			0.326
Video4	*prim*	0.271	0.297	0.201	0.26			0.257
	aggr	0.418	**0.704**	**0.571**	**0.595**			**0.572**
	adapt	**0.464**	0.059	0.0	0.509			0.258
	adapt $\mathcal{D}_a^p$	0.16	0.0	0.0	0.581			0.185
Video5	*prim*	0.192	0.601	0.489	0.547			0.457
	aggr	**0.376**	**0.656**	**0.676**	**0.564**			**0.568**
	adapt	0.081	0.604	0.56	0.0			0.311
	adapt $\mathcal{D}_a^p$	0.063	0.821	0.661	0.0			0.386
Video6	*prim*	0.071	0.383	**0.372**				0.275
	aggr	**0.487**	**0.492**	0.24				**0.406**
	adapt	0.025	0.036	0.072				0.044
	adapt $\mathcal{D}_a^p$	0.05	0.529	0.77				0.45
Video7	*prim*	**0.873**	0.782	0.795	0.83	**0.643**		0.785
	aggr	0.866	**0.84**	**0.865**	**0.857**	0.504		**0.786**
	adapt	0.194	0.185	0.737	0.018	0.052		0.237
	adapt $\mathcal{D}_a^p$	0.009	0.047	0.743	0.034	0.004		0.167
Video8	*prim*	0.031	0.177	0.306				0.171
	aggr	**0.387**	**0.478**	**0.347**				**0.404**
	adapt	0.0	0.0	0.073				0.024
	adapt $\mathcal{D}_a^p$	0.714	0.553	0.249				0.505
Video9	*prim*	0.603	0.479	0.508	0.776	0.439		0.561
	aggr	**0.814**	**0.753**	**0.69**	**0.852**	**0.791**		**0.78**
	adapt	0.295	0.073	0.182	0.728	0.33		0.322
	adapt $\mathcal{D}_a^p$	0.357	0.052	0.239	0.041	0.059		0.15
Video10	*prim*	0.529	0.211	**0.118**	**0.224**			0.271
	aggr	**0.588**	**0.561**	0.0	0.0			**0.287**
	adapt	0.0	0.44	0.0	0.0			0.11
	adapt $\mathcal{D}_a^p$	0.131	0.191	0.691	0.011			0.256

Table 12.: Performance of *aux* strategy on the GENKI data set.

	Distances	*Angles*	*Combined*	*All*
F1-Score	0.941	0.909	0.958	0.977

Table 13.: Performance of *adapt* on the GENKI data set.

$\mathcal{D}_l^p$	Feature Sets				Average F1-score
	Distances	*Angles*	*Combined*	*All*	
Video1	0.113 ± 0.13	0.123 ± 0.12	0.463 ± 0.15	0.453 ± 0.28	0.288 ± 0.25
Video2	0.383 ± 0.38	0.207 ± 0.2	0.294 ± 0.27	0.253 ± 0.1	0.284 ± 0.27
Video3	0.119 ± 0.1	0.162 ± 0.13	0.326 ± 0.19	0.159 ± 0.24	0.191 ± 0.19
Video4	0.269 ± 0.28	0.399 ± 0.35	0.185 ± 0.24	0.366 ± 0.26	0.305 ± 0.3
Video5	0.009 ± 0.02	0.266 ± 0.3	0.386 ± 0.36	0.419 ± 0.37	0.27 ± 0.34
Video6	0.0 ± 0.0	0.641 ± 0.08	0.45 ± 0.3	0.302 ± 0.22	0.348 ± 0.3
Video7	0.162 ± 0.23	0.462 ± 0.29	0.167 ± 0.29	0.205 ± 0.3	0.249 ± 0.31
Video8	0.282 ± 0.28	0.575 ± 0.01	0.505 ± 0.19	0.234 ± 0.28	0.399 ± 0.26
Video9	0.007 ± 0.01	0.299 ± 0.25	0.15 ± 0.13	0.067 ± 0.05	0.131 ± 0.18
Video10	0.295 ± 0.23	0.541 ± 0.15	0.256 ± 0.26	0.297 ± 0.21	0.347 ± 0.24
Average F1-score	0.148 ± 0.23	0.352 ± 0.28	0.311 ± 0.27	0.279 ± 0.28	

5.6 CHOICE OF $\mathcal{D}_l^p$ REVISITED

After evaluating the *aux*, *prim*, *aggr* and *adapt* strategies, evidence seem to suggest that the last three strategies, especially *adapt*, are very susceptible to the choice of $\mathcal{D}_l^p$, and using a similar amount of smile and non-smile data is not enough to guarantee consistent performance over different folds, or good performance at all.

To get a better understanding of how performance can be improved, we analyze the issue with the current used folds. Our current strategy for selecting $\mathcal{D}_l^p$ ends up in three different types of data; a data set with a complete smile segment, where the number of smile instances is smaller than 15, the before-apex moment, and the after-apex moment. Table 14 shows the occurrences of any of these types of candidates per fold. These occurrences do not seem to coincide consistently with bad performance when comparing it to the F1-scores obtained in Table 11, so moment of apex for $\mathcal{D}_l^p$ does not seem to (only) play a deciding factor.

Table 14.: Results of transfer learning

Video	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6
Video1	before apex	before apex	after apex	segment	before apex	before apex
Video2	after apex	before apex				
Video3	before apex	after apex	segment	before apex		
Video4	before apex	after apex	before apex	before apex		
Video5	before apex	segment	after apex	before apex		
Video6	before apex	before apex	before apex			
Video7	before apex	after apex	after apex	segment	before apex	
Video8	before apex	before apex	after apex			
Video9	segment	before apex	segment	before apex	before apex	
Video10	before apex	after apex	before apex	segment		

When looking at badly performing vs. good performing folds for the same video, it becomes more clear what might be the issue; some normalized face points of the non-smile and smile instances looks exactly the same in the worse performing folds. See Figure 22, where there is almost no visible difference between all of the no-smile and smile instances in fold 4 of Video 3. To a lesser degree this can be seen for fold 4 of Video 1 as well, although with some differing instances as well. This similarity of instances with different labels is likely caused by the use of consecutive instances during training while applying a hard threshold. Consecutive frames with very little change can be labeled differently and instead of these consecutive instances helping us recognize only the apex of the smile, as previously assumed, the classifier has become more likely to not classify a smile correctly at all, especially so in the *adapt* case.

The lack of clear performance between negative and positive instances during trainings seems the most obvious issue for hindered performance, and is worth investigating. Using a continuous measure of smile likelihood (by means of a score between 0 and 1) would remove the limitations of the hard threshold, but is labor intensive and subjective to the labeler. As an alternative approach, we try to resolve this issue by keeping the same labeling and trying out two altnerative approaches for choosing $\mathcal{D}_l^p$ instead.

1. By increasing the size of $\mathcal{D}_l^p$: We include 100 non-smiling instances and 35 smiling instances, using the same selection technique as before. Complete segments are not considered.

2. By seperating the data more: We choose 60 consecutive non-smiling instances, and 15 consecutive smiling instances (same as before), but the non-smiling frames have to be at least 50 frames away of any of the smiling segments, so that it is unlikely that any transitional data is included.

The per-fold performance of the first new candidate selection using the *Combined* feature data can be observed in Table 15 on the left, the second one on the right. Unfortunately, in both cases the *adapt* case does not benefit from the extra and possible more distinct data at all, which indicates that there is another problem at hand. The aggregation does seem to benefit from the extra available data in both cases. Table 17 repeats the results from *aux*, *aggr* with our first proposed strategy for $\mathcal{D}_l^p$ and the alternative approaches for side-by-side comparison. Overall, the *aggr* with increased size of $\mathcal{D}_l^p$ performs best, with a 0.076 better performance than *aux*. Unfortunately this is not a very significant increase overall, and additionally it does not guarantee better performance over all the videos.

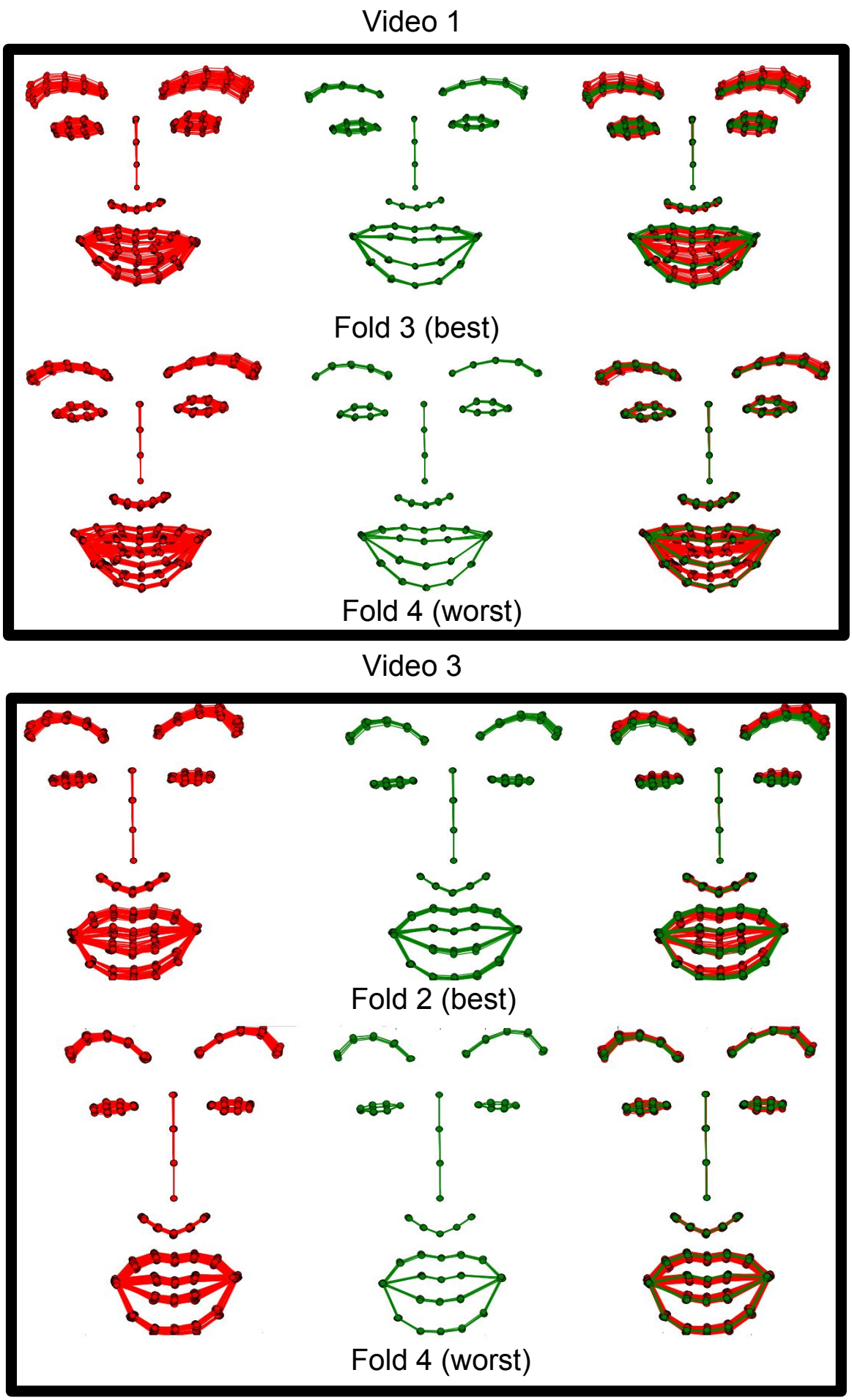


Figure 22.: Instances of normalized instances in $\mathcal{D}_l^p$ for best and worst performing folds. Green shows the smiling instances. Red shows the non smiling instances.

Table 15.: Individual fold performance for *prim*, *aggr* and *adapt* with the *Combined* feature set for the alternative proposed strategies for selecting $\mathcal{D}_l^p$.

Video		Fold 1	Fold 2	Fold 3	Average
Video1	*prim*	0.598	0.509		0.553
	aggr	**0.705**	**0.659**		**0.682**
	adapt	0.16	0.0		0.08
Video2	*prim*	0.655	0.26		0.458
	aggr	**0.679**	**0.404**		**0.542**
	adapt	0.624	0.0		0.312
Video3	*prim*	0.739	0.739		0.739
	aggr	0.726	**0.798**		**0.762**
	adapt	**0.82**	0.295		0.557
Video4	*prim*	0.25	0.222	0.057	0.176
	aggr	0.407	**0.79**	**0.475**	**0.557**
	adapt	**0.469**	0.1	0.19	0.253
Video5	*prim*	0.485	0.448	0.548	0.494
	aggr	**0.565**	**0.654**	**0.613**	**0.611**
	adapt	0.229	0.525	0.034	0.263
Video6	*prim*	**0.477**			**0.477**
	aggr	0.387			0.387
	adapt	0.02			0.02
Video7	*prim*	**0.869**	0.557		0.713
	aggr	0.86	**0.674**		**0.767**
	adapt	0.191	0.0		0.096
Video8	*prim*	0.255			0.255
	aggr	**0.324**			**0.324**
	adapt	0.0			0.0
Video9	*prim*	0.493	0.766	0.752	0.67
	aggr	**0.868**	**0.823**	**0.851**	**0.847**
	adapt	0.064	0.33	0.0	0.131
Video10	*prim*	0.386	**0.174**		0.28
	aggr	**0.485**	0.136		**0.31**
	adapt	0.178	0.0		0.089

(a) Results of increasing size of $\mathcal{D}_l^p$ to 100 (Alternative 1).

Video		Fold 1	Fold 2	Fold 3	Fold 4	Average
Video1	*prim*	0.414	**0.486**	**0.582**		**0.494**
	aggr	**0.451**	0.437	0.431		0.44
	adapt	0.02	0.0	0.0		0.007
Video2	*prim*	0.416				0.416
	aggr	**0.717**				**0.717**
	adapt	0.0				0.0
Video3	*prim*	0.625	0.731			0.678
	aggr	**0.714**	**0.749**			**0.732**
	adapt	0.0	0.0			0.0
Video4	*prim*	0.258	0.377	0.244	0.226	0.276
	aggr	**0.315**	**0.728**	**0.5**	**0.528**	**0.518**
	adapt	0.026	0.0	0.48	0.0	0.127
Video5	*prim*	0.387	0.614	0.418		0.473
	aggr	**0.681**	**0.668**	**0.678**		**0.676**
	adapt	0.459	0.26	0.051		0.257
Video6	*prim*	0.568	0.719	0.328		0.538
	aggr	**0.706**	**0.726**	**0.7**		**0.711**
	adapt	0.0	0.0	0.0		0.0
Video7	*prim*	0.796				0.796
	aggr	**0.862**				**0.862**
	adapt	0.0				0.0
Video8	*prim*	0.249	0.281			0.265
	aggr	**0.311**	**0.287**			**0.299**
	adapt	0.257	0.0			0.129
Video9	*prim*	0.692	0.691			0.692
	aggr	**0.715**	**0.729**			**0.722**
	adapt	0.108	0.0			0.054
Video10	*prim*	**0.46**	0.246	0.17		0.292
	aggr	0.401	**0.315**	**0.596**		**0.437**
	adapt	0.0	0.0	0.225		0.075

(b) Results of nonconsecutive smile and non-smile instances selection in $\mathcal{D}_l^p$ (Alternative 2).

Table 17.: Performance of *aux* and *aggr* strategy using different approaches to choosing instances for $\mathcal{D}_l^p$ with the *Combined* feature set.

Video	*aux*	*aggr* original $\mathcal{D}_l^p$	*aggr* increased size of $\mathcal{D}_l^p$	*aggr* nonconsecutive instance selection $\mathcal{D}_l^p$
Video 1	0.356	0.555	**0.682**	0.44
Video 2	0.678	0.59	0.542	**0.717**
Video 3	0.69	0.569	**0.762**	0.732
Video 4	0.396	**0.572**	0.557	0.518
Video 5	**0.69**	0.568	0.611	0.676
Video 6	0.047	0.406	0.387	**0.711**
Video 7	**0.866**	0.786	0.767	0.862
Video 8	0.372	**0.404**	0.324	0.299
Video 9	0.804	0.78	**0.847**	0.722
Video 10	**0.471**	0.287	0.31	0.437
Average F1-score	0.537	0.569	**0.613**	0.581

5.7 REVISITING THE RESEARCH QUESTIONS.

After reporting and discussing the results of the different evaluation strategies, and the exploration of some additional improvements, we are ready to revisit the research question of this thesis:

Research Question. *Can transfer learning improve the detection rate of spontaneous smiles when applied on a generically trained classifier trained with geometric features?*

First off, performing the *aux* strategy has shown us that there is indeed merit to the idea that a general classifier does not perform well out of the box for each and every face. The GENKI data set has been used in multiple researches as a general, all purpose data set with high performance rate, but now that it is used in a context where it has to consistently perform well on a single face over multiple frames and smile segments, it becomes clear that smiles in faces like Video 6 and Video 10 are difficult to detect, causing consistent failure. However, the application of transfer learning as a solve-all technique is problematic, as can be understood from the above discussed results. We will support this further by focusing on Research SQ 2:

Sub-Question 2. *How does transfer learning compare to alternative approaches of individualized smile detection?*

The proposed alternatives to transfer learning have been *prim* and *aggr* of which the second has outperformed our transfer learning efforts for each of the primary data sets, and was more reliable across different folds. Yang et al. have noticed in their own research that the *aggr* strategy performs similar to *adapt*, but this is not the case for us. Possibly this might mean a problem in options that have no been further explored like parameter choice during training, or negative transfer learning due to the primary data not being representative enough for the rest of the data. However, when using transfer learning for individualized smile detection in real life applications, such parameter tweaking or finding a representative subset is not something that can be easily performed. On the other side, given the promising result of *aggr* without putting in much fine-tune effort seems to be a better alternative, even though training might take longer. Additionally the influence of *aggr* is more easily tunable by duplicating the used primary data less or more. For this reason, transfer learning by means of an A-SVM does not seem like the most appropriate method for individualized smile detection.

Nevertheless, it is important to mention that *aggr* is a better, but a not perfect solution either for individualized smile detection in our evaluation, as it does overall improve performance with regards to the auxiliary data, but only slightly, and with out any guarantee that it will outperform the *aux* strategy for every case. In its current form adding a portion of the primary data during is more trouble than the performance boost that it *might* give. Further experiments with auxiliary and primary data ratios for aggregated learning might prove effective in improving the benefit of the *aggr* strategy. Additionally, more research towards choosing a good heuristic for $\mathcal{D}_l^p$ could mean a more consistent performance, even for different faces.

Another issue that influenced our results negatively is the different labeling efforts in the AM-FED and GENKI data set that we have witnessed, for example, in Video 1, where closed smiles were not labeled as smiles, but were considered smiles nonetheless, in all likelihood because the auxiliary data includes samples of closed smiles labeled as smiles. In retrospect an instance-based transfer learning approach might have helped us exclude these instances for consideration during transfer learning in these cases.

We now discuss the SQ:

Sub-Question 1. *What geometric features prove effective in personalized smile detection?*

From the feature sets we tested, we can say that *Combined* and *All* perform well enough to compare to the state of the art when enough training data is available during training on the primary data set (see Table 3 and 4), so as a baseline they perform well. We created the *Combined* set as a compressed case of *All* to contain the most distinct features to distinguish between smiles and non smiles. The fact that their performance is similar seems to mean we succeeded in this regard. The lower score of *Distances*, *Angles* and *Orientation* compared to *Combined* indicate that there is information to be gained from the different feature sets; which is as expected. A concern of our geometric-based feature approach was possible noise in face point placement causing errors in smile detection for faces showing only very subtle smiles. This, however, did not pose a problem, as the cross-validation for each of the AM-FED videos resulted in at least 87% accuracy with the *Combined* feature set, even for Video 6 and 10 that only show subtle smiles.

That the feature sets are appropriate for individualized smile detection can be concluded from the improved performance for each of the videos in the *aggr* strategy compared to the *aux* one. The performance becomes better when more instances from the primary data are available in the training set, which shows that some important variances are available in the feature set that helps the classification process for new faces.

6

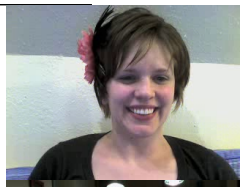
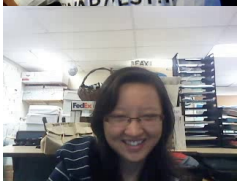
CONCLUSION

In our research we aimed to improve smile detection performance on individual faces by applying transfer learning as an additional step in the machine learning process. We started out with training a Support Vector Machine on a large generic set of smile data, called the auxiliary data, and used this to train an Adaptive SVM, together with a subset of data of the target face, called the primary data. We then tried out a variety of geometric low level feature sets like angular, distance and head-orientation data, which we expected to capture the differences between different people's smiles well. The data used during evaluation consists of the GENKI data set as our auxiliary data set over all trials, and 10 videos from the AM-FED data set as 10 separate primary data sets.

We evaluated our approach of transfer learning by means of an A-SVM (*adapt*) by comparing it to the performance of the SVM trained on auxiliary data alone (*aux*) an SVM trained on some primary data alone (*prim*), and to an SVM trained on the combination of the auxiliary data and primary (*aggr*). We found that the *aggr* case performed best over all different strategies, with least fluctuation and best F1-score improvement compared to the *aux* case over each of the different methods. Previous research has suggested that an approach like *aggr* should perform similar to *adapt*, but in our research we were not able to get the same performance. The most likely cause seems to be the result of negative transfer, likely caused by bad choices for $\mathcal{D}_l^p$. More research in how to choose good values for $\mathcal{D}_l^p$ that are representative for $\mathcal{D}^p$ but doesn't require complete knowledge about the primary data might be a possible solution. This should also be the strategy to take for improving the *aggr* strategy performance, as in its current form the effort of adding primary data during training does not out weight the benefits. Additionally, our current research seemed to suffer from inconsistent labeling between the AM-FED and GENKI data set, which might be better resolved using an instance-based transfer learning approach than by applying the A-SVM.

A

APPENDIX

Name	AMFED ID	Frames Before Pruning	Frames After Pruning	Smile Frames	Segments	Apex
Video 1	1e7bf94c-02a5-48de-92bd-b0234354dbd5	740	740	104	5	
Video 2	d880e27b-f6f3-4c35-b400-af0e19f05d41	856	638	355	3	
Video 3	502ba501-f30c-4abb-af2f-d725456d1b5a	876	748	288	4	
Video 4	516fc1bb-cfcc-4a70-b466-a0a69112e074	887	887	147	3	
Video 5	4294f380-8e04-41ff-a453-3d7ad4b0a0d3	871	827	316	4	
Video 6	24d782fd-cf97-4c64-b54a-cb422ac479d6	876	876	334	3	
Video 7	ee5e0ebd-5f86-4a0c-8690-a13c8639cc82	869	869	506	4	

Name	AMFED ID	Frames Before Pruning	Frames After Pruning	Smile Frames	Segments	Apex
Video 8	845dd4d7-2e42-4d31-a96d-47c5ecf722e6	873	873	53	2	
Video 9	2bde5de6-0d0c-497a-82bd-8cc441b8b0f3	846	836	277	5	
Video 10	5540d693-8f69-47bd-aa95-24227a949290	874	874	76	3	

Table 18.: Amfed Data used

BIBLIOGRAPHY

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

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, June 2014.

[3] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2610–2617. IEEE, 2012.

[4] Vinay Bettadapura. Face expression recognition and analysis: the state of the art. *arXiv preprint arXiv:1203.6722*, 2012.

[5] Nigel Bosch, Sidney D'Mello, Ryan Baker, Jaclyn Ocumpaugh, Valerie Shute, Matthew Ventura, Lubin Wang, and Weinan Zhao. Automatic detection of learning-centered affective states in the wild. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 379–388. ACM, 2015.

[6] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[7] Ginevra Castellano, Iolanda Leite, André Pereira, Carlos Martinho, Ana Paiva, and Peter W Mcowan. Multimodal affect modeling and recognition for empathic robot companions. *International Journal of Humanoid Robotics*, 10(01):1350010, 2013.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] Jeffrey F Cohn, Zara Ambadar, and Paul Ekman. Observer-based measurement of facial expression with the facial action coding system. *The handbook of emotion elicitation and assessment*, pages 203–221, 2007.

[10] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537–556, 2013.

[11] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. Translated learning: Transfer learning across different feature spaces. In *Advances in neural information processing systems*, pages 353–360, 2008.

[12] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[13] Sébastien David, Miguel A Ferrer, Carlos M Travieso, Jesús B Alonso, and Dpto De Señales y Comunicaciones. gpdshmm: A hidden markov model toolbox in the matlab environment. *CSIMTA, Complex Systems Intelligence and Modern Technological Applications*, pages 476–479, 2004.

[14] Hamdi Dibeklioglu, Roberto Valenti, Albert Ali Salah, and Theo Gevers. Eyes do not lie: spontaneous versus posed smiles. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 703–706. ACM, 2010.

[15] Paul Ekman, Wallace V Freisen, and Sonia Ancoli. Facial signs of emotional experience. *Journal of personality and social psychology*, 39(6):1125, 1980.

[16] Andreas Ernst, Tobias Ruf, and Christian Kueblbeck. A modular framework to detect and analyze faces for audience measurement systems. In *2nd Workshop on Pervasive Advertising at Informatik*, pages 75–87. Citeseer, 2009.

[17] Hatice Gunes and Maja Pantic. Dimensional emotion prediction from spontaneous head gestures for interaction with sensitive artificial listeners. In *International conference on intelligent virtual agents*, pages 371–377. Springer, 2010.

[18] Mohammed Ehsan Hoque, Daniel J. McDuff, and Rosalind W. Picard. Exploring temporal patterns in classifying frustrated and delighted smiles. *IEEE Trans. Affect. Comput.*, 3(3):323–334, July 2012.

[19] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

[20] T. Kanade, J. F. Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53, 2000.

[21] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[22] Gwen Littlewort, Marian Stewart Bartlett, Ian Fasel, Joshua Susskind, and Javier Movellan. Dynamics of facial expression extracted automatically from video. *Image Vision Comput.*, 24(6):615–625, June 2006.

[23] Patrick Lucey, Jeffrey F Cohn, Iain Matthews, Simon Lucey, Sridha Sridharan, Jessica Howlett, and Kenneth M Prkachin. Automatically detecting pain in video through facial action units. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3):664–674, 2011.

[24] 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 *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 881–888, June 2013.

[25] Daniel McDuff, Rana El Kaliouby, and Rosalind W Picard. Crowdsourcing facial responses to online videos. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 512–518. IEEE, 2015.

[26] Hongying Meng, Di Huang, Heng Wang, Hongyu Yang, Mohammed AI-Shuraifi, and Yunhong Wang. Depression recognition based on dynamic facial and vocal expression features using partial least square regression. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 21–30. ACM, 2013.

[27] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[29] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[30] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[31] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[32] Marc Schroder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark Ter Maat, Gary McKeown, Sathish Pammi, Maja Pantic, et al. Building autonomous sensitive artificial listeners. *IEEE Transactions on Affective Computing*, 3(2):165–183, 2012.

[33] Thibaud Sénéchal, Jay Turcot, and Rana El Kaliouby. Smile or smirk? automatic detection of spontaneous asymmetric smiles to understand viewer experience. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE, 2013.

[34] Caifeng Shan. Smile detection by boosting pixel differences. *IEEE transactions on image processing*, 21(1):431–436, 2012.

[35] Michel F. Valstar, Hatice Gunes, and Maja Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, ICMI '07, pages 38–45, New York, NY, USA, 2007. ACM.

[36] Michel F Valstar, Maja Pantic, Zara Ambadar, and Jeffrey F Cohn. Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 162–170. ACM, 2006.

[37] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[38] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. Toward practical smile detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2106–2111, Nov 2009.

[39] Jacob Whitehill and Christian W Omlin. Haar features for facs au recognition. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 5–pp. IEEE, 2006.

[40] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 188–197, New York, NY, USA, 2007. ACM.

[41] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1855–1862. IEEE, 2010.

[42] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, Jan 2009.

[43] Ligang Zhang, Dian Tjondronegoro, and Vinod Chandran. Geometry vs. appearance for discriminating between posed and spontaneous emotions. In *International Conference on Neural Information Processing*, pages 431–440. Springer, 2011.

[44] Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 454–459. IEEE, 1998.

[45] Erheng Zhong, Wei Fan, Jing Peng, Kun Zhang, Jiangtao Ren, Deepak Turaga, and Olivier Verscheure. Cross domain distribution adaptation via kernel mapping. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1027–1036. ACM, 2009.