A combined modular system for face detection, head pose estimation, face tracking and emotion recognition in thermal infrared images

Marcin Kopaczka, Justus Schock, Jan Nestler, Kevin Kielholz and Dorit Merhof
Institute of Imaging and Computer Vision
RWTH Aachen University
Aachen, Germany
marcin.kopaczka@lfb.rwth-aachen.de

Abstract—**Thermal infrared imaging is an emerging imaging modality allowing capturing heat radiation not detectable in the visible spectrum. In recent years, numerous applications of thermal infrared imaging for the processing of face images have been published. Many of these approaches only allow minimal head movement due to the lack of sufficiently robust face tracking and interpretation algorithms for thermal infrared images. To address this issue, we present a suite of interconnected algorithms for a number of typical facial image processing tasks - face detection, head pose estimation, detection and tracking of facial landmarks and facial expression analysis. The modules can be used independently or in combination with each other. For combined use and as a demonstration of the versatility of our solution, we present a multiprocess solution based on a networking middleware that allows using all proposed algorithms to perform real-time face tracking and emotion recognition in thermal infrared images. The full code is made freely available on gitub under GNU license to allow incorporating our solutions into own projects.**

Index Terms—**thermal infrared, face detection, head pose estimation, face tracking, emotion recognition**

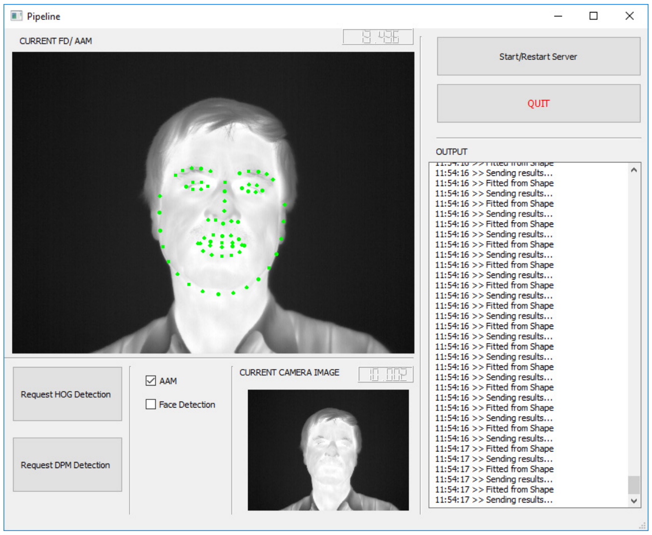
I. Introduction

Automated processing of facial images is a highly active research area in computer vision. Numerous methods have been presented for the automatic detection, tracking and ultimately analysis of facial features. Research in this area is fueled by the fact that facial image analysis is required by a large number of applications. Person identification, emotion and facial expression recognition or behavioral analysis are examples for tasks requiring complex image processing pipelines, especially when algorithms are to be applied to unconstrained video data. Currently, most image processing algorithms for analysis of facial images are developed exclusively for visual data (RGB) or visual data enhanced by depth information (RGB-D) that can be obtained by additional depth sensors such as the widely used kinect sensor. For these technologies, a number of sophisticated algorithms are readily available, an overview of which can be found in [1]. At the same time, only a small number of advanced algorithms for thermal infrared imaging has been proposed, despite the increased availability of thermal infrared sensors and the advantages of this imaging modality. Thermal infrared or long-wave infrared imaging has several aspects making it an interesting choice for several applications. These advantages include lack of illumination sensitivity, which means that images acquired using a thermal sensor do not change their appearance regardless of the current lighting situation, allowing them to work under varying lighting conditions. Changes in light intensity or direction have no effect on the final image, which means that thermal infrared image processing does not need to target effects such as shadows or over- and underexposed areas, a common problem when working with regular images. Even more, since thermal sensors capture radiation emitted by objects themselves, in contrast to other imaging modalities they do not require any additional light source but can operate in dim light or even complete darkness. An additional advantage for analysis of facial images is the fact that certain physiological effects that are invisible in the visual domain can be detected in thermal data. Perspiration can be detected by analyzing the temperature around the nostrils, which means that the breathing rate can be assessed with little effort when appropriate tracking and detection algorithms are used [2]. Algorithms for the detection of blood vessels or even heart rate measurement using thermal infrared images have also been presented [3].

Of the algorithms available for the analysis of facial images in thermal infrared data, most impose heavy limitations with respect to the allowable amount of head movement. Only a limited number of tracking algorithms are available, most of them covering only small movements and no arbitrary head motion. Face detection algorithms have also been presented, but most of them have hard-coded algorithmic requirements for certain head and body poses, for example strict frontal views of the face or requiring the shoulders to be visible as well. At the same time, no system combining both aspects - preprocessing in terms of face detection and face tracking and subsequent robust image analysis of the tracked faces - exists, leaving both approaches unlinked.

To address this shortcoming, we present a modular system allowing application of facial image analysis algorithms not only to highly restricted poses but to images and videos displaying a large amount of head movement. The system consists of several modules for distinct tasks - in our case image

978-1-5386-6628-9/18/$31.00 ©2018 IEEE



☑ AAM
☐ Face Detection

```
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:16 >> Fitted from Shape
11:54:16 >> Sending results...
11:54:17 >> Fitted from Shape
11:54:17 >> Sending results...
11:54:17 >> Fitted from Shape
11:54:17 >> Sending results...
11:54:17 >> Fitted from Shape
11:54:17 >> Sending results...
```

Fig. 1. Screenshot of the GUI provided by our system.

acquisition from a thermal camera or a recorded source, face detection in the thermal data, precise detection and tracking of facial landmarks and subsequent application of analysis algorithms to the tracked data. Each module applies state-of-the-art feature- and model-based methods for its task and the underlying architecture allows conveniently exchanging parts of the pipeline for other algorithms that may better fit the respective image processing problem. Interaction and data transfer between the modules is implemented using ZeroMQ, a lightweight networking middleware allowing processes to communicate across a network, connecting tasks independently of operating system, bus architecture (32 or 64 bit) and programming languages used. This approach maximizes the modularity and agility of our system as it allows using the operating system and language best suitable for certain tasks. For example, the camera connection can be implemented natively in C++ using the API provided by the camera vendor while image processing algorithms can be written using Python, taking advantage of Python's powerful image processing and machine learning methods. At the same time, the GUI can be implemented conveniently for Windows workstations while demanding computation tasks may be outsourced to a Linux cluster with powerful hardware. A screenshot of the GUI is shown in Fig. 1

This paper is structured as follows: We describe the algorithms for implemented specific tasks in Sec. II-A. The modular architecture is described in Sec. II-A5. The methods are benchmarked and evaluated in Sec. III, followed by a discussion in Sec. IV. Finally, we give a conclusion in Sec. V.

II. Methods

Here, we describe both the general architecture as well as the underlying algorithms and motivate their choice.

A. *Algorithms Used*

The used algorithms are all based on machine learning methods and therefore require high quality training data. We have used the freely available annotated thermal face database presented in [4] to train our implementations.

1) Face Detection: A number of algorithms has been proposed for face detection in thermal infrared images. They can be divided into two groups: Algorithm-based and data-based approaches. Algorithm-based methods have intrinsic requirements on image properties, such as a certain pose of the person (usually fully frontal) and perform detection based on a fixed, manually defined set of rules. Contrary to that, data-based methods use machine learning and sets of labeled data to learn detecting faces and make up decision rules themselves. As shown in [5], data-driven methods require extensive amounts of labeled images for training, but if this is given, they can significantly outperform algorithm-based approaches if trained properly. In our work, we use the data-driven HOG-SVM method [6] for face detection. We have chosen this algorithm as it is a well understood and researched method that has already been applied to various object detection tasks in various imaging domains. Furthermore, a HOG-SVM can be trained with little effort and therefore quickly adapted to new data if necessary.

Detector training is performed using the manual annotations of the face database. Positive examples are generated by using the bounding boxes defined by the manual annotations and extending them by 10% in all directions. Extending is applied since the it allows the gradient-based HOG detector to pick up object boundaries better - gradients close to the borders of the bounding boxes may be distorted and lower the descriptor performance if the bounding boxes are chosen too small. After the extended bounding boxes have been cropped from the full image, HOG features are computed on the resulting patches. By applying HOG to a patch, the patch is transformed into a set of local gradient vectors that describe its content based on the edges of the structures within it. While detailed pixel information is lost, the resulting vector contains a more high-level description of the objects in the image. The descriptor parameters are chosen identically to the original HOG publication since these values are the result of a through evaluation and have been proven to perform well in numerous applications.

Using this method, HOG feature vectors for all images in the database are computed. Since the method requires an additional set of realistic non-face images for training, we additionally sampled 10.000 patches from the face database that do not contain faces. We sampled both patches containing no face at all as well as patches that were containing partial faces with a maximal overlap of 50% with the manually defined bounding box. Including partial faces in the negative set trained the detector to classify patches containing partial faces as negative, thereby restricting positive results to patches that contain a whole face. HOG vectors were computed for the negative patches as well and both patch sets were fed into a linear SVM to train the detector.

In our architecture, the detector runs in a separate process and face detection is performed automatically when receiving an image. Since face detection needs not to be performed for all frames, the detector is activated only when a new video feed is loaded or upon user request. It returns the bounding box coordinates of the detected face or a set of zeroes if no face is detected in the image. While the detector itself can detect multiple faces in an image, we only return the bounding box of the first detected face since in our use-case we are not using images displaying more than one person.

2) Facial Landmark Detection: Here, we describe the facial landmark detection before introducing our head pose estimation algorithm. In our pipeline, head pose estimation is performed prior to landmark detection. However, an understanding of the facial landmark detection algorithm is required before head pose estimation can be presented.

While a large number of algorithms for facial landmark detection is available for regular images, only a small number of publications addresses this topic for thermal data. The key reason is that facial landmark detection in thermal images is a much more challenging task due to the properties of this imaging modality. Faces in thermal infrared images lack contrast, both in skin tone as well as between different facial areas. For example, there is virtually no intensity difference between the skin and lips in thermal images, while a clear difference can be seen in regular photographs. To the best of our knowledge, the only available algorithm for precise facial landmark detection in thermal infrared images has been proposed in [7], where feature-based active appearance models [8] are used. Being an extension of the widely adapted active appearance models (AAMs) [9] [10], feature-based AAMs use feature descriptors to improve tracking robustness and allow precise facial landmark detection in thermal infrared images [7].

Active appearance models use statistical shape and texture modeling combined with optimization algorithms to fit a pre-trained model to an unseen face. They can be methodically motivated by combining the approaches of active shape models [11] and eigenfaces [12]. AAMs are trained using a face database with detailed manual annotations for all landmark points. In a first step, principal component analysis (PCA) is applied to the landmark data. The PCA result is a mean shape of all faces in the database and a set of eigenshapes. New face shapes can be modeled by linear combination of the mean shape and the modulating eigenshapes. To allow texture modeling, all faces are transformed into the previously computed mean shape using piecewise affine transformations. The result of this transformation is a set of perfectly aligned face images with identical shape and varying texture. Subsequently, a second PCA is applied to the pixel data (appearance) of the images. Similar to the shape PCA, the appearance PCA yields a mean texture and a number of aligned eigenfaces that can be used to model face appearances. Note that in contrast to the original eigenfaces approach presented in [12], the images fed into in the appearance PCA are shape-normalized by the shape PCA and therefore only need to model actual texture variation, making AAMs significantly more robust towards pose variation.

After training, an AAM can be used to model an unseen face by altering its shape and appearance values to match the new face. In practice, this can be expressed as an optimization problem where the optimal shape parameters and appearance parameters need to be found to minimize the difference between the target image and the model. Numerous ways of solving this task by re-formulating the optimization problem have been proposed, see [13] for the most comprehensive overview over the available algorithms. In our work, we use the simultaneous inverse compositional approach (SIC) as it delivers the most precise results.

Feature-based AAMs extend the approach described above by performing fitting not on the image itself, but on a set of feature images acquired by densely computing feature vectors for all image pixels. Well-performing feature descriptors include dense HOG and SIFT [14]. We follow the results of [7] and implement a dense SIFT desscriptor that converts each image into 36 individual channels. Fitting is then performed on all channels simultaneusly.

For speeding up tracking performance, we have reduced the number of model parameters. This increases the frame rate by reducing computation complexity. Additionally, fitting is performed on a model with a diagonal of 70 pixels for additional speed-up. In our current implementation, we use 9 shape and 5 appearance channels for fitting, allowing the algorithm to run with 10 frames per second. In our modular implementation, the AAM module receives a bounding box and optionally landmark estimates from the head pose estimation module. The module returns the 68 detected landmark positions. If no initial landmarks are provided, then the optimization is initialized with the AAM's mean shape. While this is the default for most AAM implementations, we have additionally developed a head pose estimation system for improved AAM initialization.

3) Head Pose Estimation: As stated above, AAMs are usually fitted by starting with a mean shape and performing incremental optimization from there. This may lead to issues in case the face is strongly rotated. Initial and final landmark positions may be strongly different, causing the AAM optimizer to converge to a local minimum, resulting in strongly misdetected landmarks. Therefore, we added a head pose estimation step to our pipeline. Placed between face detection and AAM-driven landmark detection, the head pose estimator performs a prediction of the first two AAM parameters. This makes it possible to initialize the AAM with a shape that is already close to its final position, thereby reducing optimization time and the risk of converging to a local minimum. To perform the estimation, a random forest regressor with 150 trees is trained to predict the in-plane head rotation and the first two parameters of the underlying AAM. To obtain a ground truth for training and validation, a set of videos with strong variation in head pose is tracked with a precise AAM that allows high-quality landmark detection. The face images are subsequently rescaled to 64 x 64 pixels, forwarded to a HOG descriptor and ultimately fed into the random forest for training. We have decided to use 64 x 64 pixels as image size since preliminary experiments have shown that larger images increase computation time, however do not result in an improved regression performance. The predicted values are forwarded to the AAM where they are used to initialize the model's parameters.

4) Facial Expression Recognition: Facial expression recognition is performed using HOG-SVM in a similar way as the implemented face detector. We forward the bounding box defined by the landmarks detected by the AAM module to the facial expression recognition module, where the image patch defined by the bounding box is classified using a multi-class SVM trained on the facial expression database. The patch is then re-scaled to 96 x 96 pixels to allow HOG extraction at defined coordinates. The feature vector is formed by computing HOG features of the rescaled image patch, however not systematically over the whole image as in the case of face detection, but from this patch exclusively. This allows a dramatic speed-up compared to the face detection. In our implementation, four basic expressions from the database (neutral, happiness, anger and surprise) are detectable in real time. The system returns a continuous value between 0 and 1 for all four expressions which are then fed back to the GUI module.

5) System Architecture: The system consists of several separately compiled or executed modules. Following the processing pipeline shown in Fig.2, the first module is an image provider. This program connects either to a physical camera or has access to video data stored on disc. Independently of the data source, the providing service feeds video frames at a defined rate into the network. In our case, typical frame rates are 10 or 30 fps. Frames are sent as 8 bit unsigned int with one image channel to save bandwidth, which means that they may require windowing or other preprocessing if the data source has a higher bit rate. Our camera recorded 1024 x 768 pixel images at 16 bit unsigned int, therefore we performed linear windowing between 25 °C and 36 °C before sampling to 8 bit to maximize contrast in the temperature range that is most relevant for faces. Frames are broadcasted uncropped and at full resolution.

The images are received by the master module that also provides the GUI and acts as the central data link for all other modules. This module performs no image processing, instead it takes care of dispatching the data to the connected service providers and displaying all returned results.

In a first step, images can be sent to the face detection module that performs bounding box localization using HOG-SVM as described in Section II-A1. Performing face detection requires approximately one second and is only required once when initializing the tracker to obtain an initial bounding box, therefore it is triggered manually. The face detector returns the bounding box to the master module.

The images and the initial bounding box are subsequently forwarded to the facial landmark detection module that uses the parameter-optimized feature-based AAM described in Sec. II-A2. Upon receiving the first image of the recording, the module uses the detected bounding box and the AAM's mean shape to initialize landmark positions for tracking. After landmarks have been detected for the first image, the module changes its data source and always uses its detected positions of the previous frame to initialize tracking for the current frame.

III. Experiments and results

The modules for face detection, facial landmark detection and facial expression recognition are using algorithms we previously published and evaluated throughly in other work. An overview of different face detection algorithms, including the HOG-SVM method used here, can be found in [5]. Thermal infrared landmark detection with feature-based AAMs is introduced and evaluated in [7]. Finaly, the database itself and an evaluation of the used facial expression recognition method are presented in [4]. For the sake of briefness and to keep the conference's page limit, we do not present the results of this previous work here. Instead, we focus on evaluating the head pose estimation, which is the main novel algorithm presented in this work. Additionally, we perform run-time benchmarking of the architecture to analyze its real-time capabilities.

A. Head Pose Estimation

The system was trained with five videos showing large head pose variation. The videos were tracked with the high-quality AAM presented in [7] to obtain AAM parameters. Subsequently, the system was evaluated by using it to predict head poses from the database. As evaluation metric, we used the normalized root mean square error between the ground truth and AAM fitting either from the mean shape or from an initialization from the head pose estimation system. As Figure 3 shows, the fitting accuracy is strongly improved by using our head pose estimation method. Commonly, head pose estimation is performed by measuring the three main rotational degrees of freedom (roll, pitch and yaw), however no sufficiently annotated thermal infrared database is available. To evaluate our method's capability to predict these values, we trained the head pose estimation system on a database in the visible spectrum that was labeled with a ground truth for these values, where we predicted the three angles simultaneously. The results shown in Fig. 4 indicate that the system also allows precise prediction of freedom of movement in three directions if given an extensively labeled database. Overall, head pose estimation strongly increases the system's capability to be initialized with arbitrary head poses.

B. Runtime Benchmark

We have tested our system's real-time capabilities when fed with images from a live camera. The camera was recording at its maximum capability of 1024 x 768 pixels and 30 frames per second and 16 bit depth, resulting in 47 MB/s of data. The image server was set to window the images to 8 bit

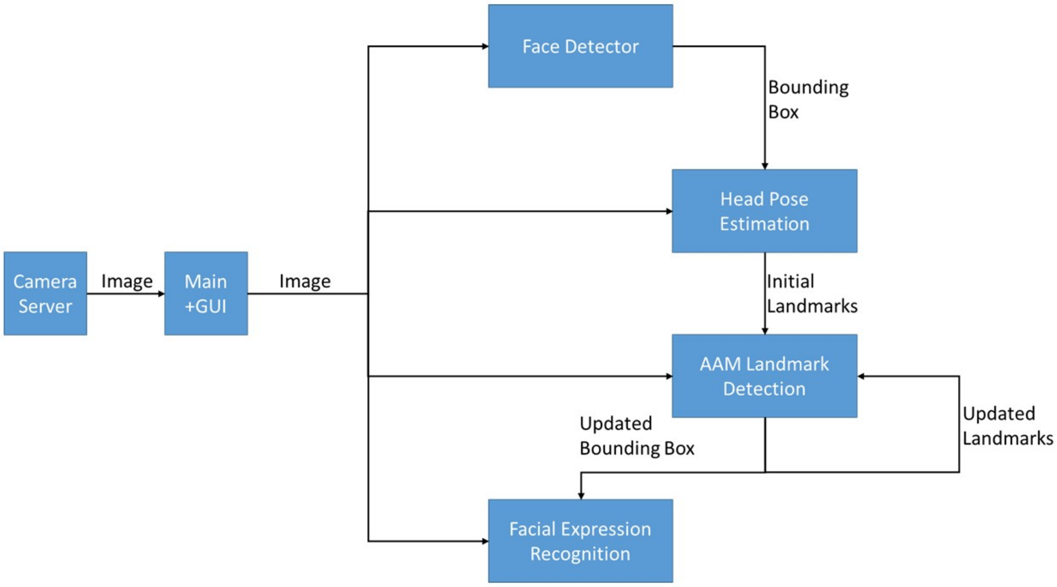


Fig. 2. Schematic overview of our system's pipeline, giving an overvoew of the modules and the data flow between them. See Section II-A5 for a detailed description.

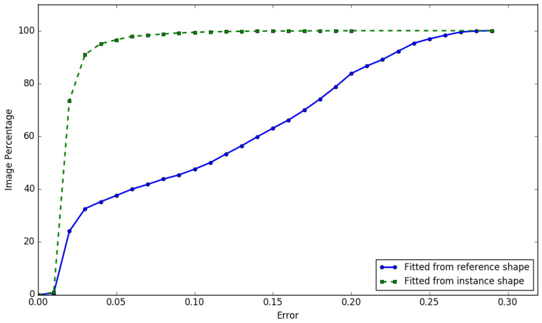


Fig. 3. Normalized root mean squared error betweed database faces fitted from the AAM's mean shape and from shapes generated using random forest estimation.

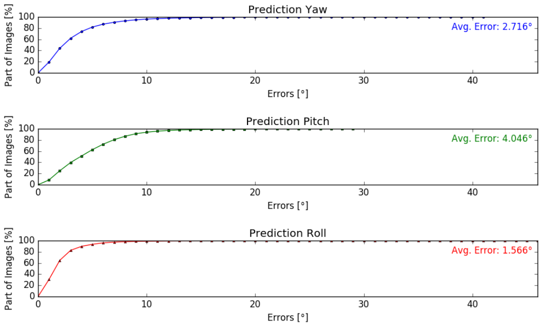


Fig. 4. Head pose estimation trained to predict roll, pitch and yaw on a visual face database.

and forward them to the main module, which then distributed them to the individual modules. On a workstation PC (Intel i7-6700, 4.0 GHZ), AAM tracking on four of the cores (we assume that the number of cores was limited by settings of the underlying Intel MKL library we used) was able to perform landmark detection and facial expression recognition at 8 to 9 fps. When requesting face detection and head pose estimation, the system needed 0.25 seconds to return a bounding box for the face and estimate the first two AAM parameters. Since face detection and head pose estimation are usually only required when initializing the AAM, the system is considered to run at 9 fps once initialization has been performed. A qualitative evaluation of system performance on several videos has shown that the tracking is stable under regular unconstrained head movement.

IV. Discussion

The presented system allows reliable face tracking of facial landmarks and facial expression recognition in thermal infrared videos at 10 fps. The tracker is able to track faces showing regular movement. The additional stage consisting of a face detection and head pose estimation allows initializing the tracker with start values that allow fitting with precision that is superior to using the mean shape as initialization.

Our system, extended with a module for the analysis of breathing patterns in thermal images, has been successfully adapted to detect breathing anomalies [15], where we have shown that our AAM tracking approach outperforms state-of-the-art tracking approaches and allows precise detection of abnormal events.

V. Conclusion

We have presented a modular system for real-time face detection, head pose estimation, facial landmark detection and emotion recognition. The architecture allows processing of images from a live video feed or stored videos using a lightweight network middleware. Additionally, our system allows face detection and head pose estimation to improve its capabilities to cope with arbitrary head poses.

References

[1] Ciprian Adrian Corneanu, Marc Oliu Simón, Jeffrey F Cohn, and Sergio Escalera Guerrero, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 8, pp. 1548–1568, 2016.

[2] Carina Barbosa Pereira, Xinchi Yu, Michael Czaplik, Rolf Rossaint, Vladimir Blazek, and Steffen Leonhardt, "Remote monitoring of breathing dynamics using infrared thermography," *Biomedical optics express*, vol. 6, no. 11, pp. 4378–4394, 2015.

[3] Travis Gault and Aly Farag, "A fully automatic method to extract the heart rate from thermal video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 336–341.

[4] Marcin Kopaczka, Raphael Kolk, and Dorit Merhof, "A fully annotated thermal face database and its application for thermal facial expression recognition," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2018.

[5] Marcin Kopaczka, Jan Nestler, and Dorit Merhof, "Face detection in thermal infrared images: A comparison of algorithm- and machine-learning-based approaches," in *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2017.

[6] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[7] Marcin Kopaczka, Kemal Acar, and Dorit Merhof, "Robust facial landmark detection and face tracking in thermal infrared images using active appearance models," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Rome, Italy, February 2016, pp. 150–158.

[8] E. Antonakos, J. Alabort i medina, G. Tzimiropoulos, and S. Zafeiriou, "Feature-based lucas-kanade and active appearance models," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2617–2632, September 2015.

[9] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 6, pp. 681–685, 2001.

[10] Iain Matthews and Simon Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[11] Timothy F Cootes and Christopher J Taylor, "Active shape models—'smart snakes'," in *BMVC92*, pp. 266–275. Springer, 1992.

[12] Lawrence Sirovich and Michael Kirby, "Low-dimensional procedure for the characterization of human faces," *Josa a*, vol. 4, no. 3, pp. 519–524, 1987.

[13] Joan Alabort-i Medina and Stefanos Zafeiriou, "A unified framework for compositional fitting of active appearance models," *International Journal of Computer Vision*, vol. 121, no. 1, pp. 26–64, 2017.

[14] David G Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

[15] Marcin Kopaczka, Oezcan Oezkan, and Dorit Merhof, "Face tracking and respiratory signal analysis for the detection of sleep apnea in thermal infrared videos with head movement," in *International Conference on Image Analysis and Processing Workshop (ICIAP)*, 2017.