OpenFace: an open source facial behavior analysis toolkit

Tadas Baltrušaitis
Tadas.Baltrusaitis@cl.cam.ac.uk

Peter Robinson
Peter.Robinson@cl.cam.ac.uk

Louis-Philippe Morency
morency@cs.cmu.edu

Abstract

Over the past few years, there has been an increased interest in automatic facial behavior analysis and understanding. We present OpenFace – an open source tool intended for computer vision and machine learning researchers, affective computing community and people interested in building interactive applications based on facial behavior analysis. OpenFace is the first open source tool capable of facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. The computer vision algorithms which represent the core of OpenFace demonstrate state-of-the-art results in all of the above mentioned tasks. Furthermore, our tool is capable of real-time performance and is able to run from a simple webcam without any specialist hardware. Finally, OpenFace allows for easy integration with other applications and devices through a lightweight messaging system.

1. Introduction

Over the past few years, there has been an increased interest in machine understanding and recognition of affective and cognitive mental states and interpretation of social signals especially based on facial expression and more broadly facial behavior [18, 51, 39]. As the face is a very important channel of nonverbal communication [20, 18], facial behavior analysis has been used in different applications to facilitate human computer interaction [10, 43, 48, 66]. More recently, there has been a number of developments demonstrating the feasibility of automated facial behavior analysis systems for better understanding of medical conditions such as depression [25] and post traumatic stress disorders [53]. Other uses of automatic facial behavior analysis include automotive industries [14], education [42, 26], and entertainment [47].

In our work we define facial behavior as consisting of: *facial landmark motion*, *head pose* (orientation and motion), *facial expressions*, and *eye gaze*. Each of these modalities play an important role in human behavior, both individually and together. For example automatic detection and analysis of facial Action Units [19] (AUs) is an important building block in nonverbal behavior and emotion recognition systems [18, 51]. This includes detecting both the presence and the intensity of AUs, allowing us to analyse their occurrence, co-occurrence and dynamics. In addition to AUs, head pose and gesture also play an important role in emotion and social signal perception and expression [56, 1, 29]. Finally, gaze direction is important when evaluating things like attentiveness, social skills and mental health, as well as intensity of emotions [35].

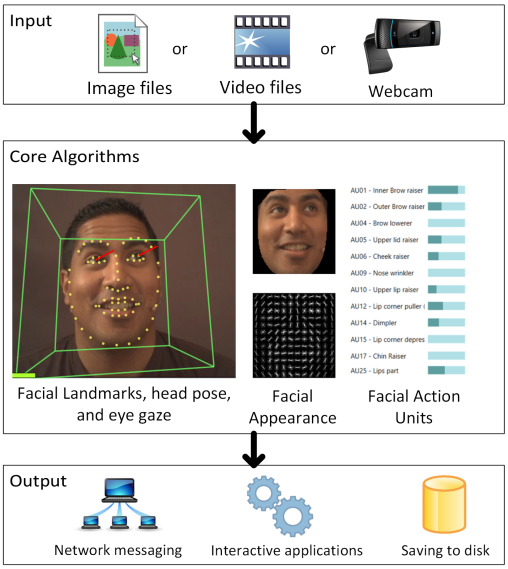


Figure 1: OpenFace is an open source framework that implements state-of-the-art facial behavior analysis algorithms including: facial landmark detection, head pose tracking, eye gaze and facial Action Unit estimation.

Over the past years there has been a huge amount of progress in facial behavior understanding [18, 51, 39]. However, there is still no open source system available to the research community that can do all of the above mentioned tasks (see Table 1). There is a big gap between state-of-the-art algorithms and freely available toolkits. This is especially true if real-time performance is wanted - a necessity for interactive systems .

Furthermore, even though there exist a number of ap-

Tool	Approach	Landmark	Head pose	AU	Gaze	Train	Fit	Binary	Real-time
COFW[13]	RCPR[13]	✓				✓	✓		✓
FaceTracker	CLM[50]	✓	✓				✓	✓	✓
dlib [34]	[32]	✓				✓	✓		✓
DRMF[4]	DRMF[4]	✓	✓					✓	✓
Chehra	[5]	✓	✓					✓	✓
GNDPM	GNDPM[58]	✓						✓	
PO-CR[57]	PO-CR [57]	✓						✓	
Menpo [3]	AAM, CLM, SDM[1]	✓				✓	✓		[2]
CFAN [67]	[67]	✓						✓	✓
[65]	Reg. For [65]	✓	✓			✓	✓	✓	✓
TCDCN	CNN [70]	✓	✓					✓	✓
EyeTab	[63]				✓	N/A	✓	✓	✓
Intraface	SDM [64]	✓	✓					?[3]	✓
OKAO	?	✓	✓	✓	✓			✓	
FACET	?	✓	✓	✓				✓	✓
Affdex	?	✓	✓	✓				✓	✓
Tree DPM [71]	[71]	✓				✓	✓		
LEAR	LEAR [40]	✓						✓	✓
TAUD	TAUD [31]			✓				✓	
OpenFace	[7, 6]	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of facial behavior analysis tools. We do not consider fitting code to be available if the only code provided is a wrapper around a compiled executable. Note that most tools only provide binary versions (executables) rather than the model training and fitting source code. [1] The implementation differs from the originally proposed one based on the used features, [2] the algorithms implemented are capable of real-time performance but the tool does not provide it, [3] the executable is no longer available on the author's website.

proaches for tackling each individual problem, very few of them are available in source code form and would require significant amount of effort to re-implement. In some cases exact re-implementation is virtually impossible due to lack of details in papers. Examples of often omitted details include: values of hyper-parameters, data normalization and cleaning procedures, exact training protocol, model initialization and re-initialization procedures, and optimization techniques to make systems real-time. These details are often as important as the algorithms themselves in order to build systems that work on real world data. Source code is a great way of providing such details. Finally, even the approaches that claim they provide code instead only provide a thin wrapper around a compiled binary making it impossible to know what is actually being computed internally.

OpenFace is not only the first open source tool for facial behavior analysis, it demonstrates state-of-the art performance in facial landmark detection, head pose tracking, AU recognition and eye gaze estimation. It is also able to perform all of these tasks together in real-time. Main contributions of OpenFace are: 1) implements and extends state-of-the-art algorithms; 2) open source tool that includes model training code; 3) comes with ready to use trained models; 4) is capable of real-time performance, without the need of a GPU; 5) includes a messaging system allowing for easy to implement real-time interactive applications; 6) available as a Graphical User Interface (for Windows) and as a command line tool (for Ubuntu, Mac OS X and Windows).

Our work is intended to bridge that gap between existing state-of-the-art research and easy to use out-of-the-box solutions for facial behavior analysis. We believe our tool will stimulate the community by lowering the bar of entry into the field and enabling new and interesting applications[1].

First, we present a brief outline of the recent advances in face analysis tools (section 2). Then we move on to describe our facial behavior analysis pipeline (section 3). We follow, by a description of a large number of experiments to asses our framework (section 4). Finally, we provide a brief description of the interface provided by OpenFace (section 5).

2. Previous work

A full review of work in facial landmark detection, head pose, eye gaze, and action unit estimation is outside the scope of this paper, we refer the reader to recent reviews of the field [17, 18, 30, 46, 51, 61]. We instead provide an

[1] https://www.cl.cam.ac.uk/research/rainbow/projects/openface/

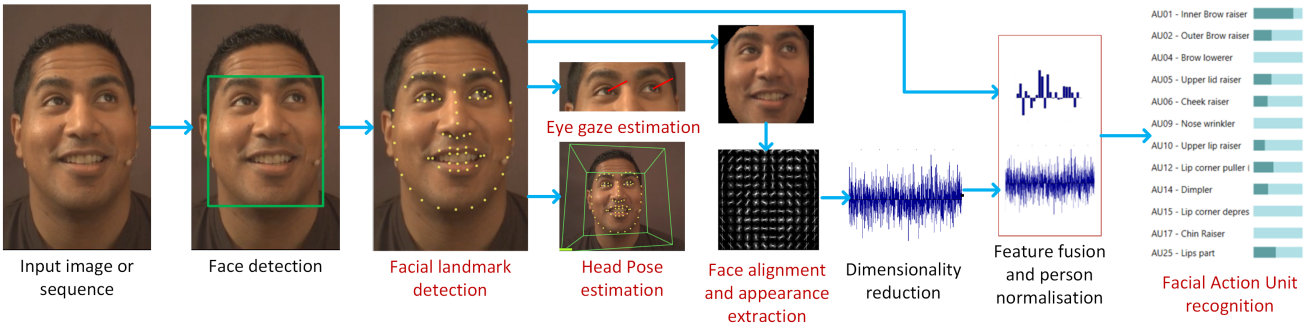


Figure 2: OpenFace facial behavior analysis pipeline, including: *facial landmark detection*, *head pose* and *eye gaze* estimation, facial *action unit* recognition. The outputs from all of these systems (indicated by red) can be saved to disk or sent over a network.

overview of available tools for accomplishing the individual facial behavior analysis tasks. For a summary of available tools see Table 1.

Facial landmark detection - there exists a broad selection of freely available tools to perform facial landmark detection in images or videos. However, very few of the approaches provide the source code and instead only provide executable binaries. This makes the reproduction of experiments on different training sets or using different landmark annotation schemes difficult. Furthermore, binaries only allow for certain predefined functionality and are often not cross-platform, making real-time integration of the systems that would rely on landmark detection almost impossible. Although, there exist several exceptions that provide both training and testing code [3, 71], those approaches do not allow for real-time landmark tracking in videos - an important requirement for interactive systems.

Head pose estimation has not received the same amount of interest as facial landmark detection. An earlier example of a dedicated head pose estimation is the Watson system, which is an implementation of the Generalized Adaptive View-based Appearance Model [45]. There also exist several frameworks that allow for head pose estimation using depth data [21], however they cannot work on webcams. While some facial landmark detectors include head pose estimation capabilities [4, 5], most ignore this problem.

AU recognition - there are very few freely available tools for action unit recognition. However, there are a number of commercial systems that amongst other functionality perform Action Unit Recognition: FACET[2], Affdex[3], and OKAO[4]. However, the drawback of such systems is the sometimes prohibitive cost, unknown algorithms, and often unknown training data. Furthermore, some tools are inconvenient to use by being restricted to a single machine (due to MAC address locking or requiring of USB dongles). Finally, and most importantly, the commercial product may be discontinued leading to impossible to reproduce results due to lack of product transparency (this is illustrated by the recent unavailability of FACET).

Gaze estimation - there are a number of tools and commercial systems for eye-gaze estimation, however, majority of them require specialist hardware such as infrared cameras or head mounted cameras [30, 37, 54]. Although, there exist a couple of systems available for webcam based gaze estimation [72, 24, 63], they struggle in real-world scenarios and some require cumbersome manual calibration steps.

In contrast to other available tools OpenFace provides both training and testing code allowing for easy reproducibility of experiments. Furthermore, our system shows state-of-the-art results on in-the-wild data and does not require any specialist hardware or person specific calibration. Finally, our system runs in real-time with all of the facial behavior analysis modules working together.

3. OpenFace pipeline

In this section we outline the core technologies used by OpenFace for facial behavior analysis (see Figure 2 for a summary). First, we provide an explanation of how we detect and track facial landmarks, together with a hierarchical model extension to an existing algorithm. We then provide an outline of how these features are used for head pose estimation and eye gaze tracking. Finally, we describe our Facial Action Unit intensity and presence detection system, which includes a novel person calibration extension to an existing model.

3.1. Facial landmark detection and tracking

OpenFace uses the recently proposed Conditional Local Neural Fields (CLNF) [8] for facial landmark detection and tracking. CLNF is an instance of a Constrained Local Model (CLM) [16], that uses more advanced patch experts

[2] http://www.emotient.com/products/
[3] http://www.affectiva.com/solutions/affdex/
[4] https://www.omron.com/ecb/products/mobile/

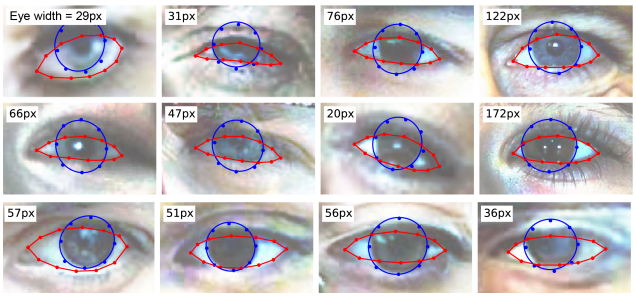


Figure 3: Sample registrations on 300-W and MPIIGaze datasets.

and optimization function. The two main components of CLNF are: Point Distribution Model (PDM) which captures landmark shape variations; patch experts which capture local appearance variations of each landmark. For more details about the algorithm refer to Baltrušaitis et al. [8].

3.1.1 Model novelties

The originally proposed CLNF model performs the detection of all 68 facial landmarks together. We extend this model by training separate sets of point distribution and patch expert models for eyes, lips and eyebrows. We later fit the landmarks detected with individual models to a joint (PDM).

Tracking a face over a long period of time may lead to drift or the person may leave the scene. In order to deal with this, we employ a face validation step. We use a simple three layer convolutional neural network (CNN) that given a face aligned using a piecewise affine warp is trained to predict the expected landmark detection error. We train the CNN on the LFPW [11] and Helen [36] training sets with correct and randomly offset landmark locations. If the validation step fails when tracking a face in a video, we know that our model needs to be reset.

In case of landmark detection in difficult *in-the-wild* images we use multiple initialization hypotheses at different orientations and pick the model with the best converged likelihood. This slows down the approach, but makes it more accurate.

3.1.2 Implementation details

The PDM used in OpenFace was trained on two datasets - LFPW [11] and Helen [36] training sets. This resulted in a model with 34 non-rigid and 6 rigid shape parameters.

For training the CLNF patch experts we used: Multi-PIE [27], LFPW [11] and Helen [36] training sets. We trained a separate set of patch experts for seven views and four scales (leading to 28 sets in total). Having multi-scale patch experts allows us to be accurate both on lower and higher resolution face images. We found optimal results are achieved when the face is at least 100px across. Training on different views allows us to track faces with out of plane motion and to model self-occlusion caused by head rotation.

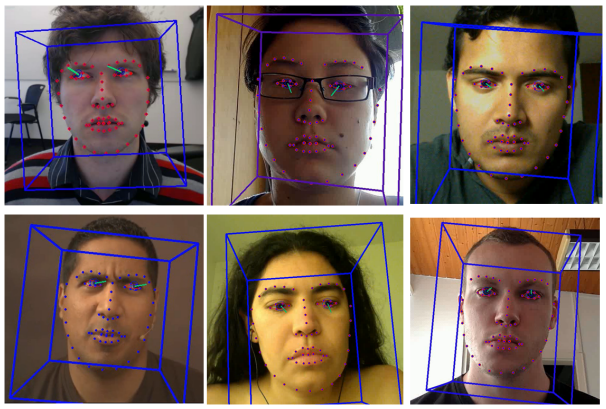


Figure 4: Sample gaze estimations on video sequences; green lines represent the estimated eye gaze vectors.

To initialize our CLNF model we use the face detector found in the dlib library [33, 34]. We learned a simple linear mapping from the bounding box provided by dlib detector to the one surrounding the 68 facial landmarks. When tracking landmarks in videos we initialize the CLNF model based on landmark detections in previous frame. If our CNN validation module reports that tracking failed we reinitialize the model using the dlib face detector.

OpenFace also allows for detection of multiple faces in an image and tracking of multiple faces in videos. For videos this is achieved by keeping a track of active face tracks and a simple logic module that checks for people leaving and entering the frame.

3.2. Head pose estimation

Our model is able to extract head pose (translation and orientation) information in addition to facial landmark detection. We are able to do this, as CLNF internally uses a 3D representation of facial landmarks and projects them to the image using orthographic camera projection. This allows us to accurately estimate the head pose once the landmarks are detected by solving the PnP problem.

For accurate head pose estimation OpenFace needs to be provided with the camera calibration parameters (focal length and principal point). In their absence OpenFace uses a rough estimate based on image size.

3.3. Eye gaze estimation

CLNF framework is a general deformable shape registration approach so we use it to detect eye-region landmarks as well. This includes eyelids, iris and the pupil. We used the SynthesEyes training dataset [62] to train the PDM and

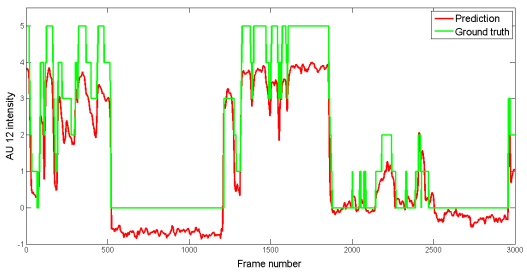


Figure 5: Prediction of AU12 on DISFA dataset [7]. Notice how the prediction is always offset by a constant value.

CLNF patch experts. This model achieves state-of-the-art results in eye-region registration task [62]. Some sample registrations can be seen in Figure 3.

Once the location of the eye and the pupil are detected using our CLNF model we use that information to compute the eye gaze vector individually for each eye. We fire a ray from the camera origin through the center of the pupil in the image plane and compute it's intersection with the eye-ball sphere. This gives us the pupil location in 3D camera coordinates. The vector from the 3D eyeball center to the pupil location is our estimated gaze vector. This is a fast and accurate method for person independent eye-gaze estimation in webcam images. See Figure 4 for sample gaze estimates.

3.4. Action Unit detection

OpenFace AU intensity and presence detection module is based on a recent state-of-the-art AU recognition framework [7, 59]. It is a direct implementation with a couple of changes that adapt it to work better on natural video sequences from unseen datasets. A more detailed explanation of the system can be found in Baltrušaitis et al. [7]. In the following section we describe our extensions to the approach and the implementation details.

3.4.1 Model novelties

In natural interactions people are not expressive very often [2]. This observation allows us to safely assume that most of the time the lowest intensity (and in turn prediction) of each action unit over a long video recording of a person should be zero. However, the existing AU predictors tend to sometimes under- or over-estimate AU values for a particular person, see Figure 5 for an illustration of this.

To correct for such prediction errors, we take the lowest n_{th} percentile (learned on validation data) of the predictions on a specific person and subtract it from all of the predictions. We call this approach – person calibration. Such a correction can be easily implemented in an online system as well by keeping a histogram of previous predictions. This extension only applies to AU intensity prediction.

AU	Full name	Prediction
AU1	Inner brow raiser	I
AU2	Outer brow raiser	I
AU4	Brow lowerer	I
AU5	Upper lid raiser	I
AU6	Cheek raiser	I
AU7	Lid tightener	P
AU9	Nose wrinkler	I
AU10	Upper lip raiser	I
AU12	Lip corner puller	I
AU14	Dimpler	I
AU15	Lip corner depressor	I
AU17	Chin raiser	I
AU20	Lip stretched	I
AU23	Lip tightener	P
AU25	Lips part	I
AU26	Jaw drop	I
AU28	Lip suck	P
AU45	Blink	P

Table 2: List of AUs in OpenFace. I - intensity, P - presence.

Another extension we propose is to combine AU presence and intensity training datasets. Some datasets only contain labels for action unit presence (SEMAINE [44] and BP4D) and others contain labels for their intensities (DISFA [41] and BP4D [69]). This makes the training on combined datasets not straightforward. We use the distance to the hyperplane of the trained SVM model as a feature for an SVR regressor. This allows us to train a single predictor using both AU presence and intensity datasets.

3.4.2 Implementation details

In order to extract facial appearance features we used a similarity transform from the currently detected landmarks to a representation of frontal landmarks from a neutral expression. This results in a 112×112 pixel image of the face with 45 pixel interpupilary distance (similar to Baltrušaitis et al.[7]).

We extract Histograms of Oriented Gradients (HOGs) features as proposed by Felzenswalb *et al.* [23] from the aligned face. We use blocks of 2×2 cells, of 8×8 pixels, leading to 12×12 blocks of 31 dimensional histograms (4464 dimensional vector describing the face). In order to reduce the feature dimensionality we use a PCA model trained on a number of facial expression datasets: CK+ [38], DISFA [41], AVEC 2011 [52], FERA 2011 [60], and FERA 2015 [59]. Applying PCA to images (sub-sampling from peak and neutral expressions) and keeping 95% of explained variability leads to a reduced basis of 1391 dimensions. This allows for a generic basis, more suitable to unseen datasets.

We note that our framework allows the saving of these intermediate features (aligned faces together with actual and dimensionality reduced HOGs), as they are useful for a number of facial behavior analysis tasks.

For AU presence prediction OpenFace uses a linear kernel SVM and for AU intensity a linear kernel SVR. As features we use the concatenation of dimensionality reduced HOGs and facial shape features (from CLNF). In order to account for personal differences the median value of the features (observed so far in online case and overall for offline processing) is subtracted from the estimates in the current frame. This has been shown to be cheap and effective way to increase model performance [7].

Our models are trained on DISFA [41], SEMAINE [44] and BP4D [69] datasets. Where the AU labels overlap across multiple datasets we train on them jointly. This leads to OpenFace recognizing the AU listed in Table 2.

4. Experimental evaluation

In this section, we evaluate each of our OpenFace subsytems: facial landmark detection, head pose estimation, eye gaze estimation, and facial Action Unit detection. For each of our experiments we also include comparisons with a number of recently proposed approaches for tackling the same problems (although none of them tackle all of them at once). Furthermore, all of the approaches we compared against provide only binaries with pre-trained models and not the full training and testing code (except for EyeTab [63] and regression forests [21]).

4.1. Landmark detection

The facial landmark detection capability was evaluated on the 300-W face validation dataset which comprises of four sub-datasets: Annotated Faces in the Wild (**AFW**)[71], **IBUG** [49], **LFPW**[11], and **Helen** [36]. For initialization we used the bounding boxes provided by the challenge organizers.

First, we evaluated the benefit of our proposed hierarchical model. The results can be seen in 6a. It can be seen that the hierarchical model leads to better facial landmark detection accuracies.

As a second experiment, we compared our approach to other facial landmark detection algorithms whose implementations are available online and which have been trained to detect the same facial landmarks (or their subsets). The baselines were: Discriminative Response Map Fitting (DRMF) [4], tree based deformable models [71], extended version of Constrained Local Models [6], Gauss-Newton Deformable Parts Model (GNDPM) [58], and Supervised Descent Method (SDM) [64].

The results can be seen in Figure 6. For reporting of 49 landmark detection results we only used the 865 images for which all of our baselines were able to detect faces, another issue with provided binaries (and not the code) is that we sometimes cannot change the face detector used. OpenFace demonstrates state-of-the-art performance and alongside tree based models [71] is the only model that provides both model training and fitting source code.

Method	Yaw	Pitch	Roll	Mean	Median
Reg. forests [22]	9.2	8.5	8.0	8.6	N/A
CLM [50]	8.2	8.2	6.5	7.7	3.3
CLM-Z [9]	8.0	6.1	6.0	6.7	3.2
Chehra [5]	13.9	14.7	10.2	12.9	5.4
OpenFace	**7.9**	**5.6**	**4.5**	**6.0**	**2.6**

Table 3: Head pose estimation results on the Biwi Kinect head pose dataset. Measured in mean absolute degree error.

Method	Yaw	Pitch	Roll	Mean	Median
CLM [50]	3.0	3.5	2.3	2.9	2.0
Chehra [5]	3.8	4.6	2.8	3.8	2.5
OpenFace	**2.8**	**3.3**	**2.3**	**2.8**	**2.0**

Table 4: Head pose estimation results on the BU dataset. Measured in mean absolute degree error. Note that BU dataset only contains RGB images so no comparison agains CLM-Z and Regression forests was perfomed.

Method	Yaw	Pitch	Roll	Mean
Reg. forests [22]	7.2	9.4	7.5	8.0
CLM-Z [9]	5.1	3.9	4.6	4.6
CLM [50]	4.8	4.2	4.5	4.5
Chehra [5]	13.9	14.7	10.3	13.0
OpenFace	**3.6**	**3.6**	**3.6**	**3.6**

Table 5: Head pose estimation results on ICT-3DHP. Measured in mean absolute degree error.

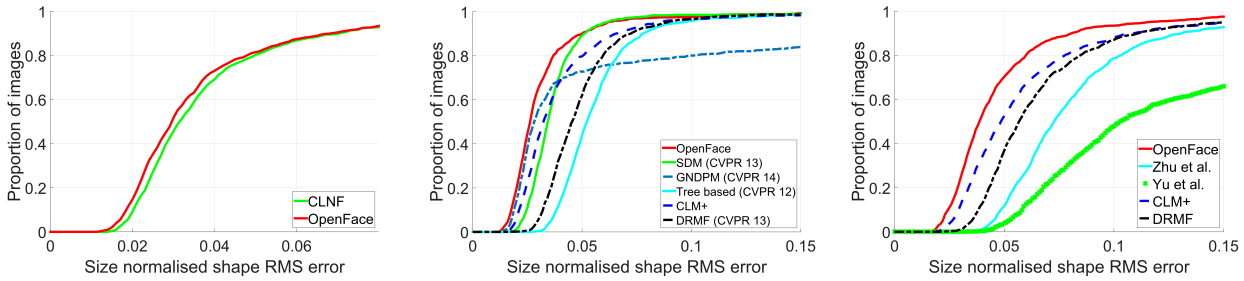
4.2. Head pose estimation

To measure OpenFace performance on a head pose estimation task we used three publicly available datasets with existing ground truth head pose data: BU [15], Biwi [21] and ICT-3DHP [9].

For comparison, we report the results of using Chehra framework [5], CLM [50], CLM-Z [9], and Regression Forests [21]. The results can be see in Table 3, Table 4 and Table 5. It can be seen that our approach demonstrates state-of-the-art performance on all three of the datasets.

4.3. Eye gaze estimation

We evaluated the ability of OpenFace to estimate eye gaze vectors by evaluating it on the challenging MPIIGaze dataset [68] intended to evaluate appearance based gaze es-



(a) Hierarchical (b) No jawline (c) All points

Figure 6: Fitting on the wild datasets using the CLNF approach included in OpenFace compared against state-of-the-art methods. All of the methods have been trained on in the wild data from different than test datasets a) Benefit of our hierarchical extension b) Comparison of detection of 49 landmark points (without the jawline) c) Comparison of detection of 68 landmark points (with the jawline). The reason some approaches were evaluated only with 49 point models is that not all authors release trained 68 point models.

AU	1	2	4	5	6	9	12	15	17	20	25	26	Mean
No callibration	0.55	**0.44**	0.58	0.36	**0.57**	0.43	0.82	0.27	0.31	0.16	0.80	0.56	0.49
Callibration	**0.57**	0.42	**0.65**	**0.57**	0.54	**0.51**	0.82	0.27	0.31	**0.23**	**0.86**	**0.63**	**0.53**

Table 6: Benefit of person specific output calibration. The difference is statistically significant (paired t test $p < 0.05$)

MODEL	GAZE ERROR
EyeTab [63]	47.1
CNN on UT [68]	13.91
CNN on SynthesEyes [62]	13.55
CNN on SynthesEyes + UT [62]	11.12
OpenFace	**9.96**

Table 7: Results comparing our method to previous work for cross dataset gaze estimation on MPIIGaze [68], measure in mean absolute degree error.

	6	10	12	14	17	μ
	Fully automatic					
BG [59]	0.67	**0.73**	0.78	**0.59**	0.14	0.58
BA [59]	0.62	0.66	0.77	0.39	0.17	0.52
DL [28]	0.66	0.73	0.79	0.55	0.33	0.61
OF	**0.69**	**0.73**	**0.83**	0.50	**0.37**	**0.62**
	Pre-segmented					
BG [59]	0.48	**0.51**	0.69	**0.59**	0.05	0.46
BA [59]	0.33	0.48	0.60	0.50	0.11	0.40
DL [28]	0.42	0.54	0.61	0.50	0.22	0.46
OF	**0.58**	0.49	**0.70**	0.52	**0.41**	**0.54**

Table 8: AU intensity results (intra-class correlation coefficient) on FERA 2015 test dataset comparing against their proposed appearance and geometry based baselines[59].

timation. MPIIGaze was collected in realistic laptop use scenarios and poses a challenging and practically-relevant task for eye gaze estimation. Sample images from the dataset can be seen in the right two columns of Figure 4.

We evaluated our approach on a 750 face image subset of the dataset - leading to 1500 eye images (one per eye). We did not use the manually labeled eye corner location provided with the dataset but used the full pipeline from OpenFace. The error rates of our model can be seen in Table 7.

4.4. Action Unit recognition

We performed AU recognition experiments on three publicly available datasets: SEMAINE, DISFA, and BP4D. The evaluation was done in a person independent manner.

In our first experiment we validated our person calibration extension on the DISFA dataset The results can be seen in Table 6. It can be clearly seen that our calibration scheme leads to more better overall AU intensity prediction.

As a second experiment, we submitted an earlier version of OpenFace to the 2015 Facial Expression Recognition and Analysis (FERA2015) challenge [59]. The challenge organizers evaluated it on an unseen (and unreleased) subset of SEMAINE and BP4D datasets. The system was evaluated in both AU presence and intensity prediction tasks. The results on the challenge data can be seen in Table 9 and Table 8.

Note that the OpenFace system has been extended since then (as outlined in the previous sections), but as the challenge data was not released we are unable to provide the

	BP4D											SEMAINE						
AU	1	2	4	6	7	10	12	14	15	17	23	2	12	17	25	28	45	Mean
BG [59]	0.19	0.19	0.20	0.65	**0.80**	0.80	0.80	**0.72**	0.24	0.31	0.32	0.57	0.60	0.09	0.45	0.25	0.40	0.45
BA [59]	0.18	0.16	0.23	0.67	0.75	0.80	0.79	0.67	0.14	0.25	0.24	**0.76**	0.52	0.07	0.40	0.01	0.21	0.40
DL [28]	**0.40**	**0.35**	**0.32**	0.72	0.78	0.80	0.79	0.68	0.23	**0.37**	0.31	0.37	**0.71**	0.07	0.60	0.04	0.26	0.46
OF	0.26	0.25	0.25	**0.73**	**0.80**	**0.84**	**0.82**	**0.72**	**0.34**	0.33	**0.34**	0.41	0.57	**0.20**	**0.69**	**0.26**	**0.42**	**0.48**

Table 9: AU occurrence results on FERA 2015 test dataset (F1). Only OpenFace (OF) provides a full out-of-the-box system.

AU	1	2	4	5	6	9	10	12	14	15	17	20	25	26	Mean
OpenFace$_d$	0.40	0.46	0.72	0.74	0.52	0.69	0.61	0.88	0.28	0.53	0.28	0.24	0.87	0.65	0.56
OpenFace$_s$	0.27	0.02	0.66	0.55	0.41	0.23	0.68	0.87	0.38	0.05	0.32	0.30	0.85	0.53	0.43

Table 10: Evaluating OpenFace on DISFA (5 unseen subjects), and BP4D (for AU10 and AU14). The target subjects were chosen using stratified cross-validation. Dynamic models (OpenFace$_d$) use calibration and neutral expression subtraction, whereas static models (OpenFace$_s$) rely on a single image of an individual. The dynamic models seem to be particularly important for AUs that might involve wrinkling of the face. The results are reported in Canonical Correlation Coefficients.

AU	7	23	28	45	Mean
Dynamic	0.74	0.37	0.36	0.40	0.47
Static	0.75	0.36	0.30	0.31	0.43

Table 11: Evaluating OpenFace classifiers (F1 scores) on SEMAINE (28, 45) and BP4D (AU7 AU23) FERA 2015 validation sets.

results of the newest system on the FERA2015 test sets. Because of this, we evaluated OpenFace on three publicly available datasets. The results for AU intensity can be found in Table 10 and presence in Table 11. Our system was specifically tailored for Action Unit recognition in videos rather than individual images, hence the performance of the dynamic models is much higher.

The recognition of certain AUs is not as reliable as that of others partly due to lack of representation in training data and inherent difficulty of the problem. This is an area of OpenFace that is still under active development and that will continue to be refined with time, especially as more datasets become available.

5. Interface

OpenFace is an easy to use toolbox for the analysis of facial behavior. There are three main ways of using OpenFace: Graphical User Interface, command line, and real-time messaging system (based on ZeroMQ). As the system is open source it is also possible to integrate it in any C++ or C$^\sharp$ based project. To make the system easier to use we provide sample Matlab scripts that demonstrate how to extract, save, read and visualize each of the behaviors. The system is cross-platform and has been tested on Windows, Ubuntu and Mac OS X.

OpenFace can operate on real-time data video feeds from a webcam, recorded video files, image sequences and individual images. It is possible to save the outputs of the processed data as CSV files in case of facial landmarks, shape parameters, Action Units and gaze vectors. HOG features are saved as Matlab readable binary streams, and aligned face images are saved as either image sequences or videos. Moreover, it is possible to load the saved behaviors into ELAN [12] for easy visualization. Example use case of saving facial behaviors using OpenFace would involve using them as features for emotion prediction, medical condition analysis, and social signal analysis systems.

Finally, OpenFace can be easily used to build real-time interactive applications that rely on various facial analysis subsystems. This is achieved by using a lightweight messaging system - ZeroMQ [5]. It allows to send estimated facial behaviors over a network to anyone requesting the features. Such a system has already been used in ophthalmology research [55]. We also provide examples in Python and C++ to show examples of listening to ZeroMQ messages from OpenFace in real time.

6. Conclusion

In this paper we presented OpenFace – a first fully open source real-time facial behavior analysis system. OpenFace is a useful tool for the computer vision, machine learning and affective computing communities and will stimulate research in facial behavior analysis an understanding. Furthermore, the future development of the tool will continue and it will attempt to incorporate the newest and most reliable approaches for the problem at hand while remaining a transparent open source tool and retaining its real-time capacity. We hope that this tool will encourage other researchers in the field to share their code.

[5] http://zeromq.org/

References

[1] A. Adams, M. Mahmoud, T. Baltrušaitis, and P. Robinson. Decoupling facial expressions and head motions in complex emotions. In *Affective Computing and Intelligent Interaction*, 2015.

[2] S. Afzal and P. Robinson. Natural Affect Data - Collection & Annotation in a Learning Context. *Design*, 2009.

[3] J. Alabort-i medina, E. Antonakos, J. Booth, and P. Snape. Menpo : A Comprehensive Platform for Parametric Image Alignment and Visual Deformable Models Categories and Subject Descriptors. pages 3–6, 2014.

[4] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013.

[5] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental Face Alignment in the Wild. In *CVPR*, 2014.

[6] T. Baltrušaitis, N. Banda, and P. Robinson. Dimensional affect recognition using continuous conditional random fields. In *FG*, 2013.

[7] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specic normalisation for automatic Action Unit detection. In *Facial Expression Recognition and Analysis Challenge, in conjunction with FG*, 2015.

[8] T. Baltrušaitis, L.-P. Morency, and P. Robinson. Constrained local neural fields for robust facial landmark detection in the wild. In *ICCVW*, 2013.

[9] T. Baltrušaitis, P. Robinson, and L.-P. Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *CVPR*, 2012.

[10] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *CVPR Workshops*, 2003.

[11] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[12] H. Brugman and A. Russel. Annotating multi-media / multimodal resources with ELAN. *Proceedings of the 4th International Conference on Language Resources and Language Evaluation (LREC 2004)*, pages 2065–2068, 2004.

[13] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *International Conference on Computer Vision*, 2013.

[14] C. Busso and J. J. Jain. Advances in Multimodal Tracking of Driver Distraction. In *Digital Signal Processing for in-Vehicle Systems and Safety*, pages 253–270. 2012.

[15] M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, Reliable Head Tracking under Varying Illumination : An Approach Based on Registration of Texture-Mapped 3D Models. *TPAMI*, 22(4), 2000.

[16] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006.

[17] B. Czupryski and A. Strupczewski. High accuracy head pose tracking survey. In D. lzak, G. Schaefer, S. Vuong, and Y.-S. Kim, editors, *Active Media Technology*, volume 8610 of *Lecture Notes in Computer Science*, pages 407–420. Springer International Publishing, 2014.

[18] F. De la Torre and J. F. Cohn. Facial Expression Analysis. In *Guide to Visual Analysis of Humans: Looking at People*. 2011.

[19] P. Ekman and W. V. Friesen. *Manual for the Facial Action Coding System*. Palo Alto: Consulting Psychologists Press, 1977.

[20] P. Ekman, W. V. Friesen, M. O'Sullivan, and K. R. Scherer. Relative importance of face, body, and speech in judgments of personality and affect. *Journal of Personality and Social Psychology*, 38:270–277, 1980.

[21] G. Fanelli, J. Gall, and L. V. Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624, 2011.

[22] G. Fanelli, T. Weise, J. Gall, and L. van Gool. Real time head pose estimation from consumer depth cameras. In *DAGM*, 2011.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminative Trained Part Based Models. *IEEE TPAMI*, 32, 2010.

[24] O. Ferhat and F. Vilariño. A Cheap Portable Eye–tracker Solution for Common Setups. *3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, 2013.

[25] J. M. Girard, J. F. Cohn, M. H. Mahoor, S. Mavadati, and D. P. Rosenwald. Social risk and depression: Evidence from manual and automatic facial expression analysis. In *FG*, 2013.

[26] A. Graesser and A. Witherspoon. Detection of Emotions during Learning with AutoTutor. *Annual Meetings of the Cognitive Science Society*, pages 285–290, 2005.

[27] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *IVC*, 2010.

[28] A. Gudi, H. E. Tasli, T. M. D. Uyl, and A. Maroulis. Deep Learning based FACS Action Unit Occurrence and Intensity Estimation. In *Facial Expression Recognition and Analysis Challenge, in conjunction with FG*, 2015.

[29] Z. Hammal, J. F. Cohn, C. Heike, and M. L. Speltz. What Can Head and Facial Movements Convey about Positive and Negative Affect? In *Affective Computing and Intelligent Interaction*, 2015.

[30] D. W. Hansen and Q. Ji. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[31] B. Jiang, M. F. Valstar, and M. Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. *20FG*, 2011.

[32] V. Kazemi and J. Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014.

[33] D. E. King. Dlib-ml: A machine learning toolkit. *JMLR*, 2009.

[34] D. E. King. Max-margin object detection. *CoRR*, 2015.

[35] C. L. Kleinke. Gaze and eye contact: a research review. *Psychological bulletin*, 100(1):78–100, 1986.

[36] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012.

[37] M. Lidegaard, D. W. Hansen, and N. Krüger. Head mounted device for point-of-gaze estimation in three dimensions. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*, 2014.

[38] P. Lucey, J. F. Cohn, T. Kanade, J. M. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, 2010.

[39] B. Martinez and M. Valstar. Advances, challenges, and opportunities in automatic facial expression recognition. In B. S. M. Kawulok, E. Celebi, editor, *Advances in Face Detection and Facial Image Analysis*. Springer, 2015. In press.

[40] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic. Local evidence aggregation for regression based facial point detection. *TPAMI*, 35, 2013.

[41] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. Disfa : A spontaneous facial action intensity database. *IEEE T-AFFC*, 2013.

[42] B. McDaniel, S. D'Mello, B. King, P. Chipman, K. Tapp, and a. Graesser. Facial Features for Affective State Detection in Learning Environments. *29th Annual meeting of the cognitive science society*, pages 467–472, 2007.

[43] D. McDuff, R. el Kaliouby, D. Demirdjian, and R. Picard. Predicting online media effectiveness based on smile responses gathered over the internet. In *FG*, 2013.

[44] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic. The SEMAINE corpus of emotionally coloured character interactions. In *IEEE International Conference on Multimedia and Expo*, 2010.

[45] L.-P. Morency, J. Whitehill, and J. R. Movellan. Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *FG*, 2008.

[46] E. Murphy-Chutorian and M. M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. *TPAMI*, 31:607–626, 2009.

[47] Paris Mavromoustakos Blom, S. Bakkes, C. T. Tan, S. Whiteson, D. Roijers, R. Valenti, and T. Gevers. Towards Personalised Gaming via Facial Expression Recognition. *Proceedings of Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, (AIIDE):30–36, 2014.

[48] P. Robinson and R. el Kaliouby. Computation of emotions in man and machines. *Philosophical Transactions B*, pages 3441–3447, 2009.

[49] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV*, 2013.

[50] J. Saragih, S. Lucey, and J. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *IJCV*, 2011.

[51] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation and recognition. *IEEE TPAMI*, 2014.

[52] B. Schuller, M. F. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic. AVEC 2011 The First International Audio / Visual Emotion Challenge. In *ACII*, 2011.

[53] G. Stratou, S. Scherer, J. Gratch, and L.-P. Morency. Automatic nonverbal behavior indicators of depression and ptsd: Exploring gender differences. In *ACII*, 2013.

[54] L. Świrski, A. Bulling, and N. A. Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of ETRA*, 2012.

[55] P. Thomas, T. Baltrušaitis, P. Robinson, and A. Vivian. Measuring head posture in 3 dimensions with the Cambridge Face Tracker: accurate, inexpensive, easy to use. In *World Congress of Pediatric Ophthalmology and Strabismus*, 2015.

[56] J. L. Tracy and D. Matsumoto. The spontaneous expression of pride and shame: Evidence for biologically innate nonverbal displays. *Proceedings of the National Academy of Sciences*, 105(33):11655–11660, 2008.

[57] G. Tzimiropoulos. Project-Out Cascaded Regression with an application to Face Alignment. *Cvpr*, 2015.

[58] G. Tzimiropoulos and M. Pantic. Gauss-Newton Deformable Part Models for Face Alignment In-the-Wild. *Computer Vision and Pattern Recognition*, (c):1851–1858, 2014.

[59] M. Valstar, J. Girard, T. Almaev, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. Cohn. FERA 2015 - Second Facial Expression Recognition and Analysis Challenge. In *IEEE FG*, 2015.

[60] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. R. Scherer. The First Facial Expression Recognition and Analysis Challenge. In *IEEE FG*, 2011.

[61] N. Wang, X. Gao, D. Tao, and X. Li. Facial feature point detection: A comprehensive survey. *CoRR*, abs/1410.1037, 2014.

[62] E. Wood, T. Baltrušaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *ICCV*, 2015.

[63] E. Wood and A. Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of ETRA*, Mar. 2014.

[64] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[65] H. Yang and I. Patras. Sieving Regression Forest Votes for Facial Feature Detection in the Wild. In *ICCV*, 2013.

[66] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *TPAMI*, 31(1):39–58, 2009.

[67] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-Fine Auto-encoder Networks (CFAN) for Real-time Face Alignment. In *ECCV*, pages 1–16, 2014.

[68] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. June 2015.

[69] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. BP4D-Spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *IVC*, 2014.

[70] Z. Zhang, P. Luo, C.-C. Loy, and X. Tang. Facial Landmark Detection by Deep Multi-task Learning. *Eccv*, 2014.

[71] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[72] P. Zieliński. Opengazer: open-source gaze tracker for ordinary webcams, 2007.