OpenFace 2.0: Facial Behavior Analysis Toolkit

Tadas Baltrušaitis[1,2], Amir Zadeh[2], Yao Chong Lim[2], and Louis-Philippe Morency[2]
[1] Microsoft, Cambridge, United Kingdom
[2] Carnegie Mellon University, Pittsburgh, United States of America

***Abstract*— Over the past few years, there has been an increased interest in automatic facial behavior analysis and understanding. We present OpenFace 2.0 — a 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 2.0 is an extension of OpenFace toolkit (created by Baltrušaitis et al. [11]) and is capable of more accurate facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. The computer vision algorithms which represent the core of OpenFace 2.0 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, unlike a lot of modern approaches or toolkits, OpenFace 2.0 source code for training models and running them is freely available for research purposes.**

I. Introduction

Recent years have seen an increased interest in machine analysis of faces [58], [45]. This includes understanding and recognition of affective and cognitive mental states, and interpretation of social signals. As the face is a very important channel of nonverbal communication [23], [20], facial behavior analysis has been used in different applications to facilitate human computer interaction [47], [50]. 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 [28], post traumatic stress disorders [61], schizophrenia [67], and suicidal ideation [40]. Other uses of automatic facial behavior analysis include automotive industries [14], education [49], and entertainment [55].

In our work we define facial behavior as consisting of: *facial landmark location*, *head pose*, *eye gaze*, and *facial expressions*. Each of these behaviors play an important role together and individually. Facial landmarks allow us to understand facial expression motion and its dynamics, they also allow for face alignment for various tasks such as gender detection and age estimation. Head pose plays an important role in emotion and social signal perception and expression [63], [1]. Gaze direction is important when evaluating things like attentiveness, social skills and mental health [65], as well as intensity of emotions [39]. Facial expressions reveal intent, display affection, express emotion, and help regulate turn-taking during conversation [3], [22].

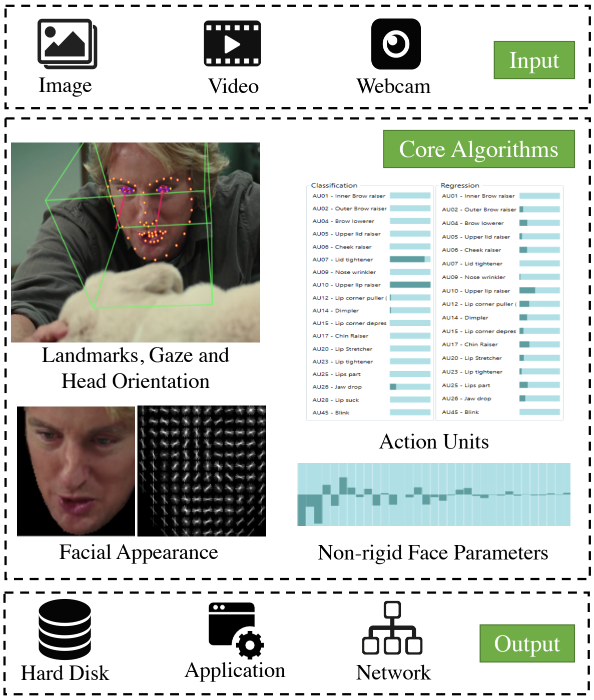


Fig. 1: OpenFace 2.0 is a framework that implements modern facial behavior analysis algorithms including: facial landmark detection, head pose tracking, eye gaze and facial action unit recognition.

Past years have seen huge progress in automatic analysis of above mentioned behaviors [20], [58], [45]. However, very few tools are available to the research community that can recognize all of them (see Table I). There is a large gap between state-of-the-art algorithms and freely available toolkits. This is especially true when real-time performance is wanted — a necessity for interactive systems.

OpenFace 2.0 is an extension of the OpenFace toolkit [11]. While OpenFace is able to perform the above mentioned tasks, it struggles when the faces are non-frontal or occluded and in low illumination conditions. OpenFace 2.0 is able to cope with such conditions through the use of a new Convolutional Neural Network based face detector and a new and optimized facial landmark detection algorithm. This leads to improved accuracy for facial landmark detection, head

This research is based upon work supported in part by the Yahoo! InMind project and the Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014-14071600011. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the ofcial policies or endorsements, either expressed or implied IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

978-1-5386-2335-0/18/$31.00 ©2018 IEEE

Tool	Approach	Landmark	Head pose	Expression	Gaze	Train	Test	Binary	Real-time	Free
COFW[13]	RCPR[13]	✓				✓	✓		✓	✓
FaceTracker	CLM[57]	✓	✓				✓	✓	✓	✓
dlib [37]	[35]	✓				✓	✓		✓	✓
Chehra	[5]	✓	✓					✓	✓	✓
Menpo [2]	AAM, CLM, SDM[1]	✓				✓	✓		[2]	✓
CFAN [77]	[77]	✓						✓	✓	✓
[73]	Reg. For [73]	✓	✓			✓	✓	✓	✓	✓
TCDCN	CNN [81]	✓	✓					✓	✓	✓
WebGazer.js	[54]				✓		✓	✓	✓	✓
EyeTab	[71]				✓	N/A	✓	✓	✓	✓
OKAO	unknown	✓	✓	✓	✓			✓		
Affdex	unknown	✓	✓	✓				✓	✓	
Tree DPM [85]	[85]	✓				✓	✓			✓
OpenPose [15]	Part affinity Fields [15]	✓	✓			✓	✓	✓	✓[3]	✓
CFSS [83]	CFSS [83]	✓				✓	✓			✓
iCCR [56]	iCCR [56]	✓						✓	✓	✓
LEAR	LEAR [46]	✓						✓	✓	✓
TAUD	TAUD [33]			✓				✓		✓
OpenFace	[8], [7]	✓	✓	✓	✓	✓	✓	✓	✓	✓
OpenFace 2.0	[70], [75], [78]	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE I: Comparison of facial behavior analysis tools. *Free* indicates that the tool is freely available for research purposes, *Train* the availability for model training source code, *Test* the availability of model fitting/testing/runtime source code, *Binary* the availability of model fitting/testing/runtime executable. Note that most tools only provide binary versions (executables) rather than the source code for model training and fitting. Notes: ([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]) requires GPU support.

pose tracking, AU recognition and eye gaze estimation. Main contributions of OpenFace 2.0 are: 1) new and improved facial landmark detection system; 2) distribution of ready to use trained models; 3) real-time performance, without the need of a GPU; 4) cross-platform support (Windows, OSX, Ubuntu); 5) code available in C++ (runtime), Matlab (runtime and model training), and Python (model training).

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].

II. Previous work

A full review of prior work in facial landmark detection, head pose, eye gaze, and action unit recognition is outside the scope of this paper, we refer the reader to recent reviews in these respective fields [18], [31], [58], [17]. As our contribution is a toolkit, we provide an overview of available tools for accomplishing the individual facial behavior analysis tasks. For a summary of available tools see Table I.

Facial landmark detection – there exists a number of freely available tools that perform facial landmark detection in images or videos, in part thanks to availability of recent good quality datasets and challenges [60], [76]. However, very few of them provide the source code and instead only provide runtime binaries, or thin wrappers around library files. Binaries only allow for certain predefined functionality (e.g. only visualizing the results), are very rarely cross-platform, and do not allow for bug fixes — an important consideration when the project is no longer actively supported. Further, lack of training code makes the reproduction of experiments on different datasets very difficult. Finally, a number of tools expect face detections (in form of bounding boxes) to be provided by an external tool, in contrast OpenFace 2.0 comes packaged with a modern face detection algorithm [78].

Head pose estimation has not received the same amount of interest as facial landmark detection. An early example of a dedicated head pose estimation toolkit is the Watson system [52]. There also exists a random forest based framework that allows for head pose estimation using depth data [24]. While some facial landmark detectors include head pose estimation capabilities [4], [5], most ignore this important behavioral cue. A more recent toolkit for head (and the rest of the body) pose estimation is OpenPose [15], however, it is computationally demanding and requires GPU acceleration to achieve real-time performance.

Facial expression is often represented using facial action units (AUs), which objectively describe facial muscle activations [21]. There are very few freely available tools for action unit recognition (see Table I). However, there are a number of commercial systems that among other functionality perform action unit recognition, such as: Affdex[2], Noldus FaceReader [3], and OKAO[4]. Such systems face a number of drawbacks: sometimes prohibitive cost, unknown algorithms, often unknown training data, and no public benchmarks. Furthermore, some tools are inconvenient to use

[1] https://github.com/TadasBaltrusaitis/OpenFace

[2] http://www.affectiva.com/solutions/affdex/

[3] http://www.noldus.com/human-behavior-research/products/facereader

[4] https://www.omron.com/ecb/products/mobile/

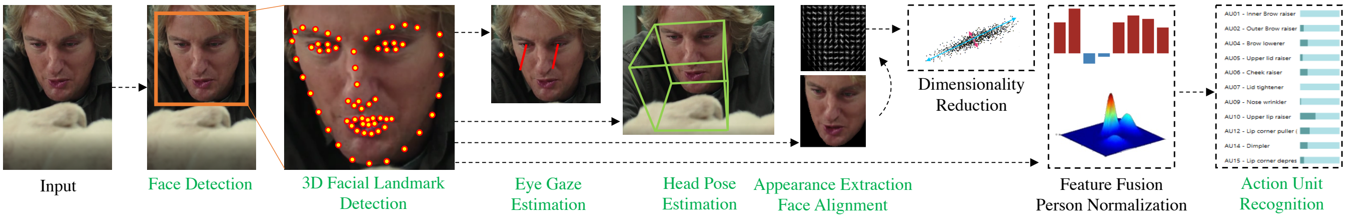


Fig. 2: OpenFace 2.0 facial behavior analysis pipeline, including: *landmark detection*, *head pose* and *eye gaze* estimation, facial *action unit* recognition. The outputs from all of these systems (indicated in green) can be saved to disk or sent via network in real-time.

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 discontinuation of FACET, FaceShift, and IntraFace).

Gaze estimation – there are a number of tools and commercial systems for gaze estimation, however, majority of them require specialized hardware such as infrared or head mounted cameras [19], [42], [62]. There also exist a couple of commercial systems available for webcam based gaze estimation, such as xLabs[5] and EyesDecide[6], but they suffer from previously mentioned issues that commercial facial expression analysis systems do. There exist several recent free webcam eye gaze tracking projects [27], [71], [54], [68], but they struggle in real-world scenarios and often require cumbersome manual calibration steps.

In contrast to other available tools (both free and commercial) OpenFace 2.0 provides both training and testing code allowing for modification, reproducibility, and transparency. Furthermore, our system shows competitive results on real world data and does not require any specialized hardware. Finally, our system runs in real-time with all of the facial behavior analysis modules working together.

III. OpenFace 2.0 pipeline

In this section we outline the core technologies used by OpenFace 2.0 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 novel speed enhancements that allow for real-time performance. 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.

A. *Facial landmark detection and tracking*

OpenFace 2.0 uses the recently proposed Convolutional Experts Constrained Local Model (CE-CLM) [75] for facial landmark detection and tracking. The two main components of CE-CLM are: Point Distribution Model (PDM) which captures landmark shape variations and patch experts which model local appearance variations of each landmark. For more details about the algorithm refer to Zadeh et al. [75], example landmark detections can be seen in Figure 3.

1) OpenFace 2.0 novelties: Our C++ implementation of CE-CLM in OpenFace 2.0 includes a number of speed optimizations that enable real-time performance. These include deep model simplification, smart multiple hypotheses, and sparse response map computation.

Deep model simplification The original implementation of CE-CLM used deep networks for patch experts with $\approx 180,000$ parameters each (for 68 landmarks at 4 scales and 7 views). We retrained the patch experts for first two scales using simpler models by narrowing the deep network to half the width, leading to $\approx 90,000$ parameters each. We chose the final model size after exploring a large range of alternatives, and chose the smallest model that still retains competitive accuracy. This reduces the model size and improves the speed by 1.5 times, with minimal loss in accuracy. Furthermore, we only store half of the patch experts, by relying on mirrored views for response computation (e.g. we store only the left eye recognizesr, instead of both eyes). This reduces the model size by a half. Both of these improvements reduced the model size from $\approx 1,200$MB to ≈ 400MB.

Smart multiple hypotheses In case of landmark detection in difficult *in-the-wild* and profile images CE-CLM uses multiple initialization hypotheses (11 in total) at different orientations. During fitting it selects the model with the best converged likelihood. However, this slows down the approach. In order to speed this up we perform an early hypothesis termination, based on current model likelihood. We start by evaluating the first scale (out of four different scales) for each initialization hypothesis sequentially. If the current likelihood is above a threshold τ_i (good enough), we do not evaluate further hypotheses. If none of the hypotheses are above τ_i, we pick three hypotheses with the highest likelihood for evaluation in further scales and pick the best resulting one. We determine the τ_i values that lead to small fitting errors on for each view on training data. This leads to a 4 time performance improvement of landmark detection in images and for initializing tracking in videos.

Sparse response maps An important part of CE-CLM is the computation of response maps for each facial landmark. Typically it is calculated in a dense grid around the current landmark estimate (e.g. 15×15 pixel grid). However, instead of computing the response map for a dense grid we can do it in a sparse grid by skipping every other pixel, followed by

[5] https://xlabsgaze.com/
[6] https://www.eyesdecide.com/

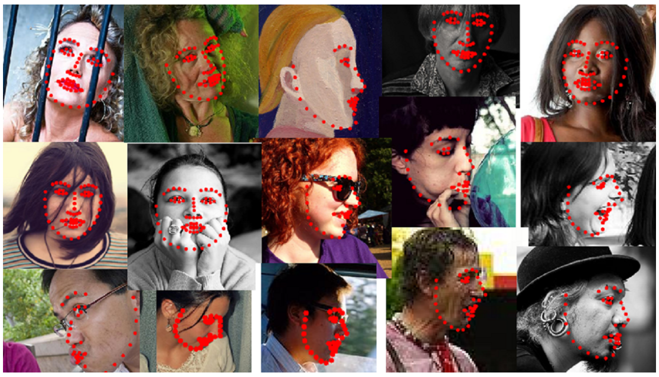


Fig. 3: Example landmark detection from OpenFace 2.0, note the ability to deal with profile faces and occlusion.

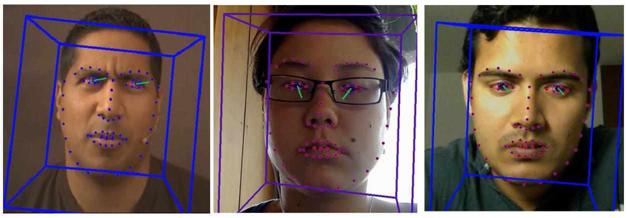


Fig. 4: Sample gaze estimations on video sequences; green lines represent the estimated eye gaze vectors, the blue boxes a 3D bounding box around the head.

a bilinear interpolation to map it back to a dense grid. This leads to 1.5 times improvement in model speed on images and videos with minimal loss of accuracy.

2) Implementation details: The PDM used in OpenFace 2.0 was trained on two datasets — LFPW [12] and Helen [41] training sets. This resulted in a model with 34 non-rigid and 6 rigid shape parameters. For training the CE-CLM patch experts we used: Multi-PIE [29], LFPW [12], Helen [41] training set, and Menpo [76]. We trained a separate set of patch experts for seven views and four scales (leading to 28 sets in total). We found optimal results are achieved when the face is at least 100 pixels ear to ear. Training on different views allows us to track faces with out of plane motion and to model self-occlusion due to head rotation. We first pretrained our model on Multi-PIE, LFPW, and Helen datasets and finished training on the Menpo dataset, as this leads to better results [75].

To initialize our CE-CLM model we use our implementation of the Multi-task Convolutional Neural Network (MTCNN) face detector [78]. The face detector we use was trained on WIDER FACE [74] and CelebA [43] datasets. This is in contrast to OpenFace which used a dlib face detector [37] which is not able to detect profile or highly occluded faces. We learned a simple linear mapping from the bounding box provided by the MTCNN detector to the one surrounding the 68 facial landmarks. When tracking landmarks in videos we initialize the CE-CLM model based on landmark detection in previous frame.

To prevent the tracking drift, we implement a simple four layer CNN network that reports if the tracking has failed based on currently detected landmarks. If our CNN validation module reports that tracking failed we reinitialize the model using the MTCNN face detector.

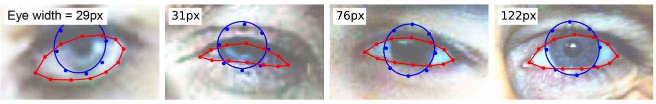


Fig. 5: Sample eye registrations on 300-W dataset.

To optimize matrix multiplications required for patch expert computation and face detection we used the OpenBLAS[7]. It allows for specific CPU architecture optimized computation. This allows us to use Convolutional Neural Network (CNN) based patch expert computation and face detection without sacrificing real-time performance on devices without dedicated GPUs. This led to a 2-5 times (based on CPU architecture) performance improvement when compared to OpenCV matrix multiplication.

All of the above mentioned performance improvements and a C++ implementation, allows CE-CLM landmark detection to achieve 30-40Hz frame rates on a quad core 3.5GHz Intel i7-2700K processor, and 20Hz frame rates on a Surface Pro 3 laptop with a 1.7GHz dual core Intel core i7-4650U processor, without any GPU support when processing 640×480 px videos. This is 30 times faster than the original Matlab implementation of CE-CLM [75].

B. *Head pose estimation*

Our model is able to extract head pose (translation and orientation) in addition to facial landmark detection. We are able to do this, as CE-CLM 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 n point in perspective problem [32], see examples of bounding boxes illustrating head pose in Figure 4.

C. *Eye gaze estimation*

In order to estimate eye gaze, we use a Constrained Local Neural Field (CLNF) landmark detector [9], [70] to detect eyelids, iris, and the pupil. For training the landmark detector we used the SynthesEyes training dataset [70]. Some sample registrations can be seen in Figure 5. We use the detected pupil and eye location 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.

D. *Facial expression recognition*

OpenFace 2.0 recognizes facial expressions through detecting facial action unit (AU) intensity and presence. We use

[7] `http://www.openblas.net`

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

TABLE II: List of AUs in OpenFace 2.0. We predict intensity and presence of all AUs, except for AU28, for which only presence predictions are made.

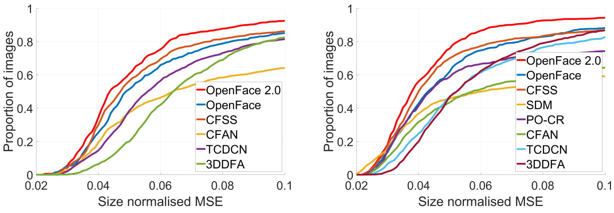
a method based on a recent AU recognition framework by Baltrušaitis et al. [8], that uses linear kernel Support Vector Machines. OpenFace 2.0 contains a direct implementation with a couple of changes that adapt it to work better on natural video sequences using person specific normalization and prediction correction [8], [11]. While initially this may appear as a simple and outdated model for AU recognition, our experiments demonstrate how competitive it is even when compared to recent deep learning methods (see Table VI), while retaining a distinct speed advantage.

As features we use the concatenation of dimensionality reduced HOGs [26] from similarity aligned 112 × 112 pixel face image and facial shape features (from CE-CLM). In order to account for personal differences when processing videos the median value of the features is subtracted from the current frame. To correct for person specific bias in AU intensity prediction, 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 [11].

Our models are trained on DISFA [48], SEMAINE [51], BP4D [80], UNBC-McMaster [44], Bosphorus [59] and FERA 2011 [66] datasets. Where the AU labels overlap across multiple datasets we train on them jointly. This leads to OpenFace 2.0 recognizing the AUs listed in Table II.

IV. Experimental evaluation

In this section, we evaluate each of our OpenFace 2.0 subsystems: facial landmark detection, head pose estimation, eye gaze estimation, and facial action unit recognition. 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). In all cases, except for facial action units (due to lack of overlapping AU categories across datasets), we perform cross-dataset experiments, allowing to better judge the generalization of our toolkit.



(a) 68 landmarks (b) 49 landmarks

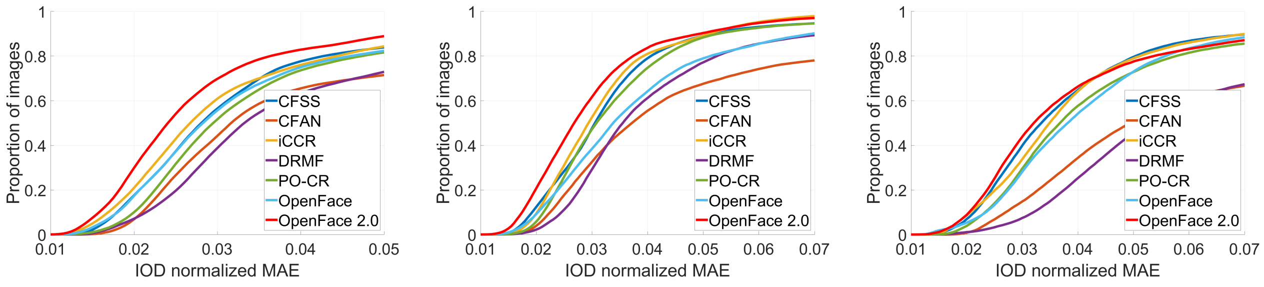
Fig. 7: Fitting on IJB-FL using OpenFace 2.0 and comparing against recent landmark detection methods. None of the approaches were trained on IJB-FL, allowing to evaluate ability to generalize.

A. Landmark detection

We evaluate our OpenFace 2.0 toolkit in a facial landmark detection task and compare it to a number of recent baselines in a cross-dataset evaluation setup. For all of the baselines, we used the code or executables provided by the authors.

Datasets The facial landmark detection capability was evaluated on two publicly available datasets: IJB-FL [36], and 300VW [60] test set. **IJB-FL** [36] is a landmark-annotated subset of IJB-A [38] — a face recognition benchmark. It contains labels for 180 images (128 frontal and 52 profile faces). This is a challenging subset containing images in non-frontal pose, with heavy occlusion and poor image quality. **300VW** [60] test set contains 64 videos labeled for 68 facial landmarks for every frame. The test videos are categorized into three types: 1) laboratory and naturalistic well-lit conditions; 2) unconstrained conditions such as varied illumination, dark rooms and overexposed shots; 3) completely unconstrained conditions including illumination and occlusions such as occlusions by hand.

Baselines 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). **CFSS** [83] — Coarse to Fine Shape Search is a recent cascaded regression approach. **PO-CR** [64] — is another recent cascaded regression approach that updates the shape model parameters rather than predicting landmark locations directly in a projected-out space. **CLNF** [9] is an extension of the Constrained Local Model that uses Continuous Conditional Neural Fields as patch experts, this model is included in the OpenFace toolbox. **DRMF** [5] — Discriminative Response Map Fitting performs regression on patch expert response maps directly rather than using optimization over the parameter space. **3DDFA** [84] — 3D Dense Face Alignment has shown great performance on facial landmark detection in profile images. **CFAN** [77] — Coarse-to-Fine Auto-encoder Network, uses cascaded regression on auto-encoder visual features. **SDM** [72] — Supervised Descent Method is a very popular



(a) Category 1 (b) Category 2 (c) Category 3

Fig. 6: Fitting on the 300VW dataset using OpenFace 2.0 and recently proposed landmark detection approaches. We only report performance on 49 landmarks as that allows us to compare to more baselines. All of the methods except for iCCR were not trained or validated on 300VW dataset.

Method	Yaw	Pitch	Roll	Mean	Median
CLM [57]	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
OpenFace 2.0	**2.4**	**3.2**	2.4	**2.6**	**1.8**

TABLE III: 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 against CLM-Z and Regression forests was performed.

Method	Yaw	Pitch	Roll	Mean
Reg. forests [25]	7.2	9.4	7.5	8.0
CLM-Z [10]	5.1	3.9	4.6	4.6
CLM [57]	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
OpenFace 2.0	**3.1**	**3.5**	**3.1**	**3.2**

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

cascaded regression approach. **iCCR** [56] — is a facial landmark tracking approach for videos that adapts to the particular person it tracks

Results of IJB-FL experiment can be seen in Figure 7, while results on 300VW Figure 6. Note how OpenFace 2.0 outperforms all of the baselines in both of the experiments.

B. Head pose estimation

To measure performance on a head pose estimation task we used two publicly available datasets with existing ground truth head pose data: BU [16] and ICT-3DHP [10].

Model	Gaze error
EyeTab [71]	47.1
CNN on UT [79]	13.91
CNN on SynthesEyes [70]	13.55
CNN on SynthesEyes + UT [70]	11.12
OpenFace	9.96
UnityEyes [69]	9.95
OpenFace 2.0	**9.10**

TABLE V: Results comparing our method to previous work for cross dataset gaze estimation on MPIIGaze [79], measure in mean absolute degree error.

For comparison, we report the results of using Chehra framework [5], CLM [57], CLM-Z [10], Regression Forests [24], and OpenFace [8]. The results can be see in Table III and Table IV. It can be seen that our approach demonstrates state-of-the-art performance on both of the datasets.

C. Eye gaze estimation

We evaluated the ability of OpenFace 2.0 to estimate eye gaze vectors by evaluating it on the challenging MPIIGaze dataset [79] intended to evaluate appearance based gaze estimation. 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 column of Figure 4. We evaluated our approach on a 750 face image subset of the dataset. We performed our experiments in a cross-dataset fashion and compared to baselines not trained on the MPIIGaze dataset.

We compared our model in a to a CNN proposed by Zhang et al. [79], to EyeTab geometry based model [71] and a k-NN approach based on the UnityEyes dataset [69]. The error rates of our model can be seen in Table V. It can be seen that our model shows state-of-the-art performance on the task for cross-dataset eye gaze estimation.

D. Action unit recognition

We evaluate our model for AU prediction against a set of recent baselines, and demonstrate the benefits of such a simple approach. As there are no recent free tools we could compare to our system (and commercial tools do not allow for public comparisons), so we compare general methods used, instead of toolkits.

Baselines Continuous Conditional Neural Fields (**CCNF**) model is a temporal approach for AU intensity estimation [6] based on non-negative matrix factorization features around facial landmark points. Iterative Regularized Kernel Regression **IRKR** [53] is a recently proposed kernel learning method for AU intensity estimation. It is an iterative nonlinear feature selection method with a Lasso-regularized version of Metric Regularized Kernel Regression. A generative latent tree (**LT**) model was proposed by Kaltwang et al. [34].

TABLE VI: Comparing our model to baselines on the DISFA dataset, results reported as Pearson Correlation Coefficient. $^{(1)}$ used a different fold split. Notes: $^{(2)}$ used 9-fold testing. $^{(3)}$ used leave-one-person-out testing.

Method	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	Mean
IRKR [53]$^{(1)}$	0.70	0.68	0.68	0.49	0.65	0.43	0.83	0.34	0.35	0.21	0.86	0.62	0.57
LT [34]$^{(2)}$	0.41	0.44	0.50	0.29	0.55	0.32	0.76	0.11	0.31	0.16	0.82	0.49	0.43
CNN [30]	0.60	0.53	0.64	0.38	0.55	0.59	0.85	0.22	0.37	0.15	0.88	0.60	0.53
D-CNN [82]	0.49	0.39	0.62	0.44	0.53	0.55	0.85	0.25	0.41	0.19	0.87	0.59	0.51
CCNF [6]$^{(3)}$	0.48	0.50	0.52	0.48	0.45	0.36	0.70	0.41	0.39	0.11	0.89	0.57	0.49
OpenFace 2.0 (SVR-HOG)	0.64	0.50	0.70	0.67	0.59	0.54	0.85	0.39	0.49	0.22	0.85	0.67	0.59

The model demonstrates good performance under noisy input. Finally, we included two recent Convolutional Neural Network (**CNN**) baselines. The shallow four-layer model proposed by Gudi et al. [30], and a deeper CNN model used by Zhao et al. [82] (called ConvNet in their work). The **CNN** model proposed by Gudi et al. [30], consists of three convolutional layers, the Zhao et al. **D-CNN** model uses five convolutional layers followed by two fully-connected layers and a final linear layer. **SVR-HOG** is the method used in OpenFace 2.0. For all methods we report results from relevant papers, except for CNN and D-CNN models which we re-implemented. In case of SVR-HOG, CNN, and D-CNN we used 5-fold person-independent testing.

Results can be found in Table VI, it can be seen that an SVR-HOG approach employed by OpenFace 2.0 outperforms the more complex and recent approaches for AU detection on this challenging dataset.

We also compare OpenFace 2.0 with OpenFace for AU detection accuracy. The average concordance correlation coefficient (CCC) on DISFA validation set across 12 AUs of OpenFace is 0.70, while using OpenFace 2.0 it is 0.73.

V. Interface

OpenFace 2.0 is an easy to use toolbox for the analysis of facial behavior. There are two main ways of using OpenFace 2.0: Graphical User Interface (for Windows), and command line (for Windows, Ubuntu, and Mac OS X). As the source code is available it is also possible to integrate it in any C++, C$^\sharp$, or Matlab 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.

OpenFace 2.0 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, head pose, action units, and gaze vectors.

VI. Conclusion

In this paper we presented OpenFace 2.0 – an extension to the OpenFace real-time facial behavior analysis system. OpenFace 2.0 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 releasing the source code and retaining its real-time capacity. We hope that this tool will encourage other researchers in the field to share their code.

References

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

[2] 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. 2014.

[3] N. Ambady and R. Rosenthal. Thin Slices of Expressive behavior as Predictors of Interpersonal Consequences : a Meta-Analysis. *Psychological Bulletin*, 111(2):256–274, 1992.

[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, P. Robinson, and L.-P. Morency. Continuous Conditional Neural Fields for Structured Regression. In *ECCV*, 2014.

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

[8] 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, FG*, 2015.

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

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

[11] T. Baltrušaitis, P. Robinson, and L.-P. Morency. OpenFace: an open source facial behavior analysis toolkit. In *IEEE WACV*, 2016.

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

[13] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *ICCV*, 2013.

[14] C. Busso and J. J. Jain. Advances in Multimodal Tracking of Driver Distraction. In *DSP for in-Vehicle Systems and Safety*. 2012.

[15] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[16] 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.

[17] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou. A Comprehensive Performance Evaluation of Deformable Face Tracking "In-the-Wild". *International Journal of Computer Vision*, 2016.

[18] B. Czupryski and A. Strupczewski. High accuracy head pose tracking survey. LNCS. 2014.

[19] E. S. Dalmaijer, S. Mathôt, and S. V. D. Stigchel. PyGaze : an open-source , cross-platform toolbox for minimal-effort programming of eye-tracking experiments. *Behavior Research Methods*, 2014.

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

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

[22] P. Ekman, W. V. Friesen, and P. Ellsworth. *Emotion in the Human Face*. Cambridge University Press, second edition, 1982.

[23] 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*, 1980.

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

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

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

[27] 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.

[28] 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.

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

[30] 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, FG*, 2015.

[31] D. W. Hansen and Q. Ji. In the eye of the beholder: a survey of models for eyes and gaze. *TPAMI*, 2010.

[32] J. A. Hesch and S. I. Roumeliotis. A Direct Least-Squares (DLS) method for PnP. In *ICCV*, 2011.

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

[34] S. Kaltwang, S. Todorovic, and M. Pantic. Latent trees for estimating intensity of facial action units. In *CVPR*, Boston, MA, USA, 2015.

[35] V. Kazemi and J. Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. *CVPR*, 2014.

[36] K. Kim, T. Baltrušaitis, A. Zadeh, L.-P. Morency, and G. Medionni. Holistically Constrained Local Model: Going Beyond Frontal Poses for Facial Landmark Detection. In *BMVC*, 2016.

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

[38] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. *CVPR*, 2015.

[39] C. L. Kleinke. Gaze and eye contact: a research review. *Psychological bulletin*, 1986.

[40] E. Laksana, T. Baltruaitis, and L.-P. Morency. Investigating facial behavior indicators of suicidal ideation. In *FG*, 2017.

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

[42] M. Lidegaard, D. W. Hansen, and N. Krüger. Head mounted device for point-of-gaze estimation in three dimensions. *ETRA*, 2014.

[43] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep Learning Face Attributes in the Wild. In *ICCV*, pages 3730–3738, 2015.

[44] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. *FG*, 2011.

[45] B. Martinez and M. Valstar. Advances, challenges, and opportunities in automatic facial expression recognition. 2016.

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

[47] Y. Matsuyama, A. Bhardwaj, R. Zhao, O. J. Romero, S. A. Akoju, and J. Cassell. Socially-Aware Animated Intelligent Personal Assistant Agent. *SIGDIAL Conference*, 2016.

[48] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. DISFA: A Spontaneous Facial Action Intensity Database. *TAFFC*, 2013.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] J. Nicolle, K. Bailly, and M. Chetouani. Real-time facial action unit intensity prediction with regularized metric learning. *IVC*, 2016.

[54] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays. WebGazer : Scalable Webcam Eye Tracking Using User Interactions. *IJCAI*, pages 3839–3845, 2016.

[55] Paris Mavromoustakos Blom, S. Bakkes, C. T. Tan, S. Whiteson, D. Roijers, R. Valenti, and T. Gevers. Towards Personalised Gaming via Facial Expression Recognition. *AIIDE*, 2014.

[56] E. Sánchez-Lozano, B. Martinez, G. Tzimiropoulos, and M. Valstar. Cascaded Continuous Regression for Real-time Incremental Face Tracking. In *ECCV*, 2016.

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

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

[59] A. Savran, N. Alyüz, H. Dibekliolu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. *Lecture Notes in Computer Science*, 5372:47–56, 2008.

[60] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. The First Facial Landmark Tracking in-the-Wild Challenge: Benchmark and Results. *ICCVW*, 2015.

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

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

[63] 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*, 2008.

[64] G. Tzimiropoulos. Project-Out Cascaded Regression with an application to Face Alignment. In *CVPR*, 2015.

[65] A. Vail, T. Baltrušaitis, L. Pennant, E. Liebson, J. Baker, and L.-P. Morency. Visual attention in schizophrenia: Eye contact and gaze aversion during clinical interactions. In *ACII*, 2017.

[66] 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.

[67] S. Vijay, T. Baltrušaitis, L. Pennant, D. Öngür, J. Baker, and L.-P. Morency. Computational study of psychosis symptoms and facial expressions. In *Computing and Mental Health Workshop at CHI*, 2016.

[68] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. A 3d morphable eye region model for gaze estimation. In *ECCV*, 2016.

[69] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesized images. In *Eye-Tracking Research and Applications*, 2016.

[70] 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.

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

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

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

[74] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *CVPR*, 2016.

[75] A. Zadeh, T. Baltrušaitis, and L.-P. Morency. Convolutional experts constrained local model for facial landmark detection. In *CVPRW*, 2017.

[76] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen. The Menpo Facial Landmark Localisation Challenge: A step towards the solution. In *CVPR workshops*, 2017.

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

[78] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. 2016.

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

[80] 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.

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

[82] K. Zhao, W. Chu, and H. Zhang. Deep Region and Multi-Label Learning for Facial Action Unit Detection. In *CVPR*, 2016.

[83] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face Alignment by Coarse-to-Fine Shape Searching. In *CVPR*, 2015.

[84] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016.

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