Fully End-to-End Composite Recurrent Convolution Network for Deformable Facial Tracking In The Wild

Decky Aspandi, Oriol Martinez, Federico Sukno and Xavier Binefa
Department of Information and Communication Technologies, Pompeu Fabra University, Barcelona, Spain

Abstract— **Human facial tracking is an important task in computer vision, which has recently lost pace compared to other facial analysis tasks. The majority of current available tracker possess two major limitations: their little use of temporal information and the widespread use of handcrafted features, without taking full advantage of the large annotated datasets that have recently become available. In this paper we present a fully end-to-end facial tracking model based on current state of the art deep model architectures that can be effectively trained from the available annotated facial landmark datasets. We build our model from the recently introduced general object tracker Re^3, which allows modeling the short and long temporal dependency between frames by means of its internal Long Short Term Memory (LSTM) layers. Facial tracking experiments on the challenging 300-VW dataset show that our model can produce state of the art accuracy and far lower failure rates than competing approaches. We specifically compare the performance of our approach modified to work in tracking-by-detection mode and showed that, as such, it can produce results that are comparable to state of the art trackers. However, upon activation of our tracking mechanism, the results improve significantly, confirming the advantage of taking into account temporal dependencies.**

I. INTRODUCTION

The human face is arguably one of the most important deformable objects for analysis, especially for tracking, with numerous real world applications, such as facial animation, human activity recognition and human - computer interaction [24]. The recent growth of facial datasets *in the wild* with annotated landmarks such as 300W [16] and 300 Videos in the Wild (300-VW) [17] has led to rapid development of facial analysis tools by introducing powerful deep learning models that are able to automatically extract more representative features from larger scale datasets. These new models have pushed forward the state of the art, outperforming the accuracy reported by earlier methods based on handcrafted features. We can find examples of such models targeting face detection [20], [32], facial classification and verification [14], [19], and facial expression analysis [11].

However, current progress in deformable facial tracking has been relatively slower when compared to other tasks and it has been less influenced by deep learning models [3]. Furthermore, currently available trackers make little use of temporal information. Indeed, most of them do not really take into account temporal information but process each frame independently and rely on doing so with sufficient precision to achieve *tracking-like* performance. In contrast, other trackers do some temporal modelings, but they are mostly limited to the adjacent frames [26],[15]. This inhibits current facial trackers to take full advantage of the temporal information contained in video sequences [27].

978-1-7281-0089-0/19/$31.00

In this paper we present a fully end-to-end facial tracking model based on current state of the art deep model architectures that can be effectively trained from the available annotated facial landmark datasets. We build our model from the recently introduced general object tracker Re^3 [4], which allows modeling the short and long temporal dependency between frames by means of its internal Long Short Term Memory (LSTM) layers. While Re^3 is too generic to be directly used as facial tracker (its performance would be suboptimal), we introduce architectural modifications that lead to a robust facial tracker achieving state of the art performance. More specifically, the contributions in this work are:

1) We replaced the original Skip Convolution Networks from Re^3 by the more robust Inception Residual Networks [18] through transfer learning.
2) We embed our main tracker together with additional layers that validate the tracking results at every frame and trigger a re-initialization strategy if drifting is detected.
3) To the best of our knowledge, we are the first to successfully train an end-to-end network that can achieve state of the art face tracking on the 300-VW benchmark.
4) We investigate the impact of different temporal windows in the performance of face tracking.

II. RELATED WORK

Currently, the most popular facial tracking technique is Tracking by Detection, which consists of performing facial detection and landmark localization at each frame. One example of this strategy is the work from Uricar et al. [21] which uses tree-based Deformable Part Models (DPM) for facial landmark detection and localisation with Kalman Filter smoothing.

Other tracking methods perform face detection only in the first frame and then apply facial landmark localization using the fitting result from the previous frame as initialization. One such example is the work from Xiao et al. [26] which adopts a multi-stage regression-based approach to initialize the shape of landmarks with high semantic meaning. Other examples include the work from Raja et al. [15] which combines a global shape model with sets of response maps for different head angles indexed on the shape model parameters and the works from Wu et al

[23] who apply shape augmented regression. There are also hybrid approaches which combine tracking by detection and initialization based on the latest fitting result. Among these, combinations of Coarse-To-Fine Shape Search (CFSS) [33] landmark localiser with multiple general-object trackers have shown to perform particularly well [3].

However, all methods derived from tracking by detection share the limitation of not considering the temporal information contained in video sequences. Furthermore, it is difficult to obtain consistent initializations from most face detectors, which tends to reduce the final landmark localisation accuracy [12]. Some approaches try to mitigate this problem by including the information from the adjacent frames to capture short temporal dependencies. For example, Yang et al. [28] used time series regression on adjacent two frames, which led them to achieve the best result reported so far on the biggest deformable facial tracking dataset: 300 Videos in the Wild (300-VW) [17].

With the recent growth of facial landmark datasets, such as 300W [16], Menpo [30], 300-VW and LS3D-W [2], current methodologies on facial analysis started to shift from systems based on handcrafted features towards incorporating deep learning architectures [31], [5]. Rapid progress can be seen on the development of various convolutional architectures as the main spatial feature extractor used on both facial detection [32] and landmark localisation models [2], [34] and achieving state of the art accuracy. In spite of this, localization is still mainly performed on every single frame, without taking into the temporal information.

On the other hand, introduction of recurrent neural networks (RNN), especially Long Short Term Memory (LSTM) [9], has allowed incorporating temporal information with great success in several applications [6]. This is the case of the recently introduced general object tracker Re^3 [4], which is robust against image occlusions and can be trained on long sequences thanks to its internal LSTM networks. Nonetheless, RNN have received little attention in the context of facial tracking. The only exception so far has been the work by Jiang et al. [7], who proved that an end-to-end RNN is capable to work on multiple domains including facial landmark tracking. However, even though they obtained very low failure rates, their accuracy was still inferior to other state of the art facial trackers.

III. Fully-end-to end Recurrent Facial Tracker

Our tracking model is a composite network that receives raw frames as input and returns the localization of facial landmarks as the final output. It is composed by four sub-networks, arranged in a way that permits the end-to-end training of the whole network, without involving any hand-crafted features. Specifically, if $\mathbf{X}_t$ and $\mathbf{X}_{t-1}$ denote the current and previous frame, respectively, our Composite Recurrent Convolution Tracker ($CRCT$) will estimate the position of n facial landmarks in the current frame $\mathbf{l}_t$:

$$\mathbf{l}_t = \{(\hat{x}_1, \hat{y}_1)...(\hat{x}_n, \hat{y}_n)\} = CRCT_{\Phi}(X_t, X_{t-1}, \mathbf{b}_{t-1}) \quad (1)$$

where $\mathbf{\Phi}$ are the parameters $\{\Phi^1,\Phi^2,\Phi^3\}$ of our composite networks $CRCT$ and $\{\hat{x}_1...\hat{x}_n, \hat{y}_1...\hat{y}_n\} \in \mathbb{R}_{>0}$.

Our $CRCT$ consists of four individual sub-networks: Multi-Task Cascaded Neural Network faces detector ($MTCNN$), facial bounding Box Tracker (BT), Facial Validator (FV) and Facial Landmark Localizer (FLL). Note that for face detection we relied on the state of the art $MTCNN$[32].

A schematic diagram of our tracker can be seen in Figure 1. We start by assuming a tracking scenario, where we have an existing estimate for the bounding box of the preceding frame.[1] This bounding box, together with the current and previous frames $\{\mathbf{X}_t, \mathbf{X}_{t-1}\}$ are fed to our BT network to produce a first estimate of the targeted landmarks ($\mathbf{l}_t^{BT}$) and bounding box ($\mathbf{b}_t^{BT}$), while at the same time updates its internal state.

Once we have our first landmarks estimate $\mathbf{l}_t^{BT}$, we use the FV network to validate the result obtained by the tracker. To do so, we train the FV network to estimate the probability that the objects tracked within $\mathbf{l}_t^{BT}$ is a face ($p(f)$). In case of obtaining a low probability, which would suggest that the BT network has lost track, we use the $MTCNN$ to perform face detection on the current frame and re-initialize the whole network for the next time step.

In contrast, if $\mathbf{l}_t^{BT}$ is successfully validated by the FV network, the current frame and its bounding box $\mathbf{b}_t^{BT}$ are fed to the FLL network, which produces the final estimates for the target landmarks, $\mathbf{l}_t^F$ and the corresponding bounding box, $\mathbf{b}_t^F$. Note that, while FLL and BT have similar convolutional layers, FLL works from an already detected and validated bounding box, which allows it to achieve a more accurate result.

A. The Recurrent Facial Bounding Box Tracker

We base our BT network on the structure of the Re^3 tracker [4], which is a full end-to-end object tracker with LSTM networks to capture the temporal dependencies from video. Given input frames $\{\mathbf{X}_t, \mathbf{X}_{t-1}\}$ cropped as $\{\mathbf{X}_t^{Pb}, \mathbf{X}_{t-1}^{Pb}\}$ with the previous Bounding Box ($P_b = b_{t-1}$), the BT network estimates the landmark positions for the current frame $\mathbf{l}_t^{BT}$ and updates the internal state of the LSTM $\mathbf{h}_t$ as follows:

$$\begin{aligned}\mathbf{h}_t, \mathbf{l}_t^{BT} &= BT_{\Phi^1}(\mathbf{X}_t, \mathbf{X}_{t-1}, P_b, h_{t-1}) \\ &= BT_{\Phi^1}(\mathbf{X}_t^{Pb}, \mathbf{X}_{t-1}^{Pb}, h_{t-1}) \\ &= LSTM_{\Phi^1}(EL_{\Phi^1}(\mathbf{X}_t^{Pb}, \mathbf{X}_{t-1}^{Pb}), h_{t-1}) \odot W_{\Phi^1}^{BT}\end{aligned} \quad (2)$$

where $LSTM$ refers to the set of internal LSTM [9] networks, EL stands for the Embedding Layer, W^{BT}, W^{EL} is the set of weight of each fully connected layers of BT and EL respectively and res is the Inception-Residual Network [18] (Inception-Resnet). The Embedding Layer is a weighted concatenation of the residual network coefficients:

$$EL = [res_{\Phi^1}(\mathbf{X}_t^{Pb}); res_{\Phi^1}(\mathbf{X}_{t-1}^{Pb})] \odot W_{\Phi^1}^{EL} \quad (3)$$

[1]For initialization, this estimate can be obtained from the $MTCNN$ detector or from an external input.

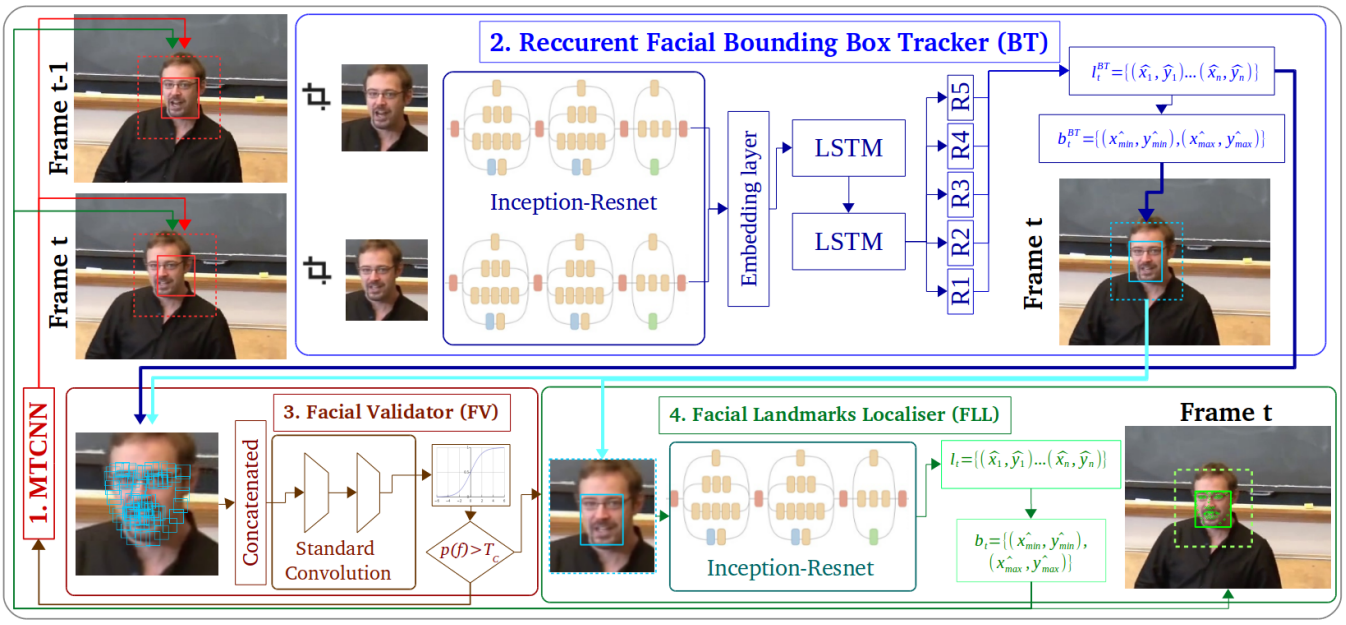


Fig. 1. General overview of our tracker

We use Φ^1 to denote the parameters of all sub-networks contained in BT. Finally, we also generate an estimate of the bounding box for the current frame $\mathbf{b}_t^{BT}$ directly from the estimated landmarks:

$$\mathbf{b}_t^{BT} = \{(\hat{x}_{min}, \hat{y}_{min}), (\hat{x}_{max}, \hat{y}_{max}) | \hat{x}, \hat{y} \in \mathbf{l}_t^{BT}\} \quad (4)$$

Note that, even though the architecture of BT is based on Re^3, we introduce several key modifications to adapt this recurrent tracker model into this new problem domain:

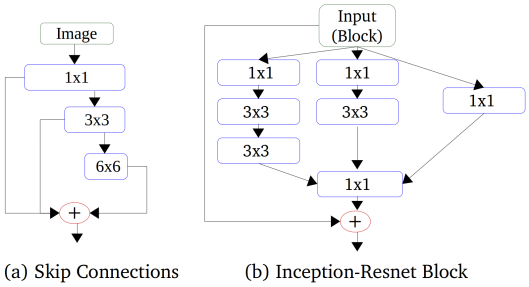
1) First we preconditioned the convolutional network of our BT to contain common facial features by replacing the internal Skip Convolution Networks (SkipNet) with the more sophisticated Inception-Resnet that has been pre-trained on the MS-Celeb [8] and CasiaWebFace [29] datasets[2] with triplet loss [14]. Figure 2 visualizes the differences between the original SkipNet on Re^3 versus the more complex structure of BT, which is inherited from the Inception-Resnet (Version 1). Each block of Inception-Resnet architecture can be expressed as below:

$$\mathbf{r}_{i+1} = H(r_i) + F(r_i, W_i) \quad (5)$$

Where r_i and r_{i+1} are the input and output of the i-th block, $H(b_i)$ is the identity matrix and F represents the combined effect of the various convolutional and ReLU layers. Notice that SkipNet does not have the advantage of residual connection as in the Inception-Resnet which eases the gradient flows in optimization [18].

2) Second we use the BT network to produce a first estimate of landmark locations (L_t^b) following the work of [7], but we split the fully-connected layer that receives the output from the LSTMs into five independent fully-connected networks so that each of them is focused on a specifically facial region. Specifically, we divide the facial landmarks in the following regions: facial silhouette (our outer contour), eyebrows, eyes, nose and lips. Thus $W^{BT} = \{W^{R1}, W^{R2}, W^{R3}, W^{R4}, W^{R5}\}$.
3) Finally, we reduce by half the number of neurons from the original Re^3, which implies an input image size to 128x128. This enables us to train the network faster while still achieving state of the art accuracy.

[2]The trained inception resnet is publicly available on: https://github.com/davidsandberg/facenet



(a) Skip Connections (b) Inception-Resnet Block

Fig. 2. Convolution architectures of Skip Network vs Inception-Residual Network block

B. The Facial Validator

After the initial estimates produced by the BT network we use the FV network to validate the results before further processing. The main reason for doing so is to avoid the drift problem, well known in the tracking literature [22]. Specifically, the FV network can be understood as a conditional function that determines whether to continue the processing pipeline based on the current estimates from BT or to reset the tracker and attempt to re-detect the facial region because

the current estimates are not reliable enough.

We follow the methodology in [3] to build a strong classifier to estimate the probability $p(f)$ that the object currently being tracked by BT is a face. To this end, we use concatenated small patch regions from the estimated landmarks (l_t^{BT}) as follows:

$$\begin{aligned}\mathbf{p}(f|\mathbf{X}_t, \mathbf{l}_t^{BT}) &= FV_{\Phi^2}(\mathbf{X}_t, \mathbf{l}_t^{BT}) \\ &= \frac{1}{1+e^{-(W_{\Phi^2}^{FV} \odot cnn_{\Phi^2}(\mathbf{X}_t, \mathbf{l}_t^{BT}))}}\end{aligned} \tag{6}$$

Where cnn is the composite function of standard stacked convolution layers followed by a bottleneck layer with W^{FV} parameterized by Φ^2 and $0 < p(f) < 1$. We use the value of T_c as the threshold level.

C. The Facial Landmark Localiser

The FLL is built by reusing the same pretrained Inception-Resnet as in BT, with the assumption that the internally extracted facial feature should also be useful to estimate the locations of the facial landmarks. This landmark localization procedure can be expressed mathematically as below:

$$\mathbf{l}_t = FLL_{\Phi^3}(\mathbf{X}_t^{Pb}) = W_{\Phi^3}^{FLL} \odot res_{\Phi^3}(\mathbf{X}_t^{Pb}) \tag{7}$$

With FLL consisting of Inception-Resnet (res) and a regression layer of weight matrix W^{FLL} parameterized by Φ^3.

D. Recurrent Facial Tracking Algorithm

The operation of our Composite Recurrent Convolution Tracker, $CRCT$, is shown in Algorithm 1. When a suitable detection of the facial region is available, e.g. from initialization or the previous frame (lines 8 and 10), the BT network produces a first estimate of facial landmarks (line 13) and bounding box (line 14). Then, the FV network is used to estimate the probability $p(f)$ that the output from BT corresponds to a face. If $p(f)$ is sufficiently high (above threshold T_c), the initial estimate is refined by the FLL network to produce the final tracker estimate (lines 18 and 19). Otherwise, it is assumed that the BT has lost track and there is a need to re-initialize the tracker (line 16).

We perform reinitialization between lines 3 and 6. We start by detecting the face in the current frame by means of the $MTCNN$ network. This detector is likely to produce multiple detections, hence its outputs are validated with respect to the bounding box of the previous frame b_{t-1}. Specifically, we compare the Euclidean distance between each new detection and the center of the previous bounding vox $\mathrm{d}(b_{t-1}, b^{MT})$ with respect to the magnitude of the previous bounding box, and keep the one that produces the minimum ratio:

$$P_b = \begin{cases} b, & \min\limits_{\forall b \in b^{MT}} \frac{d(b_{t-1}, b)}{\|b_{t-1}\|} < T_B \\ b_0, & dim(b_{t-1}) < 0 \\ b_{t-1}, & otherwise \end{cases} \tag{8}$$

as long as there is at least one detection whose ratio is below threshold T_B. Otherwise, all new detections are too far from the previous tracking result and no re-initialization is performed. The latter is necessary to tackle the cases in which the face being tracked moves out of the visual field. In such cases, without threshold T_B the system might be incorrectly re-initialized to track another face. In contrast, by using T_B the tracker remains in its latest valid coordinates awaiting for the tracked object to *come back* to the field of view.

Finally, ($SeqBT$) controls the length of the temporal window that is considered by the tracker (in frame units), which is fixed at training time (see next section). If the tracker is re-initialized or if the sequence length ($SeqT$) exceeds the temporal window ($SeqBT$), then the internal state of the the BT network is reset (line 8).

Algorithm 1 Recurrent Facial Tracking Algorithms

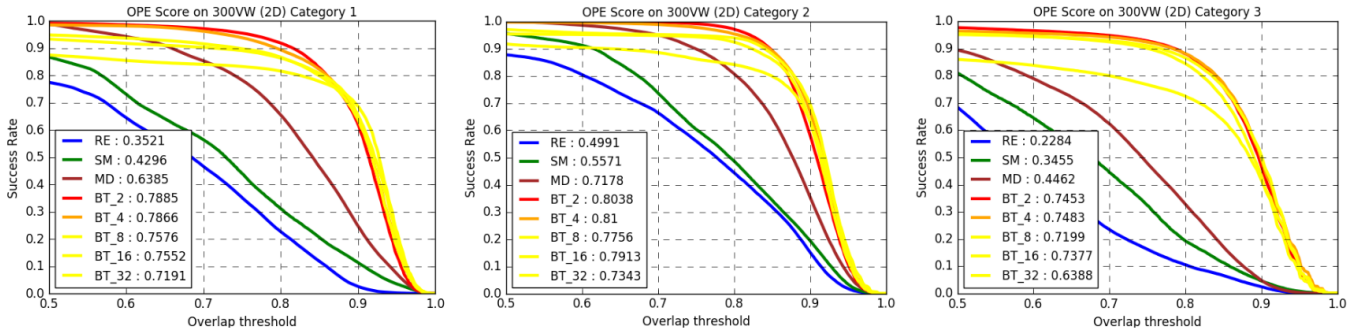
Input : Frame of $\mathbf{X}_{0..N}$
Initial value of b_0 and h_0
Threshold value of T_B, T_C, and SeqBT
Network parameters of Φ^1, Φ^2 and Φ^3
Output : Facial Landmark of $\mathbf{l}_{1..N}$

1: redetect← FALSE, SeqT ← 0, $\mathbf{b}_t \leftarrow b_0$
2: **for** $t \leftarrow 1$ to N **do**
3: **if** redetect **then**
4: $b^{MT} \leftarrow MTCNN(\mathbf{X}_t)$
5: **if** length(d(b_{t-1}, b^{MT}) $> T_B$) > 0 **then**
6: $\mathrm{P}_b \leftarrow b^{MT}$[min(d($b_{t-1}, b^{MT}$))]
7: **else**
8: $\mathrm{P}_b \leftarrow b_t$
9: **if** dim(P_b) < 0 **then**
10: $\mathrm{P}_b \leftarrow b_0$
11: **if** redetect OR SeqT $>$ SeqBT) **then**
12: $h_t \leftarrow h_0$ and SeqT ← 0
13: $\mathbf{h}_t$,$\mathbf{l}_t^{BT} \leftarrow BT_{\Phi^1}(\mathbf{X}_t$,$\mathbf{X}_{t-1}$,$\mathrm{P}_{BB})$
14: $\mathbf{b_t}^{BT} \leftarrow [max(\mathbf{l}_t^{BT}), min(\mathbf{l}_t^{BT})]$
15: **if** $FV_{\Phi^2}(\mathbf{X}_t$,$\mathbf{l}_t^{BT}) < T_C$ **then**
16: redetect ← TRUE
17: **else**
18: $\mathbf{l}_t \leftarrow FLL_{\Phi^3}(\mathbf{X}_t, \mathbf{b}_t^{BT})$
19: $\mathbf{b}_t \leftarrow [max(\mathbf{l}_t), min(\mathbf{l}_t)]$
20: SeqT ← SeqT + 1
21: redetect ← FALSE

E. Training procedure

We train BT,FLL and FV with ℓ_1,ℓ_2 and Cross Entropy Loss respectively. Specifically for BT, we follow the same curriculum learning as in Re^3 [4] using sequence lengths between $SeqBT = 2$ to $SeqBT = 32$ frames. We used multiple transfer learning to condition the pre-trained Inception-Resnet. To do so, we fine-tuned this network on the FLL network before its integration into BT.

We trained our BT network using 300-VW training dataset for 2D Landmark tracking and LS3D-W annotation [2] for 3D-2DA landmark tracking. The FV and FLL networks were trained with the 300W [16] and Menpo



(a) 300-VW 2D Category 1 (b) 300-VW 2D Category 2 (c) 300-VW 2D Category 3

Fig. 3. OPE scores on rigid facial bounding box tracking experiment

datasets [30] for both 2D and 3DA-2D landmark localization. We performed data augmentations by means of horizontal flipping, $-45°$ to $45°$ degree rotations and artificial strip boxes across the frames to simulate occlusions.

We trained our model using ADAM optimizer [10] with scheduled weight learning decay every 10.000 iterations. Two NVIDIA tesla GPUs were used for training which took approximately two to three days to train a single BT for a defined sequence length, and around two days for both FLL and FV. Our pre-trained models and results are publicly available for additional reference [3].

IV. EXPERIMENTS

A. Experiment Settings

We conducted two main facial tracking experiments: rigid facial bounding box tracking and deformable 2D and 3DA-2D facial landmark tracking. We performed the experiments on the 300-VW dataset [17] comprising 55 videos divided into three categories according to the difficulty level. We used the original 2D facial landmarks directly as ground-truth for deformable 2D facial landmark tracking and their corresponding bounding box for rigid facial bounding box tracking.

We used the projected 3DA-2D dataset video dataset [30] for deformable 3DA-2D facial landmark tracking, which consists of a subset of the videos from 300-VW dataset. To facilitate comparison to other works in all cases we report the projected result and follow the conventional 68 facial landmark locations. We set the thresholds $T_B = 1.0$ and $T_C = 0.5$ for all experiments.

B. Rigid - Facial bounding boxes tracking

In this experiment, we compare our bounding box tracker (BT) with three state of the art general object trackers:

1) MDNET[13] (abbreviated MD) which performs a series of convolutions and has a specialized regression layer on the single individual frames without taking any temporal information between frames
2) Siamese Net (abbreviated SM) [1] which uses both the previous and the current frame to be fed to its Siamese Network based tracker. This can be seen as capturing a short temporal context of 2 frames.
3) Recurrent Tracker Re^3 (abbreviated RE), as provided in [4], which is pre-trained on sequences of 32 frames.

In this test, all trackers are initialized with the same initial bounding box (the ground truth). Our system is tested without the FLL block, which means that $b_t = b_t^{BT}$, and we report results for different sequence lengths between $SeqBT = 2$ and $SeqBT = 32$ frames (BT_2, BT_4, BT_8, BT_16 and BT_32), to see the impact of longer temporal context in our BT.

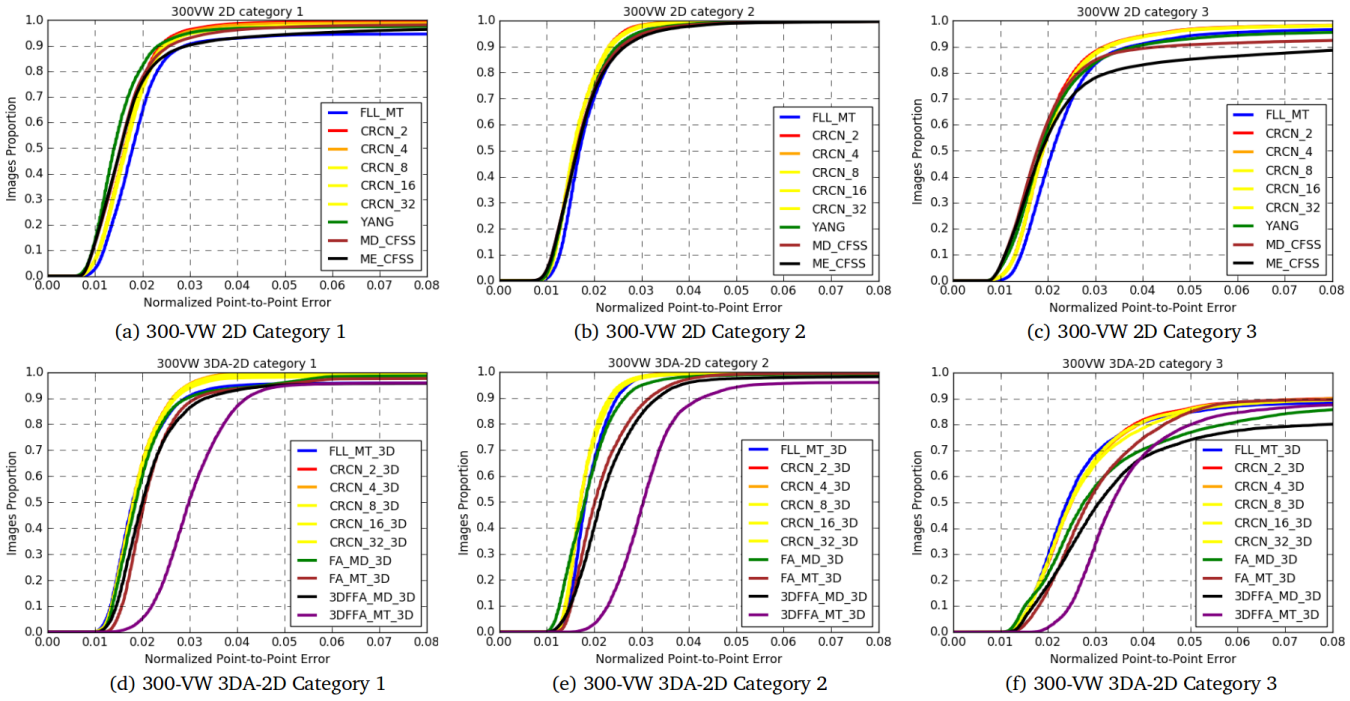
TABLE I
AUC RESULT OF ALL CATEGORIES OF RIGID FACIAL TRACKING ON 300-VW DATASET

Method	Temporal sequences					
	0	2	4	8	16	32
RE [4]	-	-	-	-	-	0.363
SM [1]	-	0.445	-	-	-	-
MD [13]	0.616	-	-	-	-	-
BT	-	0.783	**0.784**	0.754	0.761	0.705

Figure 3 shows the performance of each model, computed with One Pass Evaluation (OPE)[25] in terms of the success rate against the bounding box overlap ratios. We observe that our models, BT, achieve the best results in all three categories, outperforming all other trackers including the original Re^3. The main reason for these results is that, as opposed to our model, none of the compared trackers is specifically designed to track faces. Furthermore, with the exception of MDNET, other models lack any drift prevention mechanism, which explains the performance drop on category 3, where the extreme facial poses and illumination changes occur. As illustrated in Figure 5, our model demonstrated the ability to consistently track the facial bounding box on extreme pose and illumination conditions.

Note that we show the results for our BT tracker under different training sequence lengths (2, 4, 8, 16 and 32). The highest scores were achieved for $SeqBT = 2$ and 4 frames in all categories, with very small differences between these two settings as shown in Table I. Bigger $SeqBT$ values generally

[3]https://github.com/deckyal/RT/tree/master



(a) 300-VW 2D Category 1 (b) 300-VW 2D Category 2 (c) 300-VW 2D Category 3

(d) 300-VW 3DA-2D Category 1 (e) 300-VW 3DA-2D Category 2 (f) 300-VW 3DA-2D Category 3

Fig. 4. AUC graphs on 2D and 3DA-2D facial landmark tracking experiments.

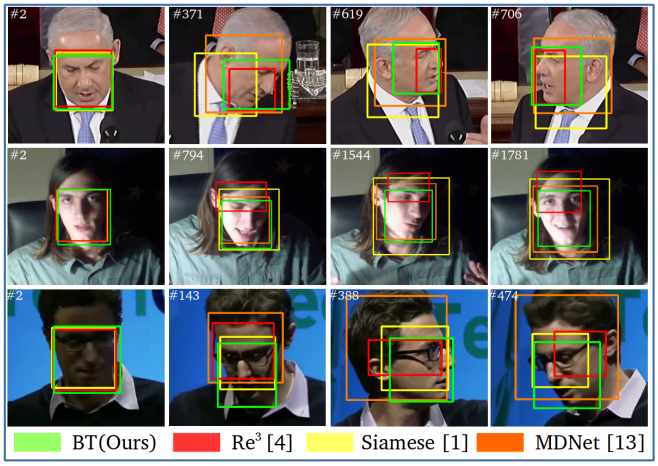


Fig. 5. Some visual results of rigid facial tracking on 300-VW dataset.

resulted in lower scores. This suggests that a rather short temporal context is sufficient to optimize facial tracking. Nevertheless, these results must be read in relation to the test sequences, which show quite irregular (not necessarily natural) facial movements. This is especially noticeable in category 3, where rapid face movement with pose changes occur in relatively short sequences. In such cases, BT_2 and BT_4, trained to capture the temporal information from shorter sequences have an advantage since they are restarted more frequently.

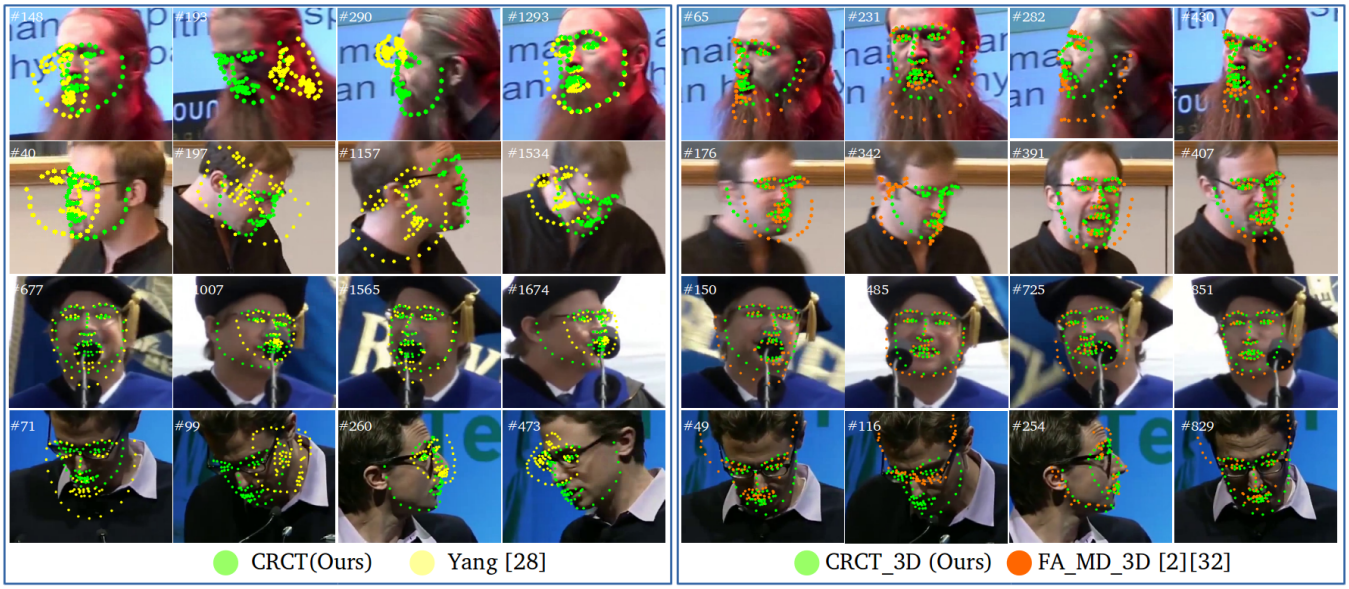
C. 2D and 3DA-2D facial landmark tracking

In this section we show the results of 2D and 3DA-2D facial landmark tracking. In the 2D setting, we compared our model with other 8 facial trackers: 1) two hybrid trackers, MEEM_CFSS and MD_CFSS, which showed the best performance in the recent facial tracking review from Chrysos et al. [3]; 2) the current state of the art tracker, from Yang et al. [28]; 3) the recent tracker from Gu et al. [7] based on Bayesian RNNs; 4) four other trackers from the original 300VW competition [17].

In 3DA-2D facial landmark tracking, we follow a similar procedure to [3] to build four hybrid trackers combining both MDNET and MTCNN with state of the art 3D facial localizers for comparison: 1) Facial Alignment Network [2] resulting in FA_MD_3D and FA_MT_3D; 2) 3DFFA[34] to create other two hybrid trackers: 3DFFA_MD_3D and 3DFFA_MT_3D.

Similarly to the previous section, we evaluate our full tracker, $CRCT$, under different operation conditions. First, analogously to the previous section, we build trackers with different lengths of training sequences, $SeqBT = 2$, 4, 8, 16 and 32. Then, we also report results for our model in tracking-by-detection mode (FLL_MT), where we use $MTCNN$ for face detection in each frame and FLL for landmark localisation. This experiment is to assess the impact of the BT network on the performance of the full tracker.

Our results are summarized in Tables II and III, while the curves for some of the trackers are also displayed in Fig. 4. In all cases, we use the Normalized Mean Error (NME) by Facial Bounding Box [30], and report the Area Under the



(a) Visual results on 2D Facial Landmark Tracking

(b) Visual results on 3DA-2D Facial Landmark Tracking

Fig. 6. Some visual results of landmark tracking on challenging case from 300-VW testset.

TABLE II
RESULTS ON THE LANDMARK 2D TRACKING DATASET

Method	Category 1		Category 2		Category 3	
	AUC	FR	AUC	FR	AUC	FR
CRCT_2	0.784	**0.50**	**0.790**	**0.05**	**0.729**	**1.75**
CRCT_4	0.778	1.01	**0.790**	0.07	0.725	1.80
CRCT_8	0.772	1.88	0.788	0.07	0.725	1.91
CRCT_16	0.773	1.64	0.787	0.07	0.725	2.02
CRCT_32	0.769	1.79	0.778	0.07	0.723	1.86
FLL_MT	0.729	5.38	0.769	2.60	0.691	3.39
MD_CFSS [3]	0.784	1.80	0.783	0.34	0.713	7.47
ME_CFSS [3]	0.758	3.56	0.772	0.38	0.659	11.3
Yang [28]	**0.791**	2.40	0.788	0.32	0.710	4.46
Jinwei [7]	0.718	1.20	0.703	0.20	0.617	4.83
Uricar [21]	0.657	7.62	0.677	4.13	0.574	7.96
Xiao [26]	0.760	5.90	0.782	3.84	0.695	7.38
Raja [15]	0.735	6.56	0.717	3.91	0.659	8.29
Wu [23]	0.674	13.9	0.732	5.60	0.602	13.1

TABLE III
RESULTS ON THE LANDMARK 3DA-2D TRACKING DATASET

Method	Category 1		Category 2		Category 3	
	AUC	FR	AUC	FR	AUC	FR
CRCT_2_3D	**0.760**	**0.09**	**0.772**	0.20	**0.605**	**9.97**
CRCT_4_3D	**0.760**	0.14	**0.772**	**0.19**	0.603	10.0
CRCT_8_3D	0.758	0.34	0.773	0.20	0.603	10.2
CRCT_16_3D	0.750	1.36	0.771	0.22	0.604	10.3
CRCT_32_3D	0.747	1.72	0.770	0.21	0.596	10.6
FLL_MT_3D	0.730	4.14	0.757	0.45	0.603	11.5
FA_MD_3D [2][32]	0.732	1.35	0.757	0.90	0.544	14.2
FA_MT_3D [2][13]	0.706	2.41	0.722	0.57	0.566	10.3
3DFFA_MD_3D [34][32]	0.721	4.30	0.702	1.85	0.504	19.8
3DFFA_MT_3D [34][13]	0.595	4.12	0.590	4.07	0.497	12.4

Curve (AUC) and Failure Rate (FR) for NME scores up to 0.08 [3].

In Table II we see that our $CRCT$ trackers trained with $SeqBT = 2$ and 4 frames achieves the highest AUC for Categories 2 and 3, while they rank within the top-3 trackers in the Category 1 dataset, slightly below [28]. Additionally, our models have far lower Failure Rates than all other compared trackers, Which for some applications is even more important than having smaller landmark localisarion errors [7]. We also see similar results in Table III for the 3DA-2D scenario, Where in overall terms our model outperforms other trackers across all categories with higher AUC and low Failure Rates.

Another observation is that our simpler tracking-by-detection model (FLL_MT) reaches fairly high AUC and low Failure Rates, with a performance comparable to other trackers. This demonstrates the maturity of tracking by detection models, as also reported in [3]. Nevertheless, these results are still inferior to those from our full $CRCT$ models, which incorporates BT to benefit from the temporal dependency between frames. This proves that the BT network provides a more consistent facial bounding box which impacts the final landmark estimation from FLL. This effect has also been demonstrated in the recent work from Lv et al. [12].

D. Visual Results Analysis

We provide several examples of 2D and 3DA-2D tracking in Figure 6, where we see that our tracker is able to

accurately localize the facial landmarks in especially difficult cases. These include extreme head poses up to full profile, blurring (e.g. due to sudden movements of the face or the camera, see 2nd row of examples), partial occlusions (1st and 3rd rows) and strong illumination changes (4th row). Specifically, for 2D landmark tracking, our model performs well in cases in which the state of the art tracker from Yang et al. [28] often gives inaccurate landmark positions. Similarly, for 3DA-2D tracking, comparison of our results to those from FA_MD_3D, highlights the robustness of our tracker to handle the difficulties mentioned above from this dataset.

V. CONCLUSIONS

In this paper we present the first composite deformable facial tracker that, while being fully end-to-end, is able to achieve state-of-the-art results for *in the wild* benchmarks. Unlike other trackers, our model benefits from the temporal information captured by our internal recurrent tracker. Further, our model can be tuned to consider shorter or longer temporal contexts and analyze their impact on facial tracking performance.

Facial tracking experiments on the challenging 300-VW dataset show that our model can produce state of the art accuracy and far lower failure rates than competing approaches. We specifically compared the performance of our approach modified to work in tracking-by-detection mode and showed that, as such, it can produce results that are comparable to state of the art trackers. However, upon activation of our tracking mechanism, the results improve significantly, confirming the advantage of taking into account temporal dependencies.

Our results suggest that the optimal temporal context to consider for this dataset is between 2 and 4 frames ($\sim$ 70 to 160 ms). Nevertheless, these results must be read in relation to the test sequences, which show quite irregular (not necessarily natural) facial movements.

VI. ACKNOWLEDGMENTS

This work is partly supported by the Spanish Ministry of Economy and Competitiveness under project grant TIN2017-90124-P, the Ramon y Cajal programme, and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

REFERENCES

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV 2016 Workshops*, pages 850–865, 2016.

[2] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). *2017 IEEE ICCV*, pages 1021–1030, 2017.

[3] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou. A Comprehensive Performance Evaluation of Deformable Face Tracking In-the-Wild. *Int J Comput Vision*, pages 1–35, 2017.

[4] D. Gordon, A. Farhadi, and D. Fox. Re 3 : Real-Time Re current Regression Networks for Object Tracking. *IEEE Robotics and Automation Letters*, 3(2):788–795, 2018.

[5] H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE T Med Imaging*, 35(5):1153–1159, May 2016.

[6] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE T Neur Net Lear*, 28(10):2222–2232, Oct 2017.

[7] J. Gu, X. Yang, S. D. Mello, and J. Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *2017 IEEE CVPR*, pages 1531–1540, July 2017.

[8] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*. Springer, 2016.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[11] S. Li and W. Deng. Deep facial expression recognition: A survey. *CoRR*, abs/1804.08348, 2018.

[12] J.-J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. *2017 IEEE CVPR*, pages 3691–3700, 2017.

[13] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE CVPR*, June 2016.

[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. *BMVC 2015*, (Section 3):41.1–41.12, 2015.

[15] G. Rajamanoharan and T. F. Cootes. Multi-view constrained local models for large head angle facial tracking. In *2015 IEEE ICCVW*, pages 971–978, Dec 2015.

[16] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE ICCVW*, pages 397–403, Dec 2013.

[17] 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. In *2015 IEEE ICCVW*, pages 1003–1011, Dec 2015.

[18] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE CVPR*, pages 1701–1708, June 2014.

[20] D. Triantafyllidou and A. Tefas. Face detection based on deep convolutional neural networks exploiting incremental facial part learning. In *2016 23rd ICPR*, pages 3560–3565, Dec 2016.

[21] M. Uricár, V. Franc, and V. Hlavác. Facial landmark tracking by tree-based deformable part model based detector. *2015 IEEE ICCVW*, pages 963–970, 2015.

[22] Q. Wang, F. Chen, W. Xu, and M. Yang. Online discriminative object tracking with local sparse representation. In *2012 IEEE WACV*, pages 425–432, Jan 2012.

[23] Y. Wu and Q. Ji. Shape augmented regression method for face alignment. *2015 IEEE ICCVW*, pages 979–985, 2015.

[24] Y. Wu and Q. Ji. Facial landmark detection: A literature survey. *Int J Comput Vision*, pages 1–28, 2018.

[25] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE CVPR*, CVPR '13, pages 2411–2418, Washington, DC, USA, 2013. IEEE Computer Society.

[26] S. Xiao, S. Yan, and A. A. Kassim. Facial landmark detection via progressive initialization. In *2015 IEEE ICCVW*, pages 986–993, Dec 2015.

[27] J. Xie, R. B. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV 2016*, pages 842–857, 2016.

[28] J. Yang, J. Deng, K. Zhang, and Q. Liu. Facial shape tracking via spatio-temporal cascade shape regression. In *2015 IEEE ICCVW*, pages 994–1002, Dec 2015.

[29] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014.

[30] S. Zafeiriou, G. G. Chrysos, A. Roussos, E. Ververas, J. Deng, and G. Trigeorgis. The 3D Menpo Facial Landmark Tracking Challenge. *Proceedings - 2017 IEEE ICCVW 2017*, 2018-Janua:2503–2511, 2018.

[31] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen. The menpo facial landmark localisation challenge: A step towards the solution. In *2017 IEEE CVPRW*, pages 2116–2125, July 2017.

[32] K. Zhang, Z. Zhang, Z. Li, S. Member, Y. Qiao, and S. Member. Joint Face Detection and Alignment using Multi - task Cascaded Convolutional Networks. *Spl*, (1):1–5, 2016.

[33] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *2015 IEEE CVPR*, pages 4998–5006, June 2015.

[34] X. Zhu, xiaoming Liu, Z. Lei, and S. Z. Li. Face Alignment In Full Pose Range: A 3D Total Solution. *IEEE T Pattern Anal*, pages 1–14, 2017.