2019 IEEE Winter Conference on Applications of Computer Vision

Video-based Face Alignment with Local Motion Modeling

R. Belmonte[1,2] N. Ihaddadene[1] P. Tirilly[3] M. Bilasco[2] C. Djeraba[3]
[1]ISEN Lille, Yncrea Hauts-de-France, France
[2]Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France
[3]Univ. Lille, CNRS, Centrale Lille, IMT Lille Douai, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France
romain.belmonte@yncrea.fr

Abstract

Face alignment remains difficult under uncontrolled conditions due to the many variations that may considerably impact facial appearance. Recently, video-based approaches have been proposed, which take advantage of temporal coherence to improve robustness. These new approaches suffer from limited temporal connectivity. We show that early, direct pixel connectivity enables the detection of local motion patterns and the learning of a hierarchy of motion features. We integrate local motion to the two predominant models in the literature, coordinate regression networks and heatmap regression networks, and combine it with late connectivity based on recurrent neural networks. The experimental results on two datasets, 300VW and SNaP-2DFe, show that local motion improves video-based face alignment and is complementary to late temporal information. Despite the simplicity of the proposed architectures, our best model provides competitive performance with more complex models from the literature.

1. Introduction

The problem of face alignment, also called facial landmark detection, receives a lot of attention due to its importance in many facial analysis tasks, e.g., identification, expression recognition, human-computer interaction, and 3D reconstruction [14]. Given the position and size of a face, the alignment process consists in modeling non-rigid facial structures. It can be done by identifying facial landmarks, which are usually located around the eyes, nose and mouth. From these landmarks, it is then easier to remove the transformation, using for example Procrustes analysis [7], to achieve the alignment of two or more faces. Under uncontrolled conditions, the variations that may impact facial appearance, e.g., variations in pose, expression, illumination, occlusion, or image blur, associated with the instability of face detection, make it a difficult problem. A large number of methods have been proposed to achieve robust detection in such scenarios [14]. Most of them are based on cascaded regression or deep neural networks [35]. The latter either regress the coordinates directly [36, 20, 29, 30] or compute heatmaps, one for each landmark, using a fully convolutional network (FCN) [3, 21].

Despite considerable progress in recent years, the performance of face alignment under uncontrolled conditions is still not fully satisfactory [35] and, even today, this problem continues to be studied largely from still images. Yet, with the ubiquity of video sensors, the vast majority of applications rely on videos. Current methods, when applied to videos, usually track landmarks by detecting them and are therefore not able to leverage the temporal dimension [27, 6]. Recent work has proved that taking into account video consistency helps to deal with the variability in the facial appearance and the ambient environment encountered under uncontrolled conditions [11, 22, 12, 19]. It generally involves a convolutional neural network (CNN) coupled to a recurrent neural network (RNN), which provides only limited temporal connectivity on feature maps with a high level of abstraction. Such architectures can model global motion (e.g., head motion) but not local motion like the movements of the eyes or the lips, which are important to detect facial landmarks accurately.

In this paper, we propose to include local motion information and model it together with the appearance within the same network. Our goal is to better exploit the dynamic nature of the face in order to obtain more stable predictions over time and more robustness to variations that considerably impact facial appearance. Early temporal connectivity using 3D convolutions is applied to both predominant models of the literature, coordinate regression networks and heatmap regression networks. We also explore the combi-

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00228

nation of early and late connectivities. Experimental results show the benefits of early connectivity. To the best of our knowledge, this is the first time that local motion modeling and spatio-temporal FCN architectures are proposed for video-based face alignment.

In the next section, we review the existing solutions with a first sub-section dedicated to image-based approaches, followed by a second dedicated to video-based approaches. Section 3 details our solution. The 2D base architectures are described as well as their extension to the temporal domain. The experimental protocol, implementation details, and results with their analysis are presented in Section 4. Experiments on two datasets, 300VW [27] and SNaP-2DFe [1], are conducted in order to evaluate the results obtained and to compare them with state-of-the-art methods. Finally, we conclude with Section 5 by highlighting some future work.

2. Related Work

2.1. Image-based Face Alignment

The vast majority of the methods proposed in the literature are based on cascaded regression [35]. Cascaded regression is a coarse-to-fine strategy that consists in learning a series of regression functions directly from the appearance of the face to progressively update the position of the landmarks. Regression can be done using a simple linear regression model or using random forests or ferns [5, 33, 17]. Beyond the choice of the regressor, what differentiates these methods is also the initial facial shape and the type of visual features used, such as HOG, SIFT, or simply pixel intensities [38]. Since these handcrafted features are generic, they are not considered as optimal for face alignment. To address this problem, learning-based features are widely used for discriminative and problem-specific feature extraction [25].

Yet these methods still encounter difficulties under uncontrolled conditions. Some work, complementary to ours, is focused specifically on some selected issues. Occlusions can be detected explicitly to improve the robustness to outliers, but it makes the annotation of datasets more onerous [4, 31]. Multiple view-specific models can be used to achieve better accuracy under extreme poses; still, model fusion or model selection are not trivial tasks [10]. Pose variations can also be handled by fitting a 3D dense model to the image [15]. These 3D approaches are used for the data augmentation required for optimal model training as well [39]. Some authors also suggest that face alignment should not be treated as an independent problem and propose to jointly learn various related tasks in order to achieve individual performance gains [40, 37, 24]. This type of approaches can make the training stage much more complex because the optimal convergence rates may vary from one task to another.

Face alignment has also benefited from recent advances in deep learning. With deep neural networks, feature extraction and regression can be trained jointly. Two main architectures have been widely used to replace the traditional approaches: networks with fully connected layers that directly output landmark coordinates [36, 20], and fully convolutional networks that perform heatmap regression and output a heatmap for each landmark [21, 3]. The latter has become popular, especially through hourglass-like architectures [3]. However, they struggle to run in real time. Binarization can be applied to improve speed and reduce the size of the model, but at the expense of accuracy [2]. Cascaded regression can still be used, either by stacking networks [3, 36], or by formulating the cascade as a recurrent process by combining CNNs and RNNs [29, 30]. In this paper, we focus on CNNs for landmark detection, as they provide an effective and versatile baseline tool to solve this problem.

2.2. Video-based Face Alignment

Image-based face alignment methods cannot use the temporal information of image sequences. Recently, a comparative analysis of video-based face alignment methods showed that the most popular strategy for this problem is tracking by detection [27]. Tracking by detection runs independently on each frame without taking into account the coherence of adjacent frames. An alternative is to use a substitute for the detection, such as a rigid tracking algorithm, which is able to capture some variations of the facial appearance during tracking [6]. However, it can easily drift, especially under uncontrolled conditions, and therefore requires the development of a reset mechanism. Cascaded regression, classically used on still images, can be adapted to the temporal domain. For example, the initialization of the current shape can be done using the similarity parameters at the previous frame [34]. Face alignment is therefore no longer dependent on the detection window but takes advantage of the previous pose instead. A more advanced approach consists in integrating an on-line model update in order to make it person-specific and thus more accurate [26]. As with rigid tracking algorithms, it may also drift over time. However, it is possible to establish a synergy between detection and tracking in order to limit the weaknesses of these two approaches but it requires performing both detection and tracking [18].

RNNs can be used to jointly estimate and track visual features over time without any specific nor complex configuration [22, 12] as with traditional Bayesian filters, which have shown their inefficiency for video-based face alignment [11]. When training is done in conjunction with a CNN, RNNs help stabilize predictions over time and improve robustness under uncontrolled conditions. The processing of the spatial and temporal information can also be

decoupled in order to explicitly exploit their complementarity [19]. However, the individual CNN streams in [19] cannot model any motion. These approaches only provide late temporal connectivity on small feature maps with a high level of abstraction at the level of the recurrent layer. The latter is then able to compute global motion features, e.g., head motion, but may not be capable of accurately detecting local motion, e.g., eye and lip movements.

In contrast, we propose to improve the temporal connectivity of deep convolutional neural networks for video-based face alignment, by including local motion to the model. To do so, we extend the convolution layers by one dimension, to perform spatio-temporal convolutions. This provides early temporal connectivity directly at the pixel level, allowing both appearance and motion to be modeled within all layers of the same network. 3D convolution has already been used for other tasks such as action or scene recognition, and has shown its ability to learn relevant spatio-temporal features [13, 28, 16, 23, 8], which makes it a natural candidate to model local motion in face alignment.

3. Face Alignment Regression Networks with Local Motion

This section describes the architectures developed in this work to extend the connectivity of CNN-based landmark detectors to include local motion through early connectivity. We consider lightweight architectures to isolate the contribution of local motions from other factors.

3.1. Problem Formulation

The aim is to model the non-linear relationship between the image and the facial shape. We study coordinate and heatmap regression techniques and we use CNNs to learn these mappings in a supervised manner. To this end, we minimize a L2 cost function that measures the difference between the prediction and the associated ground truth over a batch of n samples.

Coordinate Regression. In the case of coordinate regression, the ground truth is a vector $s = [x_1, y_1, x_2, y_2, ..., x_L, y_L]^T$ with $s \in R^{2L}$, L the number of landmarks and x, y the Cartesian coordinates of a landmark.

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i)^2 \quad (1)$$

Heatmap Regression. For heatmap regression, the ground truth $h \in R^{L\times H\times W}$ is a set of m heatmaps where the coordinates of the maximum value correspond to the coordinates of a landmark.

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(h_{ij} - \hat{h}_{ij})^2 \quad (2)$$

Since we work with RGB videos, the input of our CNN can be a 3-channel image $f \in R^{H\times W\times 3}$ as well as a 3-channel image sequence $v \in R^{T\times H\times W\times 3}$ with $H \times W$ the height and width of the image and T the temporal window.

3.2. Coordinate Regression Networks

2D Baseline. Our 2D baseline coordinate regression model, in Figure 1(a), is inspired by [9]. It contains only convolutions with a kernel size of 3×3, a stride of 1 and a padding of 2, followed by a batch normalization layer, an ELU activation layer and a max-pooling layer with a stride and a padding of 2. The number of filters increases by a factor of 2 at each of the 5 convolutions, from 32 to 512. This model produces feature maps of size 2x2x512 from an input image of size 64x64x3. These maps are vectorized and passed on to two fully-connected layers to obtain the landmark predictions. The first fully-connected layer has a capacity of 1,024 units and is followed by a batch normalization layer and an ELU activation layer. We add a dropout layer with a rate of 0.2 for regularization. The second and last fully connected layer outputs 136 values corresponding to the 68 landmark coordinates.

Convolutional Neural Network (CNN) with Temporal Connectivity. We experiment with different types of connectivities, early and late, each taking as an input a temporal window of 3 frames. As shown in Figure 1(b), local motion information is added by extending the convolution and pooling layers by one dimension [13]. This allows the network to detect local motion patterns and learn a hierarchy of motion features. We use the same parameters as for spatial dimensions, as suggested by Tran et al. [28], except for pooling. Since the temporal depth of the input is only 3, we do not apply any pooling along this dimension to preserve the temporal information. 3D convolution can be expressed as:

$$v_{ij}^{xyz} = b_{ij} + \sum_{m}\sum_{p=0}^{Pi-1}\sum_{q=0}^{Qi-1}\sum_{r=0}^{Ri-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \quad (3)$$

where v_{ij}^{xyz} is the value at position (x, y, z) on the jth feature map in the ith layer, b is the bias, m is the index of the feature map in the previous layer, P, Q, R are the height, width and depth of the kernel, and w is the value of the kernel.

Finally, we integrate late connectivity to these two models by replacing the first fully-connected layer by a long short-term memory (LSTM) [32, 11] recurrent layer with identical capacity, cf. Figures 1(c) and 1(d). The 2D CNN model is distributed in time with shared parameters. By processing the output of the streams, the recurrent layer is able to compute global motion features. Dropout is also performed on recurrent connections, with the same rate as for

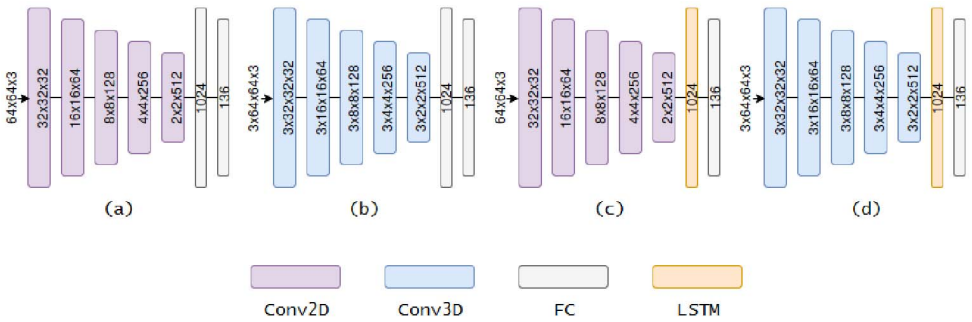

Figure 1. The proposed coordinate regression networks. (a) Baseline 2D architecture, (b) early temporal connectivity based on 3D convolution, (c) late temporal connectivity based on RNNs, and (d) both connectivity levels. These four lightweight architectures are used to evaluate the contribution of local motion for coordinate regression.

feed forward connections. The LSTM can be expressed as:

$$\begin{aligned} i_t &= \text{sigmoid}(W_{xi}X_t + W_{hi}H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\ f_t &= \text{sigmoid}(W_{xf}X_t + W_{hf}H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc}X_t + W_{hc}H_{t-1} + b_c) \\ o_t &= \text{sigmoid}(W_{xo}X_t + W_{ho}H_{t-1} + W_{co} \circ C_t + b_o) \\ H_t &= o_t \circ \tanh(C_t) \end{aligned} \tag{4}$$

where X_t and H_t are respectively the input and the hidden state at time t. W denotes the weights, b the bias and $\circ$ the Hadamard product. C_t is the memory cell of the LSTM, which is updated by the input gate i_t, the forget gate f_t and the output gate o_t. Theses gates help reduce the vanishing gradient effect which is a known issue with conventional RNNs. We keep only the last timestep of the LSTM, which encodes the spatio-temporal context.

3.3. Heatmap Regression Networks

2D Baseline. Our 2D heatmap regression model, illustrated in Figure 2(a), is designed by adapting the 2D base architecture from Section 3.2 to a fully-convolutional auto-encoder architecture. Inspired by recent advances in the literature, we adopt an hourglass-like architecture. At the encoder level, we remove the spatial pooling and replace it by convolutions with strides to obtain a fully-convolutional structure. We use a constant number of 256 filters and a stride of 2 along spatial dimensions. The decoder has an equivalent number of transposed convolutions with identical parameters. Two convolutional layers with a kernel size of 1×1, a stride of 1 and a padding of 2 are applied after the decoder in order to generate heatmap predictions. The first one has 512 filters and is followed by a batch normalization layer and an ELU activation layer. The second one outputs 68 heatmaps, one for each facial landmark.

Fully-Convolutional Network (FCN) with Temporal Connectivity. As before, the transition from 2D to 3D (Figure 2(b)) is done by extending the convolutions to the temporal domain with the same parameters as for spatial dimensions and a temporal depth of 3. Both models integrate late connectivity through the use of a convolutional LSTM layer between the encoder and the decoder, with the same parameters as the other convolutions, except the stride which is set to 1. ConvLSTM turns out to be better than FC-LSTM at handling spatio-temporal correlations [32] and allows to preserve a fully-convolutional architecture. Unlike FC-LSTM, the input and state are 3D tensors and convolutions are used for both input-to-state and state-to-state connections instead of matrix multiplications. Decoding is done in 2D by keeping only the feature maps of the last timestep of the convolutional LSTM layer. We only used dropout on recurrent connections for heatmap regression networks, with a rate of 0.2.

4. Experiments

We conduct experiments on two datasets, 300VW [27] and SNaP-2DFe [1]. 300VW is the dataset commonly used for evaluating video-based face alignment. SNaP-2DFe is a dataset where specific head movements and facial expressions are recorded for each participant; it allows us to study the impact of such motions on face alignment and to identify the respective benefits of early and late connectivities. We also analyze the performance of each architecture in terms of speed, size and number of parameters.

4.1. Datasets and Evaluation Protocols

300VW. 300VW [27] is a dataset from a competition on long-term tracking of facial landmarks in uncontrolled conditions. It contains 114 videos of about 1 minute each and featuring one person, for a total of 218,595 images annotated with 68 landmarks. 50 videos are intended for training and 64 for testing. The test set is divided into 3 categories of increasing difficulty: category 1 presents videos recorded in well-lit conditions with various head poses, category 2 contains additional lighting variations, and category 3 includes severe difficulties such as lighting, occlusions, expression,

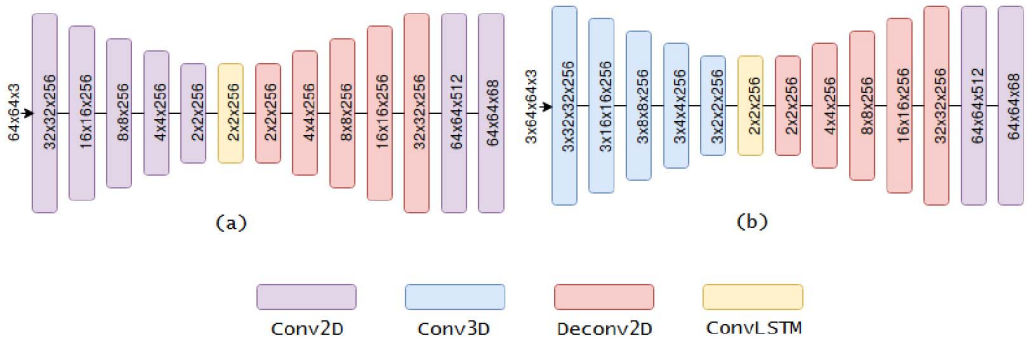

Figure 2. The proposed heatmap regression networks. (a) Late temporal connectivity based on RNNs, and (b) both connectivity levels. These two lightweight architectures are used to evaluate the contribution of local motion for heatmap regression.

and head pose. We keep the same split as the challenge [27] for training and testing on 300VW, without adding any external data. 20% of randomly selected training data was used for validation.

SNaP-2DFe. SNaP-2DFe [1] is a video dataset recently developed to quantify the impact of head movements on expression recognition performance. As it contains landmark annotations, it also provides an interesting partitioning by movement and expression to study video-based face alignment. It consists of 6 movements composed of a horizontal translation and/or a rotation (roll, pitch, yaw), each associated with 7 acted expressions (neutral, joy, fear, anger, disgust, sadness, surprise). Data from 12 participants has been collected, i.e. 37,297 images annotated with 68 landmarks. We use data from subjects 1 to 6 for fine-tuning, 7 to 8 for validation and 9 to 12 for testing.

Pre-processing. All models are trained from scratch without any pre-training on 300VW. On SNaP-2DFe, we fine-tune the models trained on 300VW due to the limited number of images present in SNaP-2DFe. Our training data is augmented by reversing each image sequence and flipping each image horizontally. The same data is used for each training session. The input of each 2D network is an RGB image cropped from the face bounding box, resized to a size of 64x64x3 and normalized by Z-score. For their 3D counterpart, we opt for an input window of size 3x64x64x3. Each video of the training set is then split into several non-overlapping image sequences of size 3. From our experiments, this value seems suitable for face-alignment and ensures a reasonable amount of training data. Ground truth coordinates are normalized between [-0.5; 0.5]. Ground truth heatmaps are generated by computing a bivariate Gaussian of bandwidth 3×3 centered at the location of the landmarks.

Evaluation metrics. In our experiments, we use the metrics of [6], i.e., mean Euclidean distance of the 68 points normalized by the diagonal of the ground truth bounding box, from which we compute the cumulative error distribution, and failure rate with a 8% threshold. We choose this metric for its robustness to pose variations, which often occur in these datasets. For a fair comparison with other existing approaches, we normalize the error by interocular distance as in the 300VW challenge.

4.2. Implementation Details

Network weights are initialized with the Xavier uniform initializer. The Adam optimizer with L2 loss is used for all models, following the original paper parameters except for the learning rate that we set to 10^{-4}. During training, the learning rate is reduced by a factor of 0.1 when the error on the validation set has not improved for 10 epochs. We performed several training sessions of 100 epochs for each model and report the best run. Batch sizes of 8 and 4 are used for the 2D and 3D models, respectively. These parameters have been found to be optimal based on empirical tuning. We observed the same convergence behavior for both types of models, i.e., coordinate regression and heatmap regression.

4.3. Evaluation on SNaP-2DFe

Table 1. Comparison of the different architectures presented in Section 3.2 on SNaP-2DFe (subjects 9 to 12). AUCs and failure rates at thresholds of 8% and 4% are reported.

Method	AUC@8	FR@8	AUC@4	FR@4
2D	78.53	0.27	57.44	0.79
3D	**79.57**	**0.15**	**59.44**	**0.64**
2DRNN	83.32	**0.08**	66.83	0.35
3DRNN	**84.12**	0.09	**68.41**	**0.28**

Table 1 summarizes the performance of coordinate regression models over the entire SNaP-2DFe test set in terms of AUC and FR with thresholds at 8% and 4%. We observe a performance gain between 2D and 3D approaches; it is larger when the threshold is 4%, showing a better ro-

Table 2. Comparison of the different architectures presented in Section 3.2 on SNaP-2DFe (subjects 9 to 12; motion only). AUCs at thresholds of 8% and 4% are reported.

Method	2D		3D		2DRNN		3DRNN	
	@8	@4	@8	@4	@8	@4	@8	@4
Nothing	**82.11**	**64.22**	81.19	62.39	86.20	72.41	**86.84**	**73.68**
Diag	82.29	64.58	**82.77**	**65.54**	85.28	70.55	**85.40**	**70.81**
Pitch	82.69	65.37	**83.56**	**67.12**	84.76	69.51	**85.26**	**70.52**
Roll	79.72	59.43	**80.90**	**61.80**	83.38	66.75	**84.01**	**68.02**
Tx	80.43	60.87	**82.14**	**64.27**	85.28	70.55	**85.33**	**70.66**
Yaw	82.64	65.29	**84.23**	**68.47**	**85.42**	**70.83**	85.23	70.47

Table 3. Comparison of the different architectures presented in Section 3.2 on SNaP-2DFe (subjects 9 to 12; emotion only). AUCs at thresholds of 8% and 4% are reported.

Method	2D		3D		2DRNN		3DRNN	
	@8	@4	@8	@4	@8	@4	@8	@4
Anger	77.97	55.94	**79.27**	**58.53**	83.54	67.09	**84.16**	**68.33**
Disgust	76.35	52.70	**78.76**	**57.52**	83.57	67.15	**84.49**	**68.98**
Fear	77.51	55.01	**78.48**	**56.95**	83.92	67.85	**84.32**	**68.64**
Happy	80.69	61.37	**83.80**	**67.60**	85.60	71.20	**86.42**	**72.83**
Neutral	**82.11**	**64.22**	81.19	62.39	86.20	72.41	**86.84**	**73.68**
Sad	76.75	53.49	**79.15**	**58.31**	82.33	64.67	**84.63**	**69.26**
Surprise	77.29	54.59	**80.93**	**61.85**	84.33	68.67	**85.36**	**70.72**

bustness in a more challenging evaluation context. This underlines the effectiveness of 3D convolutions for landmark localization. Early temporal connectivity provides a significant gain in accuracy while reducing the failure rate. The results of models with RNN layers suggest that early and late temporal connectivities may be complementary.

In order to better understand the strengths and weaknesses of these models, we compute the error for each type of head movement and facial expression. Table 2 presents the results regarding head movements. A notable performance gain is observed for all movements by applying early time connectivity only. However, when no movement occurs, there is a drop in performance, which may indicate a weakness of 3D convolution in static conditions, but could also be due to the fully-connected layer. With the addition of late temporal connectivity, there is also an improvement in performance, but it is more modest. This can be explained by the fact that the RNN layer is already capable of capturing global motion, as shown by the significant improvement from 2D to 2DRNN.

Table 3 presents the results with respect to facial expressions. The performance gain for this kind of motion is more significant than with head movements, even with the addition of late temporal connectivity. Facial expressions consist of local motion patterns that RNNs cannot capture, unlike 3D convolution. In the absence of any expression (i.e., neutral sequences), with early connectivity alone, we obtain lower performances due to the lack of motion, as observed earlier. These results clearly illustrate the value of local motion for face alignment. They also demonstrate the complementarity of local motion modeling and global motion modeling with RNNs.

4.4. Evaluation on 300VW

Coordinate Regression Networks. Table 4 shows the performance of coordinate regression models in terms of AUC and FR with a 8% threshold on the 3 categories of 300VW. Compared to the results obtained on SNaP-2DFe, the contribution of early temporal connectivity is not as clear. However, a gain is systematically observed for models without late temporal connectivity, either in terms of AUC or FR. In categories 1 and 2, we obtain more accurate predictions, while in category 3 we observe a lower failure rate. According to these results and considering the observations made on SNaP-2DFe, it seems that the 3D convolution is affected by the lack of motion and by occlusions, which are common in 300VW.

Table 4. Comparison of the different architectures presented in Section 3.2 on the 3 categories of the 300VW test set. AUC and FR at a threshold of 8% are reported.

Method	Category 1		Category 2		Category 3	
	AUC	FR	AUC	FR	AUC	FR
2D	71.82	**1.73**	67.17	**0.61**	**65.48**	4.54
3D	**72.26**	1.88	**67.62**	2.21	64.23	**4.22**
2DRNN	76.04	**1.52**	**73.45**	**0.14**	**71.25**	2.83
3DRNN	**76.21**	1.60	73.14	0.17	70.86	**2.70**

Table 5. Comparison of the different architectures presented in Section 3.3 on the 3 categories of the 300VW test set. AUC and FR at a threshold of 8% are reported.

Method	Category 1		Category 2		Category 3	
	AUC	FR	AUC	FR	AUC	FR
2DFCRNN	73.99	3.79	73.73	**0.30**	70.55	4.91
3DFCRNN	**75.25**	**2.50**	**74.23**	**0.30**	**71.60**	**4.40**

Heatmap Regression Networks. The performance of heatmap regression networks in terms of AUC and FR with a 8% threshold is reported in Table 5. Early temporal connectivity brings here significant and consistent performance gains over the 3 categories of 300VW. Despite the decoding from 2D feature maps, local motion information is preserved well in the reconstruction, which may explain the variance of the results with coordinate regression models. As previously reported, the latter might by mostly due to their fully-connected layers. Due to their fully-convolutional structure, heatmap regression networks seem more suitable to handle spatio-temporal correlations.

Table 6. Comparison of our best models, 3DRNN and 3DFCRNN, with existing models on the 3 categories of the 300VW test set. Mean error is reported.

Method	Category 1	Category 2	Category 3
SDM [33]	7.41	6.18	13.04
CFSS [38]	7.68	6.42	13.67
TCDCN [37]	7.66	6.77	14.98
TSTN [19]	5.36	**4.51**	12.84
3DRNN	**5.34**	5.01	**8.14**
3DFCRNN	5.73	4.83	8.70

,

4.5. Comparison with existing models

Table 6 shows the average error of our best models with major models from the literature on the full 300VW test set. Numbers are reported from [19]. The comparison includes static methods with handcrafted [33, 38] and learned features [37], and approaches that leverage temporal information [19]. The results show the effectiveness of our models, especially on Category 3, which is the most challenging one. Despite their simplicity, our coordinate regression 3D convolutional RNN model (3DRNN) and our heatmap regression 3D fully-convolutional RNN model (3DFCRNN) are among the top-performing methods. Thanks to their ability to capture both local and global motion, they outperform static methods and show competitive performance with other, more complex, video-based approaches.

4.6. Qualitative results

Figure 3 shows some failure case and qualitative results from 300VW data set. We observe that under static conditions (a), our 3D model has issues with accuracy. However, during expression variations (b), it shows the best performance especially around the mouth area. Notice the lack of influence of the RNN in this type of situation. Early connectivity proves to be more suitable for modeling local motions. In the presence of large head movements (c), 3DRNN provides more robustness than 3D alone. The RNN shows its ability to model global motions. The complementarity between early and late connectivities is also highlighted.

In Figure 4, we show the inputs that activate the filters and the filter weights in the first layer for the 2D, 3D and 3DRNN models. It helps to understand what kind of patterns activate the filters. The 2D model encodes local patterns and color while the 3D and 3DRNN models encode variations of local patterns and color.

4.7. Properties of the Networks

Table 7 presents different properties of the architectures proposed in this paper: number of parameters, size of the

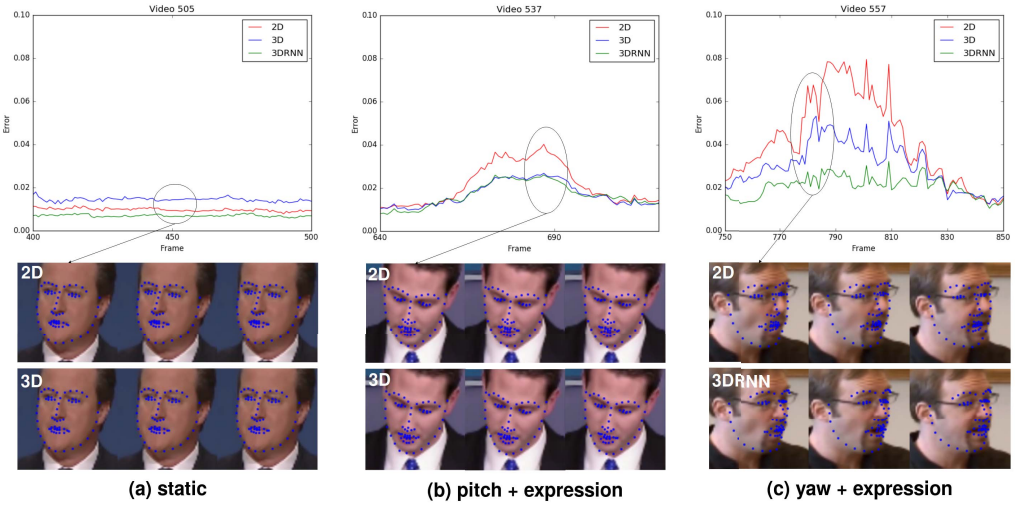

Figure 3. Failure case and qualitative results from 300VW data set. Our 3D model is less accurate under static conditions (a) but handles local motions better (b). When combined with a RNN, the robustness to large motions is improved (c).

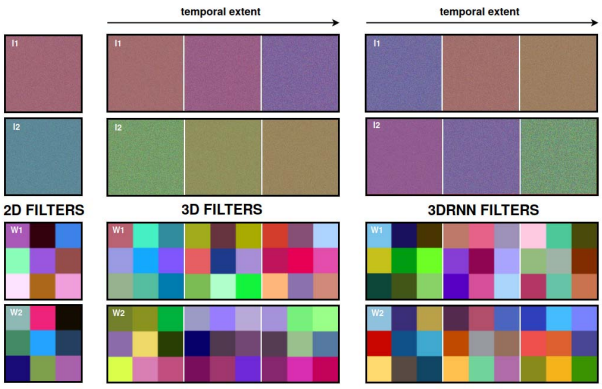

Figure 4. Activation maximization of two filters of the first layer (top) and their weights (bottom), for the 2D (left), 3D (center) and 3DRNN (right) models. The top images are sub-sampled while the bottom images are upsampled to facilitate viewing.

model, and prediction speed (in frames per second – FPS) on CPU and GPU. Runs are performed on a Intel Xeon E5 3.50GHz CPU and a Nvidia GeForce GTX 1080 Ti GPU. Although the number of parameters and the size increase considerably when the temporal dimension is added, the models remain light with less than 20M parameters and 70 MB. As a comparison, an architecture such as VGG16, used for instance in [11, 22], has more than 130M parameters and a size of more than 500 MB. We can also observe that all models run in real time on GPU, and all coordinate regression models on CPU. Moreover, we believe that, by revising the architectures or by applying techniques such as binarization [2], real time could be reachable on CPU for FCN models too.

Table 7. Comparison of the proposed architectures regarding their numbers of parameters, model sizes and speeds.

Method	#params (M)	Size (MB)	Speed (FPS)	
			CPU	GPU
2D	3.8	15	178	285
3D	11.1	43	40	205
2DRNN	14.2	55	60	252
3DRNN	17.4	67	36	205
2DFCRNN	10.2	40	14	104
3DFCRNN	14.9	58	11	99

,

5. Conclusions and Future Work

In this paper, we consider local motion for video-based face alignment. To the best of our knowledge, this is the first work to focus on local motion and to learn low-level spatio-temporal features for this problem; previous work uses RNNs, which rather encode motion at a larger scale. We designed several architectures based on the two main models in the literature: coordinate regression networks and heatmap regression networks. Experiments on two datasets confirm that modeling local motion improves the results (e.g. with expressions, see Table 3), and that it is complementary to RNNs, which model global motion. In future work, spatial and temporal information processing could be decoupled to improve accuracy under static conditions. It might also be interesting to revise spatio-temporal features decoding and to properly manage residual connections between the encoder and the decoder since, from our experience, an efficient 3D hourglass is not trivial to design.

References

[1] B. Allaert, J. Mennesson, I. Bilasco, and C. Djeraba. Impact of the face registration techniques on facial expressions recognition. Signal Processing: Image Communication, 61:44 – 53, 2018.

[2] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In ICCV, 2017.

[3] 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). In IJCV, volume 1, page 8, 2017.

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In ICCV, pages 1513–1520. IEEE, 2013.

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. IJCV, 107(2):177–190, 2014.

[6] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou. A comprehensive performance evaluation of deformable face tracking in-the-wild. IJCV, 126(2-4):198–232, 2018.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. TPAMI, 23(6):681–685, 2001.

[8] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool. Temporal 3D convnets: New architecture and transfer learning for video classification. arXiv preprint arXiv:1711.08200, 2017.

[9] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In CVPR. IEEE, 2018.

[10] Z.-H. Feng, J. Kittler, W. Christmas, P. Huber, and X.-J. Wu. Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting. In CVPR, pages 3681–3690. IEEE, 2017.

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

[12] Q. Hou, J. Wang, R. Bai, S. Zhou, and Y. Gong. Face alignment recurrent network. PR, 74:448–458, 2018.

[13] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. TPAMI, 35(1):221–231, 2013.

[14] X. Jin and X. Tan. Face alignment in-the-wild: A survey. CVIU, 162:1–22, 2017.

[15] A. Jourabloo and X. Liu. Pose-invariant face alignment via cnn-based dense 3D model fitting. IJCV, 124(2):187–203, 2017.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, pages 1725–1732, 2014.

[17] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In CVPR, pages 1867–1874, 2014.

[18] M. H. Khan, J. McDonagh, and G. Tzimiropoulos. Synergy between face alignment and tracking via discriminative global consensus optimization. In ICCV, pages 3811–3819. IEEE, 2017.

[19] H. Liu, J. Lu, J. Feng, and J. Zhou. Two-stream transformer networks for video-based face alignment. TPAMI, 2017.

[20] 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. In CVPR, 2017.

[21] D. Merget, M. Rock, and G. Rigoll. Robust facial landmark detection via a fully-convolutional local-global context network. In CVPR, June 2018.

[22] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas. Red-net: A recurrent encoder–decoder network for video-based face alignment. IJCV, May 2018.

[23] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In ICCV, pages 5534–5542. IEEE, 2017.

[24] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. TPAMI, 2017.

[25] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In CVPR, pages 1685–1692, 2014.

[26] E. Sánchez-Lozano, G. Tzimiropoulos, B. Martinez, F. De la Torre, and M. Valstar. A functional regression approach to facial landmark tracking. TPAMI, 2017.

[27] 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 ICCV Workshop, pages 50–58, 2015.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In ICCV, pages 4489–4497. IEEE, 2015.

[29] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In CVPR, pages 4177–4187, 2016.

[30] W. Wang, S. Tulyakov, and N. Sebe. Recurrent convolutional shape regression. TPAMI, pages 1–1, 2018.

[31] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In ICCV, pages 3658–3666, 2015.

[32] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In NIPS, pages 802–810, 2015.

[33] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In CVPR, pages 532–539. IEEE, 2013.

[34] J. Yang, J. Deng, K. Zhang, and Q. Liu. Facial shape tracking via spatio-temporal cascade shape regression. In ICCV Workshop, pages 41–49, 2015.

[35] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen. The menpo facial landmark localisation challenge: A step towards the solution. In CVPR Workshop, pages 2116–2125, 2017.

[36] 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. Springer, 2014.

[37] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. TPAMI, 38(5):918–930, 2016.

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

[39] X. Zhu, Z. Lei, S. Z. Li, et al. Face alignment in full pose range: A 3D total solution. TPAMI, 2017.

[40] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In CVPR, pages 2879–2886. IEEE, 2012.