International Journal of Computer Vision
https://doi.org/10.1007/s11263-018-1095-1

RED-Net: A Recurrent Encoder–Decoder Network for Video-Based Face Alignment

Xi Peng[1] · Rogerio S. Feris[2] · Xiaoyu Wang[3] · Dimitris N. Metaxas[1]

Received: 19 April 2017 / Accepted: 18 April 2018

Abstract

We propose a novel method for real-time face alignment in videos based on a recurrent encoder–decoder network model. Our proposed model predicts 2D facial point heat maps regularized by both detection and regression loss, while uniquely exploiting recurrent learning at both spatial and temporal dimensions. At the spatial level, we add a feedback loop connection between the combined output response map and the input, in order to enable iterative coarse-to-fine face alignment using a *single network model*, instead of relying on traditional cascaded model ensembles. At the temporal level, we first decouple the features in the bottleneck of the network into *temporal-variant factors*, such as pose and expression, and *temporal-invariant factors*, such as identity information. Temporal recurrent learning is then applied to the decoupled temporal-variant features. We show that such feature disentangling yields better generalization and significantly more accurate results at test time. We perform a comprehensive experimental analysis, showing the importance of each component of our proposed model, as well as superior results over the state of the art and several variations of our method in standard datasets.

Keywords Recurrent learning · Encoder–Decoder network · Face alignment

1 Introduction

Face landmark detection plays a fundamental role in many computer vision tasks, such as face recognition/verification, expression analysis, person identification, and 3D face modeling. It is also the basic technology component for a wide range of applications like video surveillance, emotion recognition, augmented reality on faces, etc. In the past few years, many methods have been proposed to address this problem, with significant progress being made towards systems that work in real-world conditions ("in the wild").

Multiple lines of research have been explored for face alignment in last two decades. Early research includes methods based on active shape models (ASMs) (Cootes and Taylor 1992; Milborrow and Nicolls 2008) and active appearance models (AAMs) (Gao et al. 2010). ASMs iteratively deform a shape model to the target face image, while AAMs impose both shape and object appearance constraints in the optimization process. Recent advances in the field are largely driven by regression-based techniques (Xiong and De la Torre 2013; Cao et al. 2014; Zhang et al. 2014a; Lai et al. 2015; Zhang et al. 2014b). These methods usually take advantage of large-scale annotated training sets (lots of faces with labeled landmark points), achieving accurate results by learning discriminative regression functions that directly map facial appearance to landmark coordinates. The features extracted for regressing landmarks can be either hand-crafted features (Xiong and De la Torre 2013; Cao et al. 2014), or features extracted from convolutional neural networks (Zhang et al. 2014a; Lai et al. 2015; Zhang et al. 2014b). Although these methods can achieve very reliable results in standard benchmark datasets, they still suffer from limited perfor-

Communicated by Xiaoou Tang.

✉ Xi Peng
xipeng.cs@rutgers.edu

Rogerio S. Feris
rsferis@us.ibm.com

Xiaoyu Wang
fanghuaxue@gmail.com.com

Dimitris N. Metaxas
dnm@cs.rutgers.edu

[1] Rutgers University, Piscataway, NJ 08854, USA

[2] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

[3] Intellifusion, Redmond, WA, USA

Published online: 23 May 2018

mance in challenging scenarios, e.g., involving large face pose variations and heavy occlusions.

A promising direction to address these challenges is to consider video-based face alignment (i.e., sequential face landmark detection) (Shen et al. 2015; Peng et al. 2017b), leveraging temporal information and identity consistency as additional constraints (Wang et al. 2016). Despite the long history of research in rigid and non-rigid face tracking (Black and Yacoob 1995; Oliver et al. 1997; Decarlo and Metaxas 2000; Patras and Pantic 2004), current efforts have mostly focused on face alignment in still images (Sagonas et al. 2013; Zhang et al. 2014a; Tzimiropoulos 2015; Zhu et al. 2015). When videos are considered as input, most methods perform landmark detection by independently applying models trained on still images in each frame in a tracking-by-detection manner (Wang et al. 2015), with notable exceptions such as (Asthana et al. 2014; Peng et al. 2015b, 2016b), which explore incremental learning based on previous frames. These methods do not take full advantage of the temporal information to predict face landmarks for each frame. How to effectively model long-term temporal constraints while handling large face pose variations and occlusions is an open research problem for video-based face alignment.

In this work, we address this problem by proposing a novel recurrent encoder–decoder deep neural network model (see Fig. 1), named as ***RED-Net***. The encoding module projects image pixels into a low-dimensional feature space, whereas the decoding module maps features in this space to 2D facial point maps, which are further regularized by a regression loss.

Our encoder–decoder framework allows us to explore spatial refining of our landmark prediction results, in order to handle faces with large pose variations. More specifically, we introduce a feedback loop connection between the aggregated 2D facial point maps and the input. The intuition is similar to cascading multiple regression functions (Xiong and De la Torre 2013; Zhang et al. 2014a) for iterative coarse-to-fine face alignment, but in our approach the iterations are modeled jointly with shared parameters, using a single network model. It provides significant parameter reduction when compared to traditional methods based on cascaded neural networks. A recurrent structure also avoids the effort to explicitly divide the task into multiple stage prediction problems. This subtle difference makes the recurrent model more elegant in terms of holistic optimization. It can implicitly track the prediction behavior in different iterations for a specific face example, while cascaded predictions can only look at the immediate previous cascade stage. Our design also shares the same spirit of residual networks (He et al. 2016). By adding feedback connections from the predicted heat map, the network only needs to implicitly predict the residual from previous predictions in subsequent iterations, which is arguably easier and more effective than directly predicting the absolute location of landmark points.

For more effective temporal modeling, we first decouple the features in the bottleneck of the network into temporal-variant factors (Peng et al. 2017a), such as pose and expression, and temporal-invariant factors, such as identity. We disentangle the features into two components, where one component is used to learn face recognition using identity labels, and the other component encodes temporal-variant factors. To utilize temporal coherence in our framework, we apply recurrent temporal learning to the temporal-variant component. We used Long Short Term Memory (LSTM) to implicitly motion patterns by looking at multiple successive video frames, and use this information to improve landmark fitting accuracy. Landmarks with large pose variation are typically outliers in a landmark training set. By looking at multiple frames, it helps to reduce the inherent prediction variance in our model.

We show in our experiments that our encoder–decoder framework and its recurrent learning in both spatial and temporal dimensions significantly improve the performance of sequential face landmark detection. In summary, our work makes the following **contributions**:

- We propose a novel recurrent encoder–decoder network model for real-time sequential face landmark detection. To the best of our knowledge, this is the first time a recurrent model is investigated to perform video-based facial landmark detection.
- Our proposed *spatial recurrent learning* enables a novel iterative coarse-to-fine face alignment using a single network model. This is critical to handle large face pose changes and a more effective alternative than cascading multiple network models in terms of accuracy and memory footprint.
- Different from traditional methods, we apply *temporal recurrent learning* to temporal-variant features which are decoupled from temporal-invariant features in the bottleneck of the network, achieving better generalization and more accurate results.
- We provide a detailed experimental analysis of each component of our model, as well as insights about key contributing factors to achieve superior performance over the state of the art. The project page is publicly available.[1]

2 Related Work

Face alignment has a long history of research in computer vision. Here we briefly discuss face alignment works related to our approach, as well as advances in deep learning, like

[1] https://sites.google.com/site/xipengcshomepage/eccv2016

the development of recurrent and encoder–decoder neural networks.

Regression-based face landmark detection Recently, regression-based face landmark detection methods (Asthana et al. 2003; Sun et al. 2013; Xiong and De la Torre 2013; Cao et al. 2014; Zhang et al. 2014a; Asthana et al. 2014; Zhu et al. 2015; Tzimiropoulos 2015; Jourabloo and Liu 2016; Wu and Ji 2016; Zhu et al. 2016) have achieved significant boost in the generalization performance of face landmark detection, compared to algorithms based on statistical models such as Active shape models (Cootes and Taylor 1992; Milborrow and Nicolls 2008) and Active appearance models (Gao et al. 2010). Regression-based approaches directly regress landmark locations based on features extracted from face images. Landmark models for different points are learned either in an independent manner or in a joint fashion (Cao et al. 2014). When all the landmark locations are learned jointly, implicit shape constraints are imposed because they share the same or partially the same regressors. This paper performs landmark detection via both a classification model and a regression model. Different from most previous methods, this work deals with face alignment in a video. It jointly optimizes detection output by utilizing multiple observations from the same person.

Cascaded models for landmark detection Additional accuracy improvement in face landmark detection performance can be obtained by learning cascaded regression models. Regression models from earlier cascade stages learn coarse detectors, while later cascade stages refine the result based on early predictions. Cascaded regression helps to gradually reduce the prediction variance, thus making the learning task easier for later stage detectors. Many methods have effectively applied cascade-like regression models for the face alignment task (Xiong and De la Torre 2013; Sun et al. 2013; Zhang et al. 2014a). The supervised descent method (Xiong and De la Torre 2013) learns cascades of regression models based on SIFT features. Sun et al. (2013) proposed to use three levels of neural networks to predict landmark locations. Zhang et al. (2014a) studied the problem via cascades of stacked auto-encoders which gradually refine the landmark position with higher resolution inputs. Compared to these efforts which explicitly define cascade structures, our method learns a spatial recurrent model which implicitly incorporates the cascade structure with shared parameters. It is also more “end-to-end” compared to previous works that divide the learning process into multiple stages.

Face alignment in videos Most face alignment algorithms utilize temporal information by initializing the location of landmarks with detection results from the previous frame, performing alignment in a tracking-by-detection fashion (Wang et al. 2015). Asthana et al. (2014) and Peng et al. (2015b, 2016b) proposed to learn a person-specific model using incremental learning. However, incremental learning (or online learning) is a challenging problem, as the incremental scheme has to be carefully designed to prevent model drifting. In our framework, we do not update our model online. All the training is performed offline and we expect our LSTM unit to capture landmark motion correlations.

Recurrent neural networks Recurrent neural networks (RNNs) are widely employed in the literature of speech recognition (Mikolov et al. 2010) and natural language processing (Mikolov et al. 2014). They have also been recently used in computer vision. For instance, in the tasks of image captioning (Karpathy and Fei-Fei 2015) and video captioning (Yao et al. 2015), RNNs are usually employed for text generation. RNNs are also popular as a tool for action classification. As an example, Veeriah et al. (2015) use RNNs to learn complex time-series representations via high-order derivatives of states for action recognition.

Encoder–decoder networks Encoder and decoder networks are well studied in machine translation (Cho et al. 2014) where the encoder learns the intermediate representation and the decoder generates the translation from the representation. It is also investigated in speech recognition (Lu et al. 2015) and computer vision (Badrinarayanan et al. 2015; Hong et al. 2015). Yang et al. (2015) proposed to decouple identity units and pose units in the bottleneck of the network for 3D view synthesis. However, how to fully utilize the decoupled units for correspondence regularization (Long et al. 2014b) is still unexplored. In this work, we employ the encoder to learn a joint representation of identity, pose, expression as well as landmarks. The decoder translates the representation to landmark heatmaps. Our spatial recurrent model loops the whole encoder–decoder framework.

3 Method

The task is to locate facial landmarks in sequential images using an end-to-end deep neural network. Figure 1 shows an overview of our approach. The network consists of a series of nonlinear and multi-layered mappings, which can be functionally categorized as four modules: **(1)** encoder–decoder f_{enc} and f_{dec}, **(2)** spatial recurrent learning f_{srn}, **(3)** temporal recurrent learning f_{trn}, and **(4)** constrained identity disentangling f_{cls}. Details of the novelty are described in following sections.

3.1 Encoder–Decoder

The input of the encoder–decoder is a single video frame $\mathbf{x} \in \mathbb{R}^{W \times H \times 3}$ and the output is a response map $\mathbf{z} \in \mathbb{R}^{W \times H \times C_z}$ which indicates landmark locations. $C_z = 7$ or 68 depending on the number of landmarks to be predicted.

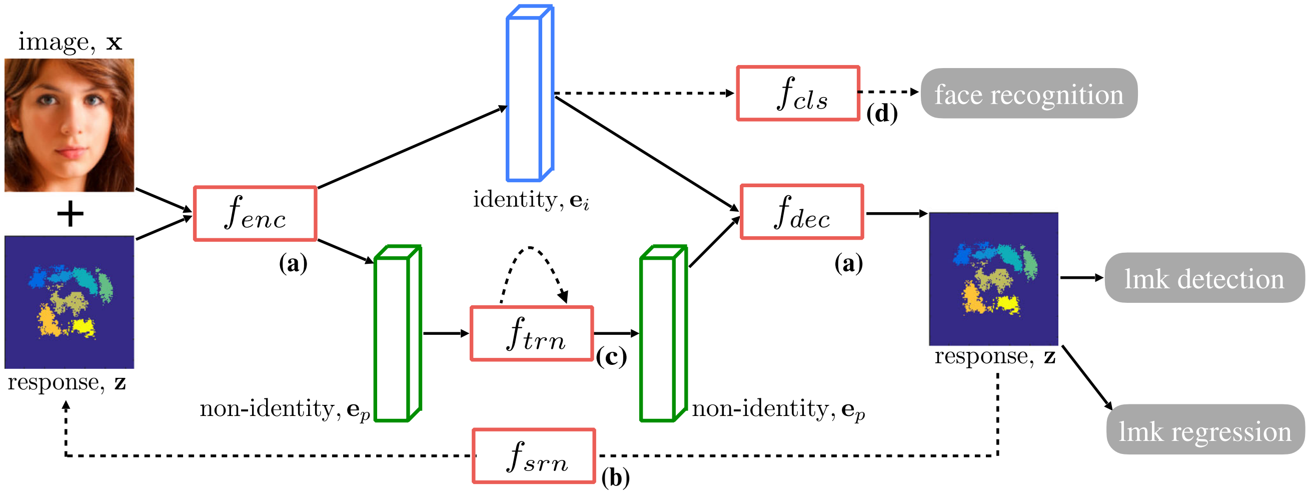

Fig. 1 Overview of the recurrent encoder–decoder network: (a) encoder–decoder (Sect. 3.1); (b) spatial recurrent learning (Sect. 3.2); (c) temporal recurrent learning (Sect. 3.3); and (d) supervised identity disentangling (Sect. 3.4). f_{enc}, f_{dec}, f_{srn}, f_{trn}, f_{cls} are potentially nonlinear and multi-layered mappings

The *encoder* performs a sequence of convolution, pooling and batch normalization (Ioffe and Szegedy 2015) to extract a low-dimensional representation $\mathbf{e}$ from both $\mathbf{x}$ and $\mathbf{z}$:

$$\mathbf{e} = f_{enc}\big(\mathbf{x}, \mathbf{z}; \theta_{enc}\big), \;\; f_{enc} : \mathbb{R}^{W \times H \times C} \rightarrow \mathbb{R}^{W_e \times H_e \times C_e}, \quad (1)$$

where $f_{enc}\big(\cdot; \theta_{enc}\big)$ denotes the encoder mapping with parameters θ_{enc}. We concatenate $\mathbf{x}$ and $\mathbf{z}$ along the channel dimension thus $C = 3 + C_z$. The concatenation is fed into the encoder as an updated input.

Symmetrically, the *decoder* performs a sequence of unpooling, convolution and batch normalization to upsample the representation code to the response map:

$$\mathbf{z} = f_{dec}(\mathbf{e}; \theta_{dec}), \;\; f_{dec} : \mathbb{R}^{W_e \times H_e \times C_e} \rightarrow \mathbb{R}^{W \times H \times C_z}, \quad (2)$$

where $f_{dec}\big(\cdot; \theta_{dec}\big)$ denotes the decoder mapping with parameters θ_{dec}. $\mathbf{z}$ has the same $W \times H$ dimension as $\mathbf{x}$ but C_z channels for C_z landmarks. Each channel presents pixel-wise confidences of the corresponding landmark.

The encoder–decoder design plays an important role in our task. **First**, the decoder's output $\mathbf{z}$ has the same resolution (but a different number of channels) as the input image $\mathbf{x}$. Thus it is easy to directly concatenate $\mathbf{z}$ with $\mathbf{x}$ along the channel dimension. The concatenation provides pixel-wise spatial cues to update the landmark prediction by the proposed *spatial recurrent learning* (f_{srn}). We will explain it soon in Sect. 3.2.

Second, the encoder–decoder network can achieve a low-dimensional representation $\mathbf{e}$ in the bottleneck. We can utilize the domain prior to decouple $\mathbf{e}$ into two parts: the identity code $\mathbf{e}_i$, which is temporal-invariant as we are tracking the same person; and the non-identity code $\mathbf{e}_p$, which models temporal-variant factors such as head pose, expression, illumination, and etc.

In Sect. 3.3, we propose the *temporal recurrent learning* (f_{trn}) to model the changes of $\mathbf{e}_p$. In Sect. 3.4, we show how to speed up the network training by carrying out the *supervised identity disentangling* (f_{cls}) on $\mathbf{e}_i$.

Third,the encoder–decoder network enables a fully convolutional design. The bottleneck embedding $\mathbf{e}$ and output response map $\mathbf{z}$ are feature maps instead of fully-connected neurons that are often used in ordinary convolutional neural networks. This design is highly memory-efficient and can significantly speed up the training and testing (Long et al. 2014a), which is preferred by video-based applications.

3.2 Spatial Recurrent Learning

The purpose of spatial recurrent learning is to pinpoint landmark locations in a coarse-to-fine manner. Unlike existing approaches (Sun et al. 2013; Zhang et al. 2014a) that employ multiple networks in cascade, we accomplish the coarse-to-fine search in a single network in which the parameters are jointly learned in successive recurrent steps.

The spatial recurrent learning is performed by iteratively feeding back the previous prediction, stacked with the image as shown in Fig. 2, to eventually push the shape prediction from an initial guess to the ground truth:

$$\mathbf{z}_k = f_{srn}\big(\mathbf{x}, \mathbf{z}_{k-1}; \theta_{srn}\big), \;\; k = 1, \ldots, K \quad (3)$$

where $f_{srn}\big(\cdot; \theta_{srn}\big)$ denotes the spatial recurrent mapping with parameters θ_{srn}. $\mathbf{z}_0$ is the initial response map, which

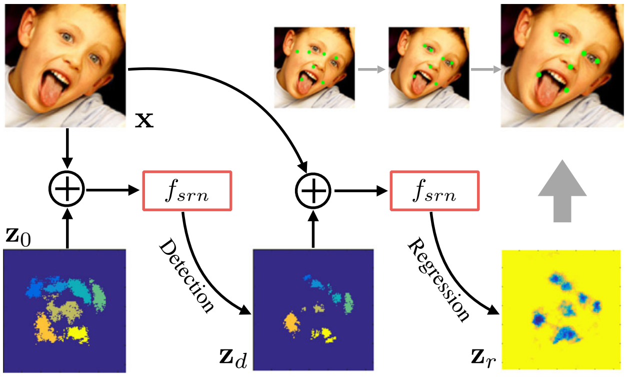

Fig. 2 An unrolled illustration of *spatial recurrent learning*. The response map is pretty coarse when the initial guess is far away from the ground truth if large pose and expression exist. It eventually gets refined in the successive recurrent steps

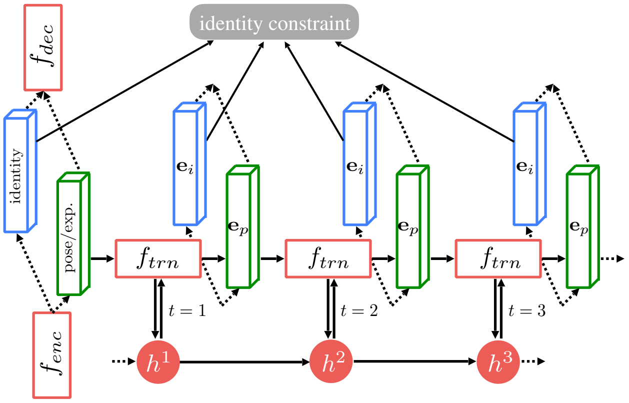

Fig. 3 An unrolled illustration of *temporal recurrent learning*. $\mathcal{C}_i$ encodes temporal-invariant factor which subjects to the same identity constraint. $\mathcal{C}_p$ encodes temporal-variant factors which is further modeled in f_{trn}

could be a response map generated by the mean shape or the output of the previous frame.

In our conference version (Peng et al. 2016a), detection-based supervision is performed in every recurrent step. It is robust to appearance variations but lacks precision, because pixels within a certain radius around the ground-truth location are labeled using the same value. To address this limitation, motivated by Bulat and Tzimiropoulos (2016), we propose to further explore the spatial recurrent learning by performing detection-followed-by-regression in successive steps.

Specially, we carry out a two-step recurrent learning by setting $K = 2$. The first step performs *landmark detection* that aims to locate 7 major facial components (i.e. $C = 7$ in Eq. (2)). The second step performs *landmark regression* that refines all 68 landmarks positions (i.e. $C = 68$). For clarity, we use C_d and C_r to denote the number of channels output by the detection and the regression steps, respectively.

The landmark detection step guarantees fitting robustness especially in large pose and partial occlusions. The encoder–decoder aims to output a binary map of C_d channels, one for each major facial component. The detection step outputs:

$$\mathbf{z}_d = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_0; \theta_{enc}); \theta_{dec}\big), \;\; \mathbf{z}_d \in \mathbb{R}^{W \times H \times C_d}, \tag{4}$$

where the detection task can be trained using pixel-wise sigmoid cross-entropy loss function:

$$\ell_d = \frac{1}{M_d} \sum_{c=1}^{C_d} \sum_{i=1}^{W} \sum_{j=1}^{H} z_{ij}^c log y_{ij}^c + (1 - z_{ij}^c) log(1 - y_{ij}^c), \tag{5}$$

where $M_d = C_d \times W \times H$. Here z_{ij}^c denotes the sigmoid output at pixel location (i, j) in $\mathbf{z}_d$ for the c-th landmark. y_{ij}^c is the ground-truth label at the same location, which is set to 1 to mark the presence of the corresponding landmark and 0 for the remaining background.

Note that this loss function is different from the N-way cross-entropy loss used in our previous conference paper (Peng et al. 2016a). It allows multiple class labels for a single pixel, which helps to tackle the landmark overlaps.

The landmark regression step improves the fitting accuracy from the outputs of the previous detection step. The encoder–decoder aims to output a heatmap of C_r channels, one for each landmark. The regression step outputs:

$$\mathbf{z}_r = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_{det}; \theta_{enc}); \theta_{dec}\big), \;\; \mathbf{z}_r \in \mathbb{R}^{W \times H \times C_r}, \tag{6}$$

where the regression task can be trained using pixel-wise L_2 loss function:

$$\ell_r = \frac{1}{M_r} \sum_{c=1}^{C_r} \sum_{i=1}^{W} \sum_{j=1}^{H} \left\| z_{ij}^c - y_{ij}^c \right\|_2^2, \tag{7}$$

where $M_r = C_d \times W \times H$. Here z_{ij}^c denotes the heatmap value of the c-th landmark at pixel location (i, j) in $\mathbf{z}_r$ for the c-th landmark. y_{ij}^c is the ground-truth value at the same location, which obeys a Gaussian distribution centered at the landmark with a pre-defined standard deviation.

Now the spatial recurrent learning (Eq. (3)) can be achieved by minimizing the detection loss (Eq. (5)) and the regression loss (Eq. (7)), simultaneously:

$$\underset{\theta_{enc}, \theta_{dec}}{\operatorname{argmin}} \, \ell_d + \lambda \ell_r, \tag{8}$$

where λ balances the loss between the two tasks. Note that the spatial recurrent learning do not introduce new parameters but sharing the same parameters of the encoder–decoder network, i.e. $\theta_{srn} = \{\theta_{enc}, \theta_{dec}\}$.

The spatial recurrent learning is highly memory efficient. It is capable of end-to-end training, which is a significant advantage compared with the cascade framework (Bulat and Tzimiropoulos 2016). More importantly, the network can jointly learn the coarse-to-fine fitting strategy in recurrent steps, instead of training cascaded networks independently (Sun et al. 2013; Zhang et al. 2014a), which guarantees robustness and accuracy in challenging conditions.

3.3 Temporal Recurrent Learning

In addition to the spatial recurrent learning, we also propose a temporal recurrent learning to model factors, e.g. head pose, expression, and illumination, that may change over time. These factors affect the landmark locations significantly (Peng et al. 2015a). Thus we can expect improved tracking accuracy by modeling their temporal variations.

As mentioned in Sect. 3.1, the bottleneck embedding $\mathbf{e}$ can be decoupled into two parts: the identity code $\mathbf{e}_i$ and the non-identity code $\mathbf{e}_p$:

$$\mathbf{e}_i \in \mathbb{R}^{W_e \times H_e \times C_i}, \mathbf{e}_p \in \mathbb{R}^{W_e \times H_e \times C_p}, C_e = C_i + C_p, \tag{9}$$

where $\mathbf{e}_i$ and $\mathbf{e}_p$ model the temporal-invariant and -variant factors, respectively. We leave $\mathbf{e}_i$ to Sect. 3.4 for additional identity supervision, and exploit variations of $\mathbf{e}_p$ via the recurrent model. Please refer to Fig. 3 for an unrolled illustration of the proposed temporal recurrent learning.

Mathematically, given T successive video frames $\{\mathbf{x}^t; t = 1, \ldots, T\}$, the encoder extracts a sequence of embeddings $\{\mathbf{e}_i^t, \mathbf{e}_p^t; t = 1, \ldots, T\}$. Our goal is to achieve a nonlinear mapping f_{trn}, which simultaneously tracks a latent state h^t and updates $\mathbf{e}_p^t$ at time t:

$$\begin{aligned} h^t &= p(\mathbf{e}_p^t, h^{t-1}; \theta_{trn}), \quad t = 1, \ldots, T \\ \mathbf{e}_p^{t*} &= q(h^t; \theta_{trn}), \end{aligned} \tag{10}$$

where $p(\cdot)$ and $q(\cdot)$ are functions of $f_{trn}\left(\cdot; \theta_{trn}\right)$ with parameters θ_{trn}. $\mathbf{e}_p^{t*}$ is the update of $\mathbf{e}_p^t$.

The temporal recurrent learning is trained using T successive frames. At each frame, the detection and regression tasks are performed for the spatial recurrent learning. The recurrent learning is performed by minimizing Eq. (8) at every time step t:

$$\underset{\theta_{enc}, \theta_{dec}, \theta_{trn}}{\text{argmin}} \sum_{t=1}^{T} \ell_d^t + \lambda \ell_r^t, \tag{11}$$

where θ_{trn} denotes network parameters of the temporal recurrent learning, e.g. parameters of LSTM units. It is worth mentioning that, we perform recurrent learning in both spatial and temporal dimensions by jointly optimizing $\{\theta_{enc}, \theta_{dec}, \theta_{trn}\}$ in Eq. (11).

The temporal recurrent module is memorizing as well as modeling the changing pattern of the temporal-variant factors. Our experiments indicated that the offline learned model can significantly improve the online fitting accuracy and robustness, especially when large variations or partial occlusions happen.

3.4 Supervised Identity Disentangling

There is no guarantee that temporal-invariant and -variant factors can be completely decoupled in the bottleneck by simply splitting the bottleneck representation $\mathbf{e}$ into two parts (Peng et al. 2017a). More supervised information is required to achieve the disentangling. To address this issue, we propose to apply a face recognition task on the identity code $\mathbf{e}_i$, in addition to the temporal recurrent learning applied on non-identity code $\mathbf{e}_p$.

The supervised identity disentangling is formulated as an N-way classification problem. N is the number of unique individuals present in the training sequences. In general, we associate the identity representation $\mathbf{e}_i$ with a one-hot encoding $\mathbf{z}_i$ to indicate the score of each identity:

$$\mathbf{z}_i = f_{cls}(\mathbf{e}_i; \theta_{cls}), \;\; f_{cls} : \mathbb{R}^{W_e \times H_e \times C_i} \rightarrow \mathbb{R}^N, \tag{12}$$

where $f_{cls}(\cdot; \theta_{cls})$ is the identity classification mapping with parameters θ_{cls}. The identity task is trained using N-way cross-entropy loss:

$$\ell_{cls} = \frac{1}{N} \sum_{n=1}^{N} z^n log y^n + (1 - z^n) log(1 - y^n), \tag{13}$$

where z^n denotes the softmax activation of the n-th element in $\mathbf{z}_i$. y^n is the n-th element of the identity annotation $\mathbf{y}_i$, which is a one-hot vector with a 1 for the correct identity and all 0s for others.

Now we can jointly train all the three tasks, i.e. f_{srn}, f_{trn}, and f_{cls}. Based on Eqs. (11) and (13), we simultaneously minimize the detection and regression loss together with the identity loss at every time step t:

$$\underset{\theta_{enc}, \theta_{dec}, \theta_{trn}, \theta_{cls}}{\text{argmin}} \sum_{t=1}^{T} \ell_{det}^t + \lambda \ell_{reg}^t + \gamma \ell_{cls}^t, \tag{14}$$

where γ weights the identity constraint. An obvious advantage of our approach is that the whole network can be trained end-to-end by optimizing all parameters $\{\theta_{enc}, \theta_{dec}, \theta_{trn}, \theta_{cls}\}$ simultaneously, which guarantees an efficient learning.

Table 1 Specification of the VGGNet-based $f_{enc/dec}$ design: block name (**Top**), feature map dimension (**Middle**), and layer configuration (**Bottom**). $[3 \times 3, 64]$ means there are 64 filters (channels), each has a size of 3 × 3. Pooling or unpooling operations are performed after or before each module. The pooling window is 2 × 2 with a stride of 2

A_0	A_1	A_2	A_3	A_4	B_4	B_3	B_2	B_1
128 × 128	64 × 64	32 × 32	16 × 16	8 × 8	16 × 16	32 × 32	64 × 64	128 × 128
2× conv	2× conv	3× conv	3× conv	3× conv	unpooling	unpooling	unpooling	unpooling
$[3 \times 3, 64]$	$[3 \times 3, 128]$	$[3 \times 3, 256]$	$[3 \times 3, 512]$	$[3 \times 3, 512]$	3× conv	3× conv	3× conv	2× conv
pooling	pooling	pooling	pooling	–	$[3 \times 3, 512]$	$[3 \times 3, 512]$	$[3 \times 3, 256]$	$[3 \times 3, 128]$

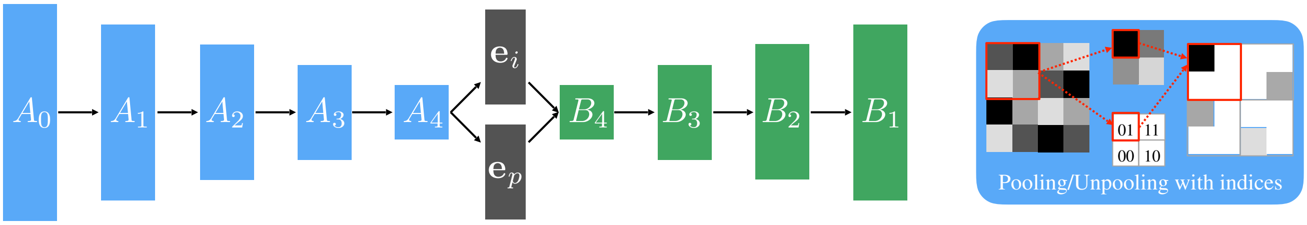

Fig. 4 **Left**: the architecture of the VGGNet-based $f_{enc/dec}$ design. The encoder (A_{0-4}) and the decoder (B_{4-1}) are nearly symmetrical except that f_{enc} has one more block A_0. A_0 downsamples the input image from 256 × 256 to 128 × 128. So $\mathbf{x}$ and $\mathbf{z}$ have the same resolution and can be easily concatenated along the channel dimension. **Right**: an illustration of the pooling/unpooing with indices. The corresponding pooling and unpooling share pooling indices using a 2-bit switch in each 2 × 2 pooling window

It has been shown in Zhang et al. (2014b) that learning the face alignment task together with correlated tasks, e.g. head pose, can improve the fitting performance. We have a similar observation when adding face recognition task to the alignment task. More importantly, we find that the additional identity task can effectively speed up the training of the entire encoder–decoder network. In addition to more supervision, the identity task helps to decouple the identity and non-identity factors more completely, which facilitates the training of the temporal recurrent learning.

4 Network Architecture

We present the architecture details of proposed modules: $f_{enc/dec}$, f_{srn}, f_{trn}, and f_{cls}. All the four modules are designed in a single network that can be trained end-to-end. We first introduce two variant designs of $f_{enc/dec}$, based on which f_{srn}, f_{trn}, and f_{cls} are designed accordingly.

4.1 The Design of f_{enc} and f_{dec}

To best evaluate the proposed method, we investigate two variant designs of the encoder–decoder: VGGNet (Simonyan and Zisserman 2014) based and ResNet (He et al. 2016) based. The VGGNet-based design has a symmetrical structure between the encoder and decoder; while the ResNet-based design has an asymmetrical structure due to the usage of the residual modules.

VGGNet-based design Table 1 presents the network specification. Figure 4 (left) shows the network architecture. The encoder is designed based on a variant of the VGG-16 network (Simonyan and Zisserman 2014; Kendall et al. 2015). It has 13 convolutional layers of constant 3 × 3 filters. We can, therefore, initialize the training process from weights trained on large datasets for object classification. We remove all fully connected layers in favor of a fully convolutional manner (Long et al. 2014a), which can effectively reduce the number of parameters from 117 to 14.8 M (Badrinarayanan et al. 2015). The bottleneck feature maps are split into two parts for the identity and non-identity codes, respectively. This design preserves rich spatial information in 3D feature maps rather than 1D feature vectors, which is important for landmark localization.

We use max-pooling to halve the feature resolution at the end of each convolutional block. The pooling window size is 2 × 2 and the stride is 2. Although max-pooling can help to achieve translation invariance, it would cause a considerable loss of spatial information especially when multiple max-pooling layers are applied in a cascade. To solve this issue, we use a 2-bit code to record the index of the maximum activation selected in a 2 × 2 pooling window (Zeiler and Fergus 2014). As illustrated in Fig. 4 (right), the memorized index is then used in the corresponding unpooling layer to place each activation back to its original location. This strategy is particularly useful when the decoder recovers the input structure from highly compressed feature maps. Besides, it is more efficient to store spatial indices than to memorize entire feature maps of float precision, which is a common setup in FCNs (Long et al. 2014a).

The decoder is nearly symmetrical to the encoder with a mirrored configuration but replacing all max-pooling with

Table 2 Specification of the ResNet-based $f_{enc/dec}$ design: block name (**Top**), feature map dimension (**Middle**), and layer configuration (**Bottom**). We use conv/decov layers with a stride of 2 to halve or double the feature map dimensions. Thus no pooling/unpooling layer is used. The skip connections E_{1-3} are specified in Table 3

C_0	C_1	C_2	C_3	C_4	D_4	D_3	D_2	D_1
128 × 128	64 × 64	32 × 32	16 × 16	8 × 8	16 × 16	32 × 32	64 × 64	128 × 128
1× conv	3× conv	8× conv	36× conv	3× conv	1× dconv	1× dconv	1× dconv	1× dconv
$\begin{bmatrix} 7 \times 7, 64 \\ \text{strid}, 2 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 512 \\ \text{stride}, 2 \\ 1 \times 1, 1024 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 256 \\ \text{stride}, 2 \\ 1 \times 1, 512 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 128 \\ \text{stride}, 2 \\ 1 \times 1, 256 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 64 \\ \text{stride}, 2 \\ 1 \times 1, 128 \end{bmatrix}$

unpooling layers. The encoder is slightly deeper than the decoder with one more convolutional block A_0 at the beginning. A_0 downsamples the input image from 256 × 256 to 128 × 128. So **x** and **z** have the same resolution and can be easily concatenated along the channel dimension. We find that batch normalization (Ioffe and Szegedy 2015) can significantly boost the training speed since it reduces internal shifting in the mini batch. Thus, we apply batch normalization as well as rectified linear unit (ReLU) (Nair and Hinton 2010) after each convolutional layer.

ResNet-based design Table 2 presents the network specification. Figure 5 (left) shows the network architecture. The encoder is designed based on a variant of the ResNet-152 (He et al. 2016), which has 50 residual units of totally 151 convolutional layers. Figure 5 (right) shows a residual unit used in C_1. 1 × 1 convolutional layers are used to cut down the number of filter parameters. According to He et al. (2016), the residual shortcut guarantees efficient training of the very deep network, as well as improved performance compared with vanilla design (Simonyan and Zisserman 2014). Stride-2 convolutions instead of max poolings are used to halve the feature map resolution at the end of each block.

Different from the VGGNet-based design, the encoder and decoder are asymmetrical. The encoder is much deeper than the decoder, and the decoder has only 4 upsampling blocks of totally 4 convolutional layers. A practical consideration behind this design is that the encoder has to tackle a complicated task, e.g. understand the image and translate it to a low-dimensional embedding, while the decoder's task is relatively simpler, e.g. recover a set of response maps to mark landmark locations from the embedding. We use stride-2 deconvolutions to double the feature map resolution in each block. Similar to the VGGNet-based design, an additional convolutional block C_0 is used to downsample the input image from 256 × 256 to 128 × 128. So **x** and **z** have the same resolution for an easy channel-wise concatenation.

Another difference between the ResNet-based design and the VGGNet-based design is the usage of skip connections E_{3-1} (Oh et al. 2015) as shown in Fig. 5 and specified in Table 3. These skip connections are used to bridge hierarchical spatial information between the encoder and decoder at different resolutions. They provide shortcuts of the gradient flow in backpropagation for efficient training. Besides, they also enable us to use a shallow decoder design since rich spatial information can be delivered through the shortcuts.

4.2 The Design of f_{srn} and f_{trn}

The design of the proposed f_{srn} and f_{trn} aims to tradeoff between network complexity and training or testing efficiency.

Spatial recurrent learning We perform a two-step spatial recurrent learning. Particularly, the first step performs land-

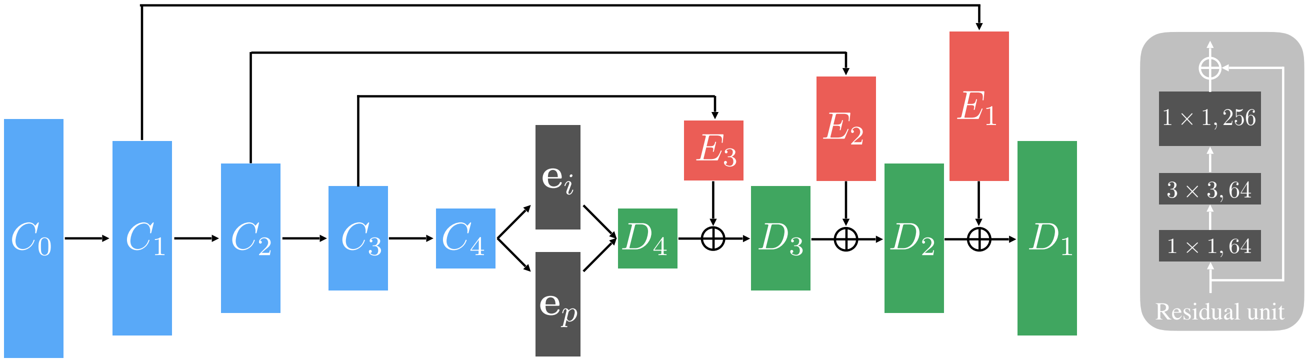

Fig. 5 **Left**: the architecture of ResNet-based $f_{enc/dec}$ design (**Left**). The encoder (C_{0-4}) and the decoder (D_{4-1}) are asymmetrical. f_{enc} is much deeper than f_{dec}, i.e. 151 versus 4 layers. C_0 downsamples the input image from 256 × 256 to 128 × 128. Skip connections (E_{1-3}) are used to bridge hierarchical spatial information at different resolutions. **Right**: an example of residual unit used in C_1. 1 × 1 convolutional layers are used in the residual unit to cut down the number of filter parameters

Table 3 Specification of the skip connections. Note that E_3 and C_1, E_2 and C_2, E_1 and C_1 share the same configurations. The bridged features are directly added to the outputs of D_{4-1} at the corresponding resolutions

E_3	E_2	E_1
16 × 16	32 × 32	64 × 64
3× conv	3× conv	3× conv
$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$

mark detection to locate 7 major facial components that are robust to variations, i.e. four corners of left/right eyes, one nose tip, and two corners of the mouth. The second step performs landmark regression to refine the predicted locations of all 68 landmarks. This coarse-to-fine strategy guarantees efficient and robust spatial recurrent learning.

As mentioned in Sect. 3.2, the ground truth of the detection task is a binary map of $C_d = 7$ channels, in which the values within a radius of 5 pixels around the ground truth are set to 1 and the values for the remaining background are set to 0. The ground truth of the regression task is a heat map of $C_r = 68$ channels, in which the correct locations are represented by Gaussian with a standard deviation of 5 pixels. The detection and regression tasks share the weights of the entire encoder–decoder except for the last convolutional layers, which use 1 × 1 convolutional layers to match different numbers of channels (7 for detection and 68 for regression).

In either landmark detection or regression, the foreground pixels are much less than the background ones, which lead to highly unbalanced loss contributions. To solve this issue, we enlarge the foreground loss defined in Eqs. (8) and (11) by multiplying a constant weight (15 in most cases) to focus more on foreground pixels.

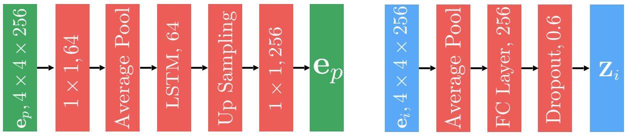

Fig. 6 **Left**: the architecture of f_{trn}. We use 4 × 4 average pooling to cut down the input dimension of LSTM and recover the dimension by upsampling. **Right**: the architecture of f_{cls}. We set $\mathbf{z}_i \in \mathbb{R}^{256}$ for a compact identity representation

Temporal recurrent learning We specify the configuration of f_{trn} in Fig. 6 (left). A Long Short Term Memory (LSTM) module (Hochreiter and Schmidhuber 1997; Oh et al. 2015) is used to model the temporal variations of the identity code. The LSTM contains 64 hidden neurons. We empirically set the number of successive frames as $T = 10$ in Eq. (11). The prediction loss is calculated at each time step. Directly feeding the non-identity code $\mathbf{e}_p$ into LSTM layers would lead to a slow training as it needs a large number of neurons for both the input and output. Instead, we apply average pooling to compress $\mathbf{e}_p$ to a 64d vector before inputting to the LSTM and recover it by unpooling with indices as shown in Fig. 4 (left).

4.3 The Design of f_{cls}

The design of f_cls is shown in Fig. 6 (right). The purpose of f_{cls} is to apply additional identity constraint on $\mathbf{e}_i$, so the identity and non-identity codes can be decoupled more completely. Specially, f_{cls} takes $\mathbf{e}_i$ as the input and output a 256d feature vector for the identity representation. Instead of using a very long feature vector in former face recognition networks (Taigman et al. 2014), e.g. 4096d, we use a compact one, *e.g.* 256d, to reduce the computational cost for efficient training (Schroff et al. 2015; Sun et al. 2015). We apply 0.6 dropout on the 256d vector to avoid overfitting. The vector is then followed by a fully-connected layer of N neurons to output an one-hot vector for the identity prediction, where N is the number of different subjects in training sequences. We use the cross-entropy loss defined in Eq. (13) to train the identity task.

5 Experiments

We first introduce the datasets and settings. Then we carry out a comprehensive study to validate the proposed method in various aspects. Finally, we compare our method with state-of-the-arts on both controlled and in-the-wild datasets.

5.1 Datasets and Settings

Datasets We conduct our experiments on both image and video datasets. These datasets are widely used in face alignment and recognition. They present challenges in multiple aspects such as large pose, extensive expression, severe occlusion and dynamic illumination. Totally 7 datasets are used:

- Annotated Facial Landmarks in the Wild (AFLW) (Koestinger et al. 2011)
- Labeled Faces in the Wild (LFW) (Learned-Miller 2014)
- Helen facial feature dataset (Helen) (Le et al. 2012; Sagonas et al. 2013)
- Labeled Face Parts in the Wild (LFPW) (Belhumeur et al. 2011; Sagonas et al. 2013)
- Talking Face (TF) (FGNet 2004)
- Face Movies (FM) (Peng et al. 2015b)
- 300 face Videos in the Wild (300-VW) (Shen et al. 2015)

We list configurations and setups of each dataset in Table 4. Different datasets have different landmark annotation protocol. For Helen, LFPW, TF, FM and 300-VW, we follow (Sagonas et al. 2013, 2016) to obtain both 68- and 7-landmark annotation. For AFLW, we generate 7-landmark annotations using the original 21 landmarks. The landmark annotation of LFW is given by Learned-Miller (2014). For identity labels, we manually label all videos in TF, FM, and 300-VW. It is easy since the identity is unique is a given video.

AFLW and 300-VW have the largest number of labeled images. They are also more challenging than others due to the extensive variations. Therefore, we use them for both training and evaluation. More specifically, 80% of the images in AFLW and 90 out of 114 videos in 300-VW are used for training, and the rest are used for evaluation. We sample videos to roughly cover the three different scenarios defined in Chrysos et al. (2015), i.e. “Scenario 1”, “Scenario 2” and

Table 4 The image and video datasets used in training and evaluation. We split AFLW and 300-VW into two parts for training and evaluation, respectively. LFW, Helen, LFPW, TF, and FM are used for training only. Note that LFW, TF, FM and 300-VW have both landmark and identity annotations; while the others have only landmark annotations

	AFLW (Koestinger et al. 2011)	LFW (Learned-Miller 2014)	Helen (Le et al. 2012)	LFPW (Belhumeur et al. 2011)	TF (FGNet 2004)	FM (Peng et al. 2015b)	300-VW (Shen et al. 2015)
In-the-wild setting	Yes	Yes	Yes	Yes	No	Yes	Yes
Image number	21,080	12,007	2330	1035	500	2150	114,000
Video number	–	–	–	–	5	6	114
Landmark annotation	21pt	7pt	194pt	68pt	68pt	68pt	68pt
Subject number	–	5,371	–	–	1	6	105
Used in training	16,864	12,007	2330	1035	0	0	90,000
Used in evaluation	4216	0	0	0	500	2150	24,000

"Scenario 3", corresponding to well-lit, mild unconstrained and completely unconstrained conditions.

We perform data augmentation by sampling ten variations from each image in the image training datasets. The sampling was achieved by random perturbation of scale (0.9–1.1), rotation (±15°), translation (7 pixels), as well as horizontal flip. To generate sequential training data, we randomly sample 100 clips from each training video, where each clip has 10 frames. It is worth mentioning that no augmentation is applied on video training data to preserve the temporal consistency in the successive frames.

Compared methods We compared the proposed method with both regression based and deep learning based approaches that reported state-of-the-art performance in unconstrained conditions. Totally 8 methods are compared:

- Discriminative Response Map Fitting (DRMF) (Asthana et al. 2003)
- Explicit Shape Regression (ESR) (Cao et al. 2014)
- Supervised Descent Method (SDM) (Xiong and De la Torre 2013)
- Incremental Face Alignment (IFA) (Asthana et al. 2014)
- Coarse-to-Fine Shape Searching (CFSS) (Zhu et al. 2015)
- Deep Convolutional Network Cascade (DCNC) (Sun et al. 2013)
- Coarse-to-fine Auto-encoder Network (CFAN) (Zhang et al. 2014a)
- Deep Multi-task Learning (TCDCN) (Zhang et al. 2014b)

For image-based evaluation, we follow Asthana et al. (2003) to provide a bounding box as the face detection output. For video-based evaluation, we follow Peng et al. (2015b) to utilize a tracking-by-detection protocol, where the face bounding box of the current frame is calculated according to the landmark of the previous frame.

Training strategy Our approach is capable of end-to-end training. However, there are only 105 different subjects presented in 300-VW, which hardly provide sufficient supervision for the identity constraint. To make full use of all datasets, we conducted the training through three steps. **First**, we pre-train the network without f_{trn} and f_{cls} using image-based datasets, i.e., AFLW (Koestinger et al. 2011), Helen (Le et al. 2012) and LFPW (Belhumeur et al. 2011). **Then**, f_{cls} is engaged for identity constraint and fine-tuned together with other modules using image-based LFW (Learned-Miller 2014). **Finally**, f_{trn} is triggered and the entire network is fine-tuned using video-based dataset, i.e. 300-VW (Shen et al. 2015).

Experimental Settings In every frame, the initial response map $\mathbf{z}_0$ (Eq. (2)) is generated using the landmark prediction of the previous frame. Parameter λ and γ (Eq. (14)) are empirically set so the ratio of $\ell_{det} : \ell_{reg} : \ell_{cls}$ is roughly equal to $1 : 10 : 1$.

For training, we optimize the network parameters by using *stochastic gradient descent* (SGD) with 0.9 momentum. We use fixed learning rate started at 0.01 and manually decreased it to an order of magnitude according to the validation accuracy. f_{enc} is initialized using pre-trained weights of VGG-16 (Simonyan and Zisserman 2014) or ResNet-152 (He et al. 2016). Other modules are initialized with Gaussian initialization (Jia et al. 2014). The training clips in a mini-batch have no identity overlap to avoid oscillations of the identity loss. We perform temporal recurrent learning in both forward and backward direction to double the usage of the sequential corpus.

For testing, we split 300-VW so that the training and testing videos do not have identity overlap (16 videos share 7 identities) to avoid overfitting. We use the inter-ocular distance to normalize the *root mean square error* (RMSE) (Sagonas et al. 2013) for accuracy evaluation. A prediction with larger than 10% mean error is reported as a failure (Shen et al. 2015).

5.2 Validation of Encoder–Decoder Variants

In Sect. 4.1, we proposed two different designs of encoder–decoder: **(1)** VGGNet-based design with symmetrical encoder and decoder, which has been mainly investigated in our former conference paper (Peng et al. 2016a); and **(2)** ResNet-based design with asymmetrical encoder, i.e., the encoder is much deeper than the decoder. In particular, skip connections are incorporated in bridging the encoder and decoder with hierarchical spatial information at different resolutions.

We compared the performance of two encoder–decoder variants on AFLW (Koestinger et al. 2011) and 300-VW (Shen et al. 2015). The results are reported in Table 5. The results show that the ResNet-based design outperforms the VGGNet-based variant with a substantial margin in terms of fitting accuracy (mean error) and robustness (standard deviation). Much deeper layers, as well as the proposed skipping shortcuts, contribute a lot to the improvement. In addition, the ResNet-based encoder–decoder has very close computational cost to the VGGNet-based variant, e.g. the average fitting time per image/frame and the memory usage of a trained model, which should be attributed to the custom residual module design and the proposed asymmetrical encoder–decoder network.

5.3 Validation of Spatial Recurrent Learning

We validated the proposed spatial recurrent learning on the validation set of AFLW (Koestinger et al. 2011). To better investigate the benefits of spatial recurrent learning, we partitioned the validation set into two image groups according to the absolute value of the yaw angle: (1) Common set where yaw ∈ [0°–30°); and (2) Challenging set where yaw ∈ (30°,

Table 5 Performance comparison of VGGNet-based and ResNet-based encoder–decoder Variants. Network configurations are described in Sect. 4.1. Row 1–2: image-based results on AFLW (Koestinger et al. 2011); Row 3–4: video-based results on 300-VW (Shen et al. 2015)

	Mean (%)	Std (%)	Time (ms)	Memory (Mb)
VGGNet-based	6.85	4.52	43.6	184
ResNet-based	6.33	3.61	54.9	257
VGGNet-based	5.16	2.57	42.5	184
ResNet-based	4.75	2.10	56.2	257

Table 6 Comparison of different combinations of the detection and regression tasks on the common and challenging set of AFLW (Koestinger et al. 2011). The proposed method (Last Row) has the best performance especially in challenging settings

	Common (%)		Challenging (%)	
	Error	Failure	Error	Failure
Single-step Detection	6.05	4.62	8.14	12.4
Single-step Regression	5.92	4.75	7.87	14.5
Recurrent Det.+Det.	5.86	3.44	7.33	8.20
Recurrent Det.+Reg.	5.71	3.30	6.97	8.75

Table 7 Comparison of cascade and recurrent learning on the entire set of AFLW (Koestinger et al. 2011). The latter improves accuracy with a half memory usage of the former

	Mean (%)	Std (%)	Memory (Mb)
Cascade Det. & Reg.	6.81	4.53	468
Recurrent Det. & Reg.	6.33	3.61	257

90°]. The training sets are ensembles of AFLW (Koestinger et al. 2011), Helen (Le et al. 2012) and LFPW (Belhumeur et al. 2011) as described in Table 4.

Validation of detection-followed-by-regression To validate the proposed recurrent detection-followed-by-regression, we investigated four different network configurations:

- Single-step prediction using loss defined in Eq. (5);
- Single-step prediction using loss defined in Eq. (7);
- Two-step recurrent detection-followed-by-detection;
- Two-step recurrent detection-followed-by-regression.

The mean fitting errors and failure rates are reported in Table 6. First, the results show that the two-step recurrent learning can instantly decrease the fitting error and failure rate, compared with either the single-step detection or regression. The improvement is more significant in challenging settings with large pose variations. Second, though landmark detection is more robust in challenging settings (low failure rate), it lacks the ability to predict precise locations (small fitting error) compared to landmark regression. This fact proves the effectiveness of the proposed recurrent detection-followed-by-regression.

Validation of recurrent learning We also conducted comparisons between the proposed spatial recurrent learning and the cascade models that are widely used in former approaches (Sun et al. 2013; Zhang et al. 2014a). For a fair comparison, we implemented a two-step cascade variant to perform detection-followed-by-regression. Each network in the cascade has exactly the same architecture as the recurrent version. But there is no weight sharing among cascades. We fully trained the cascade networks using the same training set and validated the performance in challenging settings.

The comparison is shown in Table 7. Unsurprisingly, the spatial recurrent learning can improve the fitting accuracy. The underlying reason is that the recurrent network learns the step-by-step fitting strategy jointly, while the cascade networks learn each step independently. It can better handle the challenging settings where the initial guess is usually far away from the ground truth. Moreover, the recurrent network with shared weights can instantly reduce the memory usage to one-half of the cascaded model.

Table 8 Validation of temporal recurrent learning on 300-VW (Sagonas et al. 2013). f_{trn} helps to improve the tracking robustness (smaller std and lower failure rate), as well as the tracking accuracy (smaller mean error). The improvement is more significant in challenging settings of large pose and partial occlusion as demonstrated in Fig. 7

	Common			Challenging			Full		
	Mean (%)	Std (%)	Fail (%)	Mean (%)	Std (%)	Failure (%)	Mean (%)	Std (%)	Fail (%)
w/o f_{trn}	4.52	2.24	3.48	6.27	5.33	13.3	5.83	3.42	6.43
f_{trn}	4.21	1.85	1.71	5.64	3.28	5.40	5.25	2.15	2.82

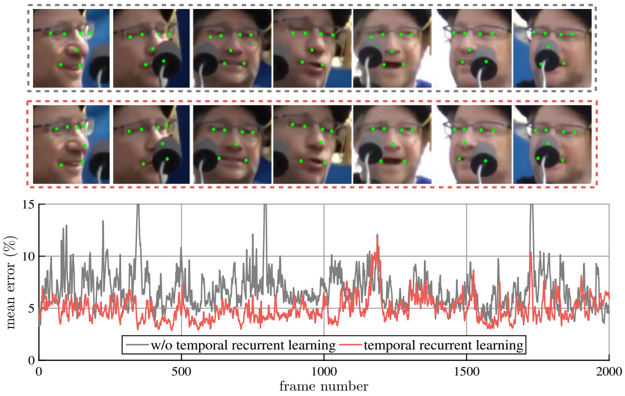

Fig. 7 Examples of temporal recurrent learning on 300-VW (Sagonas et al. 2013). The tracked subject undergoes intensive pose and expression variations as well as severe partial occlusions. f_{trn} substantially improves the tracking robustness (less variance) and fitting accuracy (low error), especially for landmarks on the nose tip and mouth corners

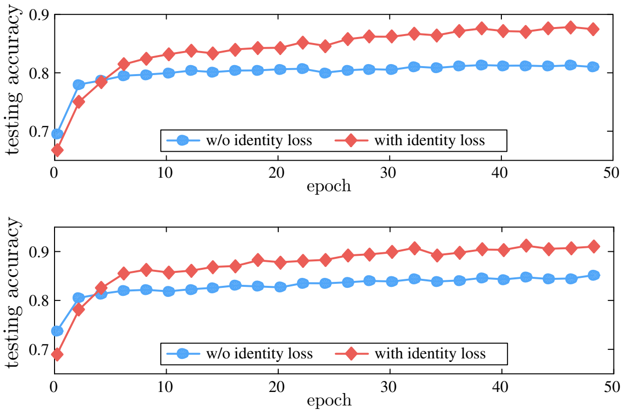

Fig. 8 Fitting accuracy of different facial components with respect to the number of training epochs on 300-VW (Shen et al. 2015). The proposed supervised identity disentangling helps to achieve a more complete factor decoupling in the bottleneck of the encoder–decoder, which yields better generalization capability and more accurate fitting results

5.4 Validation of Temporal Recurrent Learning

We validate the proposed temporal recurrent learning on the validation set of 300-VW (Shen et al. 2015). To better study the performance under different settings, we split the validation set into two groups: (1) 9 videos in common settings that roughly match "Scenario 1"; and (2) 15 videos in challenging settings that roughly match "Scenario 2" and "Scenario 3". The common, challenging and full sets were used for evaluation.

We implemented a variant of our approach that turns off the temporal recurrent learning f_{trn}. It was also pre-trained on the image training set and fine-tuned on the video training set. Since there was no temporal recurrent learning, we used frames instead of clips to conduct the fine-tuning which was

Table 9 Mean error comparison with state-of-the-arts on video-based validation sets: TF, FM, and 300-VW (Sagonas et al. 2013). The top performance in each dataset is highlighted. Our approach achieves the best fitting accuracy on both controlled and unconstrained datasets

	7 landmarks				68 landmarks		
	TF (FGNet 2004)	FM (Peng et al. 2015b)	300-VW (Shen et al. 2015)		TF (FGNet 2004)	FM (Peng et al. 2015b)	300VW (Shen et al. 2015
DRMF (Asthana et al. 2003)	4.43	8.53	9.16	ESR (Cao et al. 2014)	3.49	6.74	7.09
ESR (Cao et al. 2014)	3.81	7.58	7.83	SDM (Xiong and De la Torre 2013)	3.80	7.38	7.25
SDM (Xiong and De la Torre 2013)	4.01	7.49	7.65	CFAN (Zhang et al. 2014a)	3.31	6.47	6.64
IFA (Asthana et al. 2014)	3.45	6.39	6.78	TCDCN (Zhang et al. 2014b)	3.45	6.92	7.59
DCNC (Sun et al. 2013)	3.67	6.16	6.43	CFSS (Zhu et al. 2015)	3.04	5.67	6.13
RED-Net (Ours)	2.89	5.14	5.29	RED-Net (Ours)	2.77	4.93	5.15

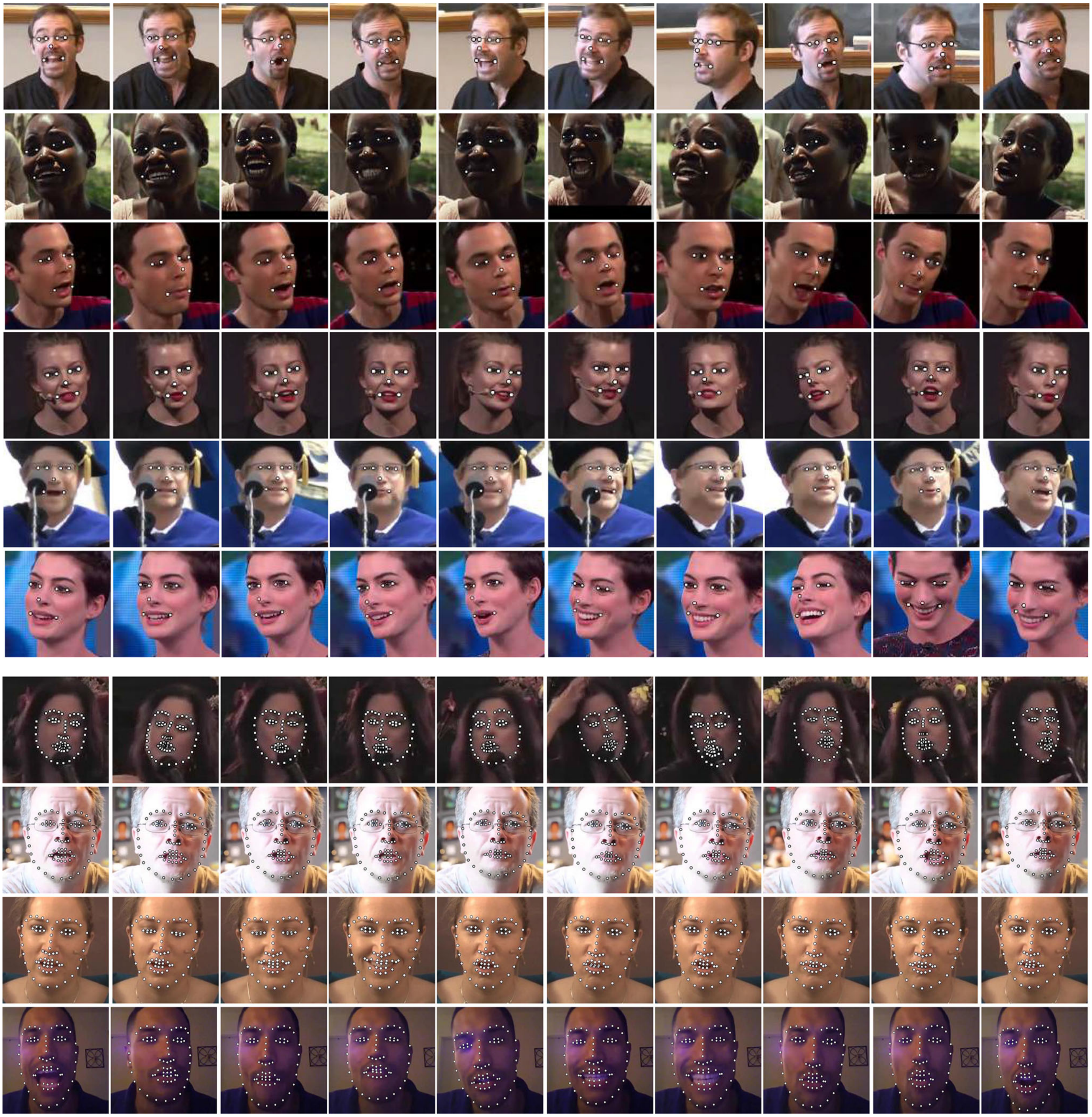

Fig. 9 Examples of 7-landmark (**Row 1–6**) and 68-landmark (**Row 7–10**) fitting results on FM (Peng et al. 2015b) and 300-VW (Shen et al. 2015). The proposed approach achieves robust and accurate fittings when the tracked subjects suffer from large pose/expression changes (**Row 1, 3, 4, 6, 10**), illumination variations (**Row 2, 8**) and partial occlusions (**Row 5, 7**)

performed for the same 50 epochs. We showed the result with and without temporal recurrent learning in Table 8.

For videos in common settings, the temporal recurrent learning achieves 6.8 and 17.4% improvement in terms of mean error and standard deviation respectively, while the failure rate is remarkably reduced by 50.8%. Temporal modeling produces better prediction by taking consideration of history observations. It may implicitly learn to model the motion dynamics in the hidden units from the training clips.

For videos in challenging settings, the temporal recurrent learning won with even bigger margin. Without f_{trn}, it is hard to capture the drastic motion or changes in consecutive frames, which inevitably results in higher mean error, standard deviation, and failure rate. Figure 7 shows an example where the subject exhibits intensive pose and expression variations as well as severe partial occlusions. The curve showed our recurrent model obviously reduced landmark errors, especially for landmarks on nose tip and mouth corners. The less oscillating error also suggests that f_{trn} significantly improves the prediction stability over frames.

5.5 Benefits of Supervised Identity Disentangling

The supervised identity disentangling is proposed to better decouple the temporal-invariant and temporal-variant factors in the bottleneck of the encoder–decoder. This facilitates the temporal recurrent training, yielding better generalization and more accurate fittings at test time.

To study the effectiveness of the identity constraint, we removed f_{cls} and follow the exact training steps. The testing accuracy comparison on the 300-VW (Sagonas et al. 2013) is shown in Fig. 8. The accuracy was calculated as the ratio of pixels that were correctly classified in the corresponding channel(s) of the response map.

The validation results of different facial components show similar trends: (1) The network demonstrates better generalization capability by using additional identity cues, which results in a more efficient training. For instance, after only 10 training epochs, the validation accuracy for landmarks located at the left eye reaches 0.84 with identity loss compared to 0.8 without identity loss. (2) The supervised identity information can substantially boost the testing accuracy. There is an approximately 9% improvement by using the additional identity loss. It worth mentioning that, at the very beginning of the training (< 5 epochs), the network has inferior testing accuracy with supervised identity disentangling. It is because the suddenly added identity loss perturbs the backpropagation process. However, the testing accuracy with identity loss increases rapidly and outperforms the one without identity loss after only a few more training epochs.

5.6 General Comparison with the State of the art

We compared our framework with both traditional approaches and deep learning based approaches. The methods with hand-crafted features include: (1) DRMF (Asthana et al. 2003), (2) ESR (Cao et al. 2014), (3) SDM (Xiong and De la Torre 2013), (4) IFA (Asthana et al. 2014), and (5) PIEFA (Peng et al. 2015b). The deep learning based methods include: (1) DCNC (Sun et al. 2013), (2) CFAN (Zhang et al. 2014a), and (3) TCDCN (Zhang et al. 2014b). All these methods were recently proposed and reported state-of-the-art performance. For fair comparison, we evaluated these methods in a tracking protocol: fitting result of current frame was used as the initial shape (DRMF, SDM and IFA) or the bounding box (ESR and PIEFA) in the next frame. The comparison was performed on both controlled, e.g. Talking Face (TF) (FGNet 2004), and in-the-wild datasets, e.g. Face Movie (FM) (Peng et al. 2015b) and 300-VW (Shen et al. 2015).

We report the evaluation results for both 7 and 68 landmark setups in Table 9. Our approach achieves state-of-the-art performance under both settings. It outperforms others with a substantial margin on all datasets under both 7-landmark and 68-landmark protocols. The performance gain is more significant on the challenging datasets (FM and 300-VW) than controlled dataset (TF). Our alignment model runs fairly fast, it takes around 40 ms to process an image using a Tesla K40 GPU accelerator. Please refer to Fig. 9 for fitting results of our approach on FM (Peng et al. 2015b) and 300-VW (Shen et al. 2015), which demonstrate the robust and accurate performance in wild conditions.

6 Conclusion

In this paper, we proposed a novel recurrent encoder–decoder network for real-time sequential face alignment. It utilizes spatial recurrency to train an end-to-end optimized coarse to fine landmark detection model. It decouples temporal-invariant and temporal-variant factors in the bottleneck of the network and exploits recurrent learning at both spatial and temporal dimensions. Extensive experiments demonstrated the effectiveness of our framework and its superior performance. The proposed method provides a general framework that can be further applied to other localization-sensitive tasks, such as human pose estimation, object detection, scene classification, and others.

References

Asthana, A., Zafeiriou, S., Cheng, S., & Pantic, M. (2003). Robust discriminative response map fitting with constrained local models. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3444–3451).

Asthana, A., Zafeiriou, S., Cheng, S., & Pantic, M. (2014). Incremental face alignment in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR.

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., & Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Black, M., & Yacoob, Y. (1995). Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Bulat, A., & Tzimiropoulos, G. (2016). *Human pose estimation via convolutional part heatmap regression* (pp. 717–732). Cham: Springer.

Cao, X., Wei, Y., Wen, F., & Sun, J. (2014). Face alignment by explicit shape regression. *International Journal of Computer Vision*, *107*(2), 177–190.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. CoRR arXiv:1409.1259.

Chrysos, G. G., Antonakos, E., Zafeiriou, S., & Snape, P. (2015). Offline deformable face tracking in arbitrary videos. In: *Proceedings of the IEEE international conference on computer vision workshop* (pp. 954–962).

Cootes, T. F., & Taylor, C. J. (1992). Active shape models-smart snakes. In *BMVC*.

Decarlo, D., & Metaxas, D. (2000). Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, *38*(2), 99–127.

FGNet: Talking face video. Technical report, Online (2004)

Gao, X., Su, Y., Li, X., & Tao, D. (2010). A review of active appearance models. *IEEE Transactions on Systems, Man, and Cybernetics*, *40*(2), 145–158.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computing*, *9*(8), 1735–1780.

Hong, S., Noh, H., & Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. CoRR arXiv:1506.04924.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR arXiv:1502.03167.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. In *ACM multimedia conference* (pp. 675–678).

Jourabloo, A., & Liu, X. (2016). Large-pose face alignment via CNN-based dense 3D model fitting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Kendall, A., Badrinarayanan, V., & Cipolla, R. (2015). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. CoRR arXiv:1511.02680.

Koestinger, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2011) Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Workshop on benchmarking facial image analysis technologies*.

Lai, H., Xiao, S., Cui, Z., Pan, Y., Xu, C., & Yan, S. (2015). Deep cascaded regression for face alignment. arXiv:1510.09083v2.

Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T. S. (2012). Interactive facial feature localization. In *European conference on computer vision* (pp. 679–692).

Learned-Miller, G. B. H. E. (2014). Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst.

Long, J., Shelhamer, E., & Darrell, T. (2014a). Fully convolutional networks for semantic segmentation. CoRR arXiv:1411.4038.

Long, J. L., Zhang, N., & Darrell, T. (2014b). Do convnets learn correspondence? In *Advances in neural information processing systems* (pp. 1601–1609).

Lu, L., Zhang, X., Cho, K., & Renals, S. (2015). A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition. In *INTERSPEECH*.

Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., & Ranzato, M. (2014). Learning longer memory in recurrent neural networks. CoRR arXiv:1412.7753.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH*.

Milborrow, S., & Nicolls, F. (2008). Locating facial features with an extended active shape model. In *European conference on computer vision* (pp. 504–513).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *The international conference on machine learning* (pp. 807–814).

Oh, J., Guo, X., Lee, H., Lewis, R. L., & Singh, S. (2015). Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems* (pp. 2845–2853).

Oliver, N., Pentland, A., & Berard, F. (1997). Lafter: Lips and face real time tracker. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 123–129).

Patras, I., & Pantic, M. (2004). Particle filtering with factorized likelihoodsfor tracking facial features. In *Automatic face and gesture recognition* (pp. 97–102).

Peng, X., Feris, R. S., Wang, X., & Metaxas, D. N. (2016a). A recurrent encoder-decoder network for sequential face alignment. In *European conference on computer vision* (pp. 38–56). Springer.

Peng, X., Hu, Q., Huang, J., & Metaxas, D. N. (2016b). Track facial points in unconstrained videos. In *Proceedings of the British machine vision conference* (pp. 129.1–129.13).

Peng, X., Huang, J., Hu, Q., Zhang, S., Elgammal, A., & Metaxas, D. (2015a). From circle to 3-sphere: Head pose estimation by instance parameterization. *Computer Vision and Image Understanding*, *136*, 92–102.

Peng, X., Yu, X., Sohn, K., Metaxas, D. N., & Chandraker, M. (2017a). Reconstruction-based disentanglement for pose-invariant face recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 1623–1632).

Peng, X., Zhang, S., Yang, Y., & Metaxas, D. N. (2015b). Piefa: Personalized incremental and ensemble face alignment. In: *Proceedings of the IEEE international conference on computer vision*.

Peng, X., Zhang, S., Yu, Y., & Metaxas, D. N. (2017b). Toward personalized modeling: Incremental and ensemble alignment for sequential faces in the wild. *International Journal of Computer Vision*, *126*, 1–14.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2016). 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, *47*, 3–18.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE international conference on computer vision workshop*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815–823).

Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., & Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE international conference on computer vision workshop*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR arXiv:1409.1556.

Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3476–3483).

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2892–2900).

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tzimiropoulos, G. (2015). Project-out cascaded regression with an application to face alignment. In *CVPR* (pp. 3659–3667).

Veeriah, V., Zhuang, N., & Qi, G. J. (2015) Differential recurrent neural networks for action recognition. In: *Proceedings of the IEEE international conference on computer vision*.

Wang, J., Cheng, Y., & Feris, R. S. (2016). Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, X., Yang, M., Zhu, S., & Lin, Y. (2015). Regionlets for generic object detection. *TPAMI*, *37*(10), 2071–2084.

Wu, Y., & Ji, Q. (2016). Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Xiong, X., & De la Torre, F. (2013). Supervised descent method and its application to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yang, J., Reed, S., Yang, M. H., & Lee, H. (2015). Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *NIPS*.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., & Courville, A. (2015) Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833).

Zhang, J., Shan, S., Kan, M., & Chen, X. (2014a). Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *European conference on computer vision* (pp. 1–16).

Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2014b). Facial landmark detection by deep multi-task learning. In *European conference on computer vision* (pp. 94–108).

Zhu, S., Li, C., Loy, C. C., & Tang, X. (2015). Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4998–5006).

Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016). Face alignment across large poses: A 3D solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.