IJCV special issue (Best papers of ECCV 2016) manuscript No.
(will be inserted by the editor)

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

Xi Peng · Rogerio S. Feris · Xiaoyu Wang · Dimitris N. Metaxas

Submitted: April 19 2017 / Revised: N/A

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

Xi Peng
Rutgers University, Piscataway, NJ, 08854
Tel.: +1(917)803-7402
E-mail: xpeng.cs@rutgers.edu

Rogerio S. Feris
IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
E-mail: rsferis@us.ibm.com

Xiaoyu Wang
Snap Research, Los Angeles, CA
E-mail: xiaoyu.wang@snapchat.com

Dimitris N. Metaxas
Rutgers University, Piscataway, NJ, 08854
E-mail: dnm@cs.rutgers.edu

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) [10, 31] and active appearance models (AAMs) [13]. 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* [53, 7, 57, 23, 58]. 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 [53, 7], or features extracted from convolutional neural networks [57, 23, 58]. Although these methods can achieve very reliable results in standard benchmark datasets, they still suffer from limited performance 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) [43], leveraging temporal information and identity consistency as additional constraints [50]. Despite the long history of research in rigid and non-rigid face tracking [5, 35, 11, 36], current efforts have mostly focused

on face alignment in still images [41, 57, 48, 59]. 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 [51], with notable exceptions such as [2, 39], 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 Figure 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 [53, 57] 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 explictly divide the task into multiple stage prediction problems. This subtle difference makes the recurrent model more elegant in terms of holistic optimization. It can implictly 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 [14]. By adding feedback connections from the predicted heatmap, the network only needs to implictly 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, 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 implictly abstract 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 aligment works related to our approach, as well as advances in deep learning, like the development of recurrent and encoder-decoder neural networks.

Regression-based face landmark detection. Recently, regression-based face landmark detection methods [1, 45, 53, 7, 57, 2, 59, 48, 19, 52, 60] have achieved significant boost in the generalization performance of face landmark detection, compared to algorithms based on statistical models such as Active shape models [10, 31] and Active appearance models [13]. 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 [7]. 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

[1] https://sites.google.com/site/xipengcshomepage/project/face-alignment

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 [53, 45, 57]. The supervised descent method [53] learns cascades of regression models based on SIFT features. Sun *et. al.* [45] proposed to use three levels of neural networks to predict landmark locations. Zhang *et. al.* [57] 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 [51]. Asthana *et. al.* [2] and Peng *et. al.* [39] 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 [30] and natural language processing [29]. They have also been recently used in computer vision. For instance, in the tasks of image captioning [20] and video captioning [55], RNNs are usually employed for text generation. RNNs are also popular as a tool for action classification. As an example, Veeriah *et. al.* [49] 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 [8] where the encoder learns the intermediate representation and the decoder generates the translation from the representation. It is also investigated in speech recognition [28] and computer vision [3, 16]. Yang *et. al.* [54] 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 [27] is still unexplored. In this work, we employ the encoder to learn a joint representation for 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 $\mathbf{z} \in \mathbb{R}^{w \times h \times C}$ which indicates landmark locations.

The *encoder* performs a sequence of convolution, pooling and batch normalization [17] to extract a representation code from the inputs:

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

where $f_{enc}\left(\cdot; \theta_{enc}\right)$ denotes the encoder mapping with parameters θ_{enc}. $\mathbf{z}$ is input together with $\mathbf{x}$ for recurrent learning which will be explained soon in next section.

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

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

where $f_{dec}\left(\cdot; \theta_{dec}\right)$ denotes the decoder mapping with parameters θ_{dec}. $\mathbf{z}$ has the same $W \times H$ dimension as $\mathbf{x}$ but different number of channels, which presents the pixel-wise confidence of the corresponding landmark.

The encoder-decoder framework plays an important role in our task. **First**, it is convenient to perform *spatial recurrent learning* (f_{srn}) since $\mathbf{z}$ has the same dimension (but different number of channels) as $\mathbf{x}$. The output of the decoder can be directly fed back into the encoder to provide pixel-wise spatial cues for the next recurrent step.

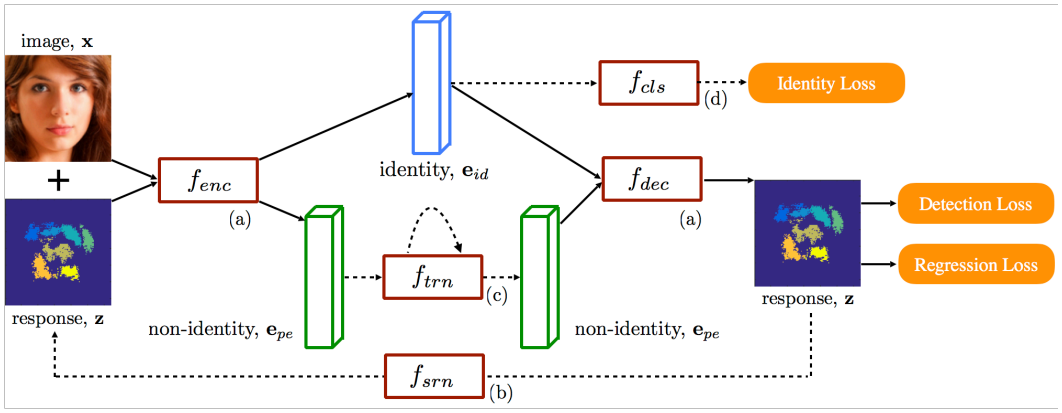

Fig. 1: Overview of the recurrent encoder-decoder network: **(a)** encoder-decoder (Section 3.1); **(b)** spatial recurrent learning (Section 3.2); **(c)** temporal recurrent learning (Section 3.3); and **(d)** supervised identity disentangling (Section 3.4). $f_{enc}, f_{dec}, f_{srn}, f_{trn}, f_{cls}$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

Second, we can decouple $\mathbf{e}$ in the bottleneck of the network into temporal-invariant and -variant factors:

$$\mathbf{e}_{id} \in \mathbb{R}^{W_e \times H_e \times C_i}, \mathbf{e}_{pe} \in \mathbb{R}^{W_e \times H_e \times C_p}, C_e = C_i + C_p, \quad (3)$$

where $\mathbf{e}_{id}$ and $\mathbf{e}_{pe}$ denote the identity and pose/expression representations, respectively. The former is further exploited in *temporal recurrent learning* (f_{trn}) for robust alignment, while the latter is used in *supervised identity disentangling* (f_{cls}) to facilitate the network training.

Third, f_{enc} and f_{dec} are designed to be full convolutional [26], *i.e.*, $\{\mathbf{x}, \mathbf{e}, \mathbf{z}\}$ are feature maps instead of fully-connected neurons that are often used in former encoder-decoder designs. This structure is highly memory and speed efficient, which is desirable to 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 [45, 57] 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 $\mathbf{z}_{k-1}$, stacked with $\mathbf{x}$ as shown in Figure 2, to eventually push the shape prediction from an initial guess to the ground truth:

$$\mathbf{z}_k = f_{srn}\big(\mathbf{x}, \mathbf{z}_{k-1}; \theta_{srn}\big), \ k = 1, \cdots, K, \quad (4)$$

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 could be a mean shape or the output of the previous frame.

In the conference version [37], 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 [6], we propose to further explore the spatial recurrent learning by performing detection-followed-by-regression in successive steps. Specially, the detection task locates major facial components (*e.g.* $C_d = 7$), while the regression task refines all landmarks (*e.g.* $C_r = 68$) positions.

The first recurrent step performs *landmark detection*, which 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 landmark, in which the values are set to 1 to mark the presence of the corresponding landmark and 0 for the remaining background:

$$\mathbf{z}_{det} = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_0; \theta_{enc}); \theta_{dec}\big), \ \mathbf{z}_{det} \in \mathbb{R}^{W \times H \times C_d}. \quad (5)$$

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

$$\ell_{det} = \frac{1}{C_d} \sum_{c=1}^{C_d} \sum_{i=1}^{W} \sum_{j=1}^{H} z_{ij}^c log y_{ij}^c + (1 - z_{ij}^c) log(1 - y_{ij}^c) \quad (6)$$

where z_{ij}^c denotes the sigmoid output of the c-th landmark at pixel location (i, j) and y_{ij}^c is the ground-truth label at the same location. Note that this loss function is different from the N-way cross-entropy loss used in our previous conference paper [37], in the sense that it allows multiple class labels for a single pixel, which helps to tackle the landmark overlaps.

The second recurrent step performs *landmark regression*, which improves the fitting accuracy of the previous detection step. The encoder-decoder aims to output a heatmap of C_r channels, one for each landmark, in which the values obey a Gaussian distribution centered at the ground-truth location with a pre-defined standard deviation:

$$\mathbf{z}_{reg} = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_{det}; \theta_{enc}); \theta_{dec}\big), \ \mathbf{z}_{reg} \in \mathbb{R}^{W \times H \times C_r}. \quad (7)$$

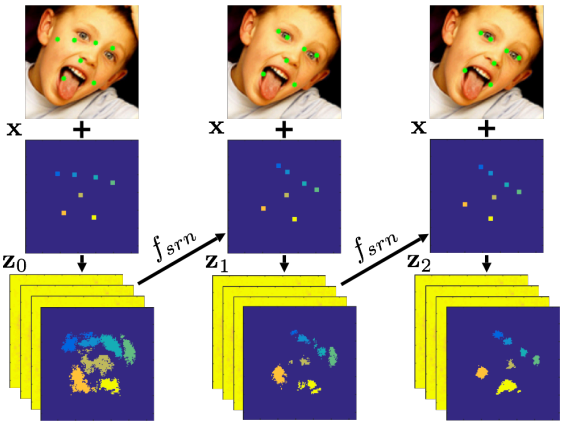

Fig. 2: An unrolled illustration of *spatial recurrent learning*. The response map is pretty coarse when the initial guess is far away from the ground truth if large pose and expression exist. It eventually gets refined in the successive recurrent steps.

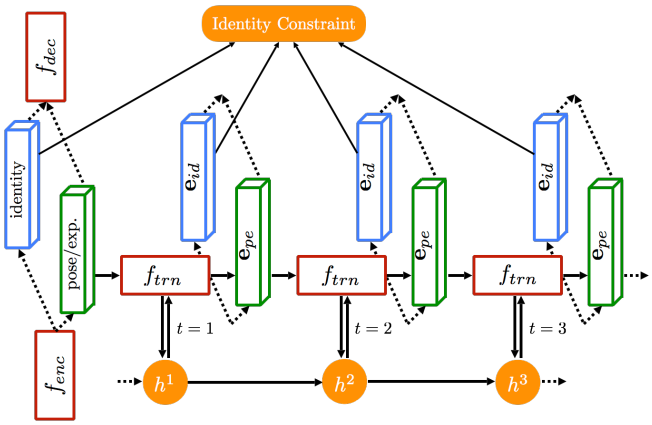

Fig. 3: An unrolled illustration of *temporal recurrent learning*. C_{id} encodes temporal-invariant factor which subjects to the same identity constraint. C_{pe} encodes temporal-variant factors which is further modeled in f_{tRNN}.

The regression task can be trained using pixel-wise L_2 loss:

$$\ell_{reg} = \frac{1}{C_r}\sum_{c=1}^{C_r}\sum_{i=1}^{W}\sum_{j=1}^{H}\|z_{ij}^c - y_{ij}^c\|_2^2, \tag{8}$$

where z_{ij}^c denotes the heatmap value of the c-th landmark at pixel location (i, j) and y_{ij}^c is the ground-truth value at the same location.

Thus the recurrent learning defined in (4) can be achieved by jointly minimizing a 2-step recurrent prediction:

$$\underset{\theta_{enc},\theta_{dec}}{\text{argmin}}\ \ell_{det} + \lambda\ell_{reg}, \tag{9}$$

where λ balances the loss between detection and regression tasks. Note that the recurrent steps share weights $\{\theta_{enc}, \theta_{dec}\}$.

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 [6]. More importantly, the network can jointly learn the coarse-to-fine fitting strategy in recurrent steps, instead of training cascaded networks independently [45, 57], which guarantees robustness and accuracy in challenging conditions.

3.3 Temporal Recurrent Learning

The recurrent learning is performed at both the spatial and temporal dimensions. Given T successive frames $\{\mathbf{x}^t; t = 1, \cdots, T\}$, the encoder extracts a sequence of representations $\{\mathbf{e}^t; t = 1, \cdots, T\}$. As we mentioned in Section 3.1, $\mathbf{e}$ can be decoupled as: identity representation $\mathbf{e}_{id}$ that is *temporal-invariant* since all frames are subject to the same identity constraint; and pose/expression representation $\mathbf{e}_{pe}$ that is *temporal-variant* since pose and expression changes over time [38]. We exploit the temporal consistence of $\mathbf{e}_{pe}$ via the temporal recurrent learning.

Figure 3 shows the unrolled illustration of the proposed temporal recurrent learning. More specifically, we aim to achieve a nonlinear mapping f_{trn}, which simultaneously tracks the latent state h^t and updates $\mathbf{e}_{pe}^t$ at time t:

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

where $p(\cdot)$ and $q(\cdot)$ are functions of temporal recurrent mapping $f_{trn}(\cdot; \theta_{trn})$. $\mathbf{e}_{pe}^{t*}$ is the update of $\mathbf{e}_{pe}^t$. θ_{trn} corresponds to mapping parameters which are learned using the same detection and regression supervision (9) but unrolled at the temporal dimension:

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

where t counts time steps. Note that in addition to the encoder-decoder parameters $\{\theta_{enc}, \theta_{dec}\}$, (11) also seeks the optimization of θ_{trn} in the same end-to-end framework.

The temporal recurrent learning memorizes the motion patterns of pose and expression variations from offline training data. It can significantly improve the fitting accuracy and robustness when large variations and partial occlusions exist.

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. More supervised information is required to achieve the disentangling. To address this issue, we propose to apply a face recognition task on the identity representation $\mathbf{e}_{id}$, in addition to the temporal recurrent learning applied on pose/expression representation $\mathbf{e}_{pe}$.

The supervised identity disentangling is formulated as an N-way classification problem. N is the number of unique

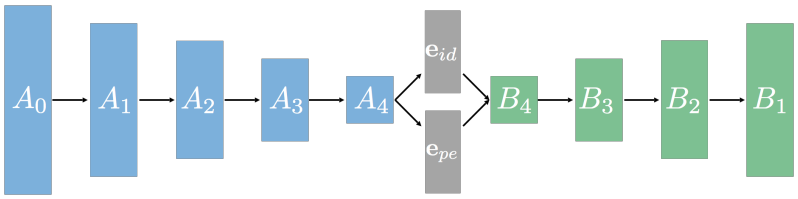

Fig. 4: Network architecture of VGGNet-based encoder f_{enc} and decoder f_{dec}. The encoder (A_{0-4}) and the decoder (B_{4-1}) are **nearly symmetrical** except that the encoder has one more stage than the decoder. Therefore the dimension of the output response map $\mathbf{z}$ is half of the dimension of the input image $\mathbf{x}$.

Table 1: Network specification of VGGNet-based encoder f_{enc} and decoder f_{dec}: module names (1^{st} row), output sizes (2^{nd} row), and network configurations (3^{rd} row). Pooling or unpooling operations are performed after each encoder module or before each decoder module. The pooling window is set to 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
$2\times$ conv	$2\times$ conv	$3\times$ conv	$3\times$ conv	$3\times$ conv	unpooling	unpooling	unpooling	unpooling
$[3 \times 3, 64]$	$[3 \times 3, 128]$	$[3 \times 3, 256]$	$[3 \times 3, 512]$	$[3 \times 3, 512]$	$3\times$ conv	$3\times$ conv	$3\times$ conv	$2\times$ conv
pooling	pooling	pooling	pooling	-	$[3 \times 3, 512]$	$[3 \times 3, 512]$	$[3 \times 3, 256]$	$[3 \times 3, 128]$

individuals present in the training sequences. In general, we associate the identity representation $\mathbf{e}_{id}$ with a one-hot encoding $\mathbf{z}_{id}$ to indicate the score of each identity:

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

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

Given (9), (11) and (13), the entire network can be trained end-to-end by optimizing:

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

where λ weights the supervision from the identity task. Note that the identity constraint is applied at every time step.

It has been shown in [58] that learning the face alignment task together with correlated tasks, *e.g.* head pose, can improve the fitting performance. We have the similar observation when adding face recognition task to the alignment task. More specifically, we found that supervised identity disentangling can significantly improve the generalization as well as fitting accuracy at test time. In this case, the factors are better decoupled, which facilitates the temporal recurrent learning to better handle variations over time.

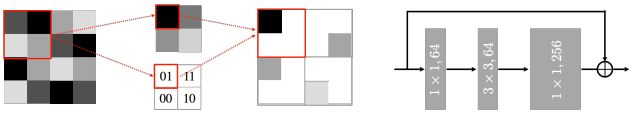

Fig. 5: Illustration of the pooling/unpooing with indices (left) and the residual unit (right) used in C_1. The corresponding pooling and unpooling share pooling indices in a 2-bit switch for each 2×2 pooling window. 1×1 convolutional layers are used in the residual unit to cut down the number of network parameters.

4 Network Architecture

All modules are embedded in a unified framework that can be trained end-to-end. We present details of network designs for efficient training and robust performance at test time.

4.1 Variant Designs of Encoder-Decoder

To best evaluate the proposed framework, we investigate two variant designs of the encoder-decoder. Specifically, the encoder is designed based on either VGGNet [44] or ResNet [14], while the decoder is designed to match the corresponding encoder.

VGGNet-based encoder-decoder. Figure 4 illustrates the network architecture and Table 1 presents the network specification. The encoder is designed based on a variant of the VGG-16 network [44, 21]. It has 13 convolutional layers with constant 3×3 filters which correspond to the first 13 convolutional layers in VGG-16. 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 [26] and output identity and pose/expression feature maps in the bottleneck. This strategy not only reduces the number of parameters from 117M to 14.8M [3], but also preserves spatial information in high-resolution feature maps instead of fully-connected feature vectors, which is crucial to our localization task.

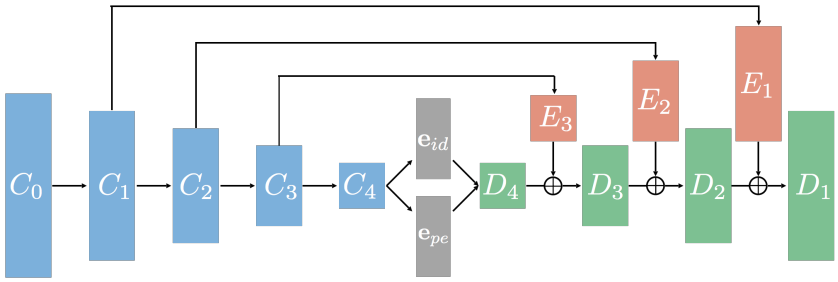

Fig. 6: Network architecture of ResNet-based encoder f_{enc} and decoder f_{dec}. The encoder (C_{0-4}) and the decoder (D_{4-1}) are **not symmetrical**. The encoder is much deeper than the decoder, *i.e.* 51 vs. 4 modules. Similar to the VGGNet-based design, the dimension of the output response map $\mathbf{z}$ is half of the dimension of the input image $\mathbf{x}$. Residual based skip connections (E_{1-3}) are designed to merge hierarchical features at different resolutions in capturing comprehensive spatial information.

Table 2: Network specification of ResNet-based encoder f_{enc} and decoder f_{dec}: module names (1^{st} row), output sizes (2^{nd} row), and network configurations (3^{rd} row). Strides of 2 are used at the end of each encoder or decoder module to halve or double the dimension of feature maps. The residual based skip module 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
$1\times$ conv	$3\times$ conv	$8\times$ conv	$36\times$ conv	$3\times$ conv	$1\times$ dconv	$1\times$ dconv	$1\times$ dconv	$1\times$ dconv
$\begin{bmatrix} 7 \times 7, 64 \\ \text{strid, } 2 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 512 \\ \text{stride, } 2 \\ 1 \times 1, 1024 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 256 \\ \text{stride, } 2 \\ 1 \times 1, 512 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 128 \\ \text{stride, } 2 \\ 1 \times 1, 256 \end{bmatrix}$	$\begin{bmatrix} 2 \times 2, 64 \\ \text{stride, } 2 \\ 1 \times 1, 128 \end{bmatrix}$

There are 5 max-pooling layers with 2×2 pooling windows and a constant stride of 2 in the encoder to halve the resolution of feature maps after each convolutional stage. Although max-pooling can help to achieve translation invariance, it inevitably results in a considerable loss of spatial information especially when several max-pooling layers are applied in succession. 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 [56]. As illustrated in Figure 5, 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 for the decoder to recover the input structure from the highly compressed feature map. Besides, it is much more efficient to store the spatial indices than to memorize the entire feature map in float precision as proposed in FCNs [26].

The decoder is **nearly symmetrical** to the encoder with a mirrored configuration but replacing all max-pooling layers with corresponding unpooling layers. The encoder is slightly deeper than the decoder with one more encoding module. Therefore the dimension of the output response map $\mathbf{z}$ is half of the dimension of the input image $\mathbf{x}$. We find that batch normalization [17] can significantly boost the training speed as it can effectively reduce internal shift within a mini batch. Therefore, batch normalization and rectified linear unit (ReLU) [32] are applied after each convolutional layer.

ResNet-based encoder-decoder. Figure 6 illustrates the network architecture and Table 2 presents the network specification. The encoder is designed based on a variant of the ResNet-152 [14], which is very deep with 50 residual units of totally 151 convolutional layers. Figure 5 shows a residual unit used in C_1. 1×1 convolutional layers are used to cut down the number of network parameters. According to [14], the residual shortcut guarantees efficient training of the very deep network, as well as improved performance compared to [44]. Stride-2 convolutions instead of max poolings are used at the end of each encoding module to halve the dimension of feature maps.

Different from the VGGNet-based design, the encoder and decoder are **not symmetrical**. The encoder is much deeper than the decoder, which has only 4 upsampling modules of totally 4 convolutional layers. The practical consideration is that the encoder has to tackle a complicated task, *e.g.* understand the image and translate it to a low-dimensional representation code, while the decoder's task is relatively simpler, *e.g.* recover a set of response maps to mark landmark locations from the representation code. We use stride-2 de-convolutions to double the dimension of feature maps in each decoding module. Similar to the VGGNet-based design, the dimension of the output response map $\mathbf{z}$ is half of the dimension of the input image $\mathbf{x}$.

Another novel design of the encoder-decoder is the residual based skip connection as shown in Figure 6. The skip modules are specified in Table 3. They bridge the feature maps of encoder and decoder at different resolutions, which can effectively merge hierarchical spatial information for accurate landmark localization [33]. The skip shortcuts can significantly speed up the training process, and enable us to use a shallow decoder instead of a symmetrical one used in the VGGNet-based design to reduce the network complexity.

Table 3: Network specification of the residual based skip module. Each residual unit has the same configuration as the one used in the corresponding layer of encoder. Therefore we can perform element-wise addition of feature maps at different resolutions.

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

4.2 Spatial and Temporal Recurrent Learning

We employ simple but effective designs for the proposed spatial and temporal recurrent learning to achieve a good tradeoff between the network complexity and performance.

Spatial recurrent learning. Landmark detection and regression are performed in recurrent steps. Particularly, the first step detects partial landmarks that are robust to pose variations and partial occlusions, *e.g.* 7 landmarks for left/right eye corners, nose tip, and mouth corners; then the second step regresses all 68 landmarks based on the input image and detection results. This coarse-to-fine strategy is important to obtain efficient and robust spatial recurrent learning.

As mentioned in Section 3.2, the landmark detection task outputs a set of $C_d = 7$ binary maps, 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 landmark regression task outputs a set of $C_d = 68$ heatmaps, in which the correct locations are represented by Gaussian with a standard deviation of 5 pixels. The two tasks share the weights of the entire encoder-decoder except for the last convolutional layer, which employs 512 1×1 filters in order to adapt to binary map or heatmap.

In either landmark detection or regression, the foreground pixels are much less the background, which lead to highly unbalanced loss contributions. To solve this issue, we enlarge the positive loss defined in (9) and (11) by multiplying a constant weight to enforce the network pays more attention to foreground pixels.

Temporal recurrent learning. The configuration is specified in Figure 7. We employ Long Short Term Memory (LSTM) network [15, 34] to model f_{trn}, in which 256 hidden units are used. We empirically set the number of successive frames as $T = 10$ in (11). The prediction loss is calculated at each time step and then accumulated after T steps for back-propagation. Directly feeding the pose/expression representation $\mathbf{e}_{pe}$ into LSTM layers would lead to a slow training speed as it needs a large number of neurons for both the input and output. Instead, we apply average pooling and upsampling with indices to compress $\mathbf{e}_{pe}$ to a $256d$ vector before and after LSTM layers.

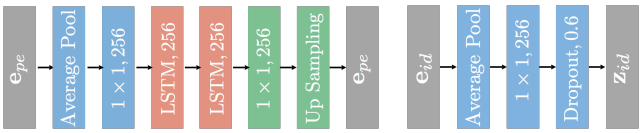

Fig. 7: Network configurations of temporal recurrent learning f_{trn} (left) and identity constrained disentangling f_{cls}. We apply average pooling and unpooling with spatial indices to cut down the input and output complexity of LSTM modules. A compact vector of $256d$ is used for identity representation to reduce the computational cost.

4.3 Identity Constraint

To facilitate the disentangling of identity and non-identity embeddings in the bottleneck of the encoder-decoder, we use a classification network to predict identity labels from $\mathbf{e}_{id}$. f_{cls} takes $\mathbf{e}_{id}$ as input and applies average pooling to obtain a $256d$ feature vector for identity representation. Instead of using a very long feature vector in former face recognition networks [47], *e.g.* $4096d$, we use a more compact vector, $256d$, to reduce the computational cost without losing recognition accuracy [42, 46]. To avoid overfitting, 0.6 dropout is applied, followed by a fully-connected layer with N neurons to predict the entity using the cross-entropy loss defined in (13), where N is the number of different subjects present in training sequences.

5 Experiments

In this section, we first demonstrate the effectiveness of each component in our framework, followed with performance comparison against the state of the art on both controlled and unconstrained 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) [22]
- Labeled Faces in the Wild (LFW) [25]
- Helen facial feature dataset (Helen) [24, 41]
- Labeled Face Parts in the Wild (LFPW) [4, 41]
- Talking Face (TF) [12]
- Face Movies (FM) [39]
- 300 face Videos in the Wild (300-VW) [43]

We list configurations and setups of each dataset in Table 4. For consistent comparisons, we followed [41, 40] for 68-landmark annotation for Helen, LFPW, TF, FM and 300-VW. Besides, 7-landmark annotations are generated based on the annotation for all datasets to locate eye corners, nose tip and

Table 4: The *image* and *video datasets* used in training and evaluation. LFW, TF, FM and 300-VW have both landmark and identity annotations. AFLW and 300-VW are split into two sets for both training and evaluation, while the rest datasets are used for training.

	AFLW [22]	LFW [25]	Helen [24]	LFPW [4]	TF [12]	FM [39]	300-VW [43]
in-the-wild setting	yes	yes	yes	yes	no	yes	yes
image number	21,080	12,007	2,330	1,035	500	2,150	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	2,330	1,035	0	0	90,000
used in evaluation	4,216	0	0	0	500	2150	24,000

mouth corners. The landmark annotation of LFW is given by [25]. We manually labeled the identity for each video in TF, FM, and 300-VW.

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 used them for both training and evaluation. More specifically, 80% of the images in AFLW and 90 out of 114 videos in 300-VW were used for training, and the rest were used for evaluation. We sampled videos to roughly cover the three different scenarios defined in [9], *i.e.* "Scenario 1", "Scenario 2" and "Scenario 3", corresponding to well-lit, mild unconstrained and completely unconstrained conditions.

We performed 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 to 1.1), rotation ($\pm 15°$), translation (7 pixels), as well as horizontal flip. To generate sequential training data, we randomly sampled 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) [1]
- Explicit Shape Regression (ESR) [7]
- Supervised Descent Method (SDM) [53]
- Incremental Face Alignment (IFA) [2]
- Coarse-to-Fine Shape Searching (CFSS) [59]
- Deep Convolutional Network Cascade (DCNC) [45]
- Coarse-to-fine Auto-encoder Network (CFAN) [57]
- Deep Multi-task Learning (TCDCN) [58]

For image-based evaluation, we followed [1] to provide a bounding box as the face detection output. For video-based evaluation, we followed [39] to utilize a tracking-by-detection protocol: a bounding box, which is calculated according to the landmark prediction in the previous frame but slightly enlarged, is used as the face detection result of the current frame.

Training strategy. Our approach is capable of end-to-end training on the video datasets. However, there are only 105 different subjects presented in 300-VW, which can hardly provide sufficient supervision for the identity constraint. To make full use of all annotated datasets, we conducted the training through three steps. **First**, we train the network without f_{trn} and f_{cls} using image-based datasets, *i.e.,* AFLW [22], Helen [24] and LFPW [4]. **Then**, f_{cls} is engaged for identity constraint and fine-tuned together with other modules using image-based LFW [25]. **Finally**, f_{trn} is triggered and the entire network is fine-tuned using video-based dataset, *i.e.* 300-VW [43].

In each step, we optimized the network parameters by using *stochastic gradient descent* (SGD) with 0.9 momentum. We used fixed learning rate started at 0.01 and manually decreased it to an order of magnitude according to the validation accuracy. f_{enc} was initialized using pre-trained weights of VGG-16 [44] or ResNet-152 [14]. Other modules were initialized with Gaussian initialization [18]. The mini-batch size was set to 5 clips that had no identity overlap to avoid oscillations of the identity loss. We performed temporal recurrent learning in both forward and backward direction to double the usage of sequential training corpus.

Evaluation protocol. To avoid overfitting, we ensure that the training and testing videos do not have identity overlap on the 300-VW (16 videos share 7 identities). We used normalized *root mean square error* (RMSE) [41] for fitting accuracy evaluation. A prediction with larger than 10% mean error was reported as a failure [43].

5.2 Comparison of Encoder-decoder Variants

In Section 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 [37]; 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 [22] and 300-VW [43]. The results are

Table 5: Performance comparison of VGGNet-based and ResNet-based encoder-decoder Variants. Network configurations are described in Section 4.1. Row 1-2: image-based results on AFLW [22]; Row 3-4: video-based results on 300-VW [43].

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

Table 6: Validation of spatial recurrent learning on AFLW [22] in Common and Challenging settings. The detection and regression tasks are defined in Section 3.2.

	Common (%)		Challenging (%)	
	Error	Failure	Error	Failure
One-shot Detection	6.05	4.62	8.14	12.4
One-shot 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

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 [22]. To better investigate the benefits of spatial recurrent learning, we partitioned the validation set into two image groups according to the absolute value of yaw angle: **(1)** Common settings where yaw $\in$ [0°-30°); and **(2)** Challenging settings where yaw $\in$ (30°,90°]. The training sets are ensembles of AFLW [22], Helen [24] and LFPW [4] as described in Table 4.

One-shot vs. Recurrent. We investigated four different network configurations: **(1)** One-shot prediction using the detection loss defined in (6); **(2)** One-shot prediction using the regression loss defined in (8); **(3)** Recurrent prediction using detection followed by detection; and **(4)** Recurrent prediction using detection followed by regression. The mean fitting errors and failure rates are reported in Table 6.

First, the results show that spatial recurrent learning can instantly decrease the fitting error and failure rate, compared with one-shot learning. 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 learning using detection-followed-by-regression loss.

Cascade vs. Recurrent. It is reasonable to compare the proposed spatial recurrent learning with the widely used cascade learning such as [45, 57]. For a fair comparison, we implemented a two-step cascade variant of our approach. Each network in the cascade has exactly the same architecture as the spatial 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 significantly improve the fitting performance. The underlying reason is 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 case where the initial guess is usually far away from the ground truth. Moreover, a single network with shared weights can instantly reduce the memory usage to one-half of the cascaded implementation.

5.4 Validation of Temporal Recurrent Learning

We validate the proposed temporal recurrent learning on the validation set of 300-VW [43]. 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 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.

Table 7: Comparison of cascade and recurrent learning in challenging settings of AFLW [22]. The latter improves accuracy with a half memory usage of the former.

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

Table 8: Validation of temporal recurrent learning on 300-VW [41]. 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 Figure 8.

	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

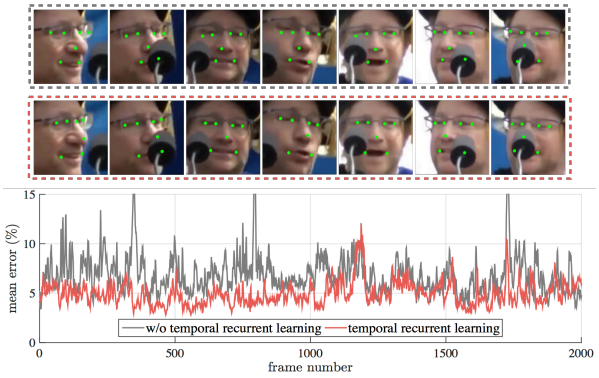

Fig. 8: Examples of temporal recurrent learning on 300-VW [41]. The tracked subject undergoes intensive pose and expression variations as well as severe partial occlusions. f_{trn} substantially improves the tracking robustness (less variance) and fitting accuracy (low error), especially for landmarks on the nose tip and mouth corners.

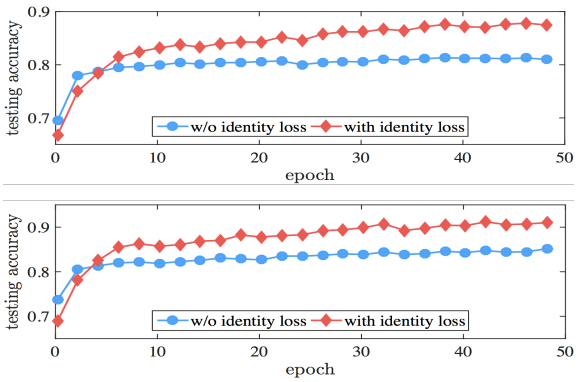

Fig. 9: Testing accuracy of different facial components with respect to the number of training epochs. The proposed supervised identity disentangling helps to achieve a more complete factor decoupling in the bottleneck of the encoder-decoder, which yields better generalization capability and more accurate testing results.

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, std and failure rate. Figure 8 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 [41] is shown in Figure 9. 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 handcrafted features include: **(1)** DRMF [1], **(2)** ESR [7], **(3)** SDM [53], **(4)** IFA [2], and **(5)** PIEFA [39]. The deep learning based methods include: **(1)** DCNC [45], **(2)** CFAN [57], and **(3)** TCDCN [58]. 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) [12], and in-the-wild datasets, *e.g.* Face Movie (FM) [39] and 300-VW [43].

Table 9: Mean error comparison with state-of-the-art methods on multiple video validation sets. 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 [12]	FM [39]	300-VW [43]		TF [12]	FM [39]	300VW [43]
DRMF [1]	4.43	8.53	9.16	ESR [7]	3.49	6.74	7.09
ESR [7]	3.81	7.58	7.83	SDM [53]	3.80	7.38	7.25
SDM [53]	4.01	7.49	7.65	CFAN [57]	3.31	6.47	6.64
IFA [2]	3.45	6.39	6.78	TCDCN [58]	3.45	6.92	7.59
DCNC [45]	3.67	6.16	6.43	CFSS [59]	3.04	5.67	6.13
RED-Net (Ours)	**2.89**	**5.14**	**5.29**	RED-Net (Ours)	**2.77**	**4.93**	**5.15**

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 40ms to process an image using a Tesla K40 GPU accelerator. Please refer to Figure 10 for fitting results of our approach on FM [39] and 300-VW [43], which demonstrate the robust and accurate performance in large pose/expression changes, illumination variations and partial occlusions.

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

1. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Robust discriminative response map fitting with constrained local models. In: CVPR, pp. 3444–3451 (2013)
2. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: CVPR (2014)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR (2015)
4. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: CVPR (2011)
5. Black, M., Yacoob, Y.: Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In: CVPR, pp. 374–381 (1995)
6. Bulat, A., Tzimiropoulos, G.: Human Pose Estimation via Convolutional Part Heatmap Regression, pp. 717–732. Springer International Publishing, Cham (2016)
7. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. IJCV **107**(2), 177–190 (2014)
8. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. CoRR **abs/1409.1259** (2014)
9. Chrysos, G.G., Antonakos, E., Zafeiriou, S., Snape, P.: Offline deformable face tracking in arbitrary videos. In: ICCVW, pp. 954–962 (2015)
10. Cootes, T.F., Taylor, C.J.: Active shape models - smart snakes. In: BMVC (1992)
11. Decarlo, D., Metaxas, D.: Optical flow constraints on deformable models with applications to face tracking. IJCV **38**(2), 99–127 (2000)
12. FGNet: Talking face video. Tech. rep., Online (2004)
13. Gao, X., Su, Y., Li, X., Tao, D.: A review of active appearance models. IEEE Transactions on Systems, Man, and Cybernetics **40**(2), 145–158 (2010)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computing **9**(8), 1735–1780 (1997)
16. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. CoRR **abs/1506.04924** (2015)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR **abs/1502.03167** (2015)
18. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACMM, pp. 675–678 (2014)
19. Jourabloo, A., Liu, X.: Large-pose face alignment via cnn-based dense 3d model fitting. In: CVPR (2016)
20. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
21. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. CoRR **abs/1511.02680** (2015)
22. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: Workshop on Benchmarking Facial Image Analysis Technologies (2011)
23. Lai, H., Xiao, S., Cui, Z., Pan, Y., Xu, C., Yan, S.: Deep cascaded regression for face alignment. In: arXiv:1510.09083v2 (2015)
24. Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.S.: Interactive facial feature localization. In: ECCV, pp. 679–692 (2012)
25. Learned-Miller, G.B.H.E.: Labeled faces in the wild: Updates and new reporting procedures. Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst (2014)
26. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CoRR **abs/1411.4038** (2014)

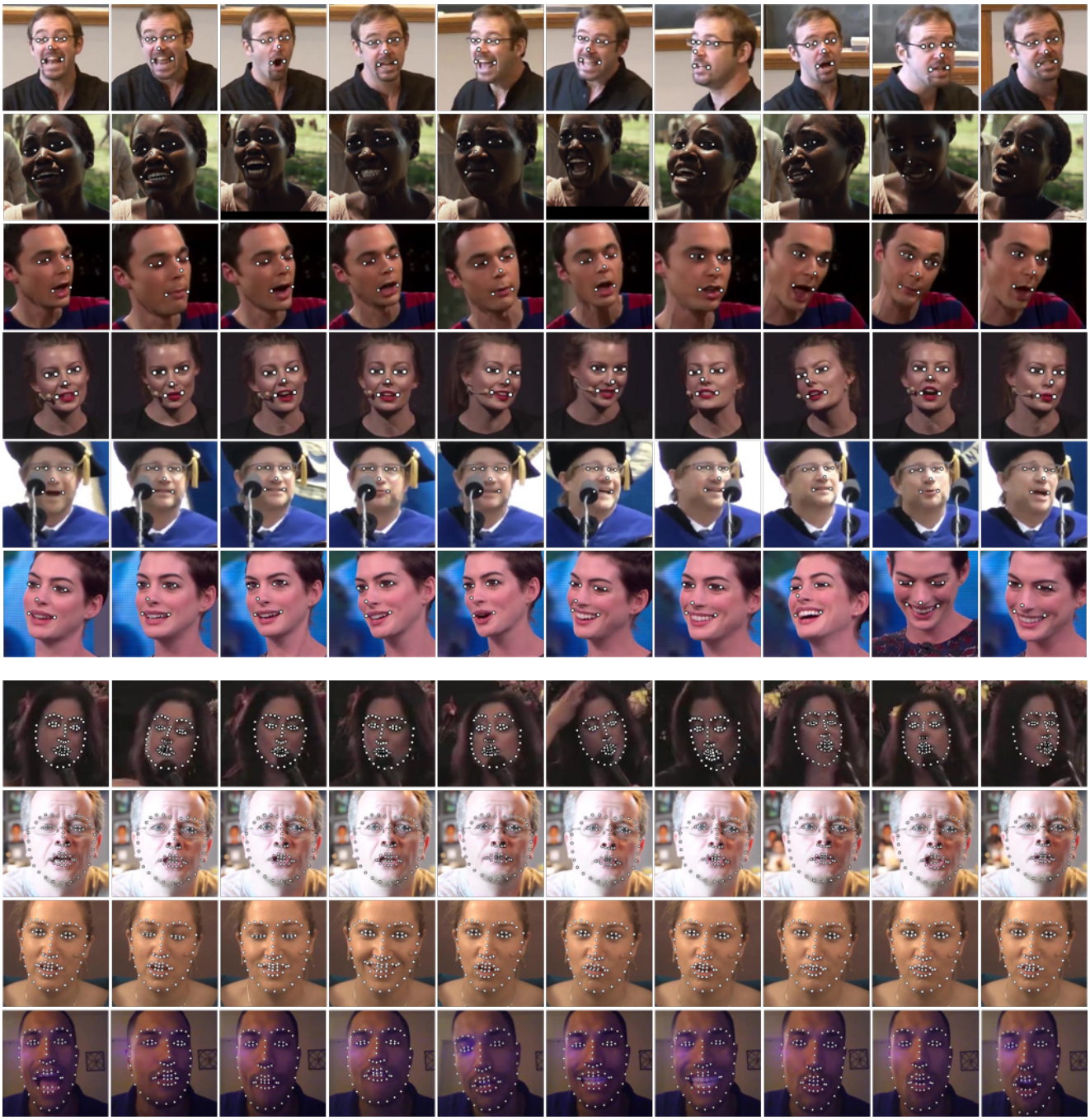

Fig. 10: Examples of 7-landmark (**row 1-6**) and 68-landmark (**row 7-10**) fitting results on FM [39] and 300-VW [43]. The proposed approach achieves robust and accurate fittings when the tracked subjects suffer from large pose/expression changes (**row 1, 3, 4, 6, 10**), illumination variations (**row 2, 8**) and partial occlusions (**row 5, 7**).

27. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: NIPS, pp. 1601–1609 (2014)
28. Lu, L., Zhang, X., Cho, K., Renals, S.: A study of the recurrent neural network encoder-decoder for lar ge vocabulary speech recognition. In: INTERSPEECH (2015)
29. Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., Ranzato, M.: Learning longer memory in recurrent neural networks. CoRR **abs/1412.7753** (2014)
30. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH (2010)
31. Milborrow, S., Nicolls, F.: Locating facial features with an extended active shape model. In: ECCV, pp. 504–513 (2008)
32. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML, pp. 807–814 (2010)
33. Newell, A., Yang, K., Deng, J.: Stacked Hourglass Networks for Human Pose Estimation, pp. 483–499 (2016)
34. Oh, J., Guo, X., Lee, H., Lewis, R.L., Singh, S.: Action-conditional video prediction using deep networks in atari games. In: NIPS, pp. 2845–2853 (2015)
35. Oliver, N., Pentland, A., Berard, F.: Lafter: Lips and face real time tracker. In: CVPR, pp. 123–129 (1997)

36. Patras, I., Pantic, M.: Particle filtering with factorized likelihoods for tracking facial features. In: Automatic Face and Gesture Recognition, pp. 97–102 (2004)
37. Peng, X., Feris, R.S., Wang, X., Metaxas, D.N.: A recurrent encoder-decoder network for sequential face alignment. In: European Conference on Computer Vision, pp. 38–56. Springer International Publishing (2016)
38. Peng, X., Huang, J., Hu, Q., Zhang, S., Elgammal, A., Metaxas, D.: From circle to 3-sphere: Head pose estimation by instance parameterization. CVIU **136**, 92–102 (2015)
39. Peng, X., Zhang, S., Yang, Y., Metaxas, D.N.: Piefa: Personalized incremental and ensemble face alignment. In: ICCV (2015)
40. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: database and results. Image and Vision Computing **47**, 3 – 18 (2016)
41. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: ICCVW (2013)
42. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR, pp. 815–823 (2015)
43. Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: Benchmark and results. In: ICCVW (2015)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
45. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: CVPR, pp. 3476–3483 (2013)
46. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: CVPR, pp. 2892–2900 (2015)
47. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: CVPR (2014)
48. Tzimiropoulos, G.: Project-out cascaded regression with an application to face alignment. In: CVPR, pp. 3659–3667 (2015)
49. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: ICCV (2015)
50. Wang, J., Cheng, Y., Feris, R.S.: Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In: CVPR (2016)
51. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. TPAMI **37**(10), 2071–2084 (2015)
52. Wu, Y., Ji, Q.: Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In: CVPR (2016)
53. Xuehan-Xiong, De la Torre, F.: Supervised descent method and its application to face alignment. In: CVPR (2013)
54. Yang, J., Reed, S., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In: NIPS (2015)
55. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A.: Describing videos by exploiting temporal structure. In: ICCV (2015)
56. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV, pp. 818–833 (2014)
57. Zhang, J., Shan, S., Kan, M., Chen, X.: Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In: ECCV, pp. 1–16 (2014)
58. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: ECCV, pp. 94–108 (2014)
59. Zhu, S., Li, C., Loy, C.C., Tang, X.: Face alignment by coarse-to-fine shape searching. In: CVPR, pp. 4998–5006 (2015)
60. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3d solution. In: CVPR (2016)