

## Self-Reinforced Cascaded Regression for Face Alignment

Xin Fan,<sup>1,2,\*</sup> Risheng Liu,<sup>1,2</sup> Kang Huan,<sup>1,2</sup> Yuyao Feng,<sup>1,2</sup> Zhongxuan Luo<sup>1,2,3</sup>

<sup>1</sup>DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Dalian, China

<sup>2</sup>Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, China

<sup>3</sup>School of Mathematical Science, Dalian University of Technology, Dalian, China

{xin.fan, rsliu, zxlue}@dlut.edu.cn, huyankang@hotmail.com, yyaofeng@gmail.com

### Abstract

Cascaded regression is prevailing in face alignment thanks to its accuracy and robustness, but typically demands manually annotated examples having low discrepancy between shape-indexed features and shape updates. In this paper, we propose a self-reinforced strategy that iteratively expands the quantity and improves the quality of training examples, thus upgrading the performance of cascaded regression itself. The reinforced term evaluates the example quality upon the consistence on both local appearance and global geometry of human faces, and constitutes the example evolution by the philosophy of “survival of the fittest”. We train a set of discriminative classifiers, each associated with one landmark label, to prune those examples with inconsistent local appearance, and further validate the geometric relationship among groups of labeled landmarks against the common global geometry derived from a projective invariant. We embed this generic strategy into typical cascaded regressions, and the alignment results on several benchmark data sets demonstrate its effectiveness to predict good examples starting from a small subset.

### Introduction

Face alignment, aiming at accurately and robustly localizing facial landmarks, plays a key role to many automatic facial analysis tasks including face recognition, expression recognition, attribute analysis, and animation. Recently, cascaded regression has become one of the most popular approaches to face alignment due to its accuracy and robustness (Ren et al. 2014; Xiong and De la Torre 2013; Kowalski, Naruniec, and Trzcinski 2017). This approach learns a series of regressors between shape-indexed features and shape updates or gradients from a set of manually labeled face images. Inevitably, the performance of cascaded regression highly depends on the quantity and quality of training examples. The quantity of unlabeled facial images is not a problem in this ‘big-data’ era, but example labeling and the quality of labels are still critical. In this study, we focus on these critical issues for cascaded regression.

Despite of its great success, the discrepancy or mismatch between limited training examples and the huge solution space typically downgrades the stability and accuracy of

cascaded regression. One typical treatment is to divide the original shape space into smaller sub-spaces (Zhu et al. 2016; Tuzel, Marks, and Tambe 2016). Researchers also attempt to group relevant input features for mitigating mismatches (Cao et al. 2014; Ren et al. 2014). The cascade Gaussian process (GP) regression trees find input features showing consistent appearance through GP kernel functions (Lee, Park, and Yoo 2015). The common strategy of these methods lies in that they ‘tighten’ the correlation between input feature and target shape from the perspective of *local appearance*.

Alternatively, researchers resort to the *global geometry* (shape) among facial landmarks in order to address the discrepancy issue. Martinez *et al.* embed nonparametric Markov networks (Martinez et al. 2013), while Liu *et al.* incorporate sparse shape constraints into regression (Liu, Deng, and Tao 2016). In addition to these explicit shape models, Li *et al.* discover the common geometry shared by human faces using a projective invariant, called characteristic number (CN), and append this geometric regression to appearance (Li et al. 2015). These various forms of facial geometric representation are able to regularize the regression, and thus improve the robustness of alignment.

It is commonly accepted in the machine learning (ML) community that training examples are central to any ML algorithms including regression. Unfortunately, the aforementioned alignment algorithms pay more attention to the regression mechanism, instead of data itself, to tackle the issue arisen from data discrepancy. Targeting at data preparation for training and validating regressors, Sagonas *et al.* develop a semi-automatic tool to annotate facial landmarks (Sagonas et al. 2013), but how these annotations may affect regression is untouched in their study. Antonakos *et al.* generate bounding boxes as face labels and validate these labels in the context of linear parametric models but not more complex cascade regression (Antonakos and Zafeiriou 2014). Recently, Zhang *et al.* develop a complicated deep network to leverage face annotations across data sets (Zhang et al. 2015). Nevertheless, a general framework is still highly demanded to fuse the discovering and upgrading training examples of low discrepancy into cascaded regression for face alignment.

Self-reinforcement refers to “a process whereby individuals control their own behavior by rewarding themselves when a certain standard of performance has been attained

\*Corresponding Author.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

or surpassed” (Artino 2011). In this paper, we propose self-reinforced cascaded regression that upgrades itself through minimizing an objective function analogous to meeting the performance standard. The optimization process iteratively updates *example labeling*, *sample survival*, and *regression* in one framework as shown in Fig. 1. The process starts from predicting unlabeled faces by the regression trained from a small number of labeled examples, and then evaluates the consistence of predicted labels on both *local appearance* and *global geometry* of human faces. Those survived examples are fed to train an upgraded regression. This process iteratively runs until convergence, yielding the cascaded regression for accurate and robust alignment.

The objective in our framework is not directly defined on the consistence between predicted labels and the ground truth as typical semi-supervised learning (Zhu and Goldberg 2009) that has the risk of overfitting, but is derived from indirect consistency with local appearance and global geometry. This independence on regressors is so general to generate the self-reinforced versions of various cascaded regression algorithms. We demonstrate that our strategy is able to automatically predict and find good examples starting from a subset as small as *one hundred* for typical regressors (Ren et al. 2014) and (Zhu et al. 2015), and even deep networks (Kowalski, Naruniec, and Trzcinski 2017). These self-reinforced regressions output comparable accuracy with the state-of-the-art on the 300W set consisting of the test sets of LFPW and Helen (Le et al. 2012) when only a small fraction of labeled examples are available, validating its effectiveness.

## Related Work

In this section, we review recent advances on labeling or generate examples in the machine learning community.

Semi-supervised learning attempts to use unlabeled data for performance improvements of classifiers trained by a small number of labeled examples (Zhu and Goldberg 2009). It has made great progress on solving the discrete classification problems in this decade (Li and Fu 2013; Li and Zhou 2015). However, it is nontrivial to directly bring the semi-supervised algorithms for discrete problems to cascaded regression where target shape updates are continuous and the solution space is quite huge. Self-paced learning (SPL), falling in the category of semi-supervised learning, include training samples in an easy-to-complex fashion (Jiang et al. 2014; Singh et al. 2015). Our approach shares commons with SPL on example selection embedded in the training process, differing in that our objective is general and decoupled from the training objective.

Generative adversarial network (GAN) (Goodfellow et al. 2014) is able to generate visually realistic images by competing two deep networks, a generator and a discriminator. Recently, GAN finds wide applications in many low level image processing tasks such as super-resolution (Ledig et al. 2016) and image attribute transfer (Huang et al. 2017). Semi-supervised learning can also be combined with GAN in order to improve the realism of a simulator’s output while preserving the annotation information (Shrivastava et al.

2016). Our example prediction and survival share the similar spirit with the generative and discriminative processes in GAN, respectively. But GAN has to initialize from a relatively larger number of examples to train two deep networks as the generator and discriminator, and provides no explicit regressor as self-reinforced regression does.

## Self-Reinforced Cascaded Regression

We describe our self-reinforced cascaded regression that defines an objective function with a local appearance and a global geometry discrepancy to iteratively expand the training set and simultaneously upgrade the regressor as shown in Fig. 1

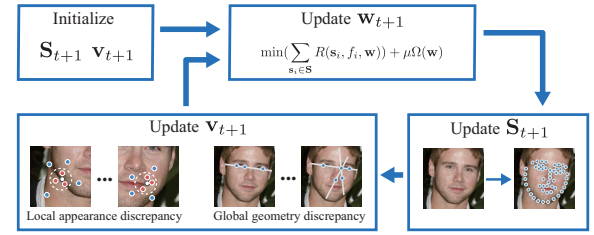


Figure 1: Overview of our self-reinforced cascaded regression, forming a closed loop with label prediction  $\mathbf{S}$  and survival  $\mathbf{v}$  as well as regression upgrading  $\mathbf{W}$ .

## General formulation

We attempt to devise a general formulation where the self-reinforcement is embedded with cascaded regression. Typical cascaded regression minimizes a loss function  $R(s_i, f_i, \mathbf{w})$ , where  $s_i$  is the annotated shape of the  $i$ th sample in the training set  $\mathbf{S}$ . The symbol  $f_i$  indicates the shape-indexed feature of the  $i$ th sample image, and  $\mathbf{w}$  denotes the parameters of the learnt regressor. We denote  $\Omega(\cdot)$  as the regularization term and  $\mu$  as a hyper parameter, and thus have a general representation for cascaded regression as follow:

$$\min(\sum_{s_i \in \mathbf{S}} R(s_i, f_i, \mathbf{w})) + \mu \Omega(\mathbf{w}). \quad (1)$$

Given the cascaded regression representation (1), we impose a regularize term to formulate the iterative reinforcement of predicted examples as:

$$(\mathbf{w}_{t+1}, \mathbf{v}_{t+1}, \mathbf{S}_{t+1}) = \arg \min_{v_i \in \{0,1\}^n} (\sum_{s_i \in \mathbf{S}_t} v_i R(s_i, f_i, \mathbf{w}_t) + \mu \Omega(\mathbf{w}_t) - \frac{1}{\alpha} \sum_{i=1}^n v_i), \quad (2)$$

where the subscript  $t$  indicates the  $t$ th iteration. The training set  $\mathbf{S}_t$  for the regression  $R$  includes either manually labeled or originally unlabeled examples with predicted annotations. The vector  $\mathbf{v}$  consists of the binary  $v_i$  that indicates whether the  $i$ th sample is accurately labeled or not, and the parameter  $\alpha$  is a weight that determines the number of survived samples. The increase of  $\alpha$  during the iteration leads to including more samples for regression.

The objective function (2) embraces the regression  $\mathbf{w}$ , shape labels  $\mathbf{s}$  and example selection  $v$  into one general framework whose optimization brings the joint upgrading of all these factors. Consequently, the optimization of this objective forms a complicated problem with the mixture of continuous and discrete variables. We resort to an iterative approximation to find the solution of (2). First, we fix  $\mathbf{v}_t$  and  $\mathbf{S}_t$  to find the optimal regression parameters  $\mathbf{w}_{t+1}$ . The problem (2) degrades to conventional cascaded regression (1), e.g., (Ren et al. 2014) and (Zhu et al. 2015) as detailed in the next section. For initialization,  $\mathbf{v}_t$  is set to 1 if the sample is manually labeled otherwise 0.

Once the trained regression  $\mathbf{w}_{t+1}$  is available, we are able to predict the unlabeled or to update the labeled subset. Given fixed  $\mathbf{v}_t$  and  $\mathbf{w}_{t+1}$  in (2), the updating of example labels  $\mathbf{S}_t$  becomes:

$$\min(\sum_{\mathbf{s}_i \in \mathbf{S}} R(\mathbf{s}_i, f_i, \mathbf{w})). \quad (3)$$

This minimization is equivalent to perform a prediction by applying the learned cascaded regression. This updating is so important in our self-reinforced regression that the process does not only expand the example quantity but also improves the labeling accuracy by the regression trained from the survived examples in the previous iteration.

Finally, we update  $\mathbf{v}_t$  with  $\mathbf{S}_{t+1}$  and  $\mathbf{w}_{t+1}$  fixed by de-generating (2) to:

$$\min(\frac{1}{\alpha} \sum_{i=1}^n v_i). \quad (4)$$

We compute the indicator  $\mathbf{v}_{t+1}$  upon local appearance and global geometry of human faces:

$$v_i = \begin{cases} 1 & -(\log a_i + \lambda \log g_i) < \alpha \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where the parameter  $\lambda$  weighs appearance and geometry. The value  $a_i$  is derived from local appearance indicating how accurately an individual landmark labels, and  $g_i$  indicates how a group of predicted labels satisfies the common geometry of human faces. The calculation of this new regularization term is independent to the regression  $R$ , providing the generalization for various regression algorithms.

*Remark:* The calculation of  $a_i$  and  $g_i$  acts as the goodness evaluation of individuals (examples), and hence initiates adjusting the behavior (accuracy) of individuals and that of cascaded regression for the next iteration, constructing the self-reinforcement process. The binary indicator  $\mathbf{v}_i$  specifies whether one label survives or not, implying the well-known law of nature “survivor of the fittest”. As nature evolves repeatedly, our self-reinforced cascaded regression iteratively upgrades from a small subset of labels until  $a_i$  and  $g_i$  are stable as shown in Fig. 1.

### Local appearance discrepancy

We define  $a_i$  as the discrepancy (similarity) among the shape-indexed features (concatenating HOG (Dalal and Triggs 2005) and FREAK (Alahi, Ortiz, and Vanderghyest

2012)) associated with an individual landmark. Figure 2 demonstrates the patches around three landmarks, i.e., the right corner, the upper boundary of the right eye, and the nose tip, from manually labeled images. The patches around the same landmark exhibit similar appearance, while greatly different from the other landmarks. Hence, the consistency of local patches around a landmark is able to indicate the accuracy of the labeled position.

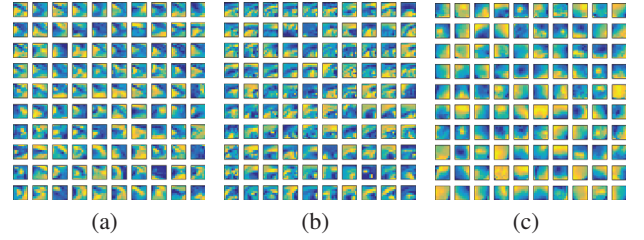


Figure 2: Local patches around (a) right corner the right eye, (b) upper boundary of the right eye, and (c) nose tip.

We take a straightforward technique to train an offline naive Bayes classifier that discriminates those labels with inconsistent neighboring appearance. We generate the positive and negative samples for training the classifier from the originally labeled subset by assuming that labeled and predicted landmarks are normally distributed. Hence, we randomly perturb the ground truth labels with a normal distribution, and compute the distance  $d_i$  between the ground truth  $\hat{l}$  and the perturbed landmark  $l_i$ . The feature around the landmark whose  $d_i$  is less than a threshold  $d_t$  (related to the standard deviation of the Gaussian distribution) is taken as one positive sample for the classifier, others as the negative. This generation scheme is illustrated in Fig. 3, where the white dot denotes the ground truth, the red ones stand for positive samples and the blue for negative ones.

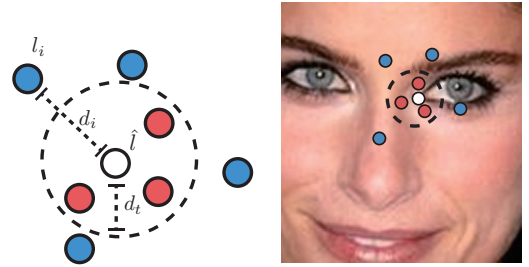


Figure 3: Generate training samples for the local appearance classifier.

Given a predicted landmark, we apply the trained classifier to determine whether the landmark is a valid prediction, and evaluate the local appearance discrepancy  $a_i$  for a predicted (or labeled) example as the portion of valid landmarks in the example:

$$a_i = \frac{\sum_{\mathbf{p} \in L_i} \arg\max_{c \in \mathcal{Y}} P(c) \prod_{k=1}^m P(p_k|c)}{|L_i|} \quad (6)$$



The symbol  $L_i$  denotes the set of local features for all landmarks in the  $i$ th sample,  $|L_i|$  is the number of landmarks, and the local feature vector  $\mathbf{p}$  has  $m$  components. The classifier output  $\mathcal{Y}$  is binary, where  $c = 1$  indicates a valid landmark and zero stands for an invalid one.

### Global geometry discrepancy

The above discrepancy can only reflect the local feature consistency around a landmark. We use the intrinsic facial geometry given by a projective invariant, named the characteristic number (CN) (Fan et al. 2015), to evaluate the discrepancy of predicted or labeled examples.

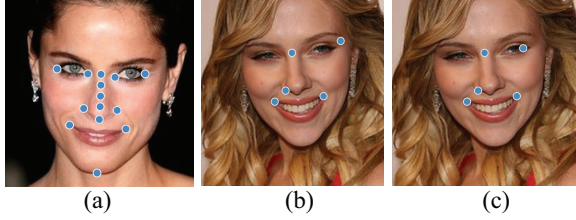


Figure 4: CN values reflect landmark geometry: (a) all chosen landmarks to generate point combinations for CN calculation, (b) one combination with the groundtruth,  $CN = 0.0204$ , and (c) one combination with an inaccurately labeled landmark,  $CN = 0.0479$ .

Fan *et al.* discover the common geometry on 8 landmarks (Li et al. 2015). Herein, we are considering to label and select examples with 68 landmarks. Unfortunately, it is prohibited for us to investigate all combinations of these 68 landmarks. We pick 14 landmarks that are all stably presented in all face examples, shown as the blue points in Fig. 4(a). We enumerate all possible three-point, five-point and six-point<sup>1</sup> combinations of these 14 landmarks, and then calculate the CN values of these combinations on all available samples. If a combination presents one common CN value with low standard deviation for all sample images, we set the value as the *intrinsic value* reflecting the common geometry underlying this landmark combination. Figure 4(b) and (c) show one sample with correctly labeled landmarks and another with an inaccurately labeled landmark, respectively. Their CN values are quite different. We have to emphasize that this process seeking combinations with stable intrinsic values only runs once for a large face data set. We verify the CN values of predicted landmark annotations on these fixed combinations in the iterative selecting process.

It is reasonable to regard a set of landmark annotations (labels) as valid when its CN value falls within a range around its corresponding intrinsic value, recorded as  $[c_k^{(min)} c_k^{(max)}]$ . Accordingly, the discrepancy  $g_i$  for the global geometry is given below:

$$g_i = \frac{\sum_{c_k \in C} p_{c_k}}{|C|} \quad (7)$$

$$\text{where } p_{c_k} = \begin{cases} 1 & c_k^{(min)} \leq c_k \leq c_k^{(max)} \\ 0 & \text{otherwise} \end{cases}$$

$c_k$  is the  $k$ th combination of CN values in the  $i$ th sample, and  $|C|$  is the total number of combinations, each of which can give one intrinsic value.

### Alignment Algorithms

The last regular term in (2) is independent on the choice of regression, and thus it is ready to embed the proposed algorithm into any cascaded regression algorithms. In this section, we exemplify the embedding to two algorithms LBF (Ren et al. 2014) and CFSS (Zhu et al. 2015) that balance accuracy and efficiency.

In every iteration, LBF have two updating stages: one for learning local binary features  $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$ , and the other for global linear regression  $\mathbf{W}^t$ . We pose the learning for the first stage as the minimization of the objective function (8), where  $\pi_l \circ \Delta \mathbf{z}_i$  is the ground truth 2-dimensional offset of the  $l$ th landmark in the  $i$ th training sample.  $I_i$  is the facial image corresponding to  $i$  sample:

$$\min_{w_l, \phi_l} \sum_{\mathbf{s}_i, \Delta \mathbf{z}_i} \|\pi_l \circ \Delta \mathbf{z}_i - w_l \phi_l(\mathbf{s}_i, I_i)\|_2^2, \quad (8)$$

where  $\mathbf{s}_i \in \mathbf{S}_{t-1}, \Delta \mathbf{z}_i \in \mathbf{S}_t$ .

Subsequently, we transform (2) into (9) in order to obtain the linear regression  $\mathbf{W}^t$  in LBF and combine it into our formulation.

$$(\mathbf{w}_{t+1}, \mathbf{v}_{t+1}, \mathbf{S}_{t+1}) = \arg \min_{\mathbf{v}_i \in \{0,1\}^n} \left( \sum_{\mathbf{s}_i, \Delta \mathbf{z}_i} v_i \|\Delta \mathbf{z}_i - \mathbf{w}_t \Phi(\mathbf{s}_i, I_i)\|_2^2 \right. \\ \left. + \mu \|\mathbf{w}_t\|_2^2 - \frac{1}{\alpha} \sum_{i=1}^n v_i \right), \quad (9)$$

where  $\mathbf{s}_i \in \mathbf{S}_{t-1}, \Delta \mathbf{z}_i \in \mathbf{S}_t$

Comparing (2) with (9), we have  $R(\cdot) = \|\Delta \mathbf{z}_i - \mathbf{w}_t \Phi(\mathbf{z}_i, I_i)\|_2^2$  and  $\Omega(\cdot) = \|\mathbf{w}_t\|_2^2$ . Consequently, we have the LBF algorithm embedded with our self-reinforcement.

The training of CFSS is to iteratively estimate a finer shape sub-region,  $(\bar{\mathbf{x}}_{(l)}, P_{(l)}^R)$ , where  $\bar{\mathbf{x}}_{(l)}$  is the center of the estimated sub-region and  $P_{(l)}^R$  is the probability distribution depicting the sub-region around the center. We simply replace the regression stage in (2) with the iterative training of CFSS. At this moment, the regression parameter  $\mathbf{w}_t$  indicates  $(\bar{\mathbf{x}}_{(l)}, P_{(l)}^R)$ , and then we can apply the self-reinforced process for CFSS.

### Experimental Results and Analysis

The experiments were performed on six widely used datasets include FRGC v2.0, LFPW, HELEN, AFW, iBUG and 300W. All faces are labeled 68 landmarks. We compute

<sup>1</sup>Four points cannot construct a projective invariant.

the alignment error for testing images using the standard mean error normalized by the inter-pupil distance (NME). The value of error indicates the percentage of the inter-pupil distance, and we simply ignore the symbol '%'.

Firstly, we verify the correlation between our discrepancy (no groundtruth label is available for its computation) and labeling error against the groundtruth. Then, we perform our self-reinforcement on two typical regressors and one recent deep model, resulting in examples of high quality at seven to twenty times, and finally compare our regression, whose training starts from a small number of labeled faces, with recent alignment algorithms.

### Correlation between discrepancy and error

We analyze the effectiveness of discrepancy that evaluates the example goodness in our self-reinforcement. The discrepancy attempts to reflect the labeling error, i.e., how inaccurate a sample is labeled. Generally, samples exhibiting larger discrepancy have higher labeling error.

To verify the correlation between the discrepancy and labeling error, we randomly chose 100 samples in LFPW, and trained an alignment regressor with these samples. Other 711 samples in LFPW were then labeled with the trained regressor. The labeling error and discrepancy of these predicted samples are plotted in Fig. 5. The  $x$  axis is sample IDs sorted by labeling error in an ascending order. The red line indicates the labeling error and one blue circle denotes the value of discrepancy for each sample. Figure 5 demonstrates that there is a strong correlation between the discrepancy and labeling error. The values of the discrepancy for corresponding samples climb up with the increase of labeling error. The red line fits the changes of the discrepancy very well. This fittingness verifies that the defined discrepancy reflects how accurate a label is. Therefore, every time we keep the samples having lower discrepancies, we have the most accurately labeled sample survived. These labels of low discrepancy introduce minimal error into training.

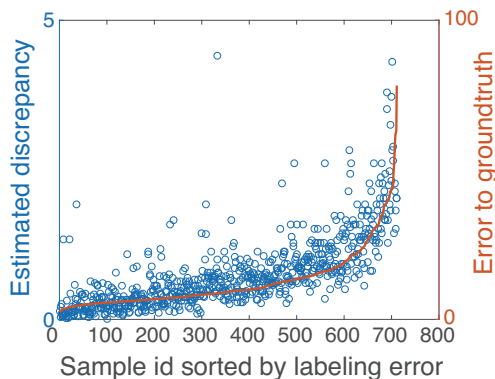


Figure 5: The values of discrepancy and labeling error (the red line). One blue circle indicates the discrepancy value of one sample.

### Unlabeled example predicting and survival

We firstly validate the self-reinforcement for typical regression, e.g., LBF and CFSS, on LFPW, and then our strategy for deep models highly data demanding on a larger mixed data set.

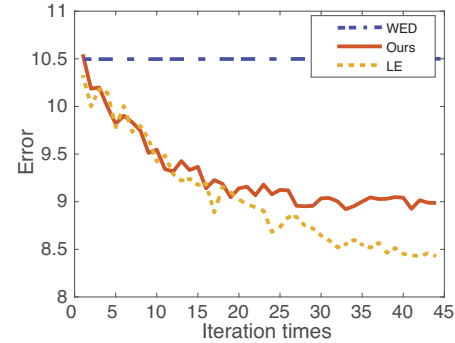


Figure 6: The test error of every iteration where the  $y$  axis indicates the error, and  $x$  denotes iteration steps. The blue dash dots are the errors of training without any extra unlabeled examples. The red solid and orange dots are those from ours and training with manually chosen examples.

**Self-reinforcement on conventional regression** LFPW contains more than one thousand images showing great variations especially on pose changes. Previous studies show that LBF and CFSS perform well on this set as long as hundreds of accurately labeled faces are available. We validate how close the self-reinforced versions of LBF and/or CFSS with unlabeled examples work to the original algorithms with labeled ones.

Firstly, we validate how the minimization of our objective (2) continuously predicts and preserves those examples of low discrepancy. Manually including examples of the lowest prediction error against the groundtruth (available in LFPW) gives the upper bound of the example survival. We started from 100 labeled examples, and implemented the self-reinforced version of LBF (SR-LBF) to automatically include 711 extra samples (regarded as unlabeled). The comparisons between manual inclusion of the lowest labeled error (LE) and our SR-LBF are plotted in Fig. 6 showing the mean alignment error in every iteration. The testing error of SR-LBF on 224 images, shown as the red solid line, decreases from 10.5 to 8.98, 14% lower than training without any extra unlabeled data (WED). The orange dots indicate the alignment errors of the regression with manually chosen samples having the lowest labeling error against the groundtruth. There is almost no difference between ours and LE in the beginning of the iteration process. The gap increases as more self-reinforced samples, automatically labeled and survived, are included, but reaches as low as 0.5 when the process converges. Our self-reinforcement is not necessarily able to generate and include the 'groundtruth' labels (not exist in practice), but definitely to improve the behavior of the regression toward the optimal.

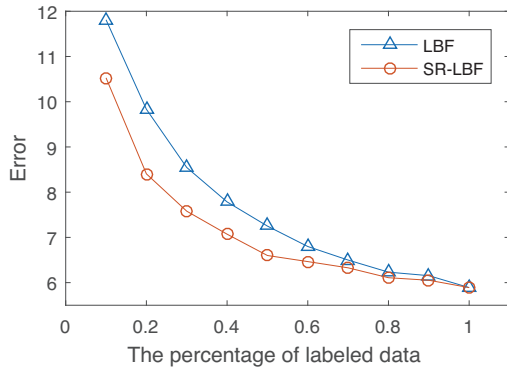


Figure 7: The mean alignment error under different ratios of manually groundtruth labels. The orange circles give the errors of SR-LBF with the mixture of groundtruth and automatically labeled examples by self-reinforcement. The blue triangles are those from LBF trained by various portions of groundtruth labels indicated by the  $x$  axis.

Secondly, we demonstrate the effectiveness of self-reinforcement by comparing SR-LBF with LBF when including different ratios of groundtruth labels for training. Besides those groundtruth labels, SR-LBF can include the rest of LFPW training images without their labels. Figure 7 illustrates the mean errors for SR-LBF and LBF on 224 testing LFPW images. As the increase of the percentage of groundtruth labels, both LBF and SR-LBF give lower errors because the quantity of training examples with high quality labels is expanding. The errors of SR-LBF are always lower than LBF, and the gaps are evident especially when only small fractions (less than 50%) of groundtruth labels are available. When all groundtruth labels are given, our regression degrades to LBF. This plot validates that the self-reinforcement is able to expand the quantity of training examples while maintaining the quality.

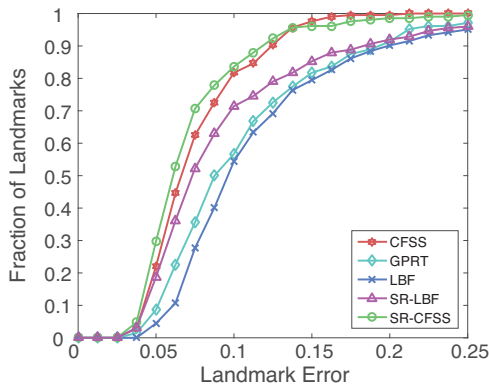


Figure 8: Cumulative errors distributions on testing images of LFPW. The  $x$ -axis is the normalized mean error (NME), and the  $y$ -axis indicates the percentage of images on which NMEs are lower than the  $x$  value.

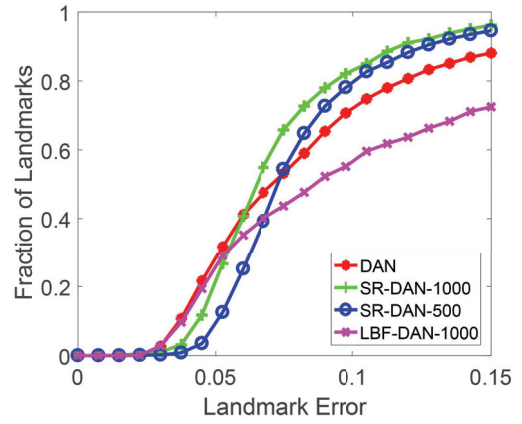


Figure 9: Cumulative errors distributions on Large dataset. The  $x$ -axis is the normalized mean error (NME), and the  $y$ -axis indicates the percentage of images on which NMEs are lower than the  $x$  value.

Thirdly, we compare the self-reinforced versions of CFSS (Zhu et al. 2015) and LBF (Ren et al. 2014) with the original algorithms as well as GPRT (Lee, Park, and Yoo 2015). Figure 8 illustrates the cumulative error distribution plots on 224 testing images of LFPW. All methods were trained with only 100 annotated images, but our self-reinforcement included 711 extra unlabeled samples. SR-CFSS has better performance than CFSS, and SR-LBF better than LBF. Both perform superior than GPRT, and SR-CFSS is the best of these five algorithms. The proposed self-reinforcement is capable of automatically labeling examples and preserving good ones. Faces annotated with alignment results are shown in Fig. 11<sup>2</sup>. The SR versions performs much better on noses and mouths presenting large variations that cannot be covered by a small number of training examples in the original regression algorithms.

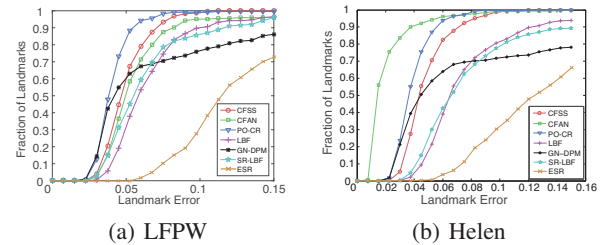


Figure 10: Cumulative errors distributions tested on LFPW and Helen.

**Self-reinforcement on deep networks** To test the capability of our self-reinforced strategy on a large amount of unlabeled facial images, we construct a large dataset which contains 8,151 images and is made up of 6 facial datasets include FRGC v2.0, LFPW, HELEN, AFW, IBUG and 300W. We compare the performance between the DAN(Kowalski,

<sup>2</sup>More images are available in the supplementary materials



Naruniec, and Trzcinski 2017) trained only by labeled examples, labeled examples with extra examples obtained by our self-reinforced strategy and labeled examples with extra examples obtained by LBF. The number of labeled examples is 100. Our self-reinforced framework use LBF as alignment algorithm and obtains over 3,000 labeled facial images (some bad samples are not chosen), then we choose 400 and 900 of them as extra examples for DAN. We also directly run LBF (Ren et al. 2014) which is trained by 100 samples on the large dataset, then perform randomly selection on the result of LBF to obtain 900 extra examples for DAN. 1,000 images from the large dataset are used for testing. Figure 9 illustrates the cumulative error distribution of these methods. As a deep learning method, DAN needs large amount of training data. The result shows that, when there are only 100 labeled training data provided, our method can enhance the performance of DBN by provide them another 400 training data. The performance can be improved when the number of extra data is added from 400 to 900. The comparisons between the regressor trained by labeled examples with extra examples obtained by our self-reinforced strategy and labeled examples with extra examples obtained by LBF prove that: selecting extra samples indiscriminately cannot only improve the performance but also result in poor accuracy.

### Quantitative comparisons with the state-of-the-art

We conducted comparisons with six face alignment algorithms on 300-W. These six face alignment regressors are pre-trained by a *huge* number of labeled images. CFAN and CFSS were trained on a combination of Helen (2000), LFPW (811) and AFW (337). The total number of these training samples is 3148. PO-CR and GN-DBM (Tzimiropoulos and Pantic 2014) were trained on the training set consisting of LFPW and Helen. ESR (Cao et al. 2014) were trained on Helen. The total number of training samples is 2811. GPRT and LBF were trained on the training set of LFPW having 811 labeled images. In contrast, our self-reinforced LBF (SR-LBF) starts from only a half of LFPW, i.e. 400 training labels, and the other half are included by our self-reinforced strategy. The cumulative error distributions of the compared methods and ours are shown in Figure 10.

The comparisons show that our regression does not necessarily give a better performance than the others. Instead, we are able to achieve comparable performance on common subsets of 300-W with an extremely small training set of labels. The number of our training labels is one half of GPRT, 25% of ESR, 14% of PO-CR and GN-DBM, and only 12% of CFAN and CFSS. Especially, our regression performs close to LBF with half of labels. Again, our self-reinforcement is open to any cascaded regression, and has the potential to improve the respective ability by automatically predicting and preserving high quality labels.

### Conclusion

We propose a self-reinforced cascaded regression that fuses the discovering and upgrading training examples of low discrepancy into cascaded regression for face alignment. The

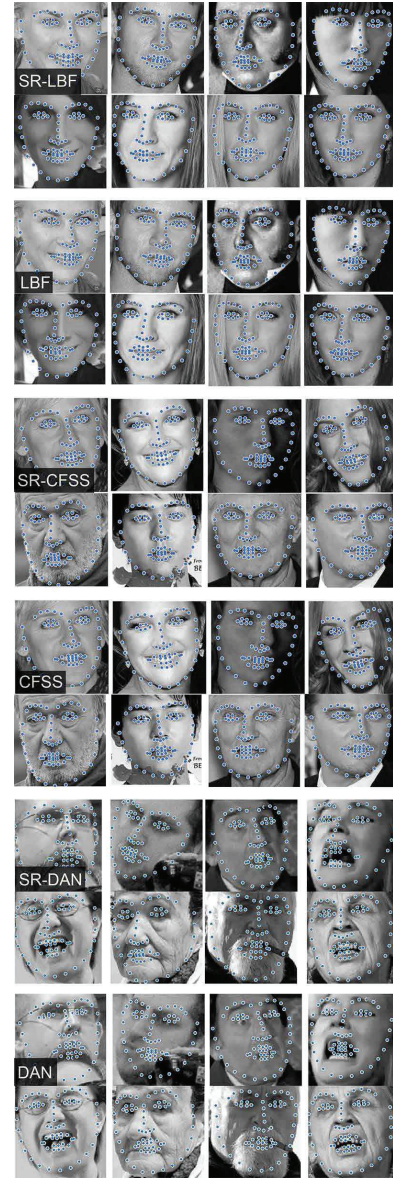


Figure 11: Example images from LBF, SR-LBF, CFSS, LR-CFSS on LFPW; Example images from DAN and LR-DAN on large dataset

framework is derived from indirect consistency with local appearance and global geometry. Finally, we validate the effectiveness of our regression. We are not intending to devise a competitive alignment algorithm trained with huge collected labels, but instead a self-reinforced strategy that automatically expands good training examples from a small subset, thus being complementary and more general to existing cascaded regression.

### Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 61572096, 61432003,

61733002, 61672125, and 61632019), and the Hong Kong Scholar Program (No. XJ2015008). Dr. Liu is also a visiting researcher with Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060.

## References

- Alahi, A.; Ortiz, R.; and Vanderghenst, P. 2012. Freak: Fast retina keypoint. In *CVPR*, 510–517.
- Antonakos, E., and Zafeiriou, S. 2014. Automatic construction of deformable models in-the-wild. In *CVPR*, 1813–1820.
- Artino, A. R. 2011. *Self-Reinforcement*. Boston, MA: Springer US. 1322–1324.
- Cao, X.; Wei, Y.; Wen, F.; and Sun, J. 2014. Face alignment by explicit shape regression. *IJCV* 107(2):177–190.
- Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, 886–893.
- Fan, X.; Wang, H.; Luo, Z.; Li, Y.; Hu, W.; and Luo, D. 2015. Fiducial facial point extraction using a novel projective invariant. *IEEE TIP* 24(3):1164–1177.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.
- Huang, R.; Zhang, S.; Li, T.; and He, R. 2017. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv:1704.04086*.
- Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, 547–556. ACM.
- Kowalski, M.; Naruniec, J.; and Trzcinski, T. 2017. Deep alignment network: A convolutional neural network for robust face alignment. *arXiv:1706.01789*.
- Le, V.; Brandt, J.; Lin, Z.; Bourdev, L.; and Huang, T. S. 2012. Interactive facial feature localization. In *ECCV*. 679–692.
- Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2016. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv:1609.04802*.
- Lee, D.; Park, H.; and Yoo, C. D. 2015. Face alignment using cascade gaussian process regression trees. In *CVPR*, 4204–4212.
- Li, S., and Fu, Y. 2013. Low-rank coding with b-matching constraint for semi-supervised classification. In *IJCAI*.
- Li, Y.-F., and Zhou, Z.-H. 2015. Towards making unlabeled data never hurt. *IEEE TPAMI* 37(1):175–188.
- Li, Y.; Fan, X.; Liu, R.; Feng, Y.; Luo, Z.; and Li, Z. 2015. Characteristic number regression for facial feature extraction. In *ICME*, 1–6.
- Liu, Q.; Deng, J.; and Tao, D. 2016. Dual sparse constrained cascade regression for robust face alignment. *IEEE TIP* 25(2):700–712.
- Martinez, B.; Valstar, M. F.; Binefa, X.; and Pantic, M. 2013. Local evidence aggregation for regression-based facial point detection. *IEEE TPAMI* 35(5):1149–1163.
- Ren, S.; Cao, X.; Wei, Y.; and Sun, J. 2014. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 1685–1692.
- Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; and Pantic, M. 2013. A semi-automatic methodology for facial landmark annotation. In *CVPR*.
- Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2016. Learning from simulated and unsupervised images through adversarial training. *arXiv:1612.07828*.
- Singh, B.; Han, X.; Wu, Z.; Morariu, V. I.; and Davis, L. S. 2015. Selecting relevant web trained concepts for automated event retrieval. In *ICCV*, 4561–4569.
- Tuzel, O.; Marks, T. K.; and Tambe, S. 2016. Robust face alignment using a mixture of invariant experts. In *ECCV*.
- Tzimiropoulos, G., and Pantic, M. 2014. Gauss-newton deformable part models for face alignment in-the-wild. In *CVPR*, 1851–1858.
- Xiong, X., and De la Torre, F. 2013. Supervised descent method and its applications to face alignment. In *CVPR*, 532–539.
- Zhang, J.; Kan, M.; Shan, S.; and Chen, X. 2015. Leveraging datasets with varying annotations for face alignment via deep regression network. In *ICCV*, 3801–3809.
- Zhu, X., and Goldberg, A. B. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning* 3(1):1–130.
- Zhu, S.; Li, C.; Change Loy, C.; and Tang, X. 2015. Face alignment by coarse-to-fine shape searching. In *CVPR*, 4998–5006.
- Zhu, S.; Li, C.; Loy, C. C.; and Tang, X. 2016. Unconstrained face alignment via cascaded compositional learning. In *CVPR*.