Face Alignment with an Ensemble of Gradient Boosting Trees

Jiashun XIAO
HKUST, Math department
`jxiaoae@connect.ust.hk`

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Abstract

Face alignment is a process that estimate the face's landmarks position or face shape from face images, and it is widely used in research and commercial applications, including, object pose estimation, face recognition, 3D reconstruction and automatic face beautification. However, the efficient and accuracy to estimate the face's landmarks position did not meet practical requirements until 2014. At that time, a milestone model was invented to solve this problem directly from a sparse subset of pixel intensities, achieving super-realtime and high prediction accuracy performance on standard dataset. The new model was based on ensemble of gradient boosting tree that optimizes the sum of landmarks position square error loss and appropriate priors of image data structure in feature selection. Here we train the model in iBUG 300W dataset, which composed of 600 face images with different pose, illumination, expression and corresponding coordinate annotations of 68 landmarks, and then test the model in a small independent dataset. The python source code is available for download at Github (https://github.com/JiaShun-Xiao/face-alignment-ert-2D)

Introduction

One main challenge about face alignment is how to extract reliable and informative features in the vector representation from raw face image with great various due to both shape deformation and nuisance factors such as changes in face expression and illumination conditions. On the one hand, we need reliable features to accurately estimate the landmarks position, and on the other hand, we need an accurate estimate of the shape to extract reliable features. Which is very similar to Expectation-maximization (EM) algorithm, here face shape is the latent variable that is unobserved, reliable features are parameters to be estimated. Therefore, it is intuitive to exploit iterative approach (the cascade) to deal with this problem. Once we have initial guess of face shape, usually the mean face pose, we can make a more accurate estimation of face shape in next iteration by transforming the image to a normalized coordinate system based on a current estimate of the shape and extract the pixel intensity value around the shape as features. This process is usually repeated several times until convergence. In each iteration, we learn the regressor via gradient boosting with a squared error loss function, and the input features are selected with prior probability on the distance between pairs of input pixels, which dramatically reduce non-informative features in training process.

Methods

The cascade of regressor

Suppose we landmarks data $x_i \in \mathbb{R}^2$ be the x, y-coordinates of ith facial landmark in an image I. Then the vector $S = (x_1^T, x_2^T, , , , , x_p^T)^T \in \mathbb{R}^{2p}$ denotes the ground truth coordinates of all the p facial landmarks in I. $\hat{S}^{(t)}$ denote the current estimate of S. In each regressor of cascade, we fit a $r_t(.,.)$ to predict an update vector from image and $\hat{S}^{(t)}$:

$$\hat{S}^{(t+1)} = \hat{S}^{(t)} + r_t(I, \hat{S}^{(t)}) \tag{1}$$

The initial shape, $\hat{S}^{(0)}$, can simply be chosen as the mean shape of the training data.

Learning each regressor in the cascade

In each regressor r_t, we have triplets training data $\{(I_i, \hat{S}_i^{(t)}, \Delta S_i^{(t)})\}_{i=1}^N$ and learning rate $0 < v < 1$, where

$$\Delta S_i^{(t)} = S_i - \hat{S}_i^{(t)} \tag{2}$$

Algorithm 1: Learning r_t in the cascade

```
1.Initialise
    $f_0(I, \hat{S}^{(t)}) = \underset{\omega}{argmin} \sum_{i=1}^N \left\| \Delta S_i^{(t)} - \omega \right\|^2$
2.for $k = 1, ..., K$:
    (a) Set $i = 1, ..., N$
            $Residual_{ik} = \Delta \hat{S}_i^{(t)} - f_{k-1}(I, \hat{S}_i^{(t)})$
    (b) Fit a regression tree to the targets $Residual_{ik}$ giving a weak regression function $g_k(I, \hat{S}^{(t)})$.
    (c) Update
            $f_k(I, \hat{S}^{(t)}) = f_{k-1}(I, \hat{S}^{(t)}) + vg_k(I, \hat{S}^{(t)})$
3. Output $r_t(I, \hat{S}^{(t)}) = f_K(I, \hat{S}^{(t)})$
```

Tree based regressor

Each regressor r_t is composed of multiple gradient boosting trees that fit to the residual targets. We approximate the underlying function of tree with a piecewise constant function where a constant vector is fit to each leaf node. In the training, we firstly randomly generate a set of candidate splits θ at each node, where each split θ is composed of paired pixel coordinate and a threshold, if the intensity difference of paired pixel great than the threshold, the training sample will sent to left node. Then, we greedily choose the best θ^* from these candidates, which has a minimum of sum of square error. This corresponds to minimizing

$$E(Q, \theta) = \sum_{s \in l, r} \sum_{i \in Q_{\theta,s}} \left\| Resisual_i - \mu_{\theta,s} \right\|^2 \tag{3}$$

where Q is the set of the indices of the training data at a node. $Q_{\theta,l}$ is the indices of the examples that are sent to the left node due to the decision induced by θ, $Residual_i$ is the vector of all the residuals computed for image i in the gradient boosting algorithm and

$$\mu_{\theta,s} = \frac{1}{|Q_{\theta,s}|} \sum_{i \in Q_{\theta,s}} Residual_i, \ for\ s \in \{l, r\} \tag{4}$$

In order to make the training process of tree more easy to understand, we provide a figure (Fig.1) below to give a more intuitive illustration about what is good split. The pair red pixels and threshold are candidate split we randomly generated. In the training samples, we can see pixel intensity difference $\Delta 1$, $\Delta 2$, ΔN should great than the threshold, while $\Delta 3$ should below the threshold since both pixel are in the face region. Therefore, the split can divide the training samples into two parts, In the left part ($\Delta 1$, $\Delta 2$, ΔN), the landmarks position should move to right, while in the right part ($\Delta 3$), the landmarks position should move to left.

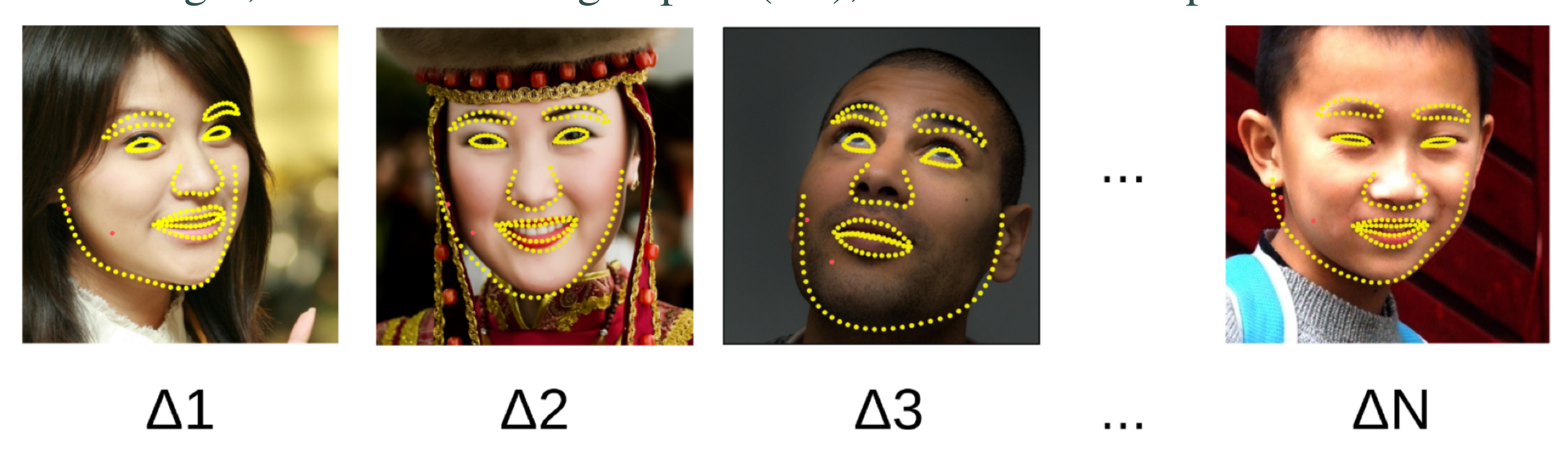

Figure 1: Illustration of Tree training

Results

We perform the experiment on iBUG 300W dataset with the following fixed parameter settings. The learning rate v is 0.1, number of regressor r_t in the casecade is T=10, and each r_t comprises of $K = 500$ trees g_k. The depth of trees is set to $F = 5$. At each level of the cascade $P = 400$ pixel locations are sampled from the image. We randomly generate 20 candidate split for each node of tree in the training process and find the best split from them. To augment the training data sample size, we use 20 different initializations for each training example. The runtime complexity of the algorithm on a single image is constant $O(TKF)$. The complexity of the training time depends linearly on the training data sample size O(NDTKFS) where N is the number of training data and D is dimension of the targets. Figure 2 shows the prediction error on test data at different levels of the cascade, which shows that algorithm can reduce the error very fast. The error is the average normalized distance of each landmark to its ground truth position. The distances are normalized by dividing by the interocular distance. Figure 3 show the landmark estimates at different levels of the cascade initialized with the mean shape centered at the output of menpo face detector. After the first few level of the cascade, the error is already greatly reduced.

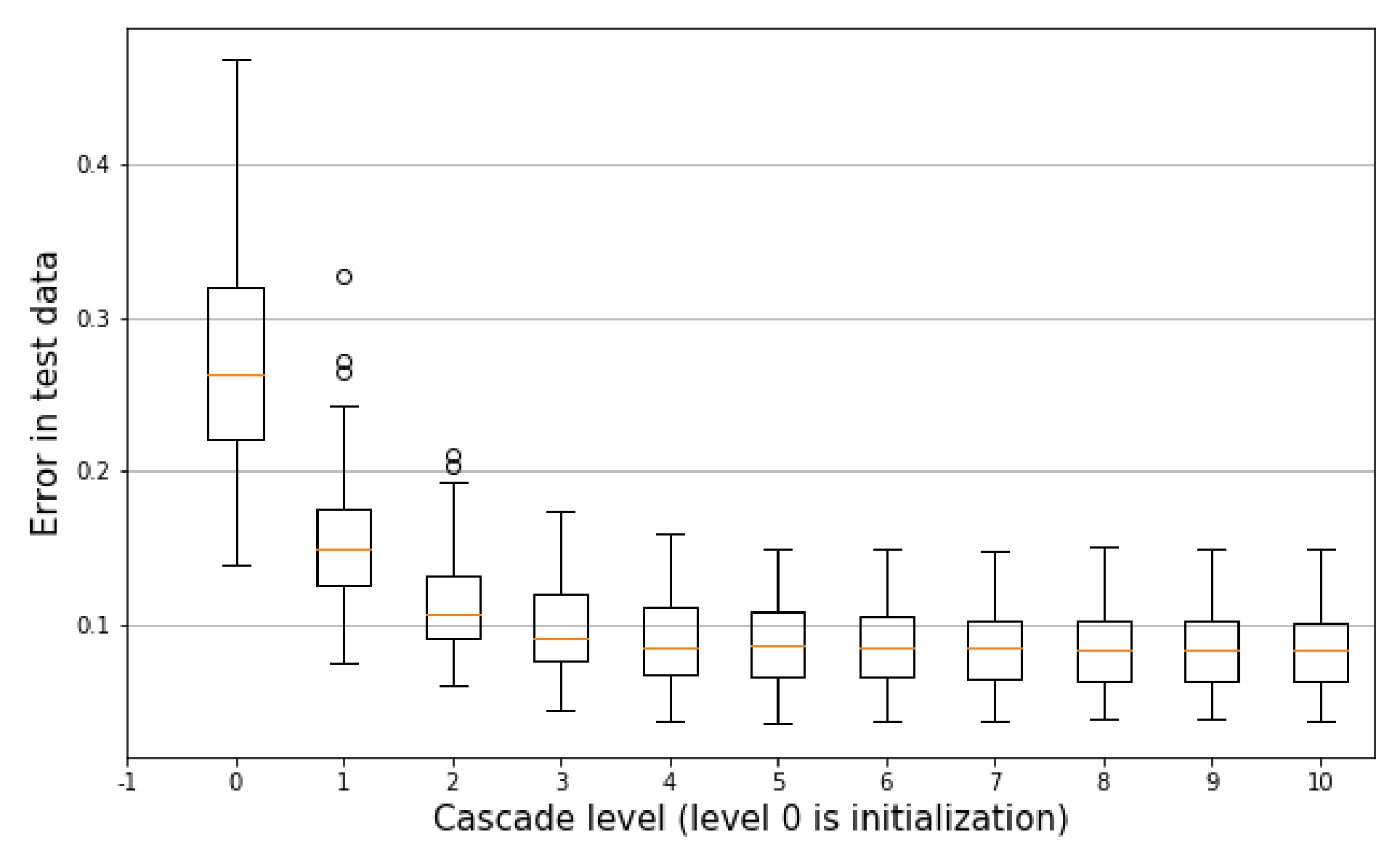

Figure 2: The Prediction error at each level of the cascade

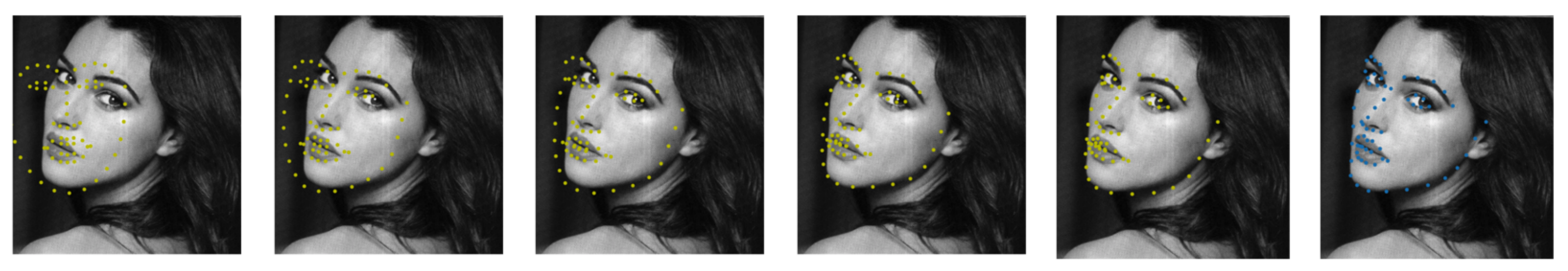
(a) T=0 (b) T=1 (c) T=2 (d) T=3 (e) T=10 (f) Ground truth

Figure 3: . Landmark estimates at different levels of the cascade

Conclusions

We demonstrate how to estimate the location of facial landmarks from a sparse subset of intensity values extracted from an input image by ensemble of gradient boosting trees, and it is faster and more accurate than previous work.

References

[1] Kazemi V, Sullivan J One millisecond face alignment with an ensemble of regression tree. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

Figure 4: . More face image with estimated landmarks position