Real-time Object Recognition Using Cascaded Regression

Le Thanh Vinh

March 2020

1 Data

COFW data set use the original non-augmented 845 LFPW faces + 500 COFW faces (1345 total), and for testing the remaining 507 COFW faces. All images were hand annotated in lab using the same 29 landmarks.

Figure 1: COFW data set with 29 landmarks

2 Theory

2.1 Feature Extraction

Table 1

	SCALE	**ROTATION**	**TRANSLATION**
LBP	False	False	False
HoG	True	False	True
SIFT	True	True	True
Deep Features	True	True	True

As we see above SIFT and Deep Features are scale-invariant, rotational invariant, translational invariant but take time and computer resources meanwhile LBP is not for all. In this implementation, I'm using HoG for purpose of fast and high accuracy in human detection, face detection and feature extraction. Althought it is not invariant rotation but the data have been processed in the right form so it is not a point.
The features in each of these landmark will have to be extracted and used as input for the regressors, With I^{W*H*C} as the notation of the image, N is the numbers of landmarks, the feature extraction:

$$\begin{aligned} \Phi : \Re^{W*H*C} * \Re^{2N} &\to \Re^{FN} \\ (I, x) &\mapsto \phi \end{aligned}$$

2.2 Cascaded Regressors

We will solve the problem for the data set with M image.The true concatenated coordinates of landmark is denoted by Y^{2N*M} .During runtime, however,since the true coordinates of the landmarks are unknown, the system will starts with a random initializer to get approximated coordinates of these keypoints X^{2N*M}.Each regressor is a linear combination of extracted features and output the update to the prior coordinates:

$$\Delta X = R\Phi(I, X) + B$$

During the training time, given a dataset of M images and landmark sets, R and B are the regressor matrix and bias, respectively. These are chosen such that:

$$R = argmin_R(R\Phi + X + B - Y)^2$$

Assume:

$$\begin{aligned} L &= (R\Phi + X + B - Y)^2 \\ \implies \frac{\partial L}{\partial R} &= 2(R\Phi + X + B - Y)\Phi' \\ \implies \frac{\partial L}{\partial R} &= R\Phi\Phi' + (X - Y + B)\Phi' \\ Lmin \iff \frac{\partial L}{\partial R} &= 0 \\ \implies R\Phi\Phi' + (X - Y + B)\Phi' &= 0 \\ \implies R\Phi\Phi' &= (Y - X - B)\Phi' \\ \implies R &= (\Phi\Phi')^{-1}(X - Y + B)\Phi' \end{aligned}$$

3 Implementation

3.1 About the code

First of all, my codes requires the 2 dataset's folders COFWtraincolor.mat and COFWtraincolor.mat in the data folder.Train dataset include 1345 image, landmarks and face box respectively 507 for test set.

The file train.py is for training the model.With input are images,landmarks ground truth, initial landmarks, and information of model such as how many week regressor we use.By default is 5.

The file obtainHOG.py use to obtain HOG feature of the image and landmarks point.

The file initialCod.py use to initiate landmarks point use for training.

The file fitsdm.py use list of matrix R,B obtain in training to predict landmarks point of input image.

The file predict.py use landmarks point predict above to draw point using OPENCV.

The file app.py connect function build later to construct a model.

For web demo, execute the file app.py

(a) Origin image.

(b) Ground truth landmarks.

(c) Predict landmarks.

Figure 2: Using just 10 weak regressor

3.2 Web interface

I used Flask to implement the web service and Boostrap 4 for frontend.Below is a screen shot:

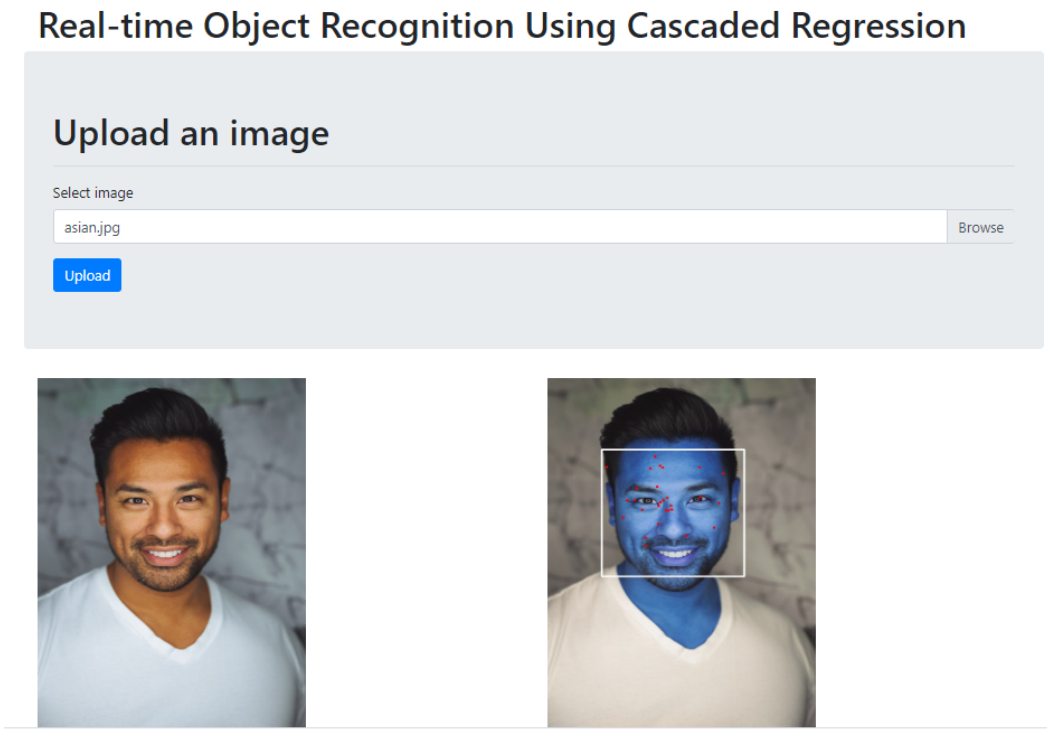

Figure 3