Reimplementation of SDM

Bui Quang Long

March 2020

Contents

1 Introduction

In this assignment, we are going to reimplement a facial landmark detector using supervised descent method (SDM). First, we will train a SDM and test it via a simple web app.

1.1 The idea

In this assignment, the core idea to detect key points is to randomly initialize a set of key points, then learn Cascaded regressors to transform random points to the right positions (which were annotated manually). Due to the fact that the regressor is **highly depended on initialization**, thus we come up with the idea of using consecutive regressors, which is so-called Cascaded Regressors.

1.2 The work flow

The flow of work is illustrated as the following: The detailed implementation will be discussed in implementation section below.

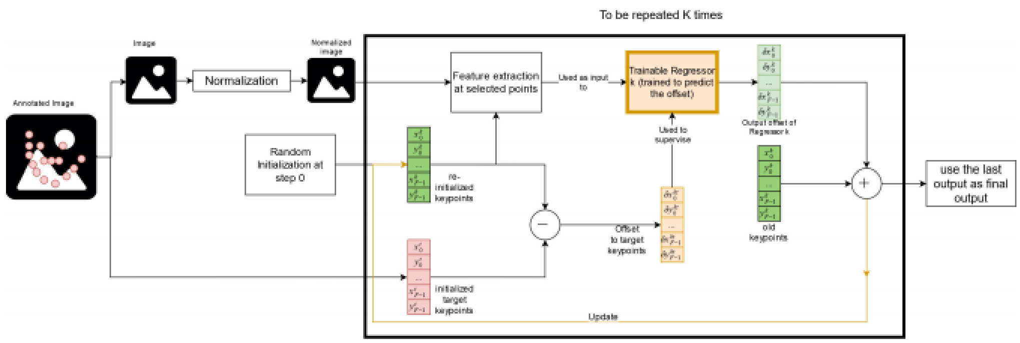

Figure 1: Work flow

2 Comparing some feature extracting techniques

Before getting into the work, let us review some of the feature extraction techniques. Notice that here we want to extract the feature of specific positions in the image. Some methods have been proposed:

- **Local binary pattern** (LBP): In short, LBP will compare one pixel with some of its neighbors in gray-scale. Then the comparison wll be denoted as a binary sequence, which later is converted to a decimal number. The cool thing of LBP is illumination invariant, meaning that if we change the light intensity of the image, the decimal number (or the feature of this pixel) will remain unchanged. For this feature, LBP is good to detect edges of object, especially texture. (A GOOD EXPLANATION VIDEO)

- **Histogram of oriented gradient** (HoG) Considering one pixel, the HoG will calculate the change in horizontal and vertical axis, then convert it to different scales, which are the magnitude and direction of gradient vectors. For this characteristic, HoG is good to detect the shape of object.

- **Scale-invariant feature transform** (SIFT): The name of SIFT has already revealed its feature: scale-invariant. This technique will can detect similar images or object in different scenario by matching similar patches in different images. Thus, this technique is good in problems with a lot of changes in size, scenario, time, ...

- **Convolutional neural network** (CNN): It can be admitted that deep learning models can perform better than above techniques if built and trained properly.

3 Implementation

3.1 Data Generation

Our 300VW data set is large, so I only use a portion of it. I used it in two ways:

- **Approach 1**: I extracted all frame from video 001. I also make a comparison of opencv-based extractor and Free Studio application. They both returned 1574 frames so I can be confident with this number. Then I will train a SDM to detect one random frame from this.
- **Approach 2**: I extracted the first frame in each of 300 videos. I think that the landmarks varies more in different videos rather than in different frames of one video. Thus I only have 300 frames to train my SDM, which is not enough. The hard thing is I need correct annotation for each frame, so I have to choose between one frame per video or all frames of one video.

3.2 Feature Extractor

In this assignment, I

- Resize images to (256 x 256)
- Convert images to gray-scale
- Extract features of key points via an uniform LBP of radius 3 and 24 neighbors. Each key point will give me a decimal number represented for the edge of in that key points.

3.3 Cascaded Regressors

In this assignment, I use MLPRegressor from sklearn for a regressor, with SGD and learning rate of 0.0001. The target for each regressor is the different between the true landmarks and the output of previous regressor.
With the data from approach 1, I used 3 regressors, then randomly take a frame for testing:

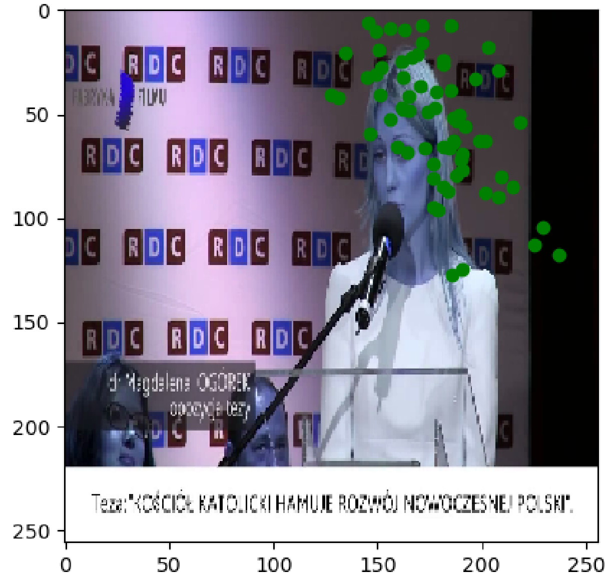

Figure 2: Testing on one frame.

When apply to data from approach 2, the result landmarks is quite far from the true ones.

3.4 Demo

I use flask to create a web app. Please run "app.py" to check it out. The front-end looks good but I haven't finished with the back-end yet. I'm still working on how they interact with each other.

3.5 An innovation of SDM

I found out that SDM has some weaknesses:

- Large computational cost
- Not robust enough because each regressor's training data comes from the output of previous one.

Thus, a better approach was proposed base on SDM, called SIR. Let us check it out.