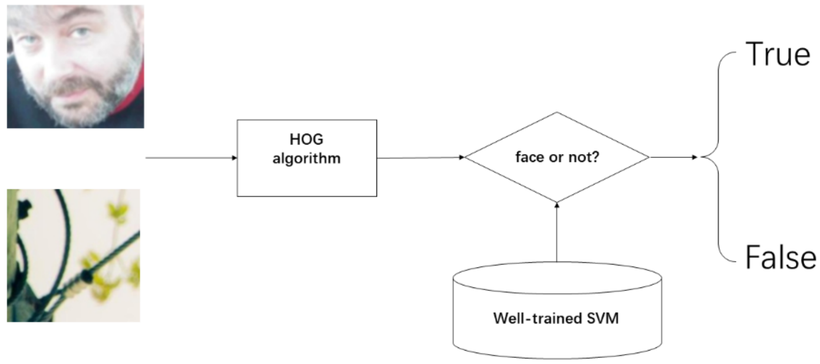
Face Detection Report

Face detection technology is becoming more and more popular in recent years, and in this summer exchange activities, we are divided into groups to do a project about this topic. After our discussion, we think the whole process could be roughly divided into four parts. Below is the workflow of our project.



1.We divide the photo sample set into three parts, they are training set, cross validation set and test set. And in each set, pictures can be divided into two parts, positive sample and negative sample. The pictures have many different sizes, and in order to be consistent, we resize them to 64*128.

In most of the time, we divided the data set into these three parts in a ratio of 3:1:1, and it can provide better help for us to select the best model.

Training set: fit the model and train the classification model by setting the parameters of the classifier.

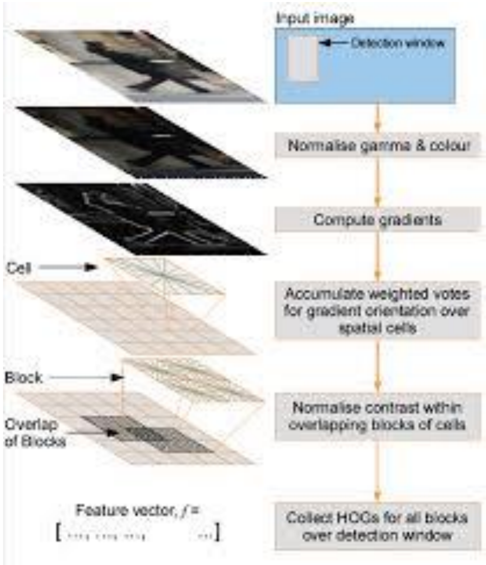
Cross Validation set: when we get the model that works best, then we can select the parameters corresponding to the model with it.

Test set: used for model prediction after we get the best model.

The sample we exactly want is positive sample, in this project, the picture with face is positive sample. And other picture without it is called negative sample.

2. Then we will analysis the features of the pictures. With HOG, the image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled, and after a series of processes, we can get the HOG eigenvectors that are stored in a one-dimensional array of these pictures.

This project adopts HOG algorithm to extract image features. The specific implementation process of HOG algorithm can be divided into the following steps: data preparation, calculation of pixel gradient, construction of histogram of gradient direction for each cell unit, statistics of gradient information of Block and output.



(1)data preparation

Read in the color image, and converted to grayscale image, image width and height. Gamma correction method is used to normalize the color space of the input image. The purpose is to adjust the contrast of the image, reduce the impact of local shadow and illumination changes, and suppress noise. The gamma value adopted is 0.5.

(original image) (grayscale image)

(2) calculation of pixel gradient

Calculate the gradient of the abscissa and ordinate directions of the image, and calculate the gradient direction value of each pixel position accordingly. The horizontal gradient, vertical gradient and pixel value at the pixel point (x, y) in the input image are calculated, so as to calculate the gradient magnitude and direction.

Amplitude and direction is the use of [1, 1] gradient operator on the original image for convolution operation, the x direction (horizontal direction, to the right as the positive direction) of gradient component sobelx, then use [1, 0, 1] T gradient operator on the original image for convolution operation, get the y direction (vertical direction, with upward as the positive direction) of gradient component sobely, through addweight and operation of phase function. They are ultimately stored in two two-dimensional lists

(3)Construct histogram of gradient direction

We divided the image into several "cells". By default, we set the cell to 8*8 pixels. Suppose we eight bin of the histogram is used to statistics the pixel gradient information, is the gradient direction of 360 degrees of cell can be divided into eight directions, when a pixel gradient direction in a certain degree, the corresponding 'bin' count plus one, in this way, each pixel within the cell to use in the histogram weighted gradient direction projection, you can get the gradient direction histogram of the cell.

The resulting information is a three-dimensional list of the histogram gradients of each cell

(4). statistics of gradient information of Block

Cell units are grouped into large blocks, and normalized gradient histograms are included in the blocks

The variation range of gradient intensity is very large due to the change of local illumination and foreground - background contrast. So we need to normalize the gradient intensity. Normalization further compresses light, shadows, and edges.

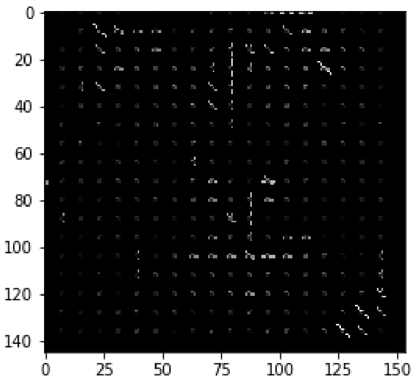
Blocks are composed of cells into large, spatially connected segments. These guys are overlapping,

In this project, a matrix-shaped block is adopted, which can be represented by three parameters: the number of cell units in each cell unit, the number of pixel points in each cell unit, and the number of histogram channels for each cell.

In this project, we adopted the following parameters: 2*2 cells/interval, 8*8 pixels/cell, and 8 histogram channels, with a step size of 1. The number of features on a block is 2*2*8.

We extracted the block information, converted it into a form acceptable to SVM, and finally output it

In addition, we can observe the extraction features of HOG algorithm by visualizing gradient histogram. We build the output matrix by calling some functions in the math package based on the gradient information in each cell, which is displayed by the plot method.



3.We put the results from the previous step into SVM classifier to do training. At the beginning, we set the initial parameters, and use pictures in the training set to train the SVM. Then we change the initial parameters, and do the previous steps over and over again, then through repeated attempts we may get the best initial parameters. We use this initial parameters to do the training once again, and we can get a well-trained SVM classifier. And we can use the pictures in the cross validation set to verify whether the result is the same as expected. If not, we may check the previous steps and do it again.

French research institute Dalal proposed HOG and SVM for pedestrian detection at CVPR in 2005. SVM is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories. An SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

Here we also draw on this idea, after extracting the HOG vector, using SVM for processing the matrix to detect the faces. The SVM model represents an example as points in space, and the mapping divides the examples of each category by the widest possible gap. The new examples are then mapped to the same space and predicted to belong to the category based on the gap they belong to.

The HOG descriptors of the positive and negative sample images are calculated to form a feature vector matrix, corresponding to a class vector specifying the category of each feature vector, and then input into the SVM for training.

First, get the data and create the SVM model and set parameters. In this process, we have to consider the choice of kernel function. In addition, the setting of c is also a consideration. When setting c, in order to get an optimal value, we try multiple values and select the best one. Then, training the model. After we get the model, save it. At last, test the model with the resulting SVM model, we can call it and predict.

4.Finally, we can use this SVM classifier to detect the face in the pictures of the test set, and it will give us feedback to show the accuracy. And we think if the accuracy is above Ninety-five percent will be a good result.

When everything is OK, we can choose any picture and run the program. Then a few seconds pass, the program will show us the result, detected face correctly or not.

Summarize:

In this project, we worked together for serval days, it’s really a difficult experience, but through our tireless efforts, we finally did it. We do not only learn something new and consolidate what we learned, but also increase the friendship between us. We have a clear division of labor, each person is responsible for a portion of the work, and finally we put it together and make some adjustments, everything is in order. And we also meet something difficult in the process of this project.

We are using cv2's api SVM function to train this model. In order to find the best setup parameter, we separated the data set to training set, validation set and test set. Then we found that we don't have enough data, therefore, we augmented the original data set by using random noise, flip the image upside down and blur the image.

Firstly, we tried a series value of C parameter (from 0.005 to 0.3, the step equals to 0.005) and gamma parameter of the svm. We trained the svm with different setup parameter by using same training data, then test their performance on validate set and choose the best one as our final model. Secondly, since we didn't use cv2's api function HoGDescriptor, but written by ourselves (we do reference some codes on github and CSDN). We changed some of the parameters to find the initial value with best performance. Besides, since we extracted HoG feature from a 64*128 pixel image, we need to resize the input image. We tried several ways to manage it. For example, we resized the original image to a slightly bigger size than 64*128,and cut a rectangle of 64*128 pixel. We also tried to simply resize the image to 64*128 pixel. It turns our that when we resize the original image to 64*128 pixel performed better.

By training our model with all above methods, our svm can guarantee over 95 percent accuracy on the test set.

Group: four (three week)

Date: 2019.07.20