پروژه ی تشخیص چهره و مات سازی پیش زمینه

گزارش نهایی

اعضای گروه :

فریدالدین حسامی

آریا پرویزی

علیرضا مهدی برزی

آذر و دی ماه ۹۹

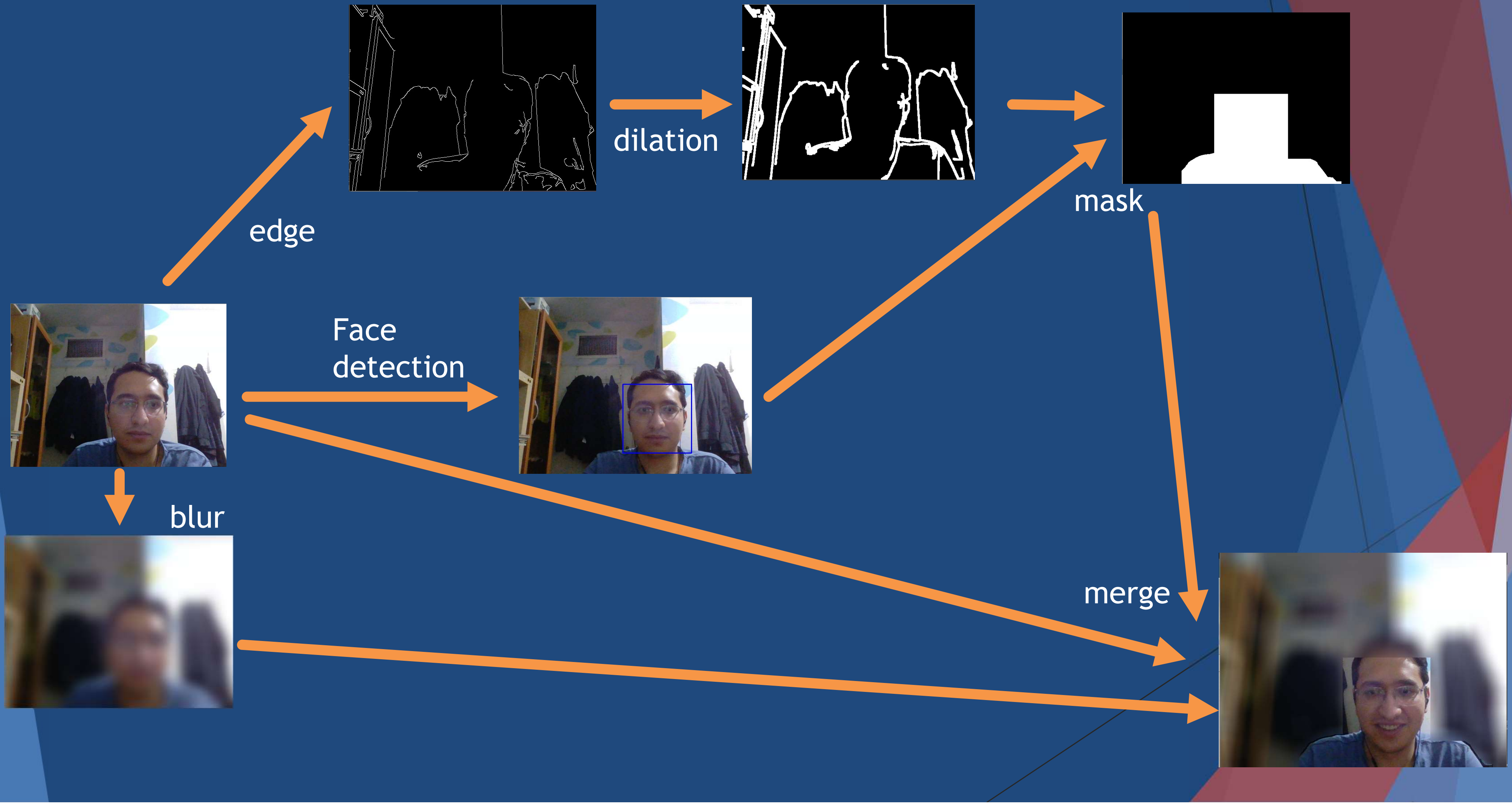
مقدمه: اهداف پروژه

فهرست بخش ها

- بخش ۱ : اجرایی از برنامه
- بخش ۲ : گام های حل مساله
- بخش ۳ : توضیحاتی در باب الگوریتم و مقاله های مورد استفاده
- بخش ۴ : کتابخانه های مورد استفاده و توضیح برنامه پروژه

بخش ۱ : اجرایی از برنامه

بخش ۲ : گام های حل مساله



edge

dilation

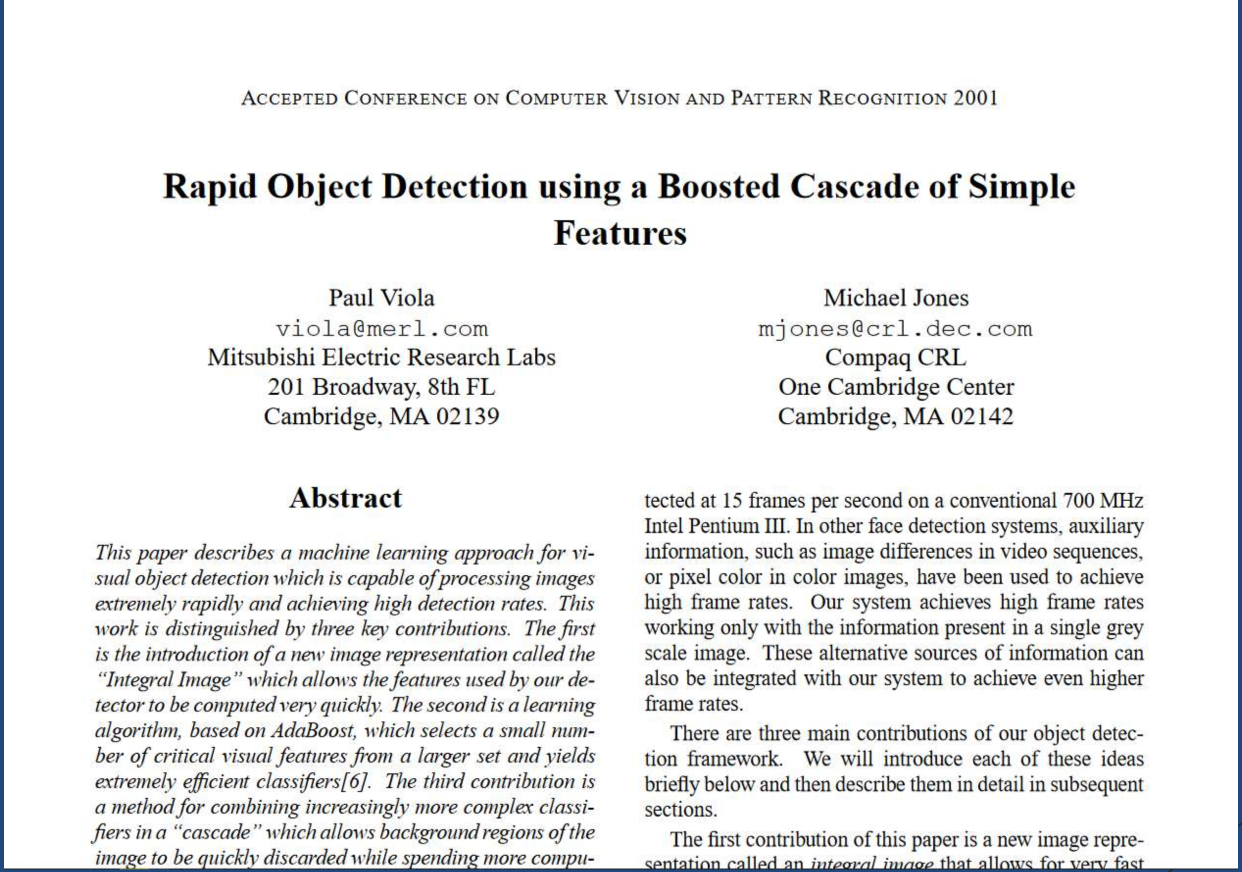
mask

Face detection

blur

merge

بخش ۳ : توضیحاتی در باب الگوریتم و مقاله های مورد استفاده



Accepted Conference on Computer Vision and Pattern Recognition 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

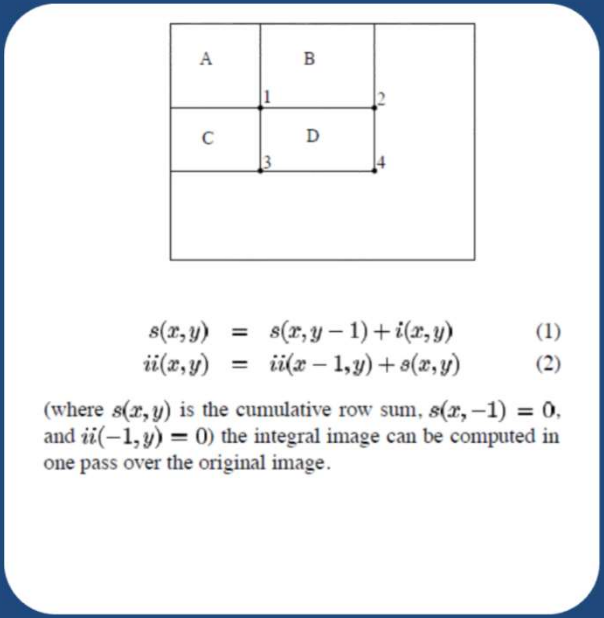
This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers[6]. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more compu-

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates.

There are three main contributions of our object detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections.

The first contribution of this paper is a new image representation called an *integral image* that allows for very fast

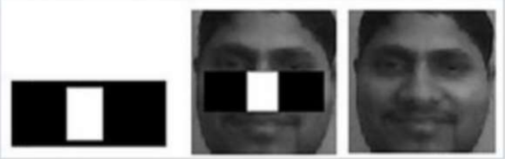
https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf



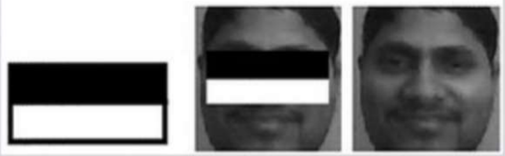
$$s(x,y) = s(x,y-1) + i(x,y) \qquad (1)$$

$$ii(x,y) = ii(x-1,y) + s(x,y) \qquad (2)$$

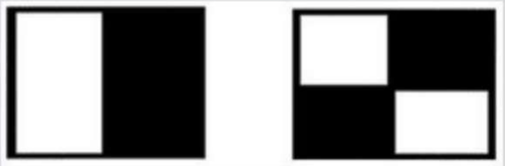
(where $s(x,y)$ is the cumulative row sum, $s(x,-1) = 0$, and $ii(-1,y) = 0$) the integral image can be computed in one pass over the original image.



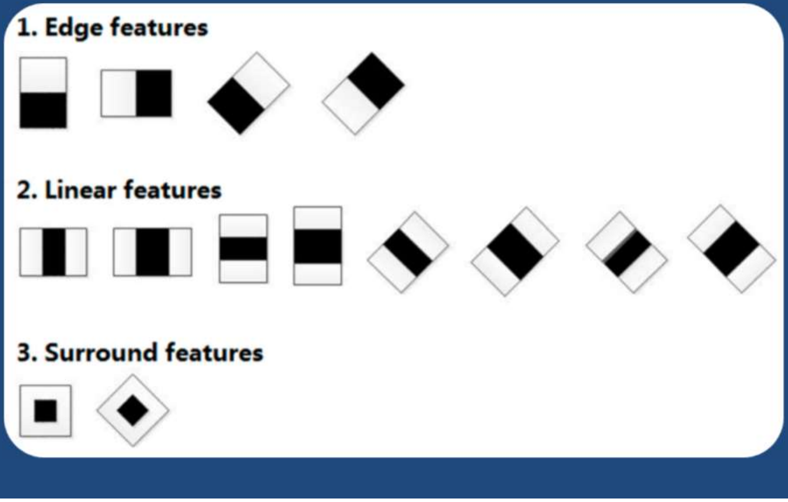
Haar Feature that looks similar to the bridge of the nose is applied onto the face



Haar Feature that looks similar to the eye region which is darker than the upper cheeks is applied onto a face



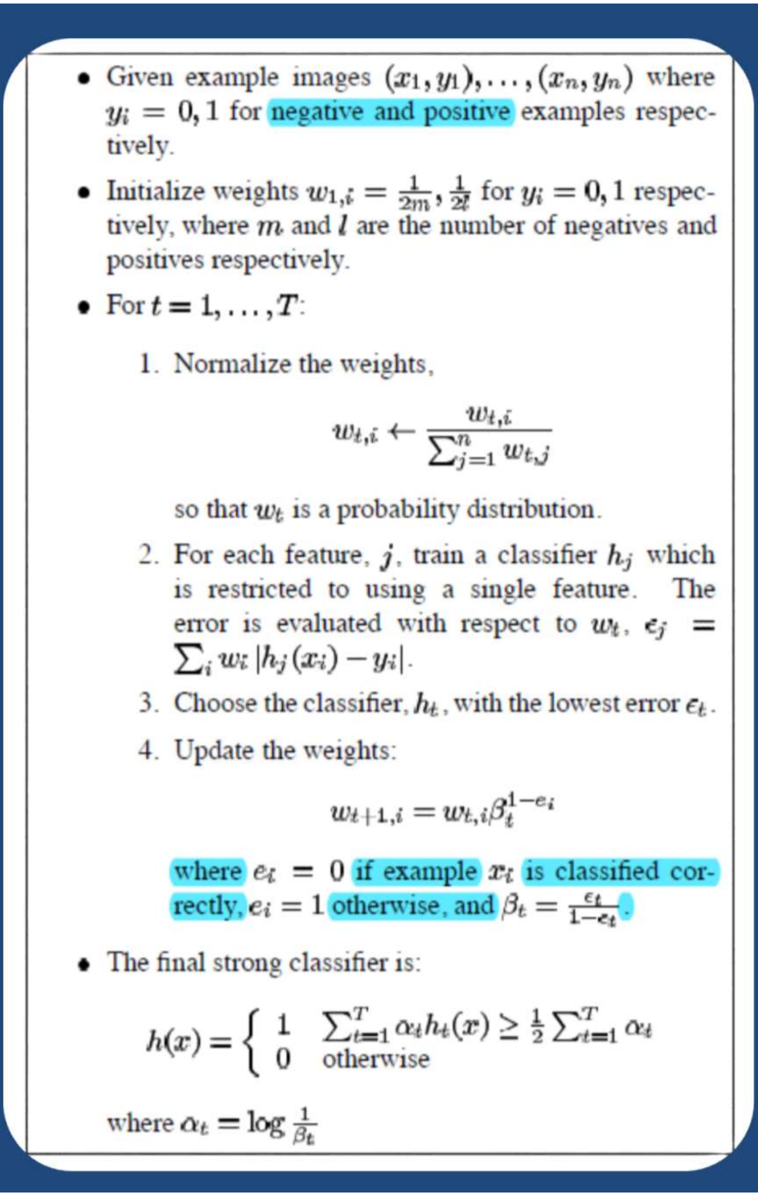
3rd and 4th kind of Haar Feature



1. Edge features

2. Linear features

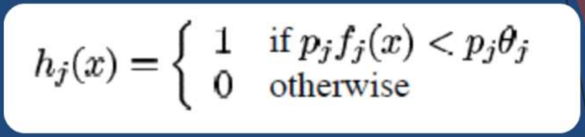
3. Surround features



- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

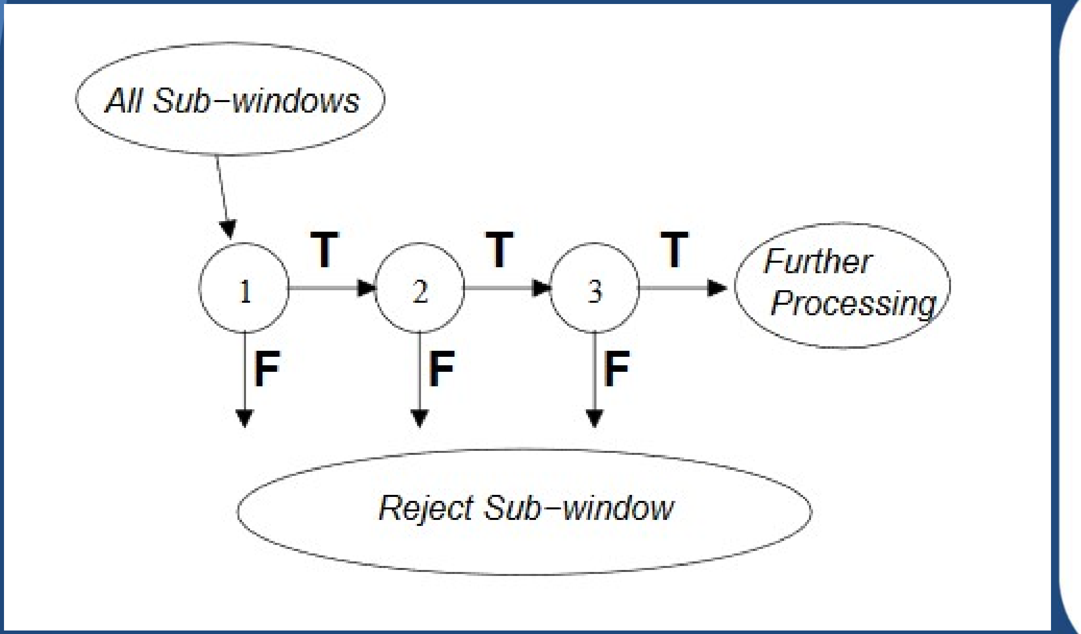
where $\alpha_t = \log \frac{1}{\beta_t}$

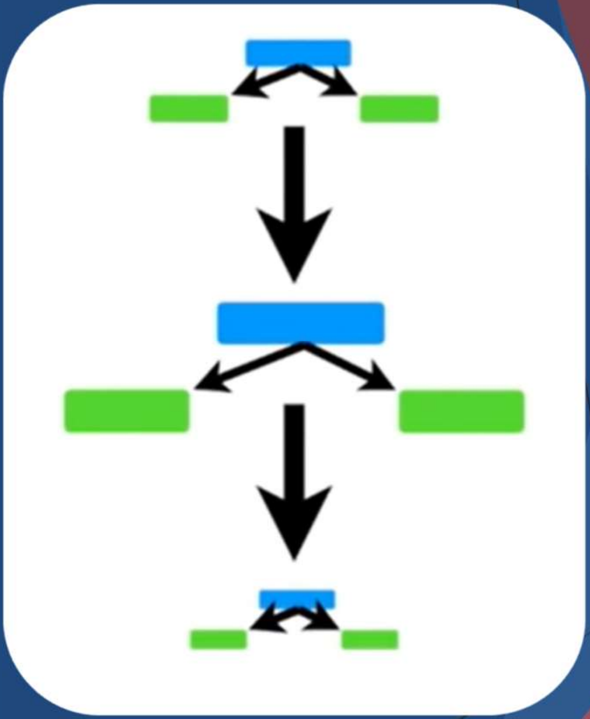


$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

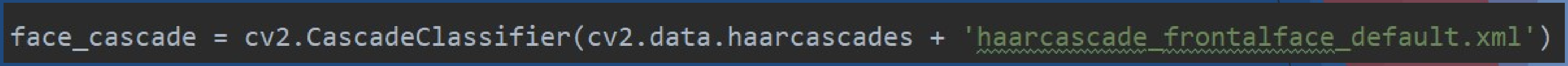
Weak Classifier

AdaBoost

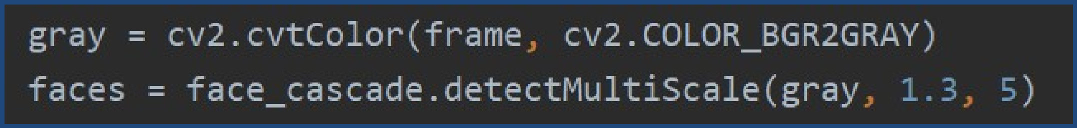

All Sub–windows
1
T
2
T
3
T
Further Processing
F
F
F
Reject Sub–window



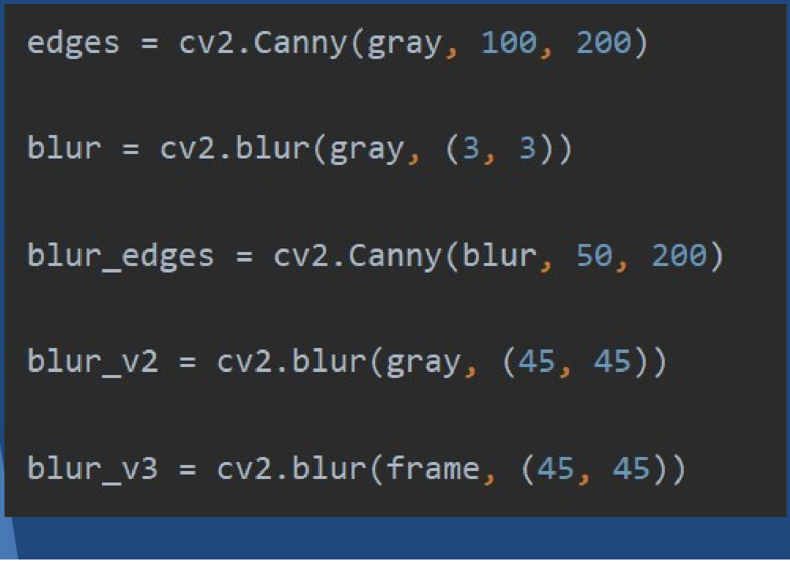
بخش ۴ : کتابخانه های مورد استفاده و توضیح برنامه پروژه



```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

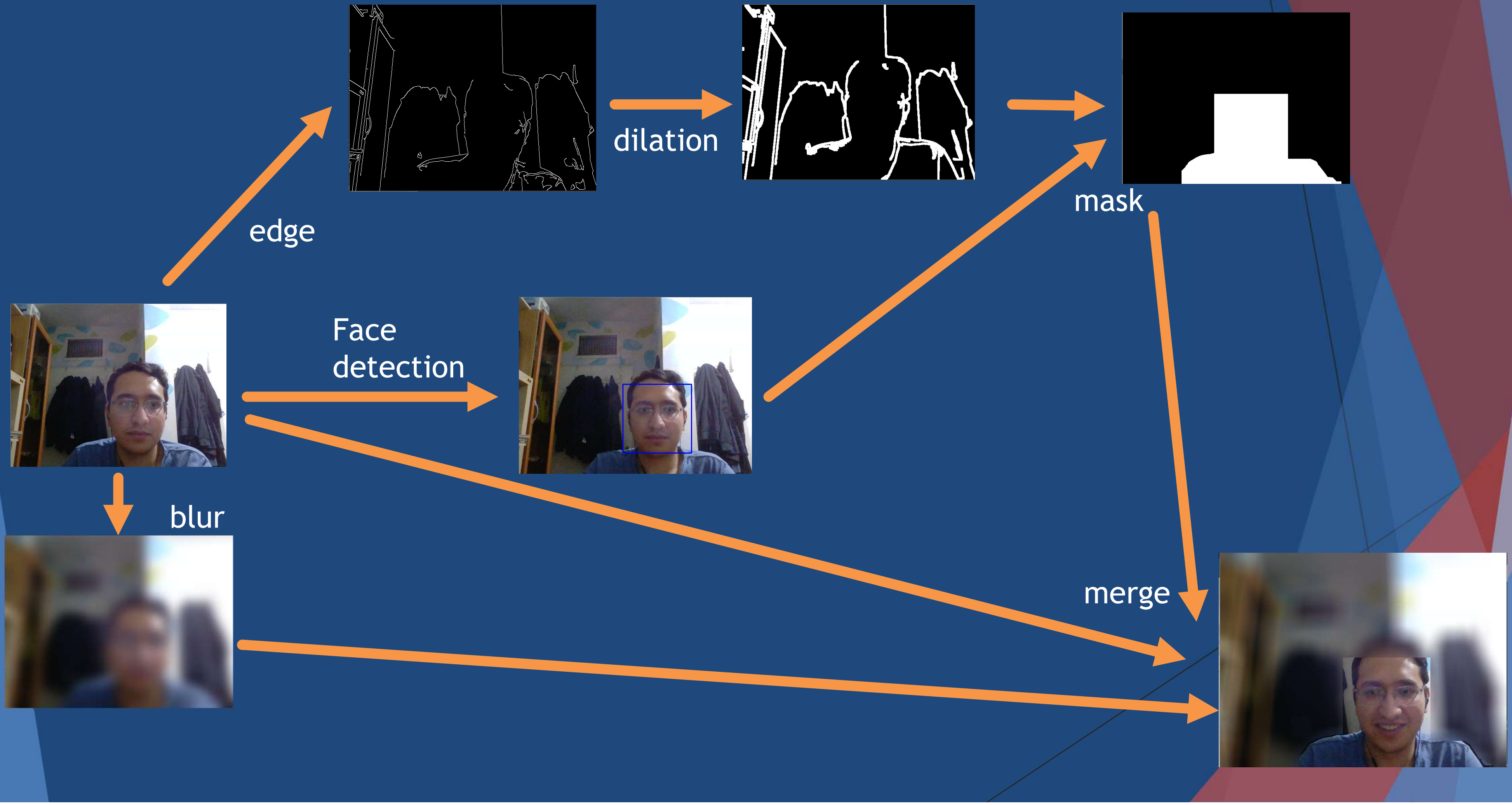


```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```



```
edges = cv2.Canny(gray, 100, 200)

blur = cv2.blur(gray, (3, 3))

blur_edges = cv2.Canny(blur, 50, 200)

blur_v2 = cv2.blur(gray, (45, 45))

blur_v3 = cv2.blur(frame, (45, 45))
```

بخش ۴ : کتابخانه های مورد استفاده و توضیح برنامه پروژه



edge

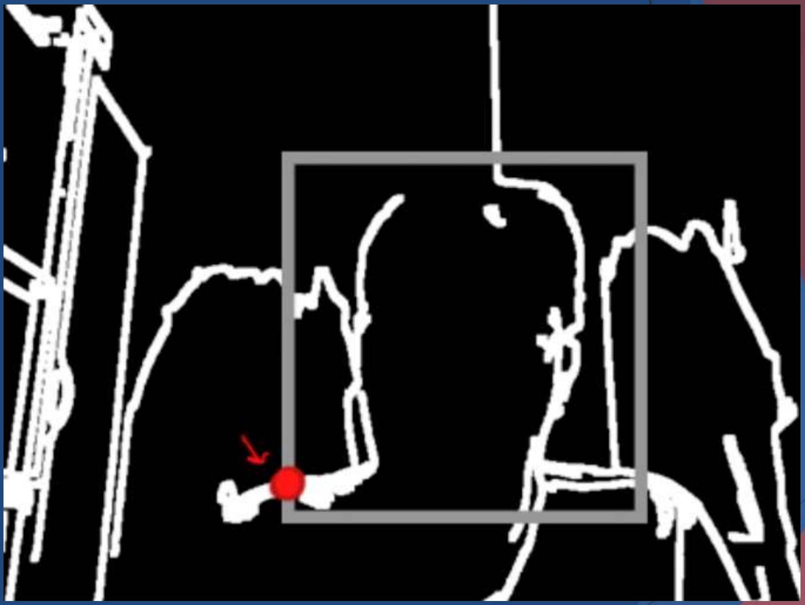
dilation

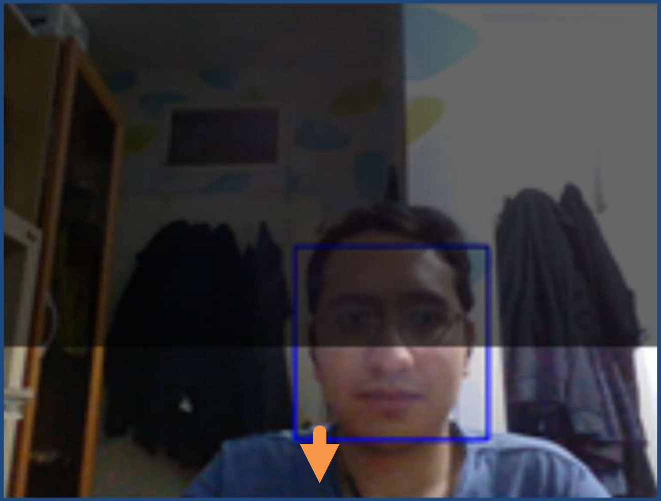
mask

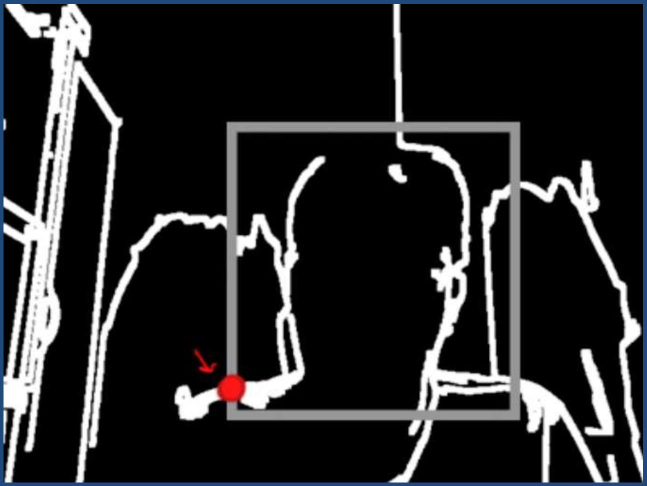
Face
detection

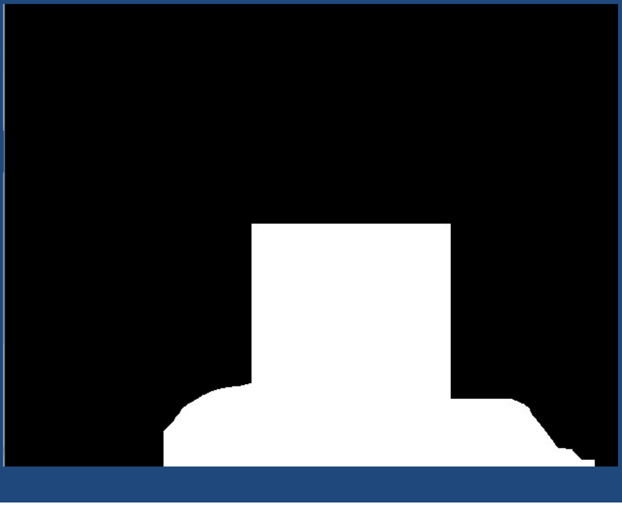
blur

merge









ممنونیم از توجهتون

Any Questions?