


# Детекција и анонимизација на лице



Дигитално процесирање на слика



# Главен Проблем

Детекција и замаглување на лице

## Потпроблеми

1. Детекција на лице
2. Замаглување на регионот на детектираното лице

# Модели за детекција на лице

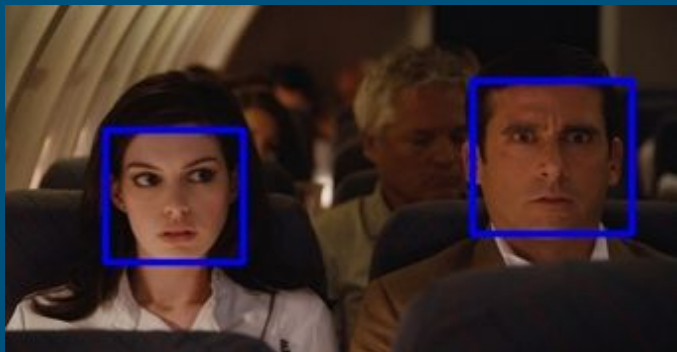
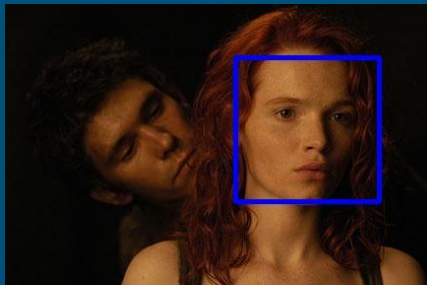
---

1. Базиран на машинско учење - **Haar Cascade** класификатор
  - Трансформирање на сликата во црно бела репрезентација
  - Haar карактеристики (Haar features)
  - Интегрална слика (Integral image)
  - AdaBoost
  - Каскаден класификатор
2. Базиран на длабоко учење - **DNN Face Detector** од DNN модулот во OpenCV библиотеката
  - [Res10\\_300x300\\_ssd\\_iter\\_140000.caffemodel](#)
  - ResNet10
  - Single Shot Multibox Detector (SSD)

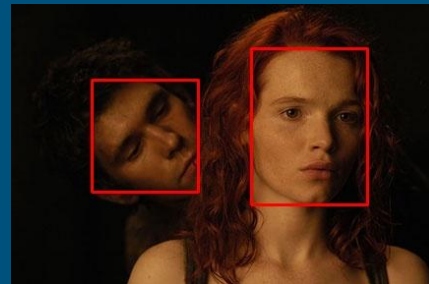
Користено е истото податочно [множество](#)

# Haar Cascade класификатор и DNN Face Detector

## - Haar Cascade класификатор



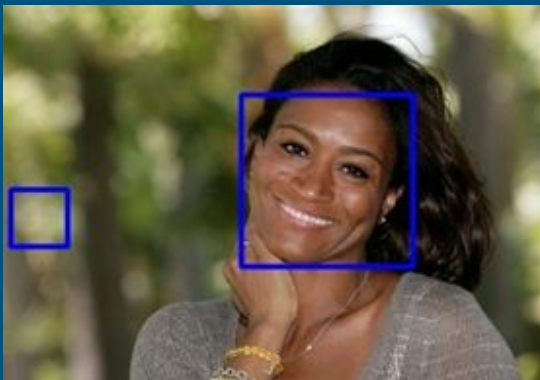
## DNN Face Detector



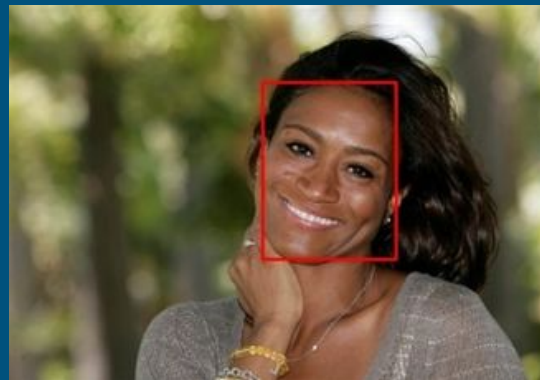
Haar Cascade класификатор - Не прецизен за детекција на лица, особено кога лицата се ротирани, замаглени или имаат различно осветлување.

# Лажни позитивни резултати од Haar Cascade

## - Haar Cascade

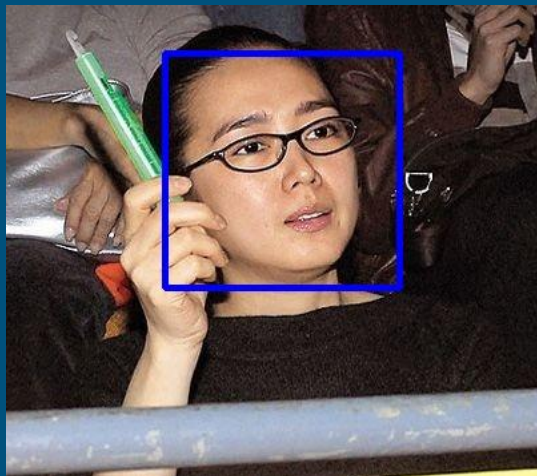


## DNN Face Detector

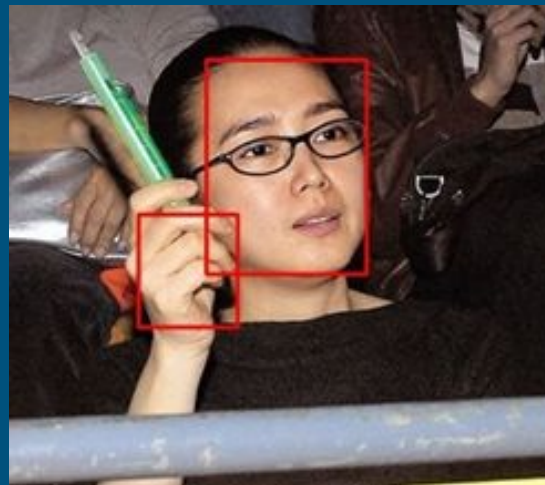


# Лажни позитивни резултати од DNN Face Detector

## Haar Cascade

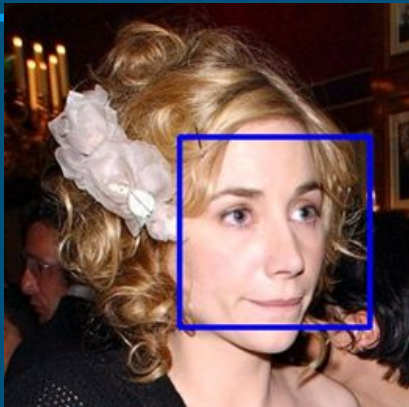


## DNN Face Detector

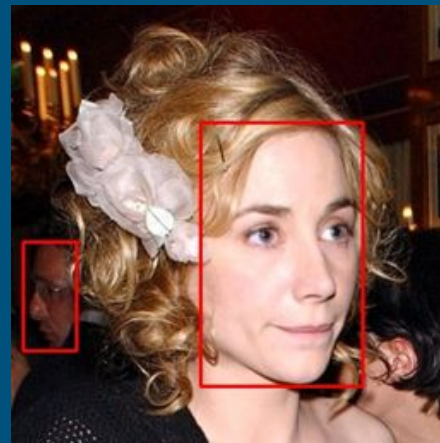


DNN Face Detector нуди подобра точност и  
флексибилност.

## Haar Cascade



## DNN Face Detector



Haar Cascade	Детекторот за лице DNN (Res10_300x300 SSD )
<p>Идеален за:</p> <ul style="list-style-type: none"><li>- корисен за <b>едноставни и брзи апликации</b> каде што <b>ресурсите се ограничени</b></li></ul>	<p>Идеален:</p> <ul style="list-style-type: none"><li>- нуди <b>подобра точност и флексибилност.</b></li></ul>
<p>Недостаток:</p> <ul style="list-style-type: none"><li>- има <b>помала прецизност</b></li></ul>	<p>Недостаток:</p> <ul style="list-style-type: none"><li>- бара <b>повеќе процесорска моќ</b></li></ul>



# Методи за замаглување

## 1. Averaging blur или boxBlur - `cv2.blur(src, ksize, anchor, borderType)`

Пресметка:

$$\frac{1}{9}(43 + 45 + 48 + 49 + 47 + 46 + 43 + 48 + 46) = 46$$

Изворна слика

43	45	48	47	43	46	49
49	47	46	47	47	45	46
43	48	46	44	47	47	44
41	47	46	42	45	47	44
51	45	44	45	44	44	47
50	46	46	47	44	46	49
41	47	49	46	45	48	48
43	46	46	46	48	45	44
43	46	47	48	42	40	47
43	45	44	45	42	42	47
46	48	44	44	46	45	48

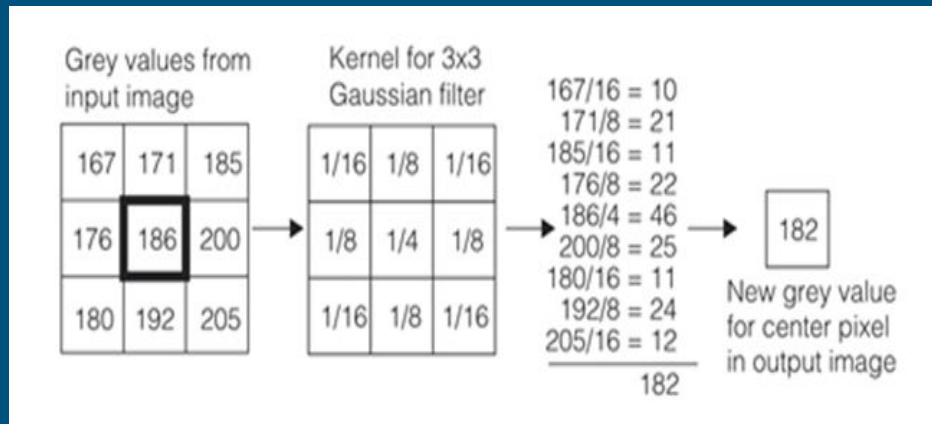
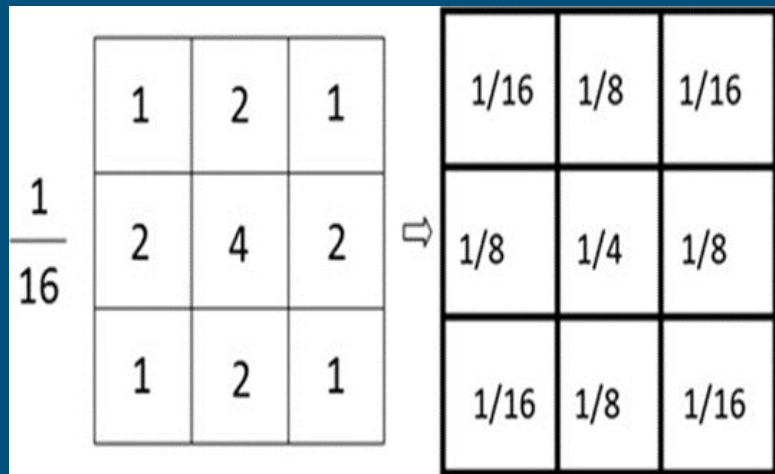
47	47	47	46	46	46	47
46	46	46	46	46	46	46
46	46	46	46	46	46	45
46	45	45	45	45	45	45
46	45	45	45	46	46	46
46	47	46	46	45	46	47
46	46	47	46	46	46	46
45	45	47	46	45	45	45
45	45	46	45	44	44	45
46	45	46	45	44	44	46
47	46	45	45	45	46	47

Филтер:

$$Kernel = \frac{1}{3 * 3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

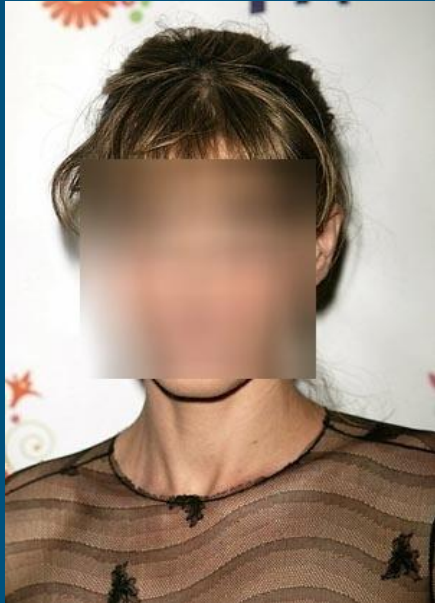
# Методи за замаглување

## 2. Gaussian blur - `cv2.GaussianBlur(src, ksize, sigmaX, dest, sigmaY, BorderType)`



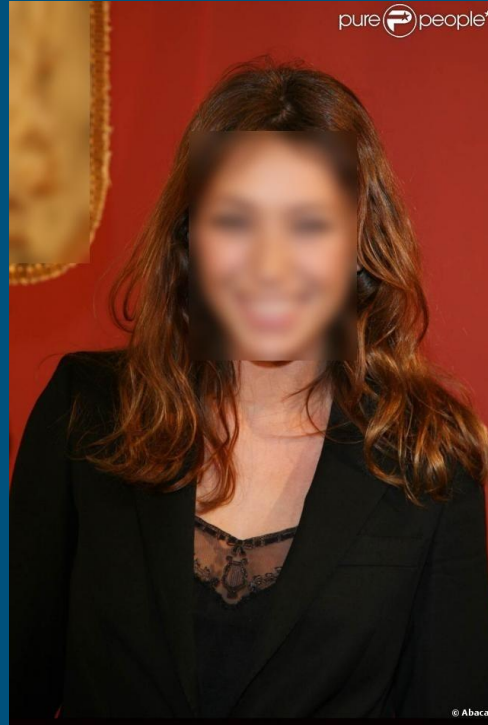
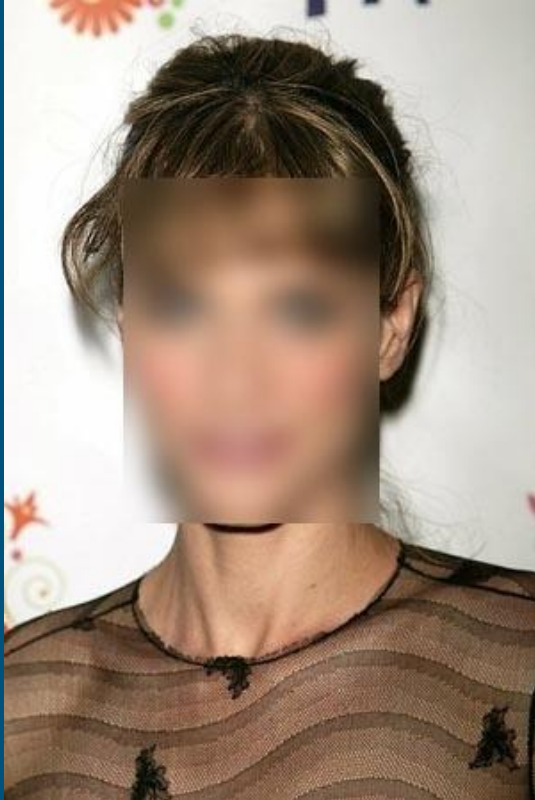
# Haar Cascade и Boxing blur

```
cv2.blur(face_roi, (51, 51))
```



# Детекторот за лице DNN (Res10\_300x300 SSD ) и Gaussian blur

```
cv2.GaussianBlur(face_roi, (51, 51), 0)
```



---

Ви благодарам на вниманието !

Изработено од:

Мила Кировски 223182