深蓝学院
shenlanxueyuan.com

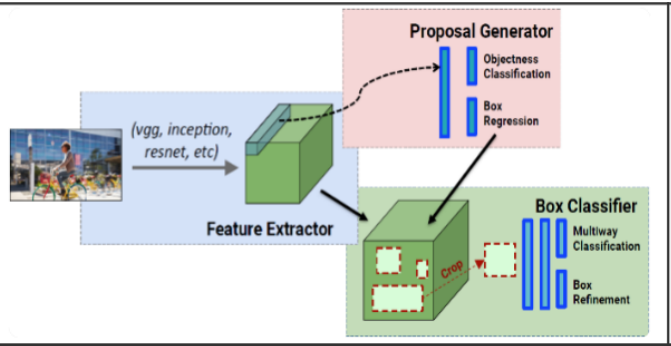
人脸检测

主讲人 张士峰

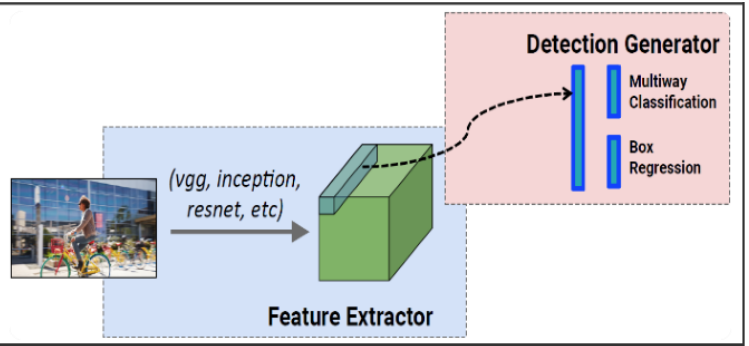
中国科学院自动化研究所
模式识别国家重点实验室

内容回顾：通用物体检测算法

基于锚框

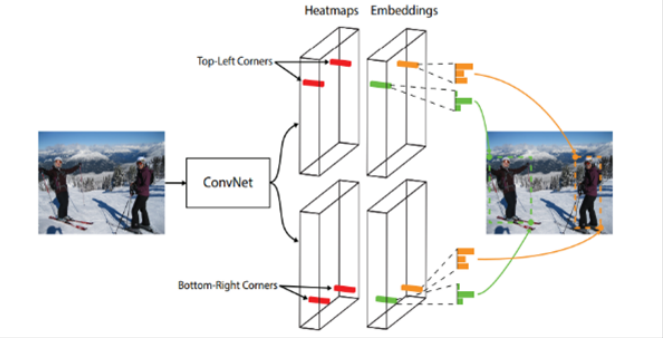


多阶段法

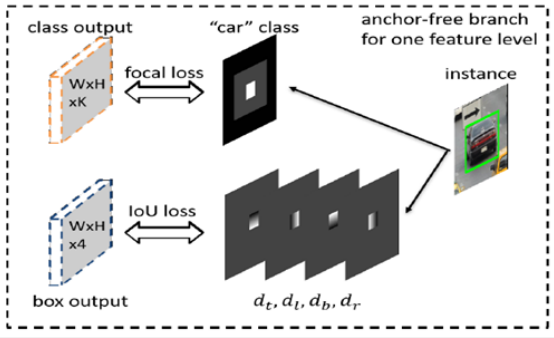


单阶段法

无需锚框



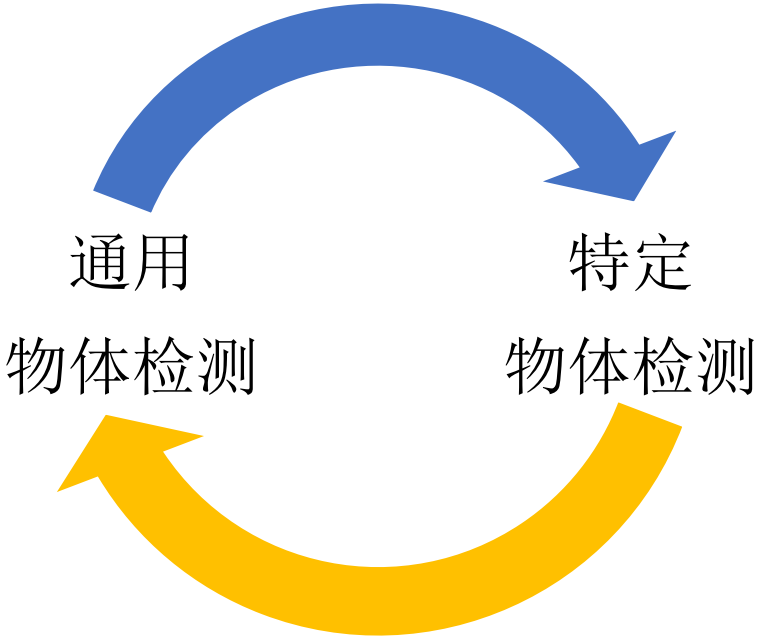
关键点法



中心域法

内容回顾：通用物体检测与特定物体检测

- 通用物体检测针对各类物体研究通用性的问题，为特定物体检测提供非常好的基础算法
- 特定物体检测利用特定物体的特殊性来解决特定问题，然后反哺通用物体检测

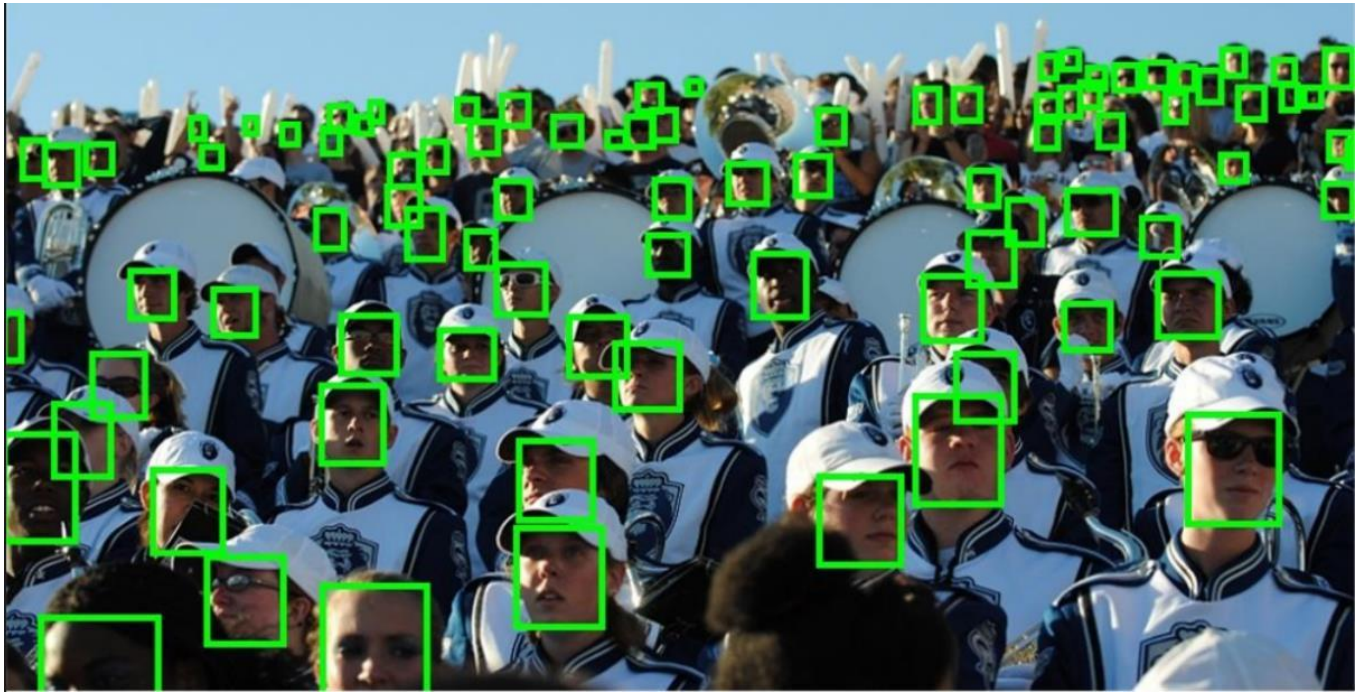


目录

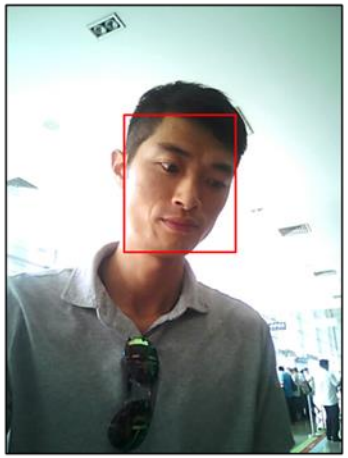
目录

人脸检测的定义

- 人脸检测：判断一副图像上是否存在人脸，如果存在，就给出所有人脸的位置

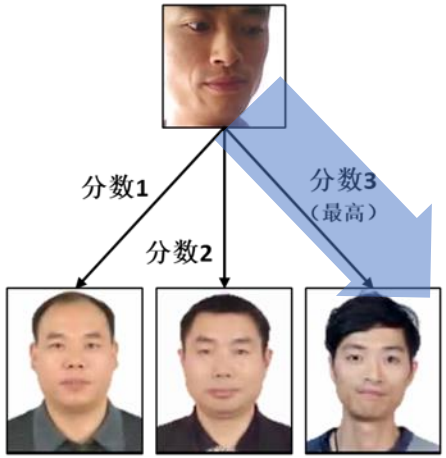


人脸识别系统



人脸检测

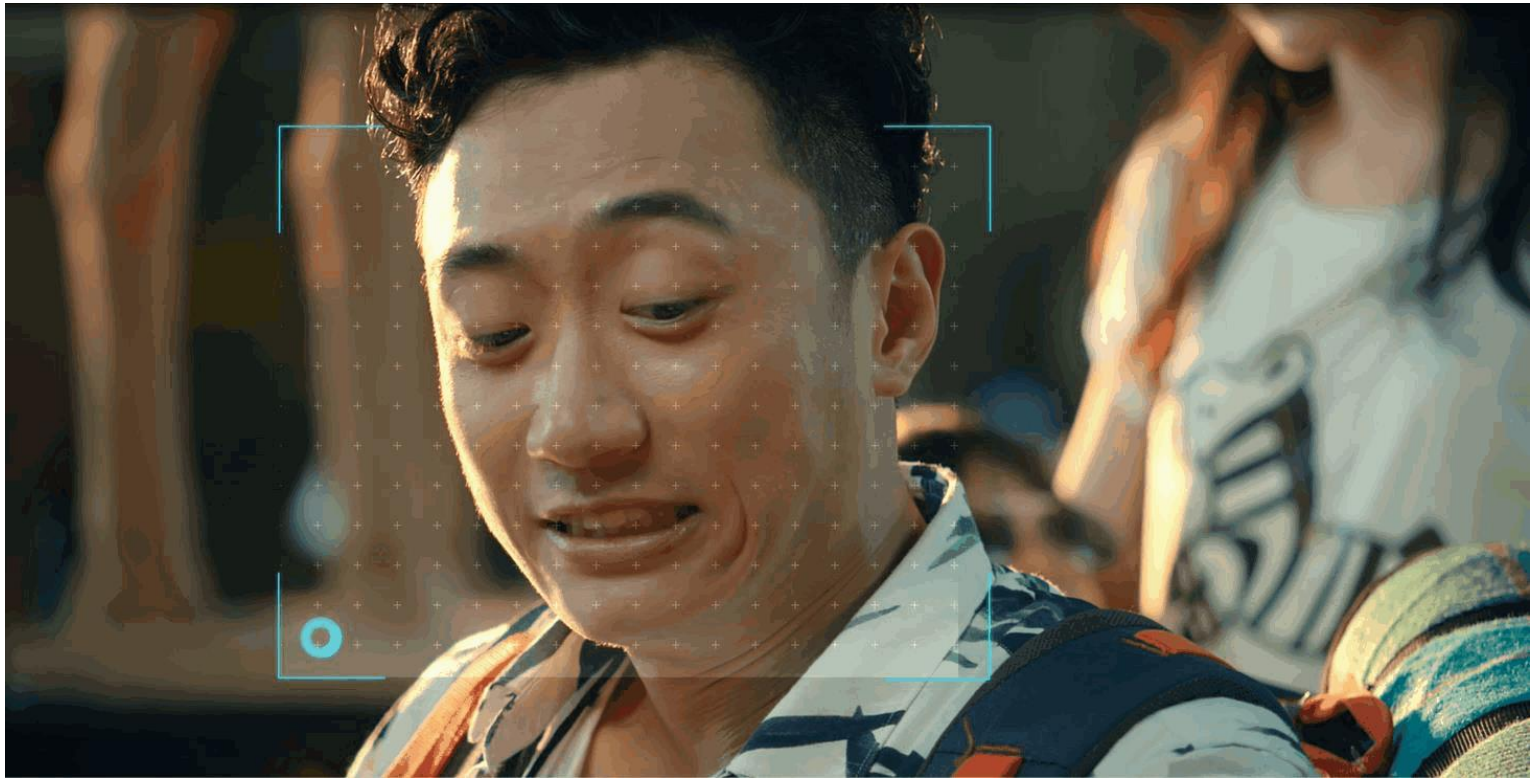
人脸关键点定位



人脸识别

人脸识别系统

人脸检测的应用

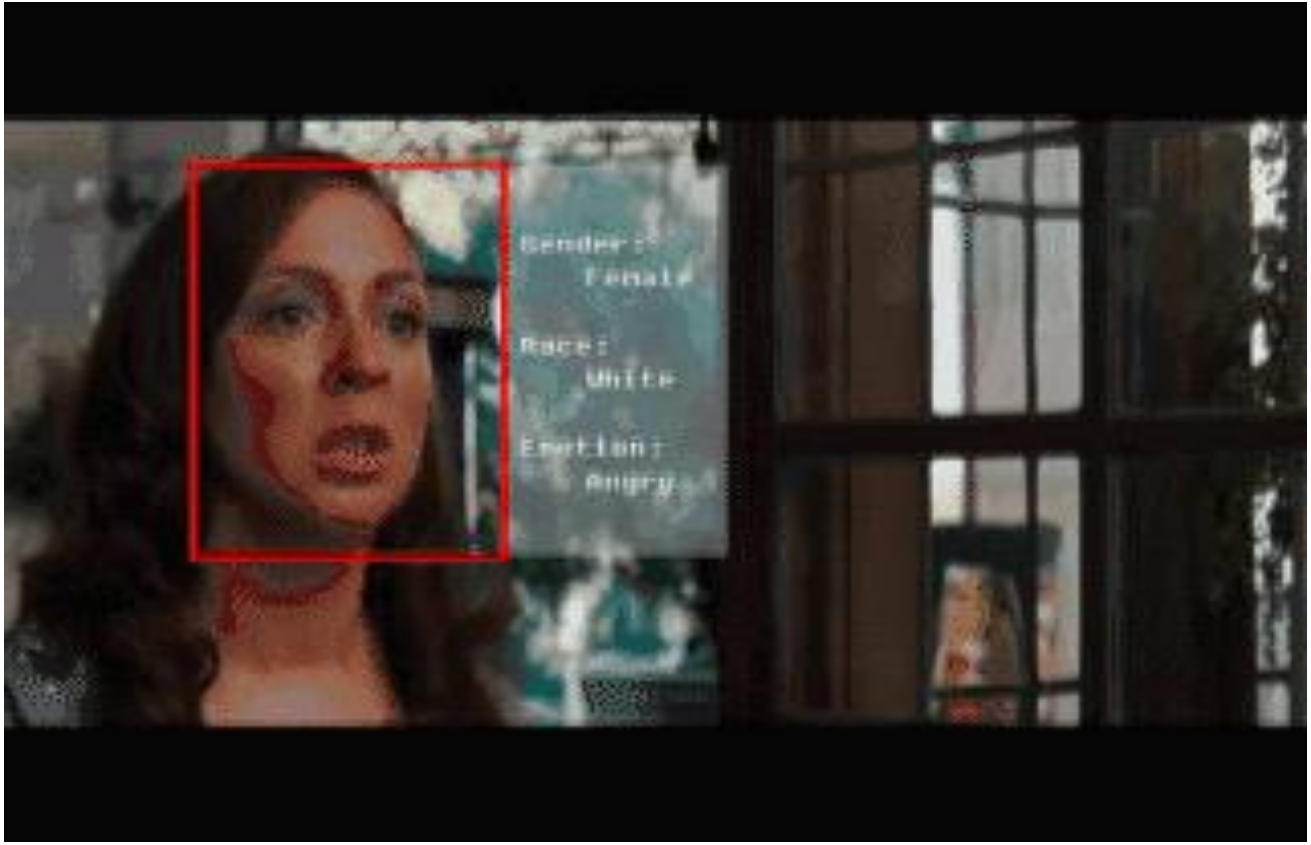


人脸入住

人脸检测的应用

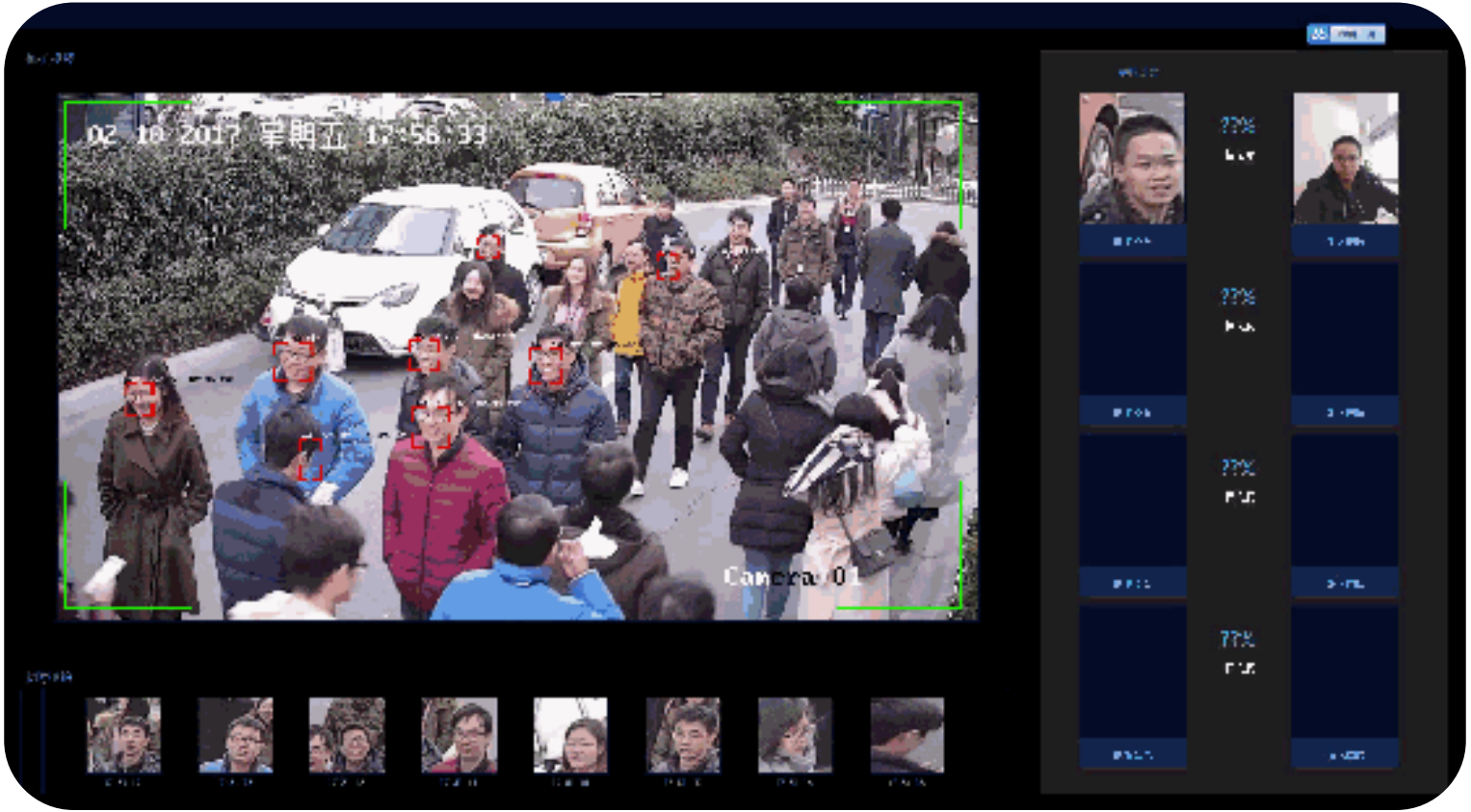
人脸解锁

人脸检测的应用



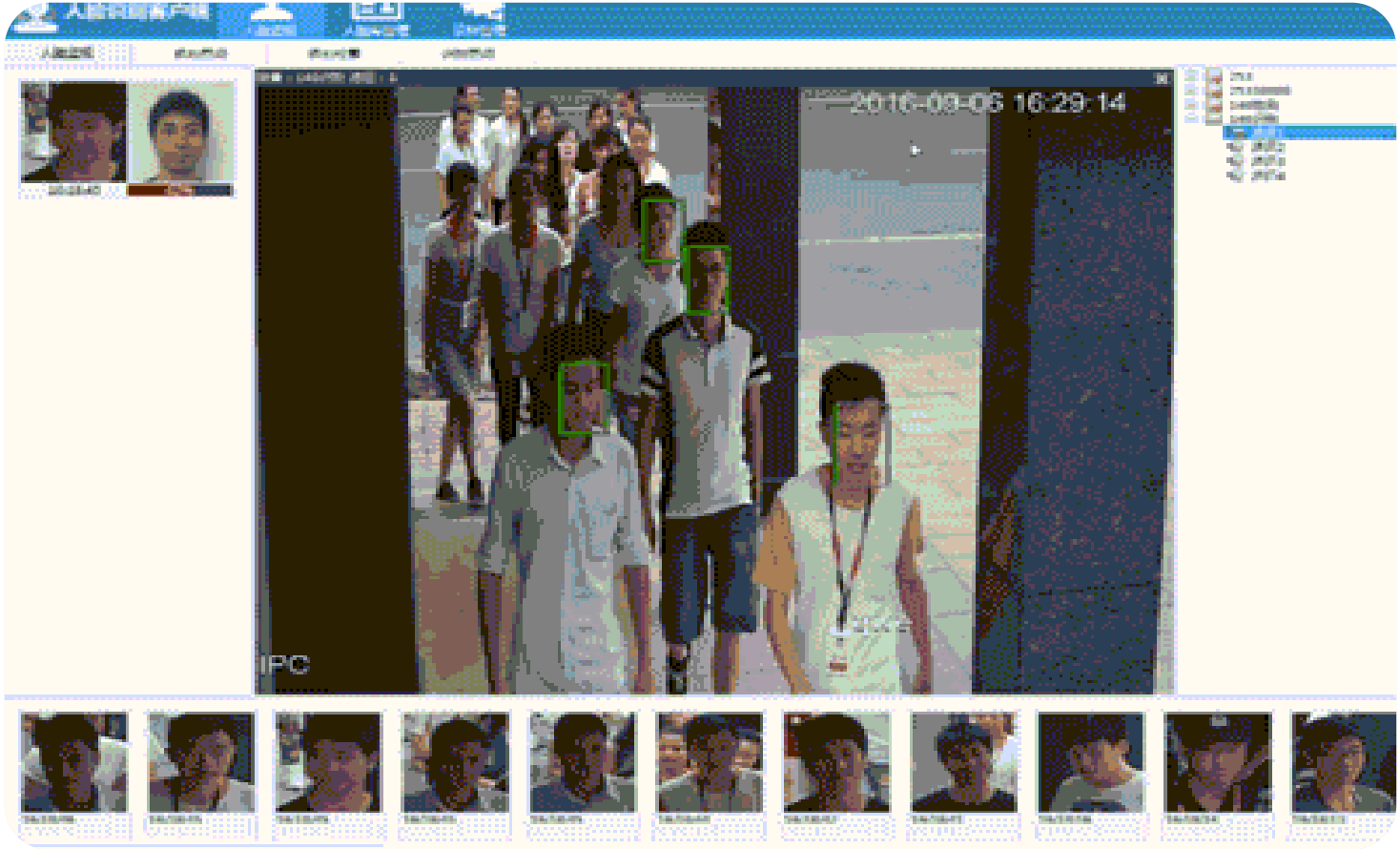
人脸属性分析

人脸检测的应用



视频监控

人脸检测的应用



出入口的人数统计

人脸检测数据集

仅是测试集

- **205**张图片、**473**张人脸
- 仅用于测试的数据集

AFW

- **851**张图片、**1335**张人脸
- 仅用于测试的数据集

PASCAL FACE

- **2845**张图片、**5171**张人脸
- 较流行的评测数据库

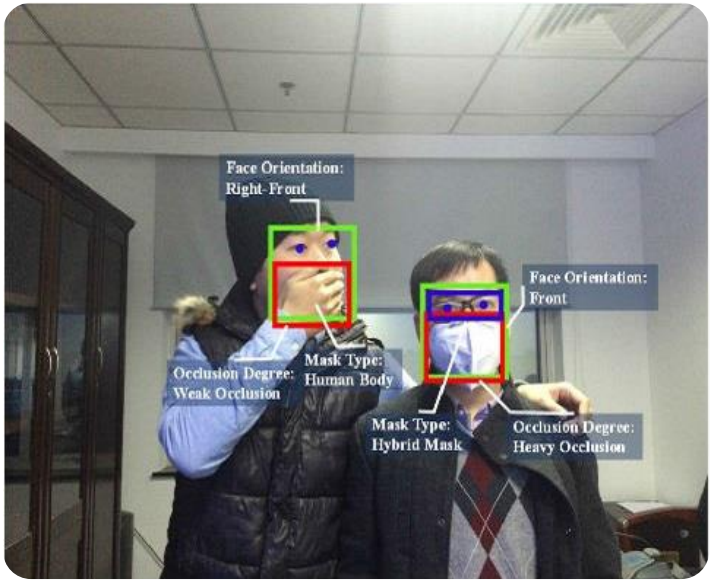
FDDB

人脸检测数据集

- MALF测试集：**5250**张图片、**11931**张人脸
- 5000张测试集，250张验证集

MALF

- **30881**张图，**35806**个人脸，遮挡人脸检测
- 训练集25876，测试集4935



MAFA

人脸检测数据集

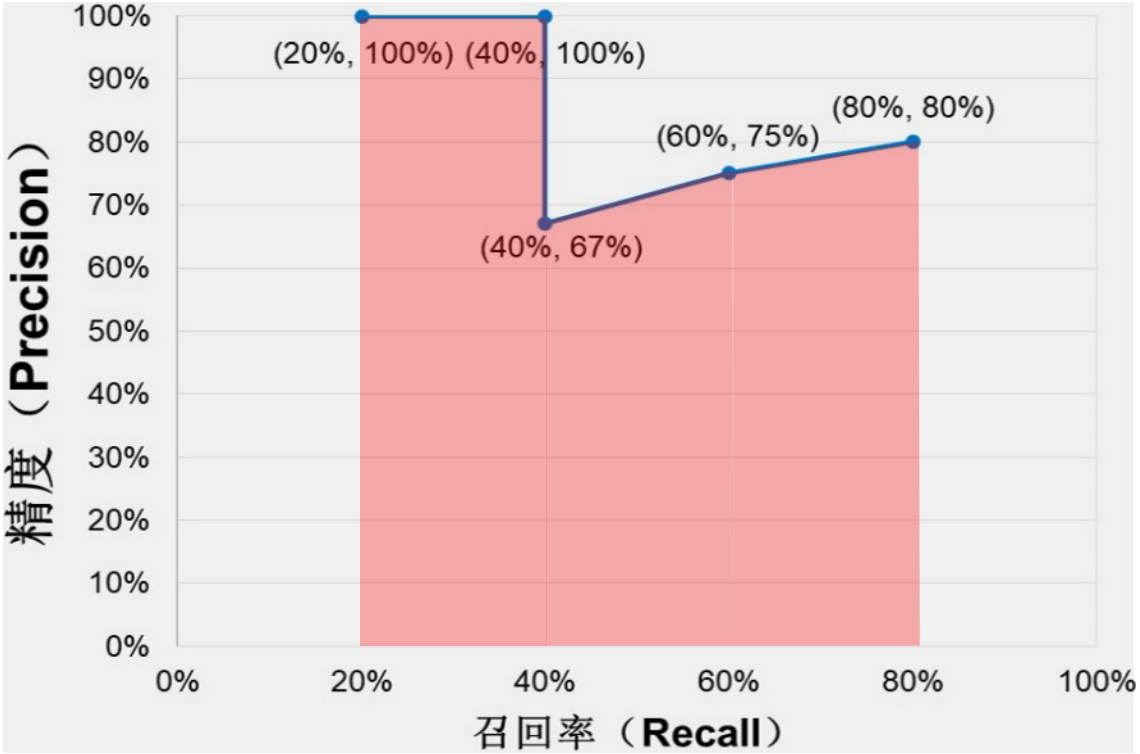
- **32203**张图片、**393703**张人脸
- 训练集12880、验证集3226、测试集16097
- 人脸检测最大最权威的数据集，难度非常大

WIDER FACE

人脸检测数据集

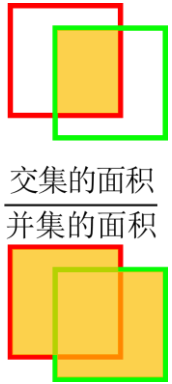
数据库	图片数量	人脸数量	测试集	验证集	训练集	难度指数
AFW	205	473	√	×	×	☆
PASCAL FACE	851	1335	√	×	×	☆☆
FDDB	2845	5171	√	×	×	☆☆
MALF	5250	11931	5000	250	×	☆☆☆
MAFA	30881	35806	4935	×	25876	☆☆☆
WIDER FACE	32203	393703	16097	3226	12880	☆☆☆☆☆

人脸检测评价指标：精度



平均精度（AP）

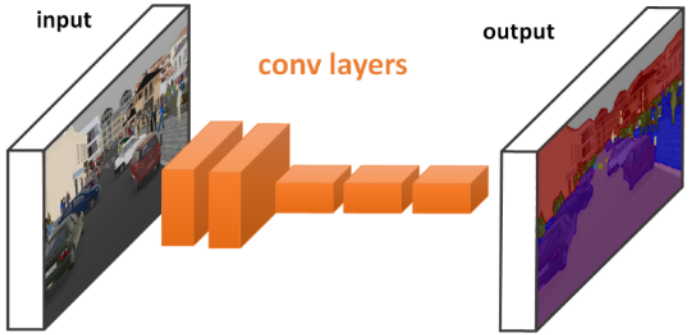
精度-召回率曲线下的面积



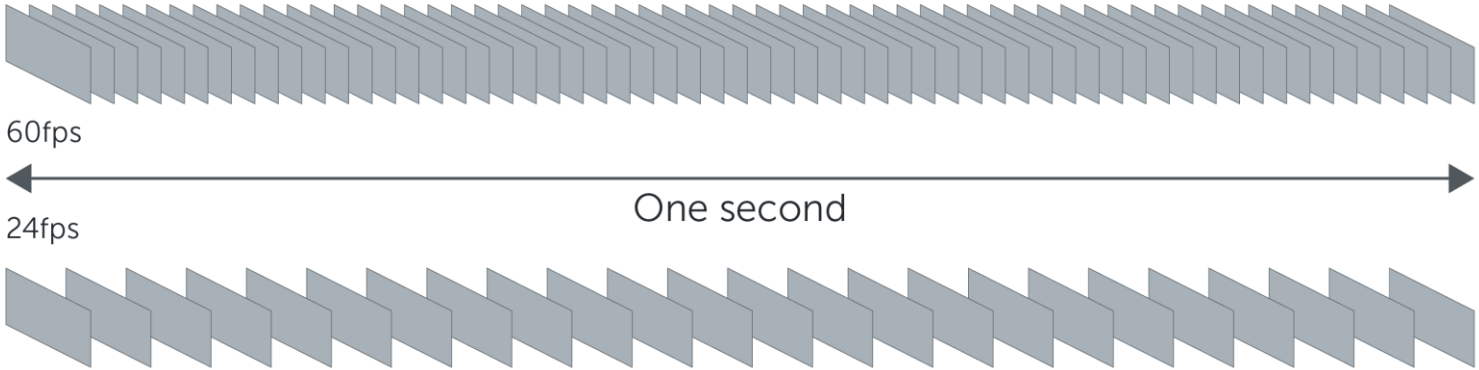
$$\text{交除并重叠比（}IoU\text{）} = \frac{\text{交集的面积}}{\text{并集的面积}} \geq 0.5$$

人脸检测评价指标：速度

- **前传耗时 (ms)**：从输入一张图像到输出最终结果所消耗的时间，这包括前处理耗时 (如图像归一化)、网络前传耗时、后处理耗时(如非极大值抑制)



- **每秒帧数 (FPS)**：每秒钟能处理的图像数量



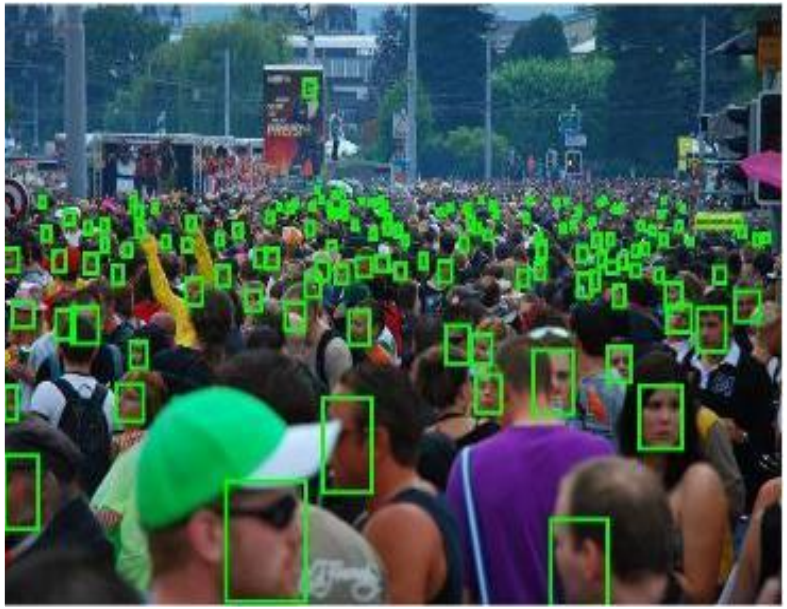
人脸检测的难点问题

- 人脸外观的复杂变化 -> 高精度

姿势　遮挡　表情　妆束　光照

人脸检测的难点问题

- 尺度和位置上的巨大搜索空间 -> 高效率



人脸检测的发展脉络

传统方法
- V-J人脸检测器

深度学习早期
- Cascade CNN

深度学习后期
- SSD, Faster R-CNN

目录

传统人脸检测算法

- 利用手工特征+分类器，以滑窗方式在图像金字塔上遍历所有位置和大小，进行人脸检测

遍历所有位置
滑窗方式

遍历不同大小
图像金字塔

传统人脸检测算法

- 利用手工特征+分类器，以滑窗方式在图像金字塔上遍历所有位置和大小，进行人脸检测

传统人脸检测算法

- 利用手工特征+分类器，以滑窗方式在图像金字塔上遍历所有位置和大小，进行人脸检测

- Viola-Jones人脸检测算法：著名的人脸检测算法、第一个商用检测算法、具备CPU实时速度

积分图	AdaBoost	级联分类器
• 用于快速提取Haar特征	• 用于特征选择	• 用于分类器设计

Rapid object detection using a boosted cascade of simple features, CVPR 2001

Viola-Jones人脸检测算法：检测流程

Viola-Jones人脸检测算法：检测流程

灰度化

Viola-Jones人脸检测算法：检测流程

灰度化

图像金字塔

图像缩放

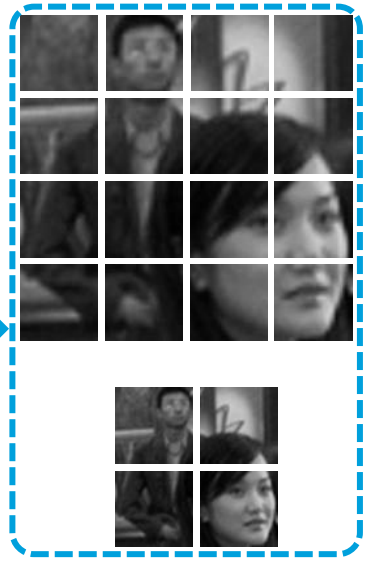
Viola-Jones人脸检测算法：检测流程

灰度化

图像金字塔

24x24
滑窗

20个24x24的子图像



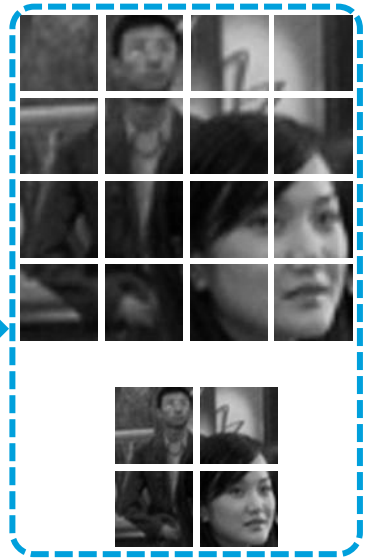
Viola-Jones人脸检测算法：检测流程

灰度化

图像金字塔

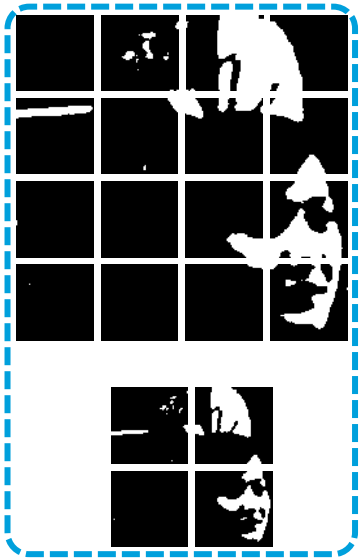
24x24
滑窗

20个24x24的子图像



特征提取

20个子图像的Haar特征



Viola-Jones人脸检测算法：检测流程

灰度化

图像金字塔

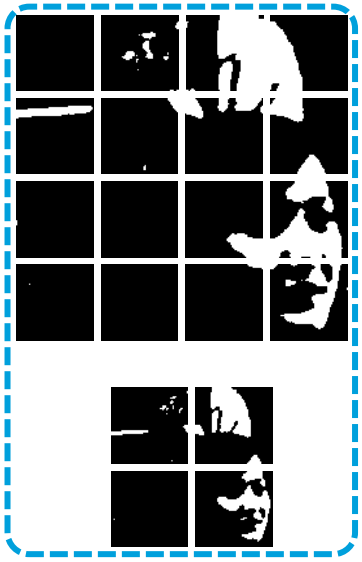
图像缩放

24x24
滑窗

20个24x24的子图像

特征提取

20个子图像的Haar特征



AdaBoost
分类器1

4个人脸候选

Viola-Jones人脸检测算法：检测流程

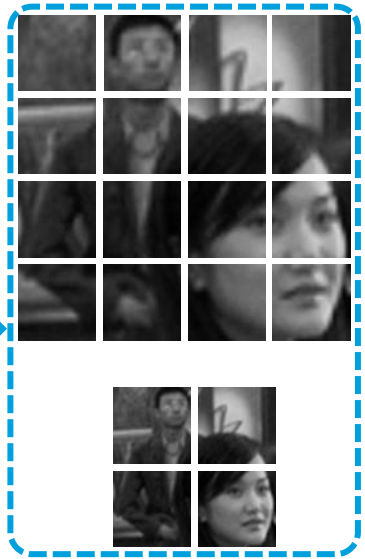
灰度化

图像金字塔

图像缩放

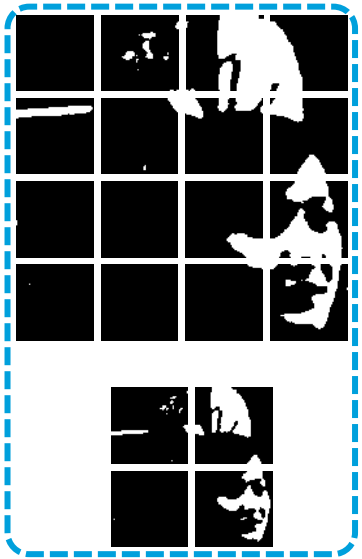
24x24
滑窗

20个24x24的子图像



特征提取

20个子图像的Haar特征



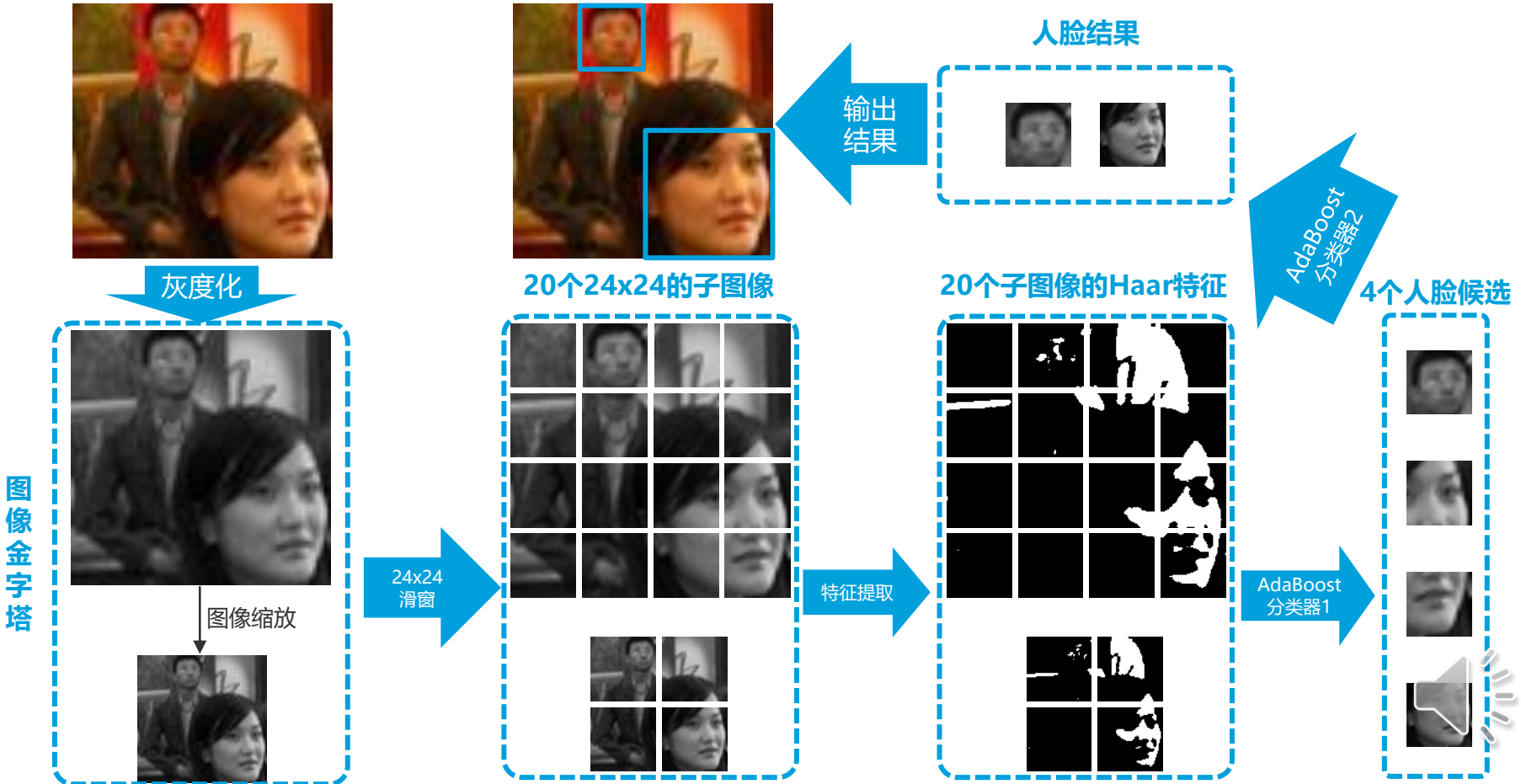
AdaBoost
分类器1

4个人脸候选

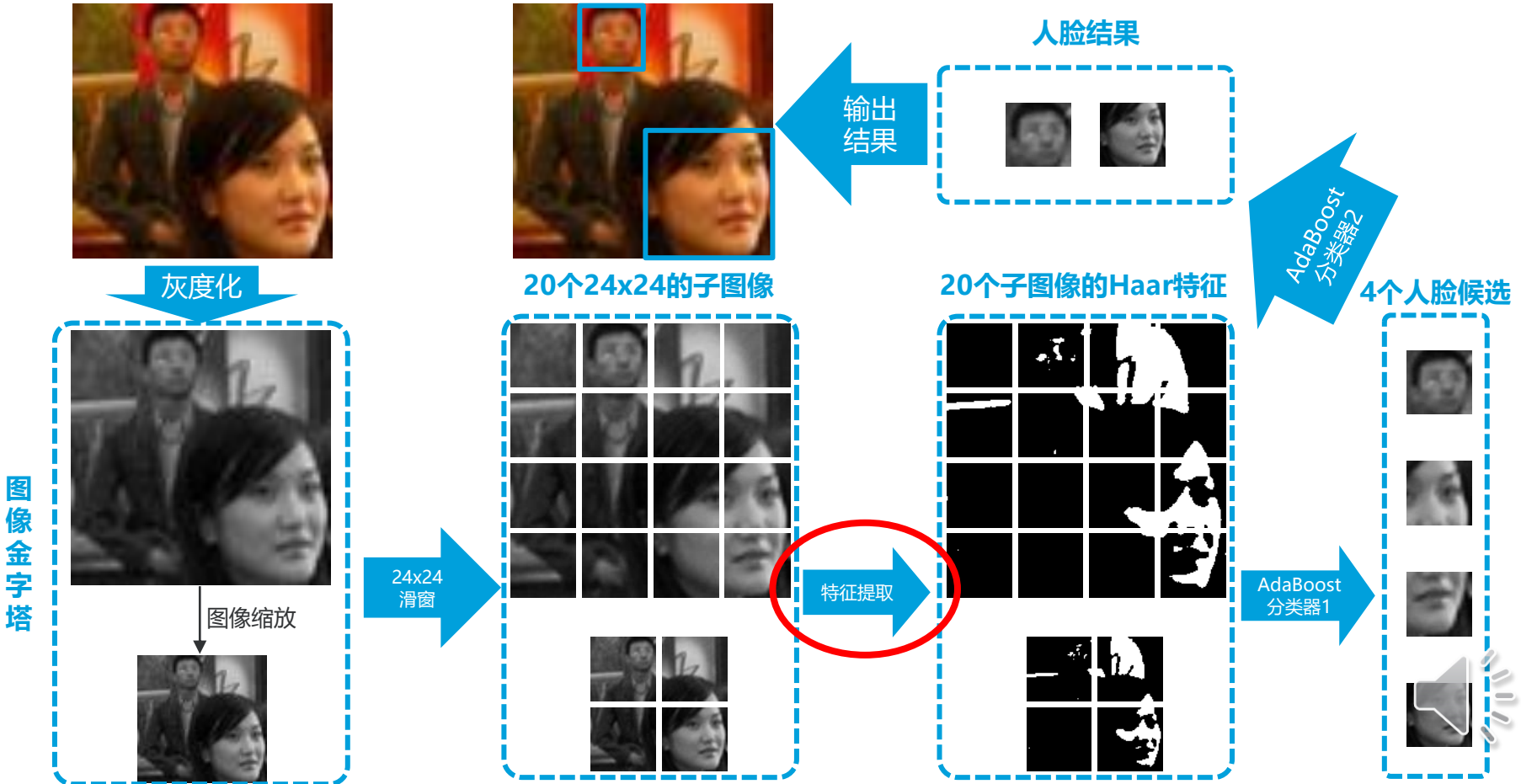
AdaBoost
分类器2

人脸结果

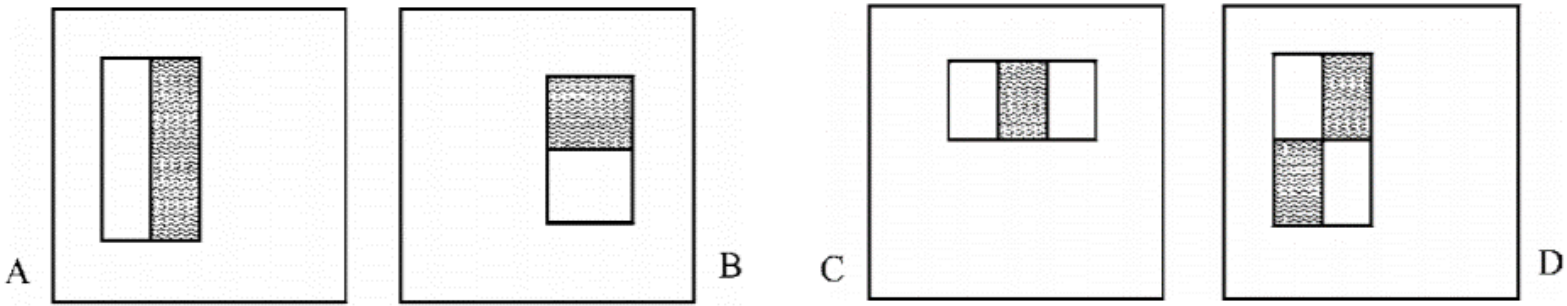
Viola-Jones人脸检测算法：检测流程



Viola-Jones人脸检测算法：检测流程

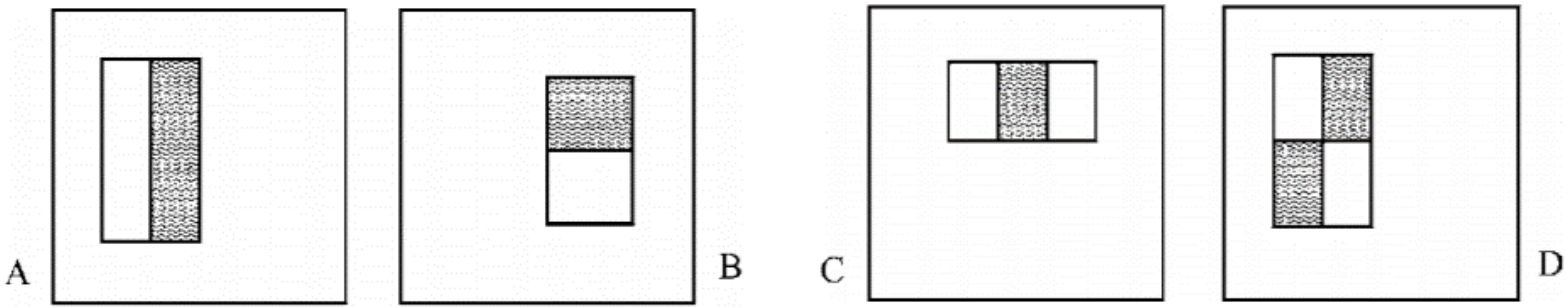


Viola-Jones人脸检测算法：Haar特征

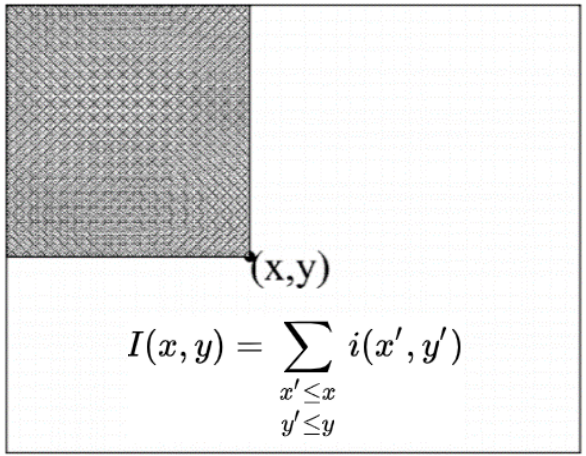


Haar特征模板

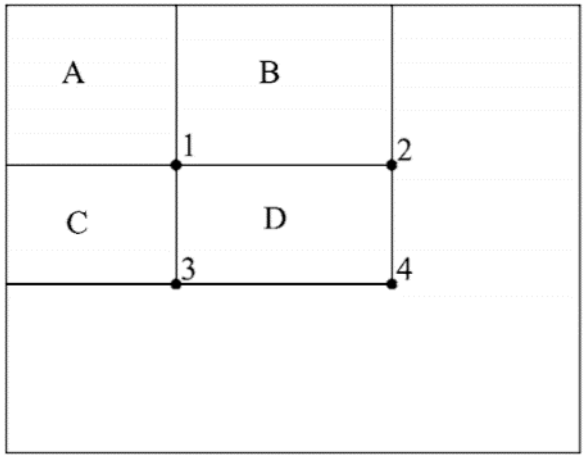
Viola-Jones人脸检测算法：Haar特征



Haar特征模板



积分图G



D的面积 = $G_4 + G_1 - G_3 - G_2$

Viola-Jones人脸检测算法：Haar特征

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

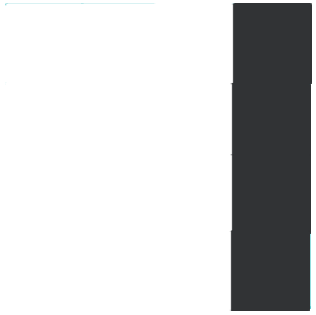
$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

1	3	5	9	10
4	10	13	22	25
6	15	21	32	39
10	20	31	46	59
16	29	42	58	74

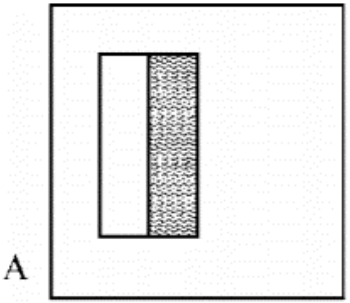
积分图的示例

Viola-Jones人脸检测算法：Haar特征

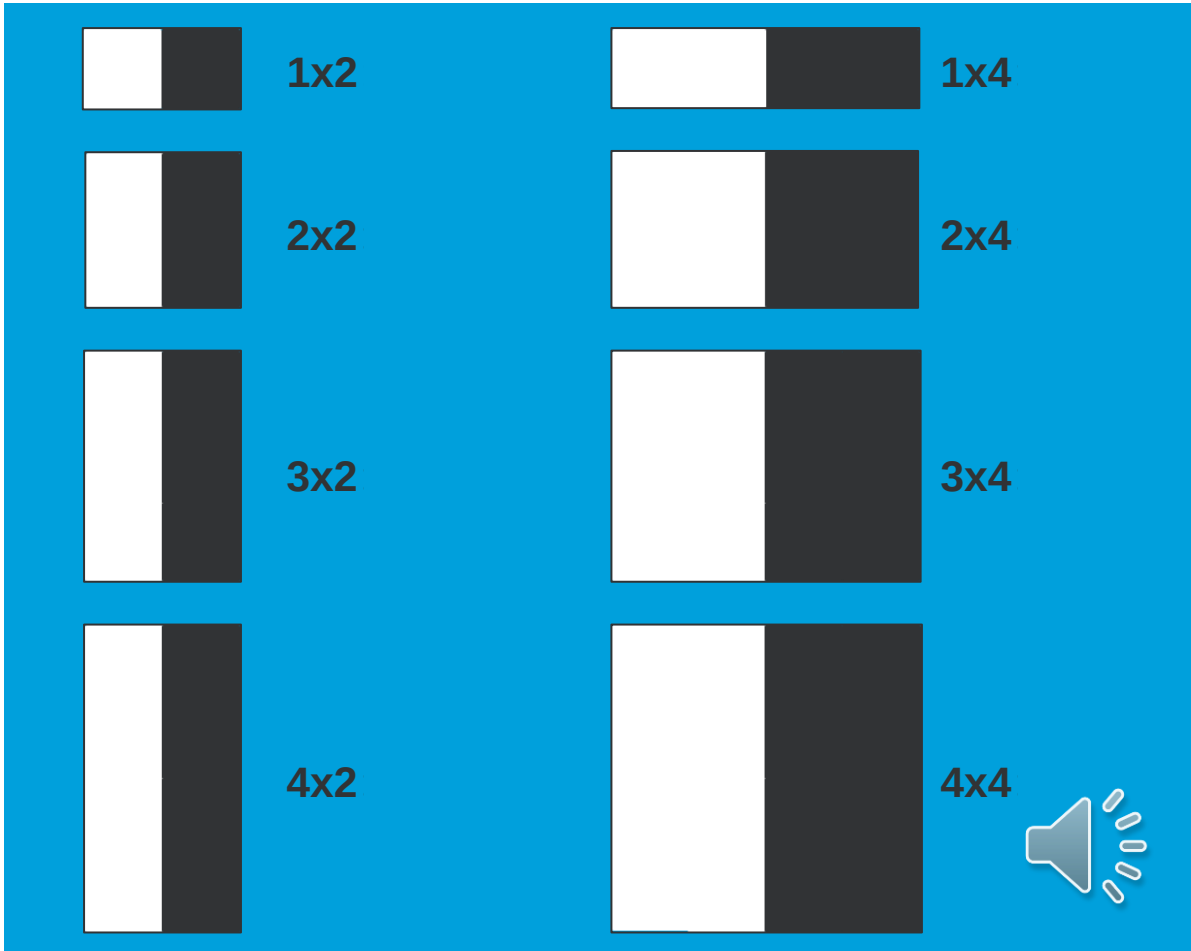
8个模板，40个Haar特征



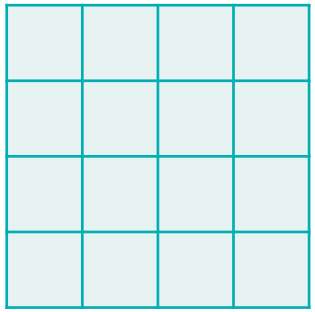
4x4图像



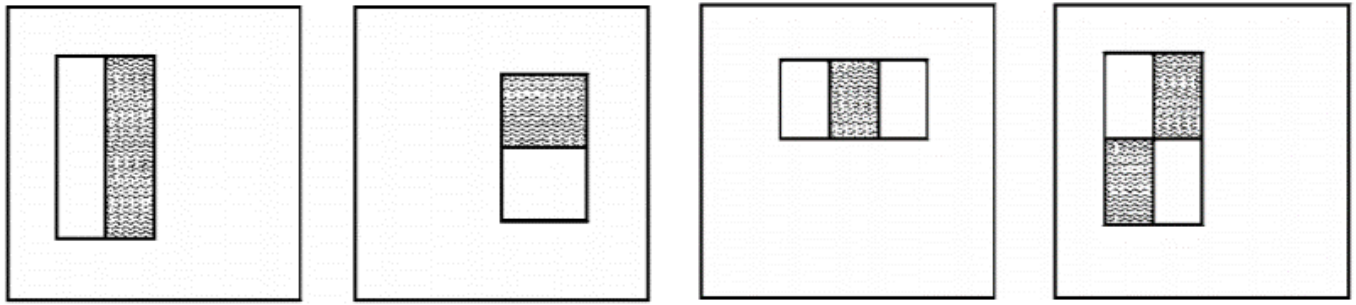
模板生成



Viola-Jones人脸检测算法：Haar特征



4x4图像

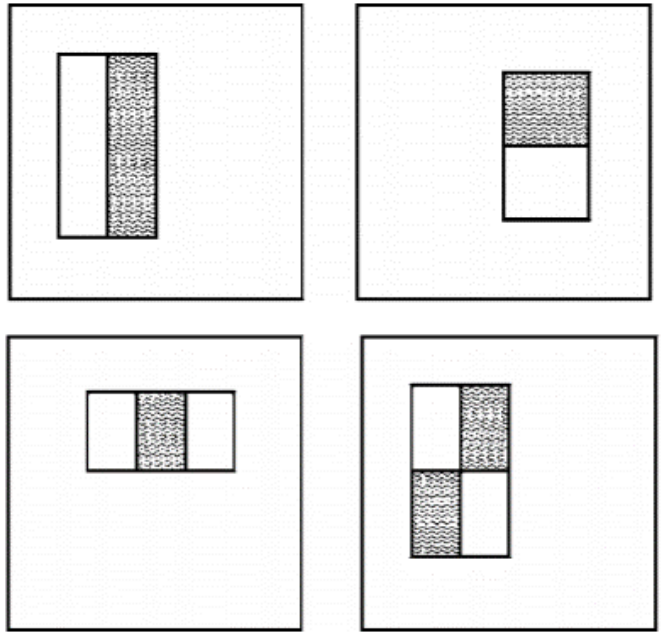


40个特征	40个特征	8个特征	10个特征

共计98个特征

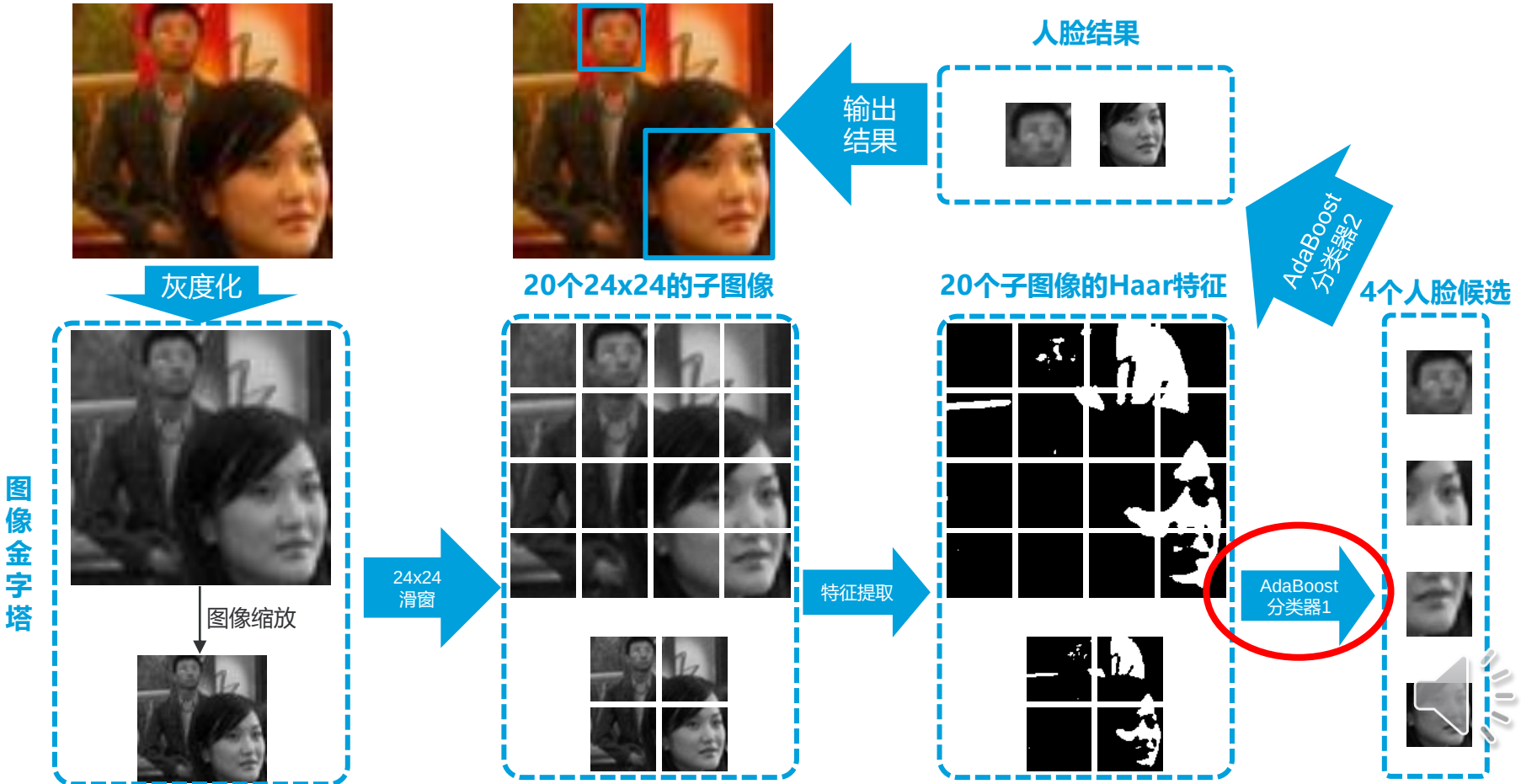
Viola-Jones人脸检测算法：Haar特征

24x24图像

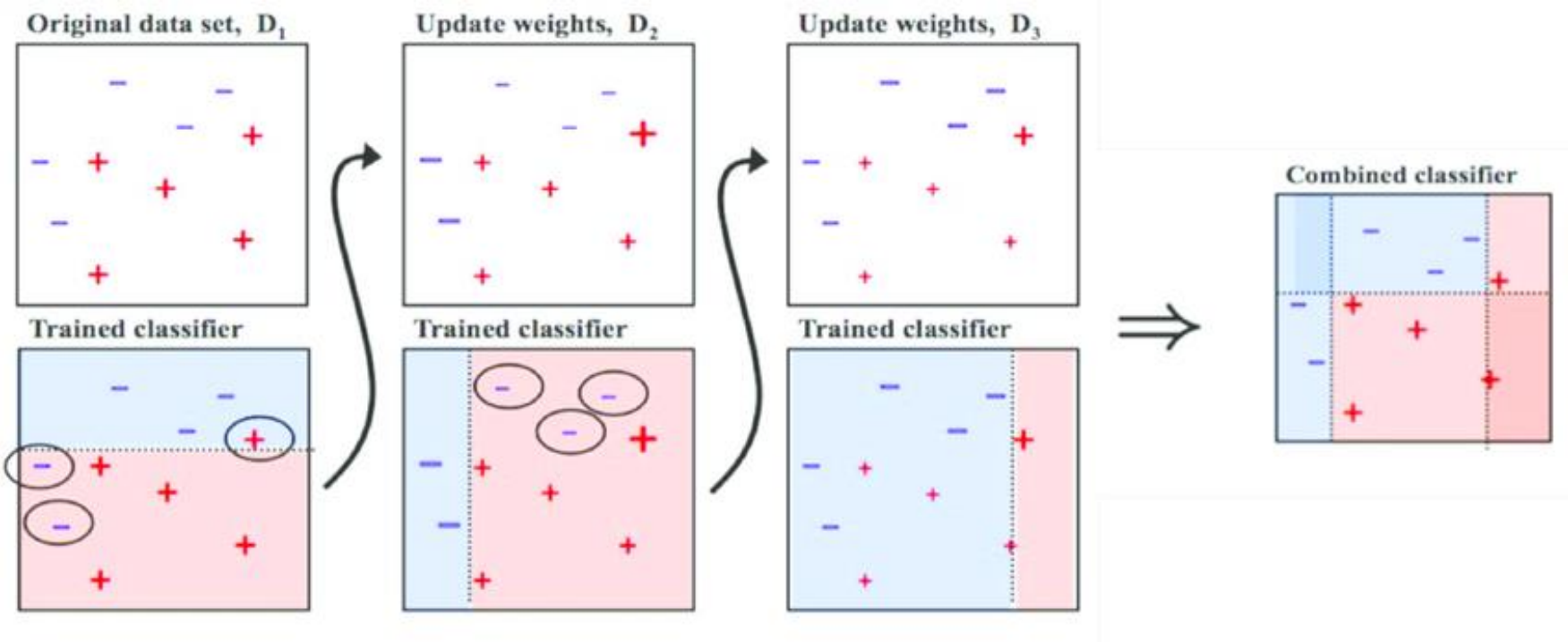


成千上万个Haar特征

Viola-Jones人脸检测算法：检测流程



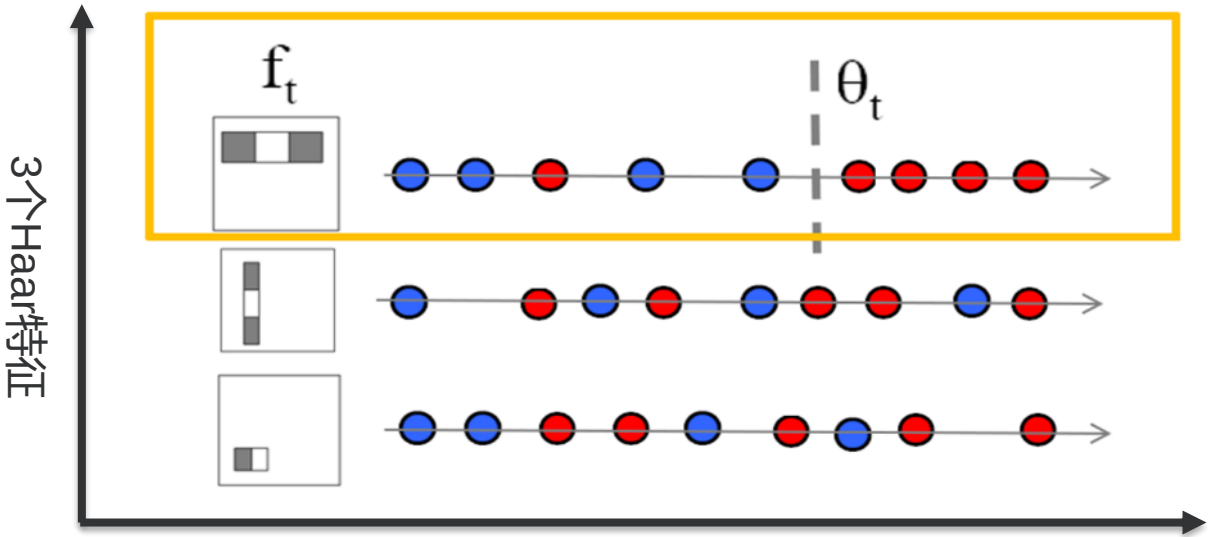
Viola-Jones人脸检测算法：AdaBoost



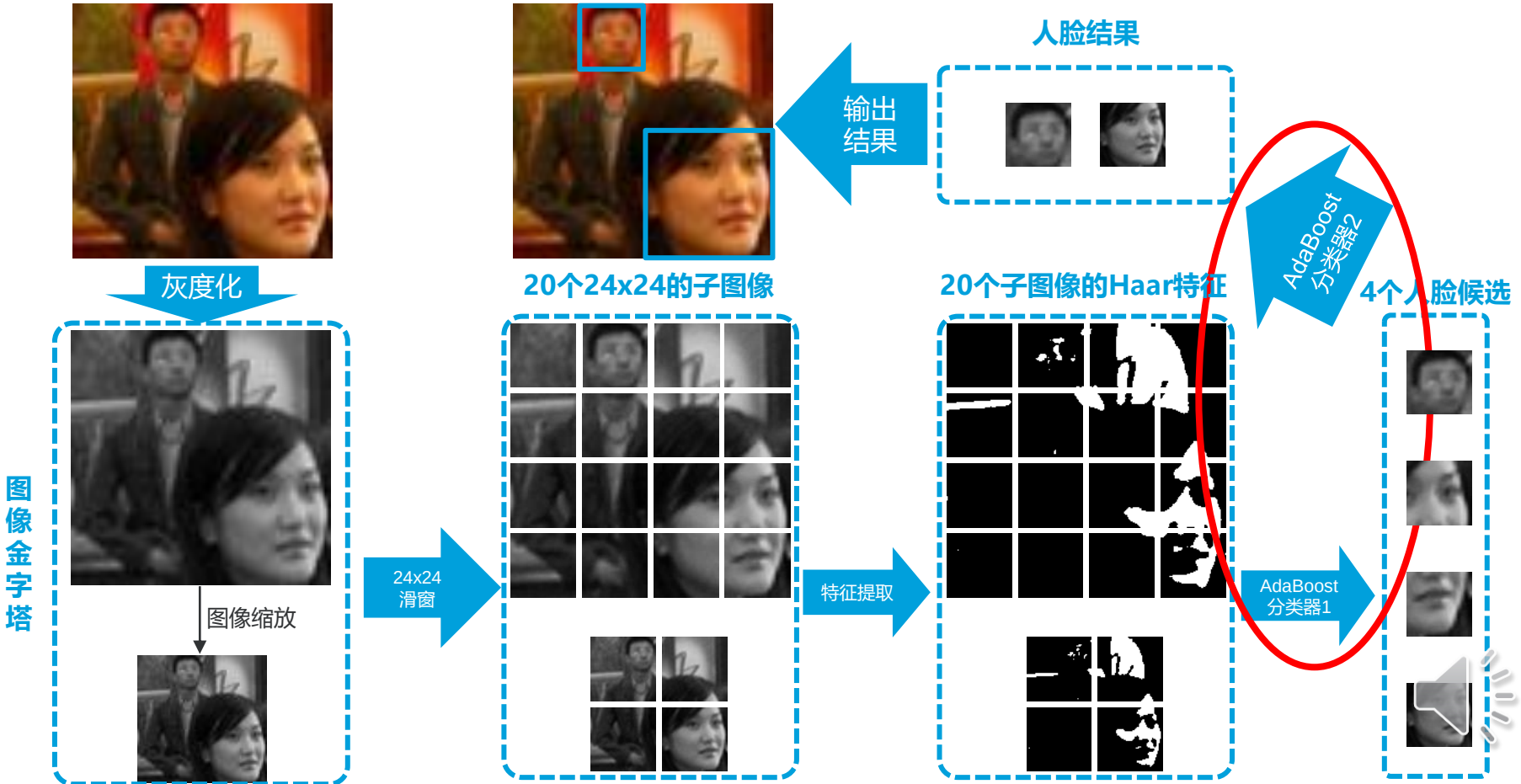
AdaBoost分类器的思想

Viola-Jones人脸检测算法：AdaBoost

- 经过特征提取后，每个24x24的训练样本都产生N个Haar特征（N维）
- 在每个维度的特征上，选一个最优阈值，构建一个人脸和非人脸分类器（N个分类器）
- 从N个分类器中，选出分类误差最小的那个，最为一个弱分类器
- 更新训练样本的权重，继续搜寻下一个弱分类器

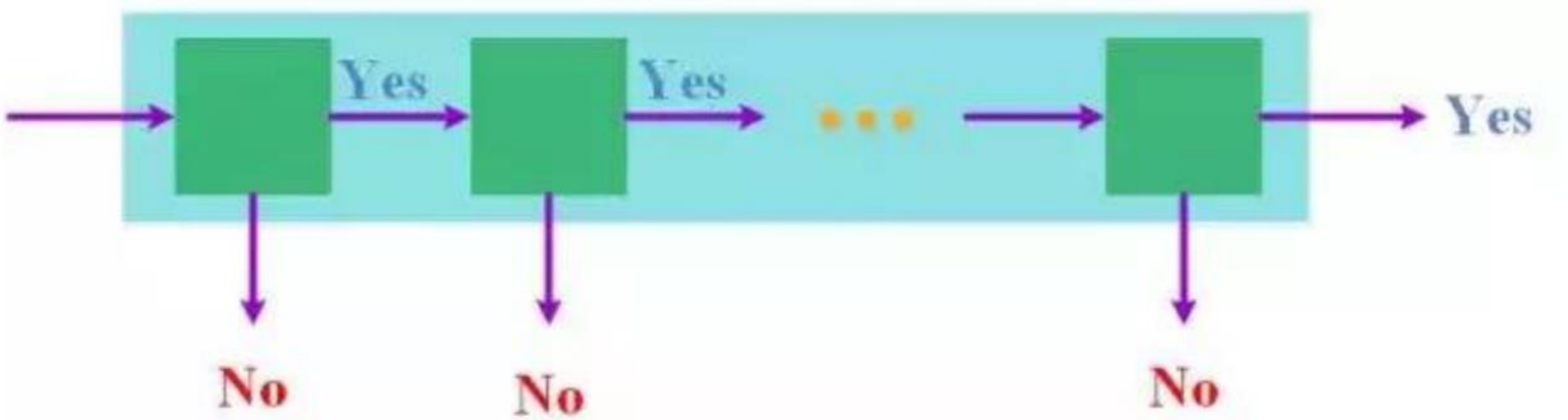


Viola-Jones人脸检测算法：检测流程



Viola-Jones人脸检测算法：级联分类器

- 把若干个AdaBoost 分类器级联起来
- 一开始使用**少量**的特征将**大量**的**简单**非人脸区域剔除掉
- 后面再利用**大量**的特征将**少量**的**复杂**非人脸区域剔除掉



- VJ 分类器最后一共是级联38个分类器，共包含 6060个特征
- 其中前面7个分类器对应的特征数为2->10->25->25->50->50->50

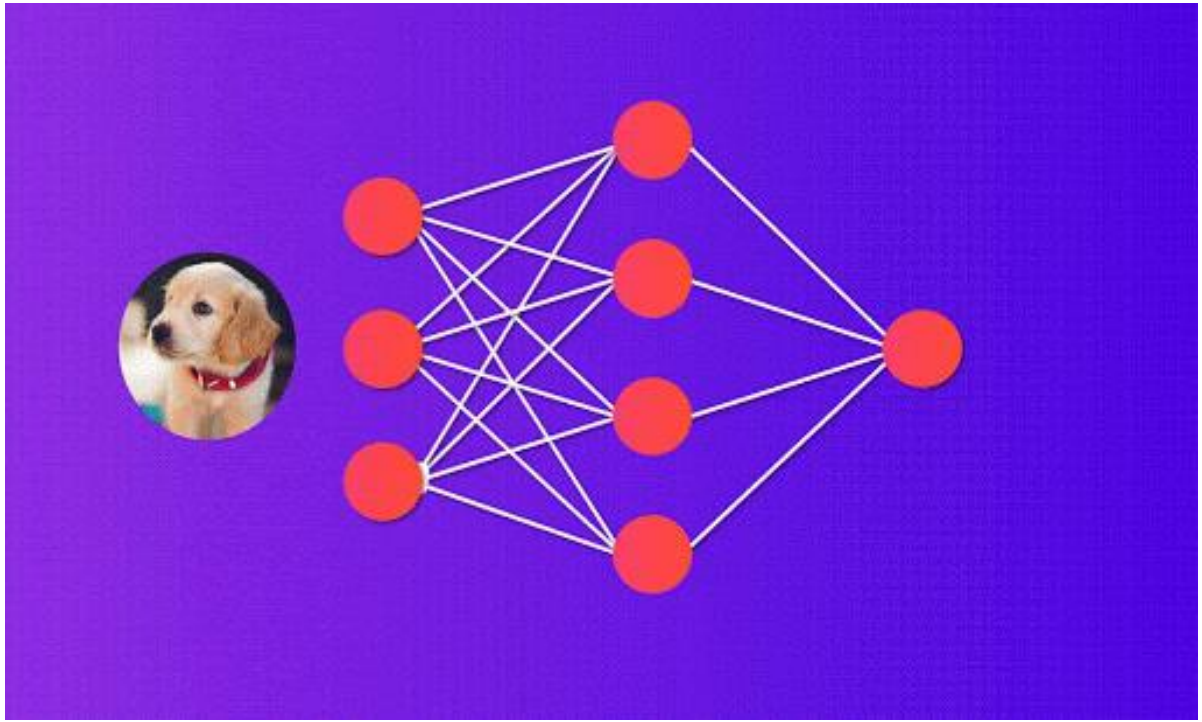
传统人脸检测算法：总结

- 滑窗操作遍历所有的位置：在图像上进行密集地窗口滑动以遍历所有的位置
- 图像金字塔遍历所有的大小：对图像进行缩放来解决人脸的尺度问题
- 手工特征提取：提取人为设计的手工特征，例如Haar
- 手工分类器：使用人为设计的分类器，例如级联的AdaBoost分类器
- 检测过程分为多个独立的阶段，训练过程比较繁琐
- 传统物体检测算法有较快的速度，但在复杂场景下精度不大理想
- 随着深度学习的出现，传统算法渐渐被取代，但仍有着重要的指导意义

目录

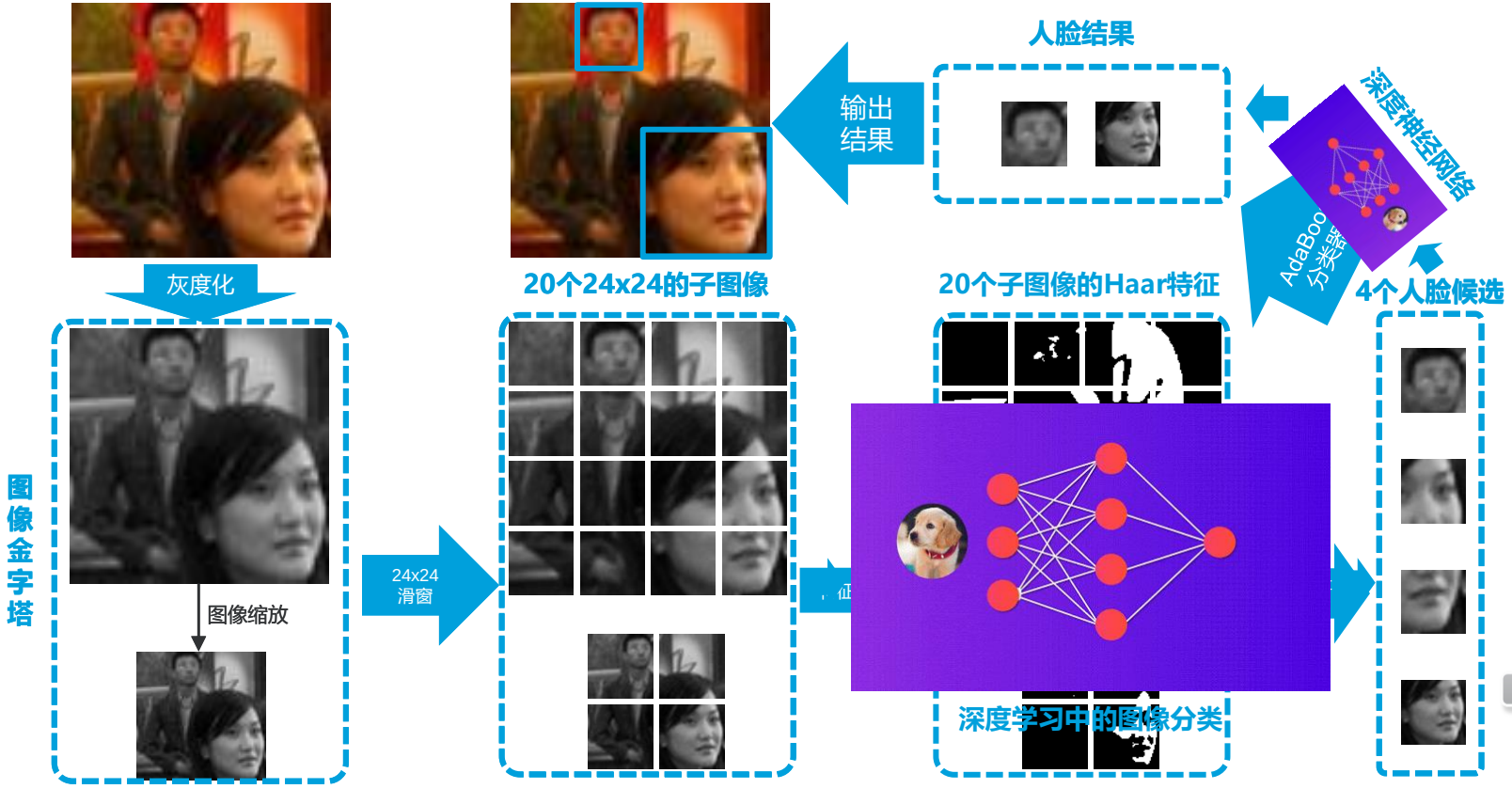
深度学习中的图像分类

- 利用神经网络，基于反向传播原理，自动地提取图像特征并对其进行分类，得到预测结果



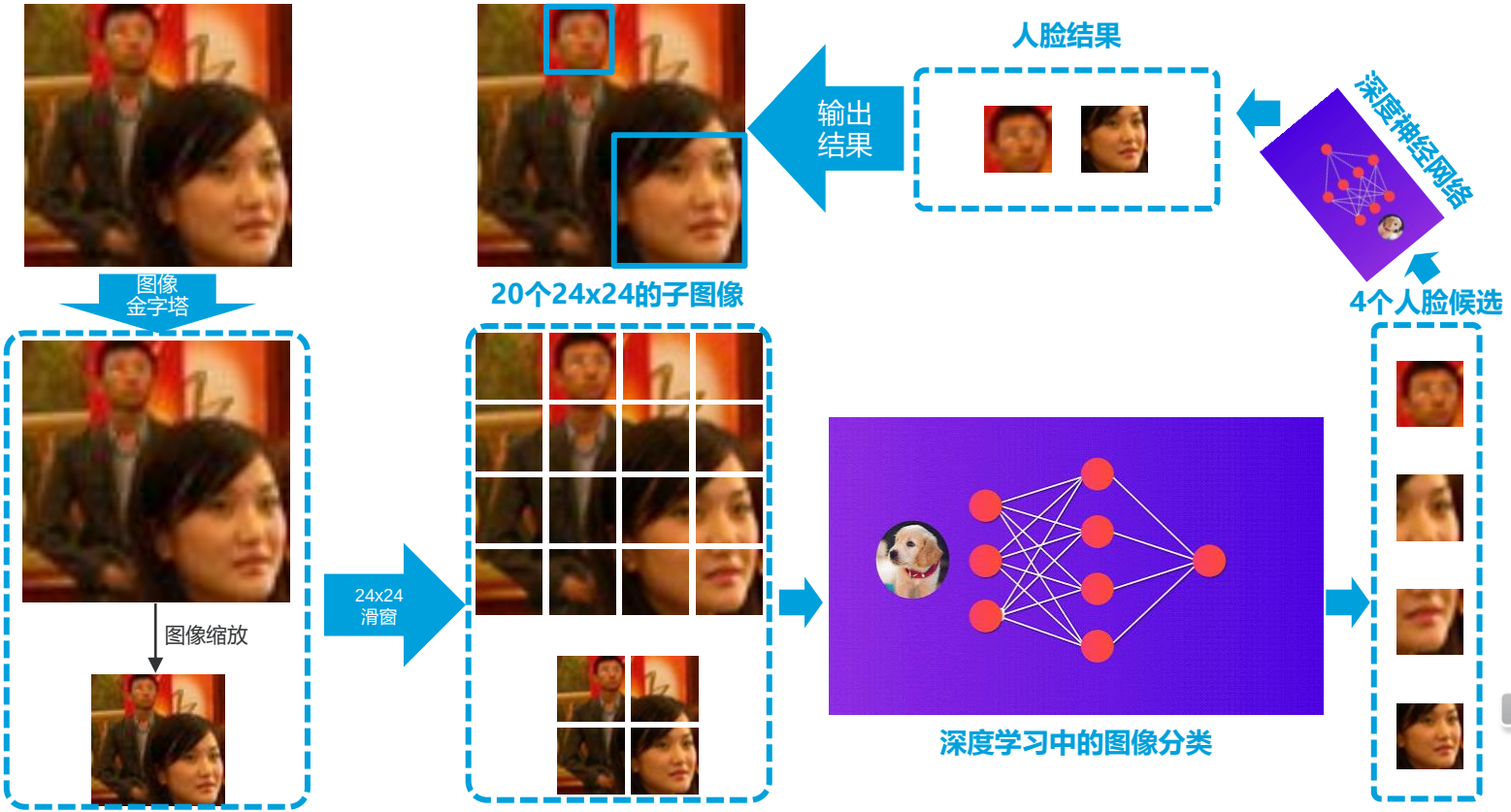
深度学习早期人脸检测算法

- 在传统人脸检测算法的流程中，把**手工设计的特征和分类器**变成**深度学习中的特征和分类器**

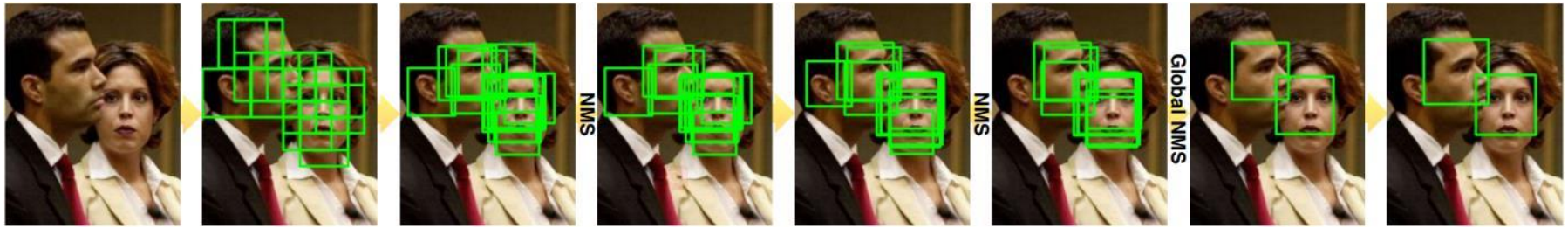


深度学习早期人脸检测算法

- 在传统人脸检测算法的流程中，把**手工设计的特征和分类器**变成**深度学习中的特征和分类器**



深度学习早期人脸检测算法：CascadeCNN



- CascadeCNN人脸检测算法 = 传统人脸检测算法Viola-Jones + 深度学习分类网络
- 滑窗操作：在图像上进行密集地窗口滑动，以遍历所有位置的人脸
- 图像金字塔：对图像进行缩放，以遍历所有大小的人脸，来解决尺度问题
- 级联思想：3级网络以级联的方式连接，每级网络由1个二分类网络和1个位置矫正网络组成

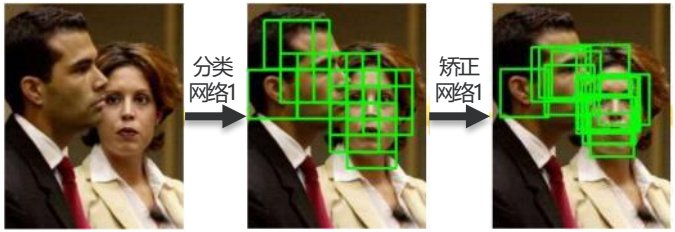
* 注：图中的NMS是指非极大值抑制（Non-Maximum Suppression）

A Convolutional Neural Network Cascade for Face Detection, CVPR 2015

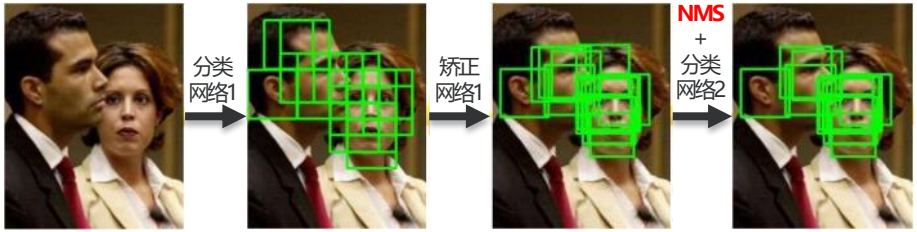
CascadeCNN人脸检测算法：检测流程

CascadeCNN人脸检测算法：检测流程

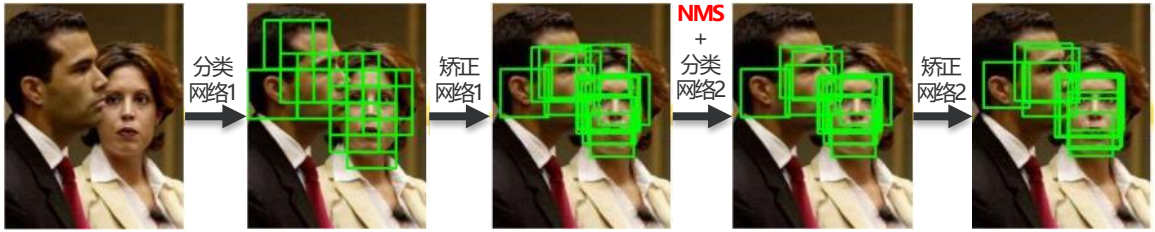
CascadeCNN人脸检测算法：检测流程



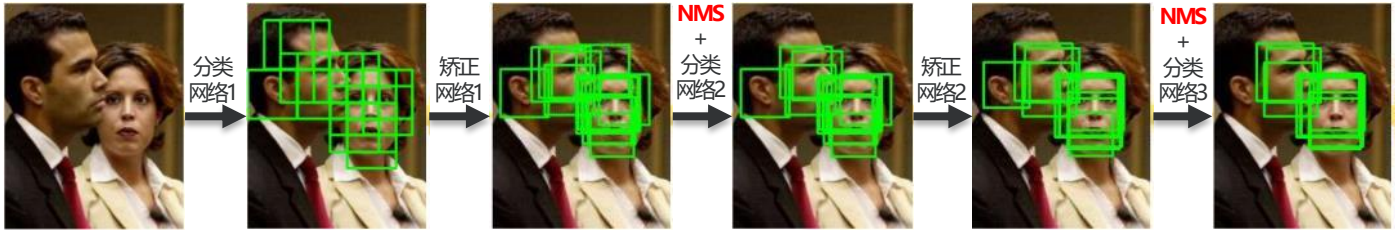
CascadeCNN人脸检测算法：检测流程



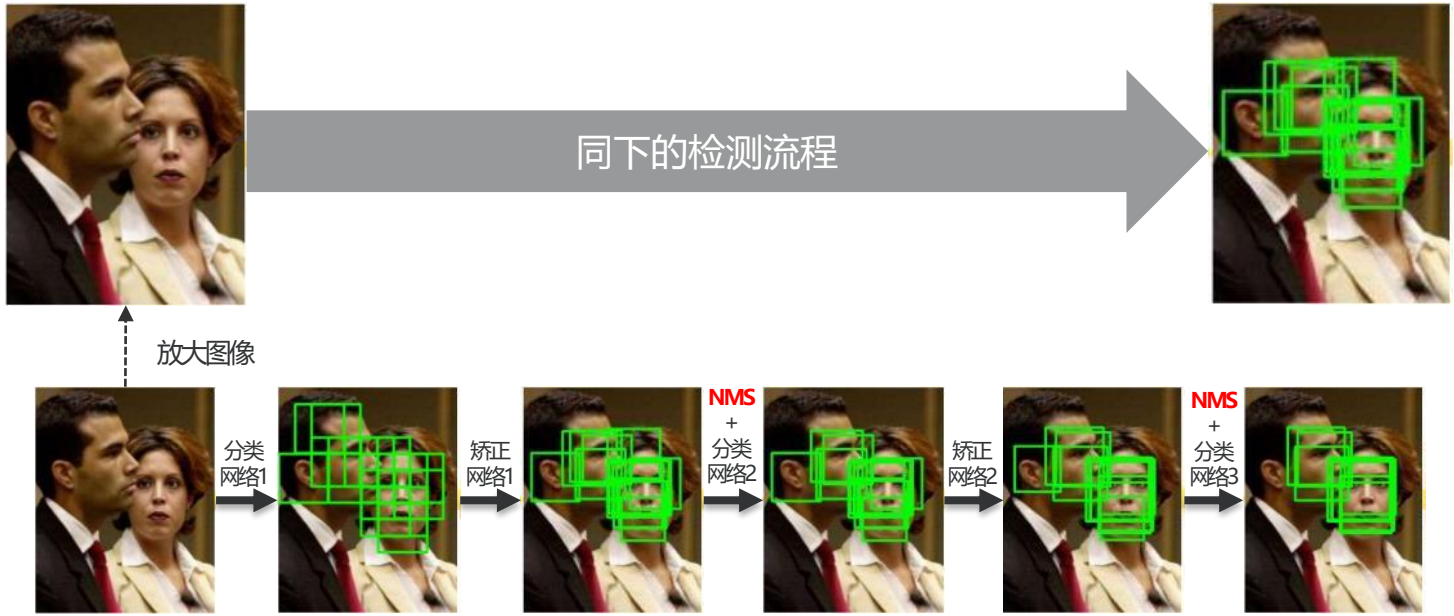
CascadeCNN人脸检测算法：检测流程



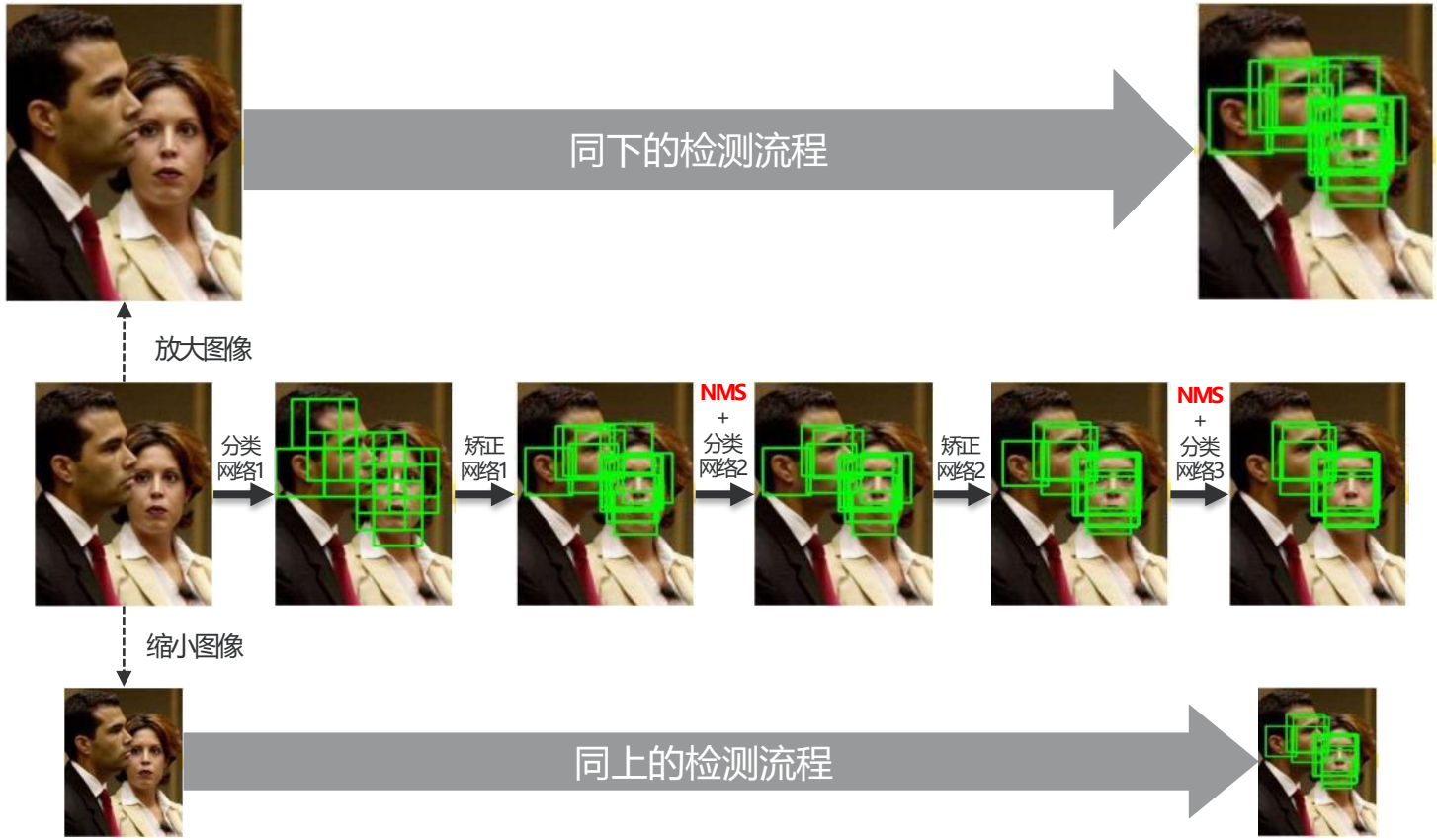
CascadeCNN人脸检测算法：检测流程



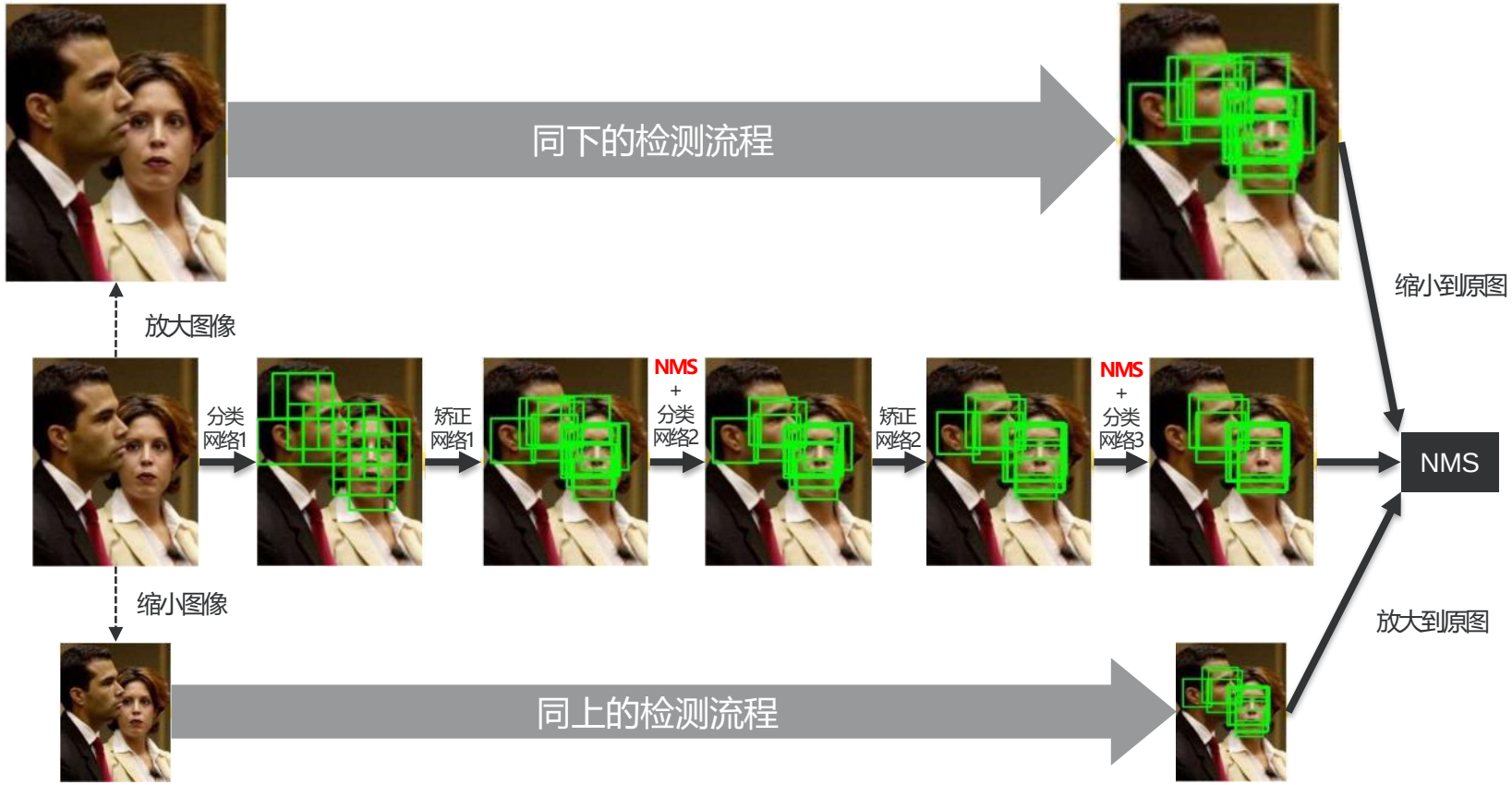
CascadeCNN人脸检测算法：检测流程



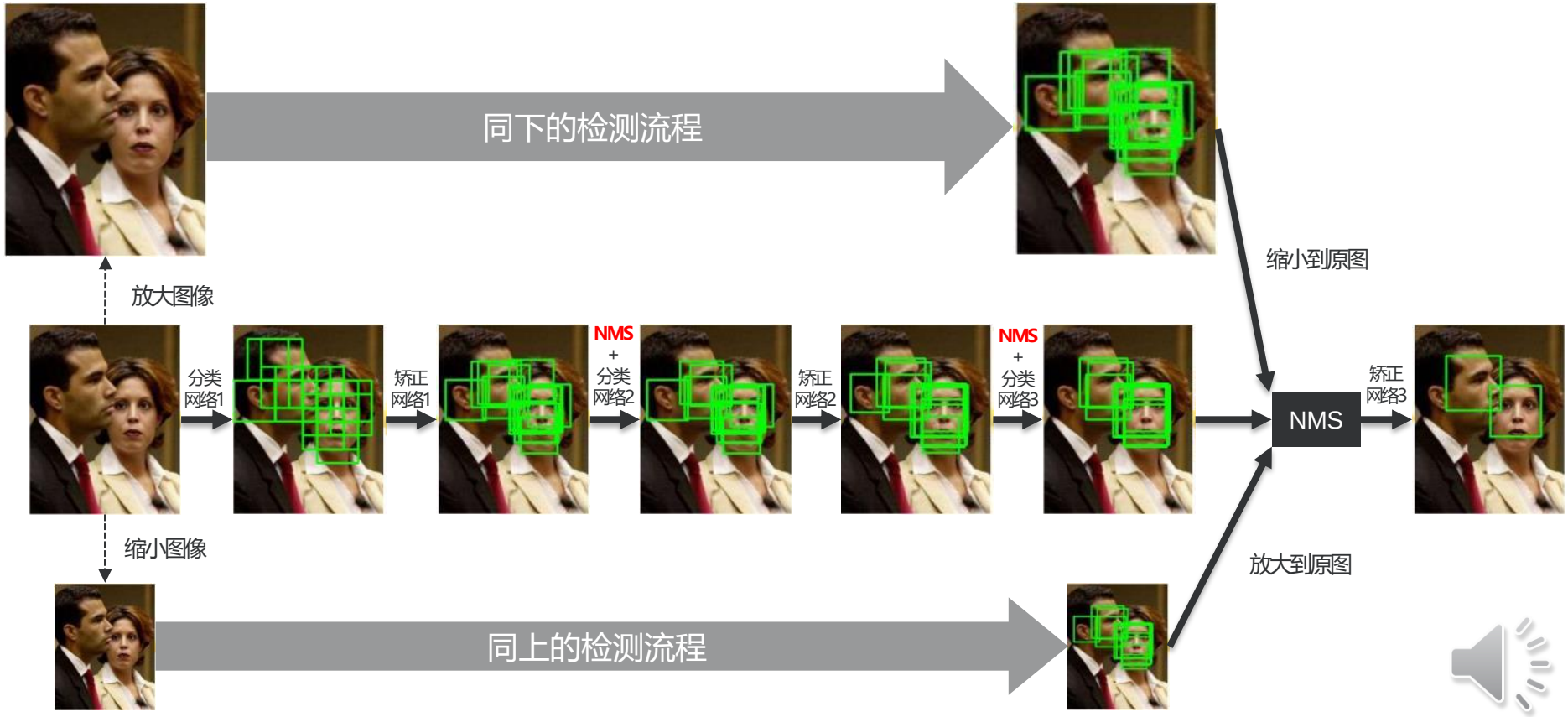
CascadeCNN人脸检测算法：检测流程



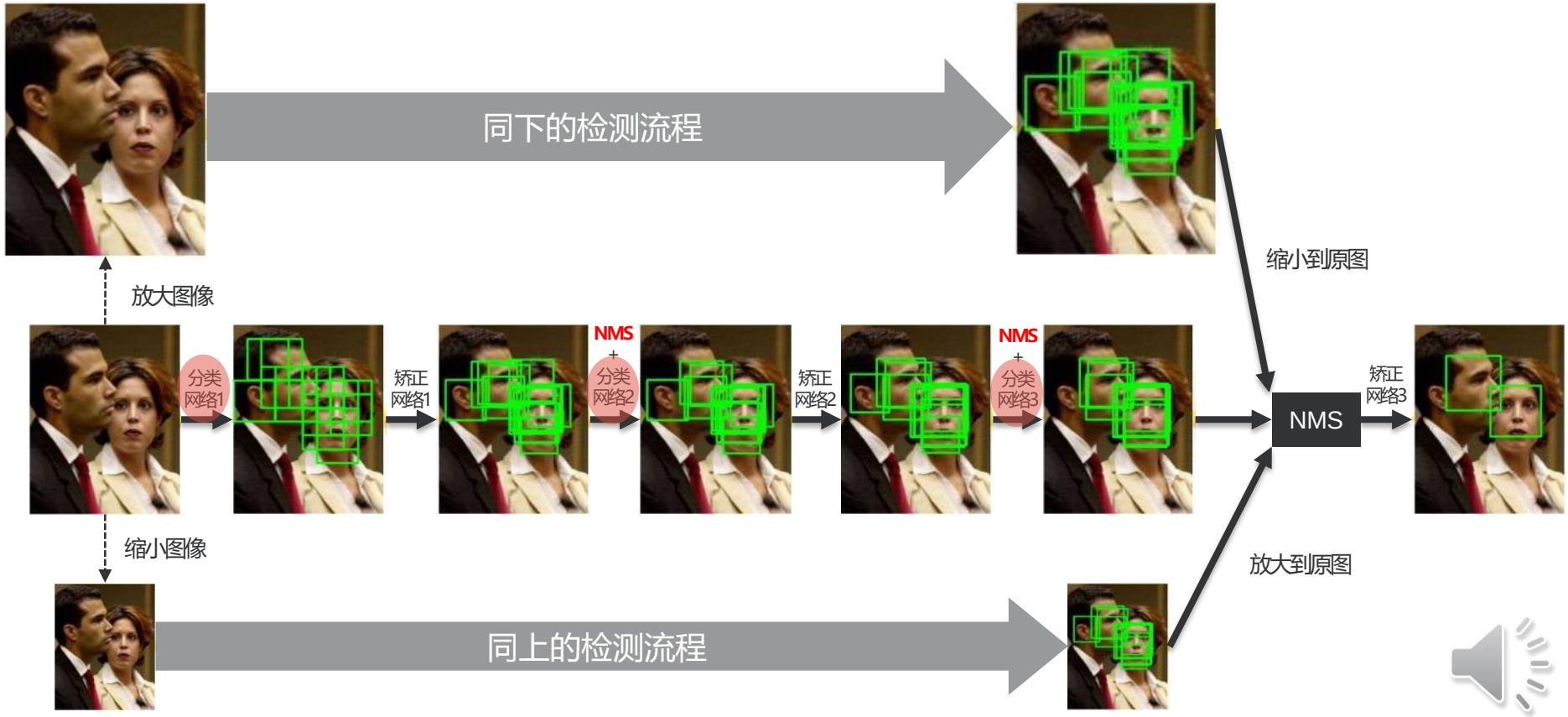
CascadeCNN人脸检测算法：检测流程



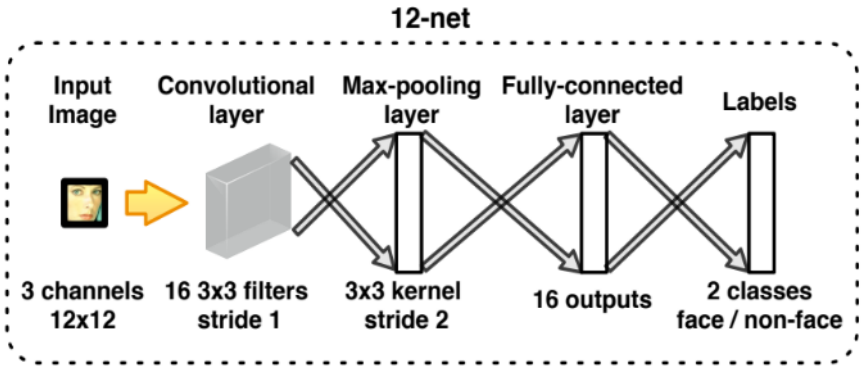
CascadeCNN人脸检测算法：检测流程



CascadeCNN人脸检测算法：检测流程

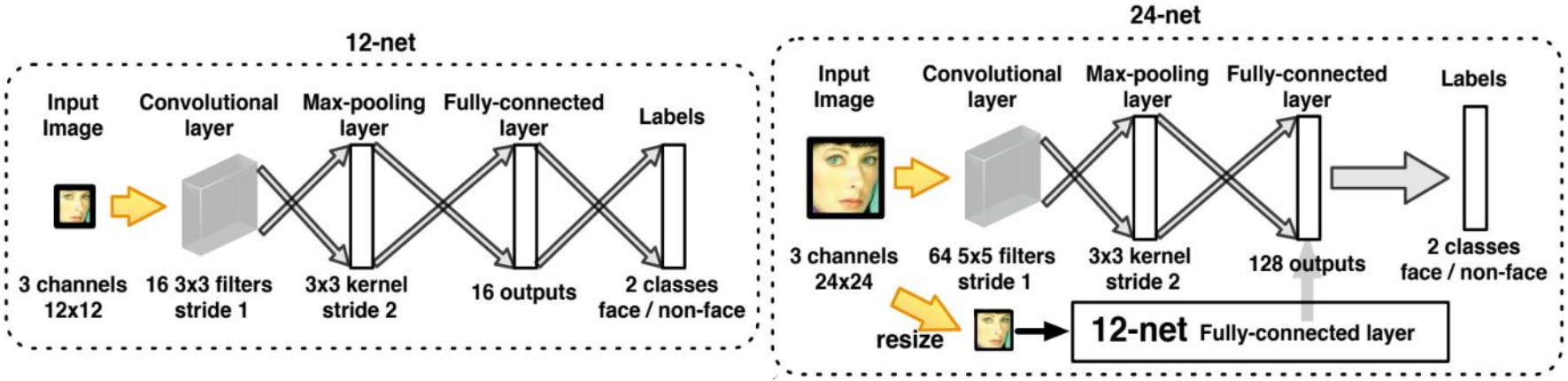


CascadeCNN人脸检测算法：分类网络1



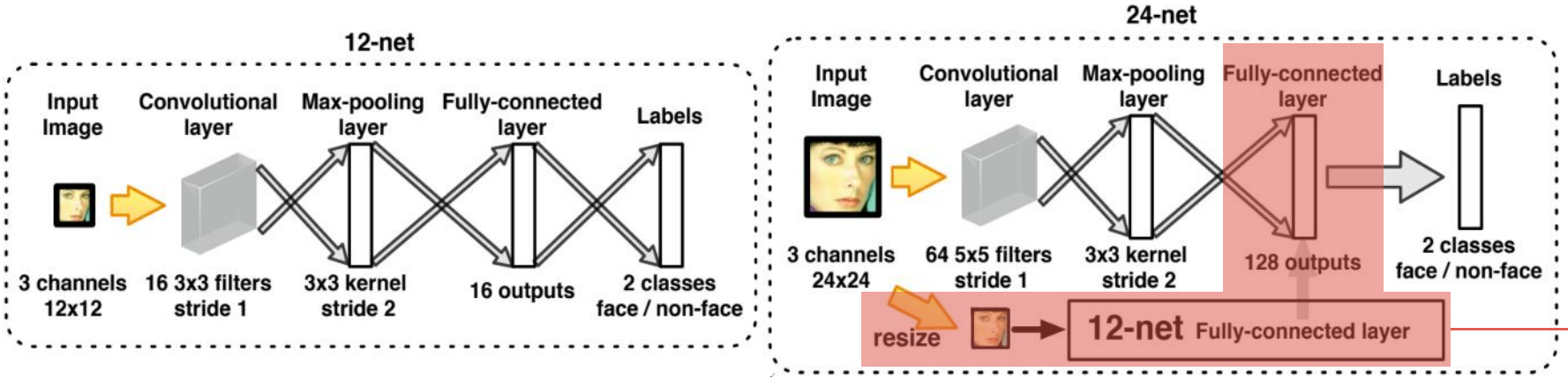
- 输入为12x12大小的RGB图像，故该分类网络记**12-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为16，1个输出为2（人脸 or 非人脸）
- 损失函数：softmax损失函数训练整个12-net网络

CascadeCNN人脸检测算法：分类网络2



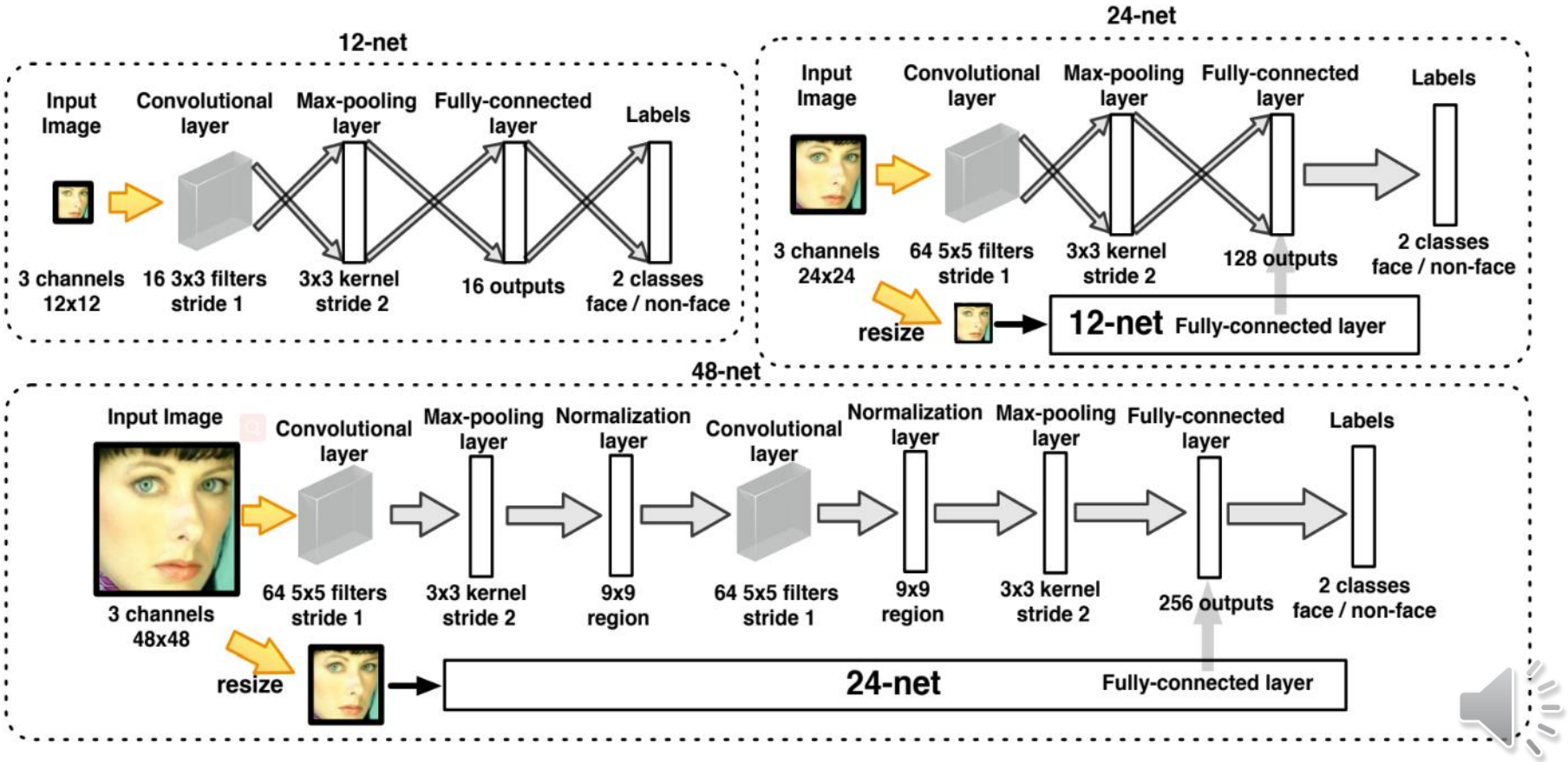
- 输入为24x24大小的RGB图像，故该分类网络记**24-net**
- 1个卷积层：卷积核为5x5，输出通道数为64，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为2（人脸 or 非人脸）
- 特征融合：缩小图像，输入12-net分类网络，输出16维特征与24-net分类网络输出的128维特征，拼接成144维特征
- 损失函数：softmax损失函数训练整个24-net网络

CascadeCNN人脸检测算法：分类网络2



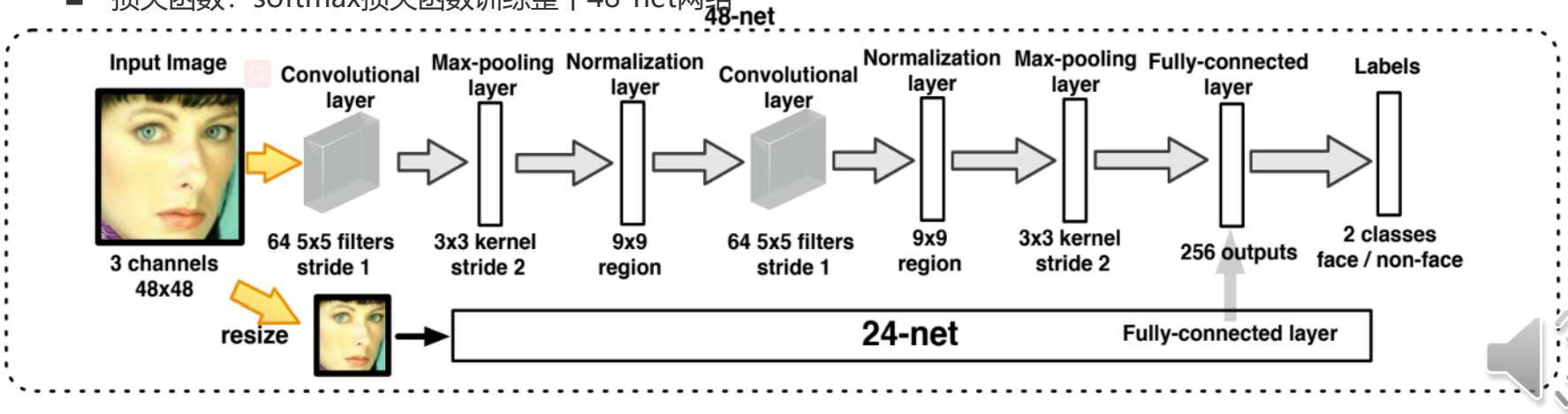
- 输入为24x24大小的RGB图像，故该分类网络记**24-net**
- 1个卷积层：卷积核为5x5，输出通道数为64，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为2（人脸 or 非人脸）
- 特征融合：缩小图像，输入12-net分类网络，输出16维特征与24-net分类网络输出的128维特征，拼接成144维特征
- 损失函数：softmax损失函数训练整个24-net网络

CascadeCNN人脸检测算法：分类网络3



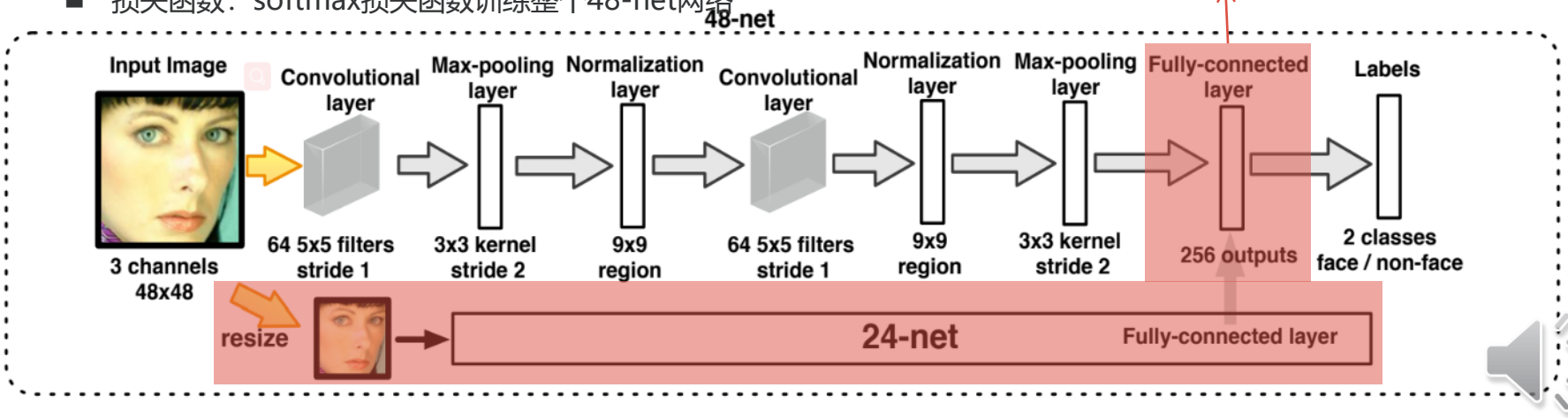
CascadeCNN人脸检测算法：分类网络3

- 输入为48x48大小的RGB图像，故该分类网络记**48-net**
- 2个卷积层：卷积核为5x5，输出通道数为64，卷积间隔为1
- 2个池化层：池化核为3x3，池化间隔为2
- 2个归一化层：归一化区域大小为9x9
- 2个全连接层：1个输出为256，1个输出为2（人脸 or 非人脸）
- 特征融合：缩小图像，输入24-net分类网络输出128维特征，与48-net分类网络输出的256维特征，拼接成384维特征
- 损失函数：softmax损失函数训练整个48-net网络

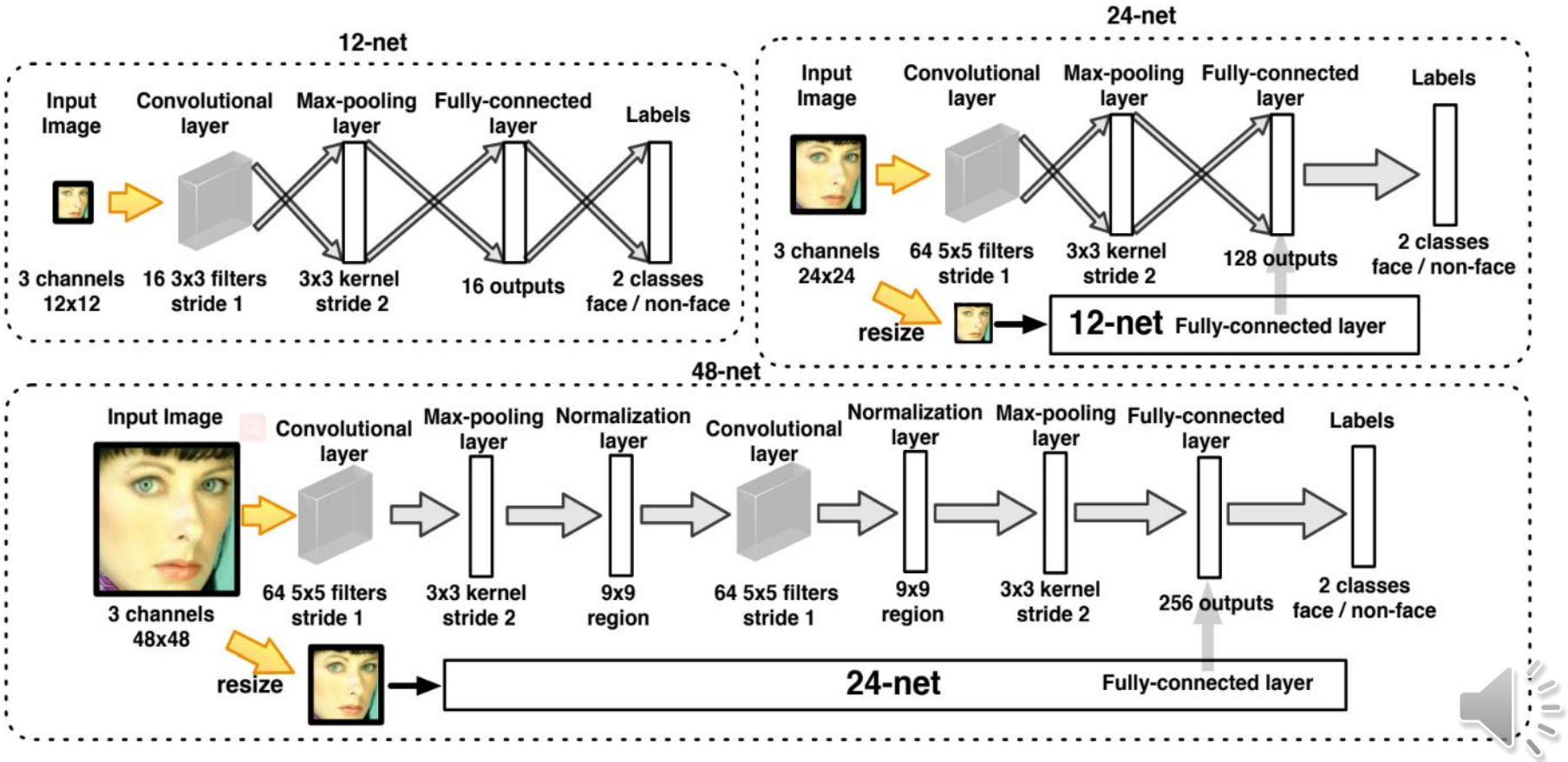


CascadeCNN人脸检测算法：分类网络3

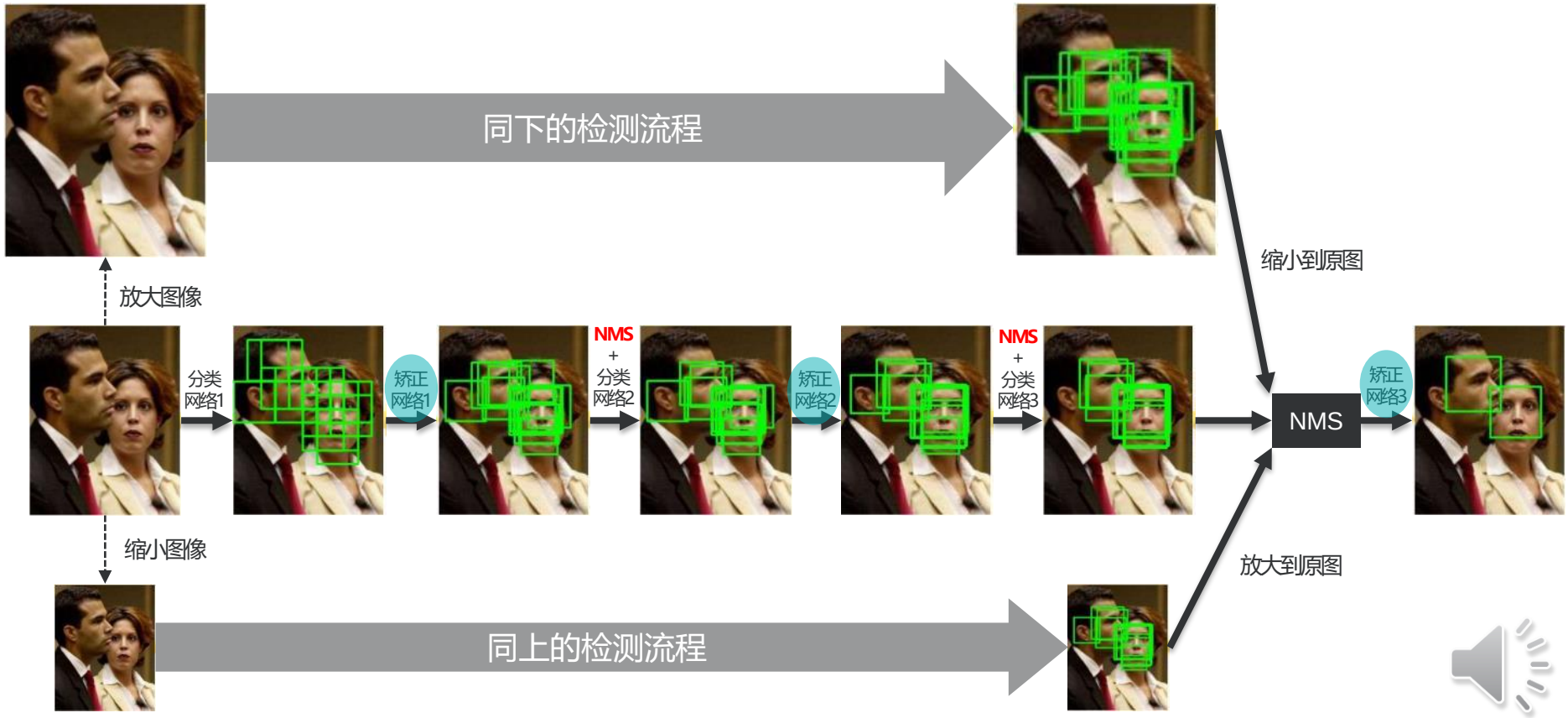
- 输入为48x48大小的RGB图像，故该分类网络记**48-net**
- 2个卷积层：卷积核为5x5，输出通道数为64，卷积间隔为1
- 2个池化层：池化核为3x3，池化间隔为2
- 2个归一化层：归一化区域大小为9x9
- 2个全连接层：1个输出为256，1个输出为2（人脸 or 非人脸）
- 特征融合：缩小图像，输入24-net分类网络输出128维特征，与48-net分类网络输出的256维特征，拼接成384维特征
- 损失函数：softmax损失函数训练整个48-net网络



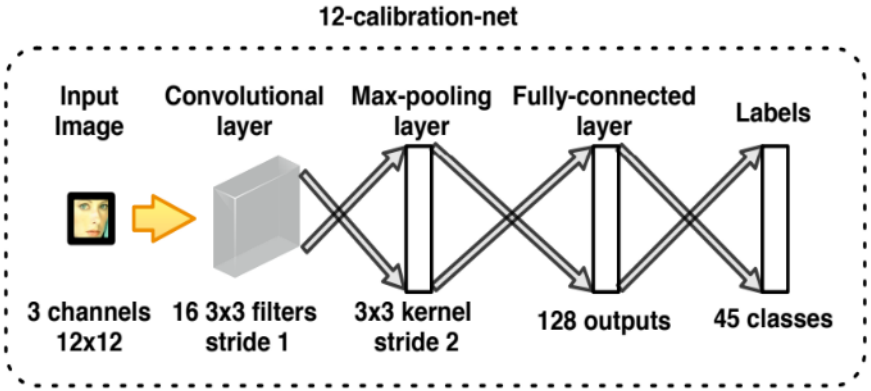
CascadeCNN人脸检测算法：分类网络



CascadeCNN人脸检测算法：检测流程

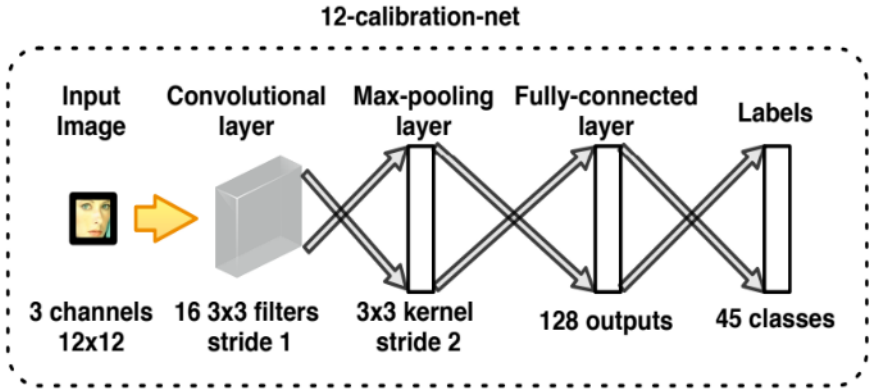


CascadeCNN人脸检测算法：矫正网络1



- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

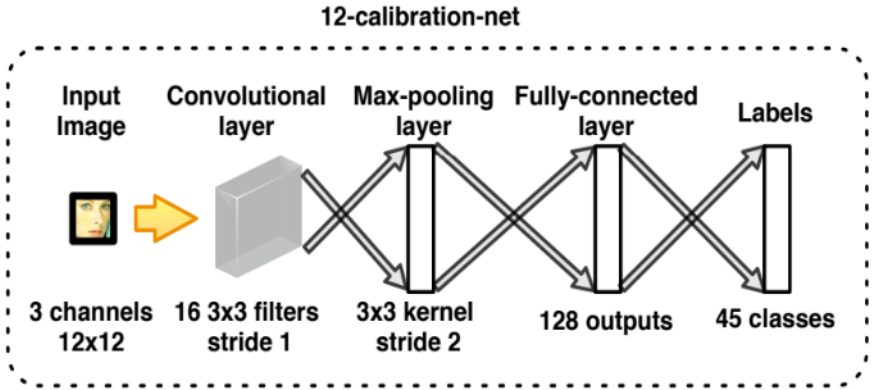
CascadeCNN人脸检测算法：矫正网络1



(x, y, w, h)

- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

CascadeCNN人脸检测算法：矫正网络1



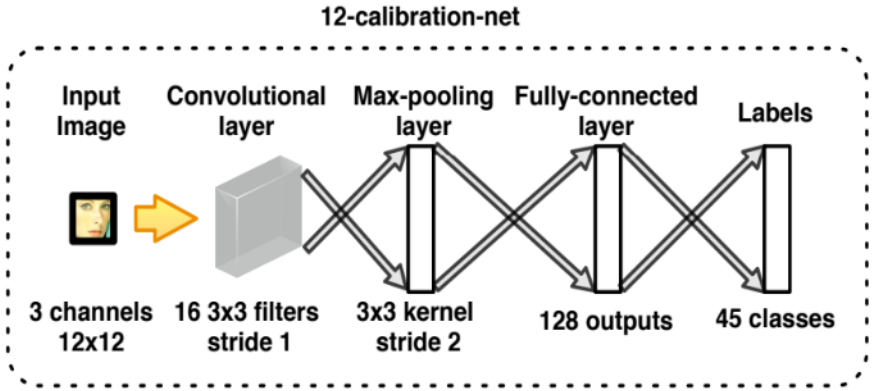
- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

(x, y, w, h)

调整位置（x,y）和大小（w,h）

$$(x - \frac{x_n w}{s_n}, y - \frac{y_n h}{s_n}, \frac{w}{s_n}, \frac{h}{s_n})$$

CascadeCNN人脸检测算法：矫正网络1



- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

(x, y, w, h)

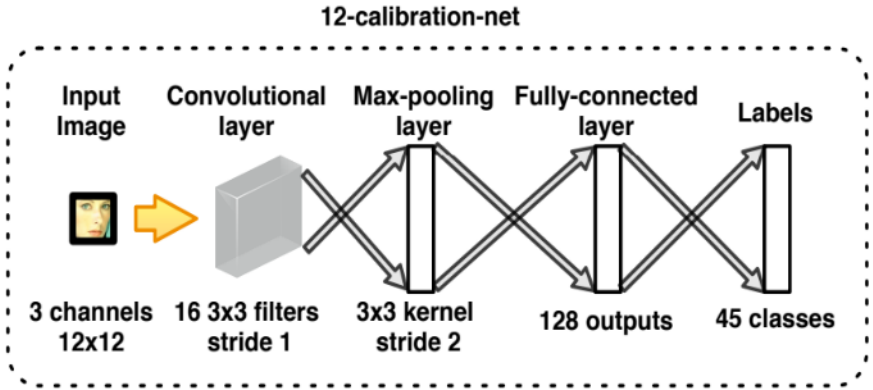
调整位置（x,y）和大小（w,h）

$$(x - \frac{x_n w}{s_n}, y - \frac{y_n h}{s_n}, \frac{w}{s_n}, \frac{h}{s_n})$$

调整的候选值

$$s_n \in \{0.83, 0.91, 1.0, 1.10, 1.21\}$$
$$x_n \in \{-0.17, 0, 0.17\}$$
$$y_n \in \{-0.17, 0, 0.17\}.$$

CascadeCNN人脸检测算法：矫正网络1



- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

(x, y, w, h)

调整位置（x,y）和大小（w,h）

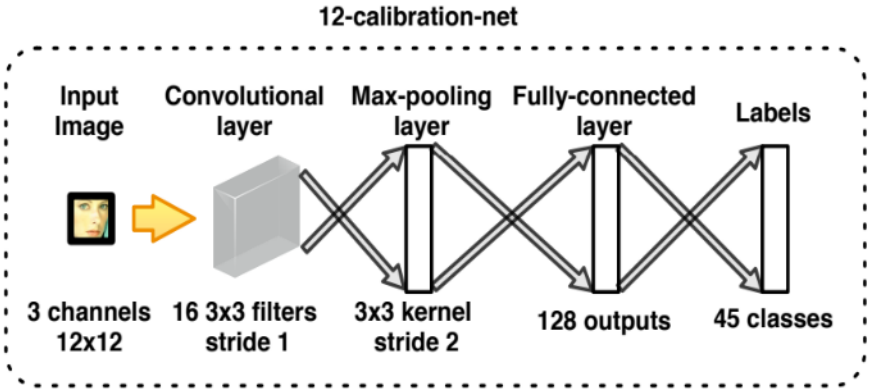
$$(x - \frac{x_n w}{s_n}, y - \frac{y_n h}{s_n}, \frac{w}{s_n}, \frac{h}{s_n})$$

调整的候选值

5x3x3=45个候选

$$\begin{aligned} s_n &\in \{0.83, 0.91, 1.0, 1.10, 1.21\} \\ x_n &\in \{-0.17, 0, 0.17\} \\ y_n &\in \{-0.17, 0, 0.17\}. \end{aligned}$$

CascadeCNN人脸检测算法：矫正网络1



- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络

45个类别的预测得分

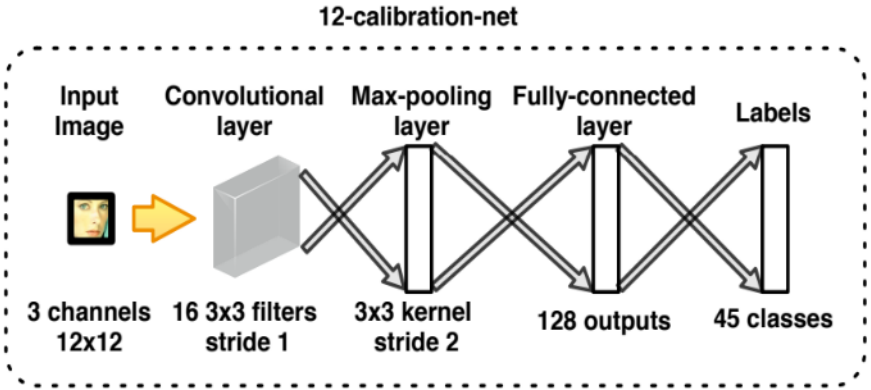
$$[s,x,y] = \frac{1}{Z}\sum_{n=1}^{N}[s_n,x_n,y_n]I(c_n > t)$$

$$Z = \sum_{n=1}^{N} I(c_n > t),$$

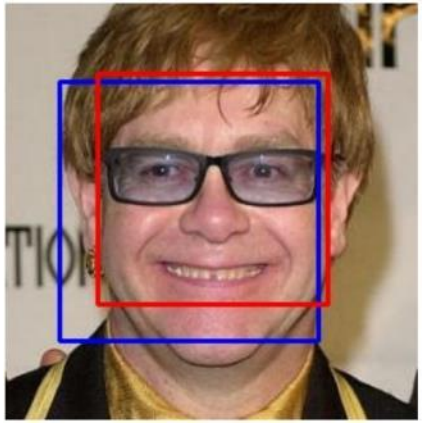
C_n 是第n个矫正类别的预测值

$$I(c_n > t) = \begin{cases} 1, & \text{if } c_n > t \\ 0, & \text{otherwise.} \end{cases}$$

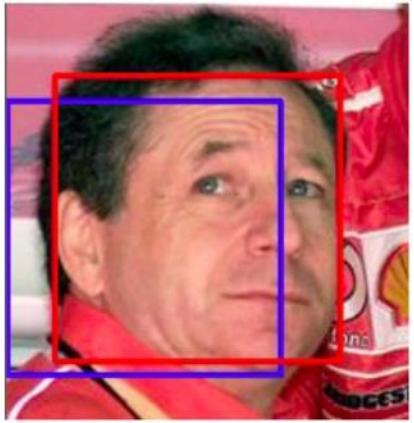
CascadeCNN人脸检测算法：矫正网络1



- 输入为12x12大小的RGB图像，故该矫正网络记**12-calibration-net**
- 1个卷积层：卷积核为3x3，输出通道数为16，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为128，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个12-calibration-net网络



矫正结果



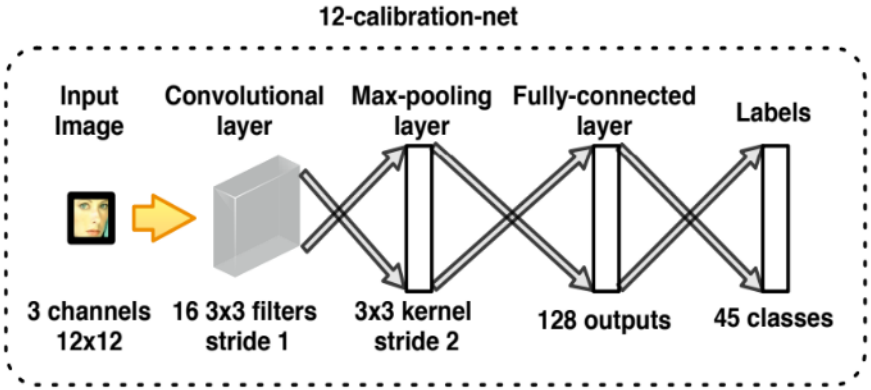
45个类别的预测得分

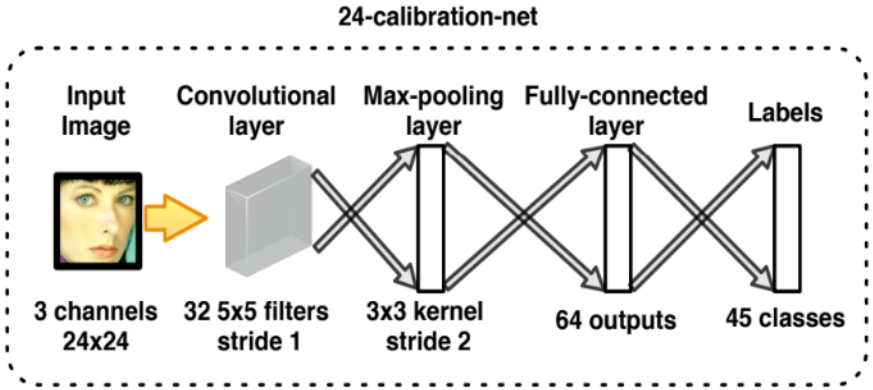
$$[s,x,y] = \frac{1}{Z}\sum_{n=1}^{N}[s_n,x_n,y_n]I(c_n > t)$$

$$Z = \sum_{n=1}^{N} I(c_n > t),$$

$$I(c_n > t) = \begin{cases} 1, & \text{if } c_n > t \\ 0, & \text{otherwise.} \end{cases}$$

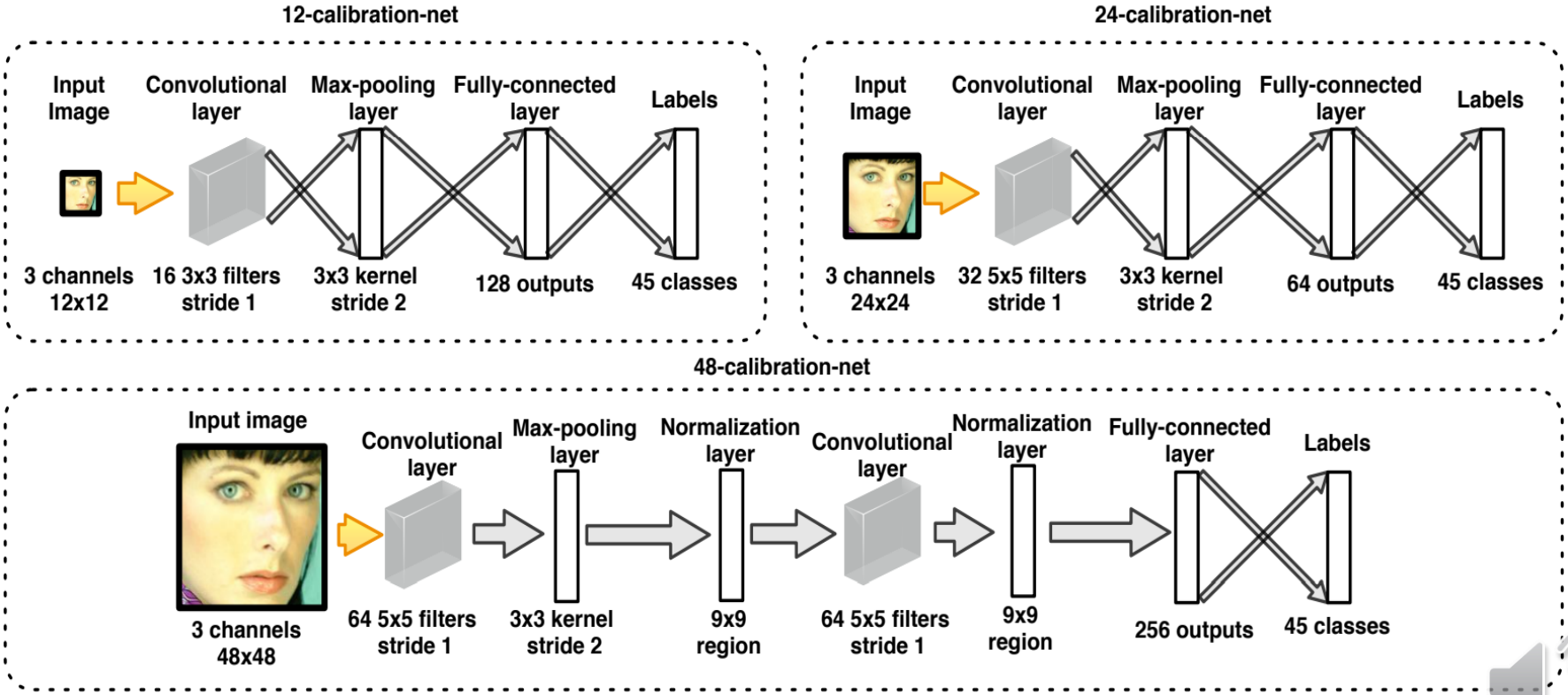
CascadeCNN人脸检测算法：矫正网络2





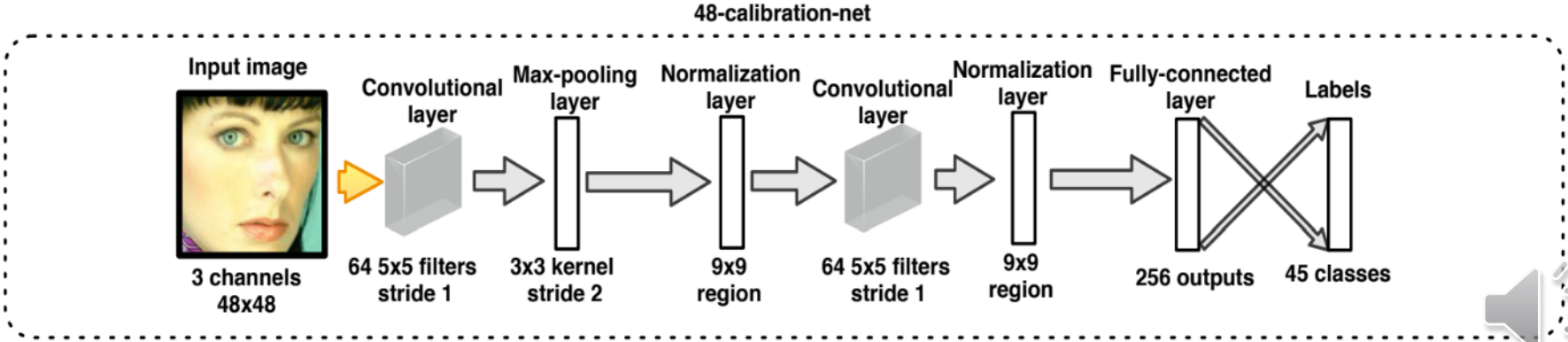
- 输入为24x24大小的RGB图像，故该矫正网络记**24-calibration-net**
- 1个卷积层：卷积核为5x5，输出通道数为32，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个全连接层：1个输出为64，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个24-calibration-net网络

CascadeCNN人脸检测算法：矫正网络3

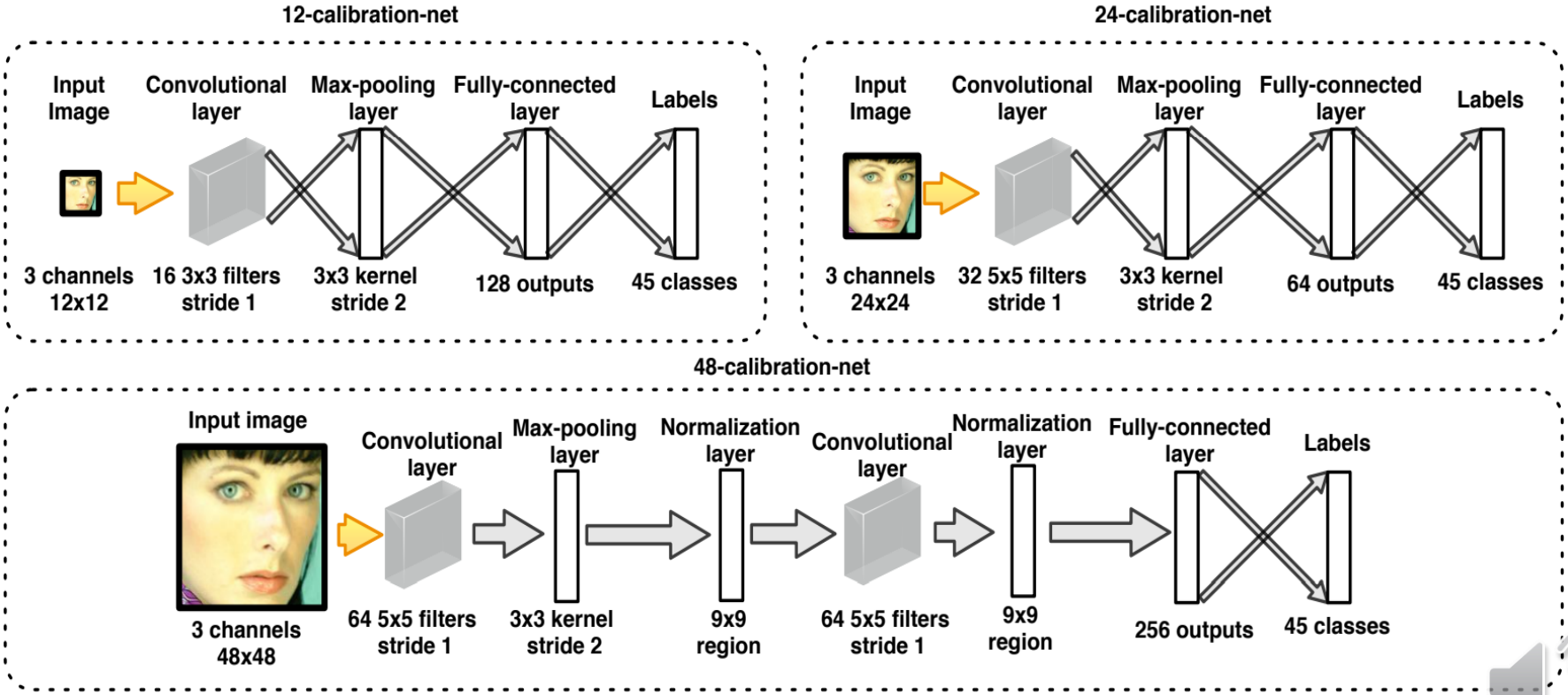


CascadeCNN人脸检测算法：矫正网络3

- 输入为48x48大小的RGB图像，故该分类网络记**48-calibration-net**
- 2个卷积层：卷积核为5x5，输出通道数为64，卷积间隔为1
- 1个池化层：池化核为3x3，池化间隔为2
- 2个归一化层：归一化区域大小为9x9
- 2个全连接层：1个输出为256，1个输出为45（45个预设的矫正类别）
- 损失函数：softmax损失函数训练整个48-calibration-net网络

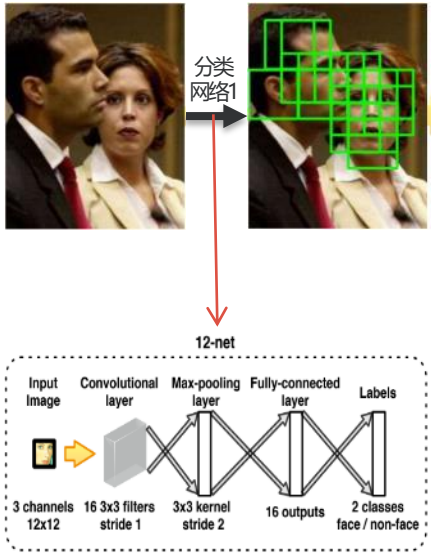


CascadeCNN人脸检测算法：矫正网络

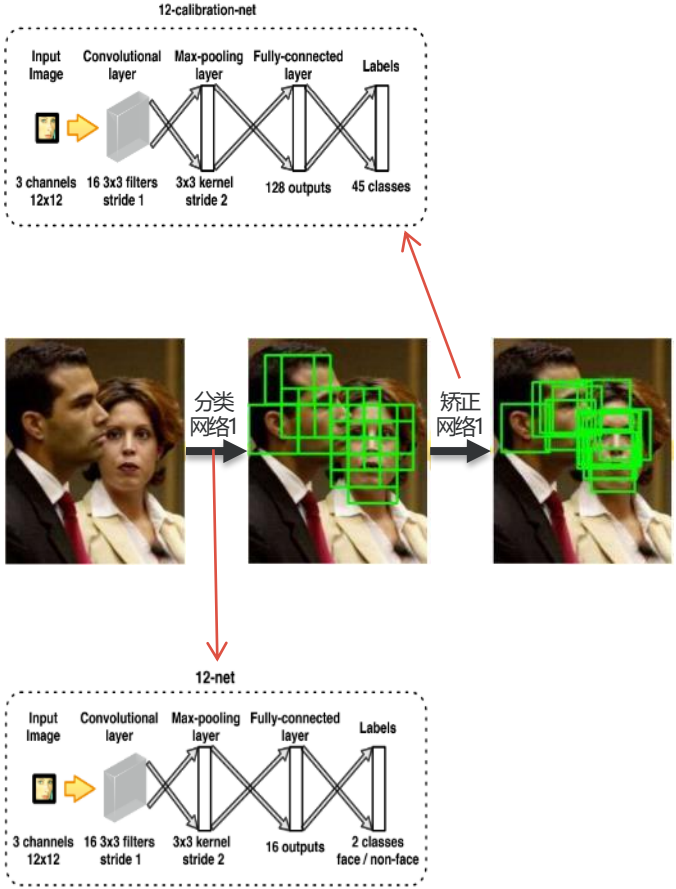


CascadeCNN人脸检测算法：整体框架

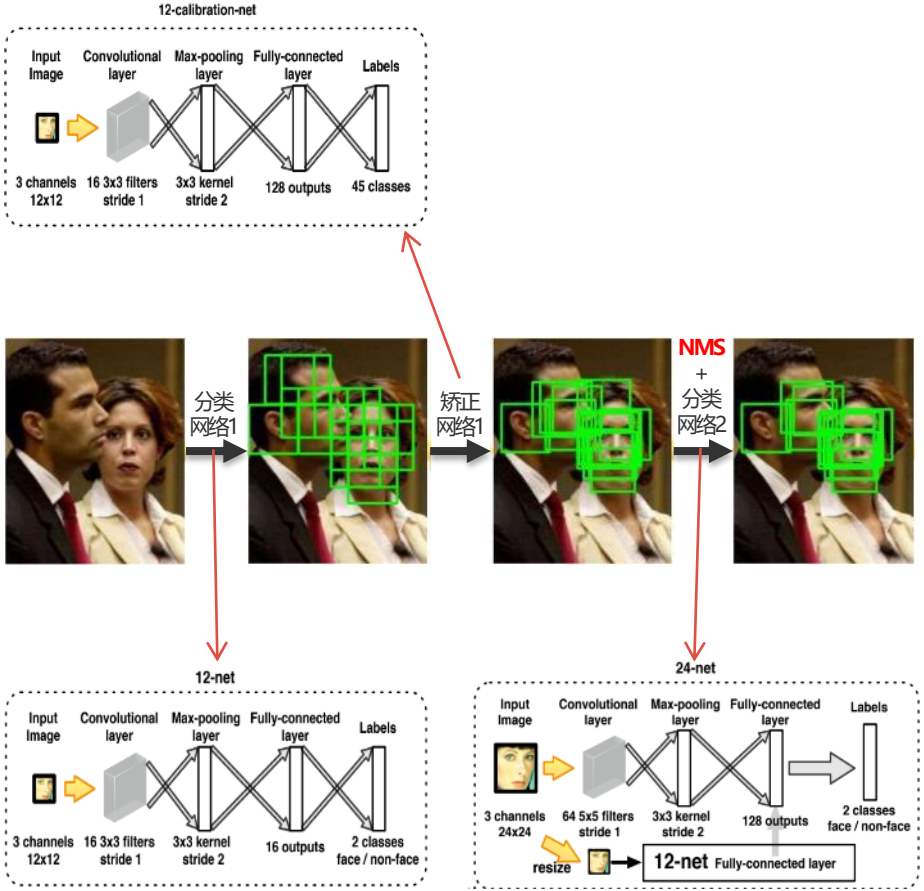
CascadeCNN人脸检测算法：整体框架



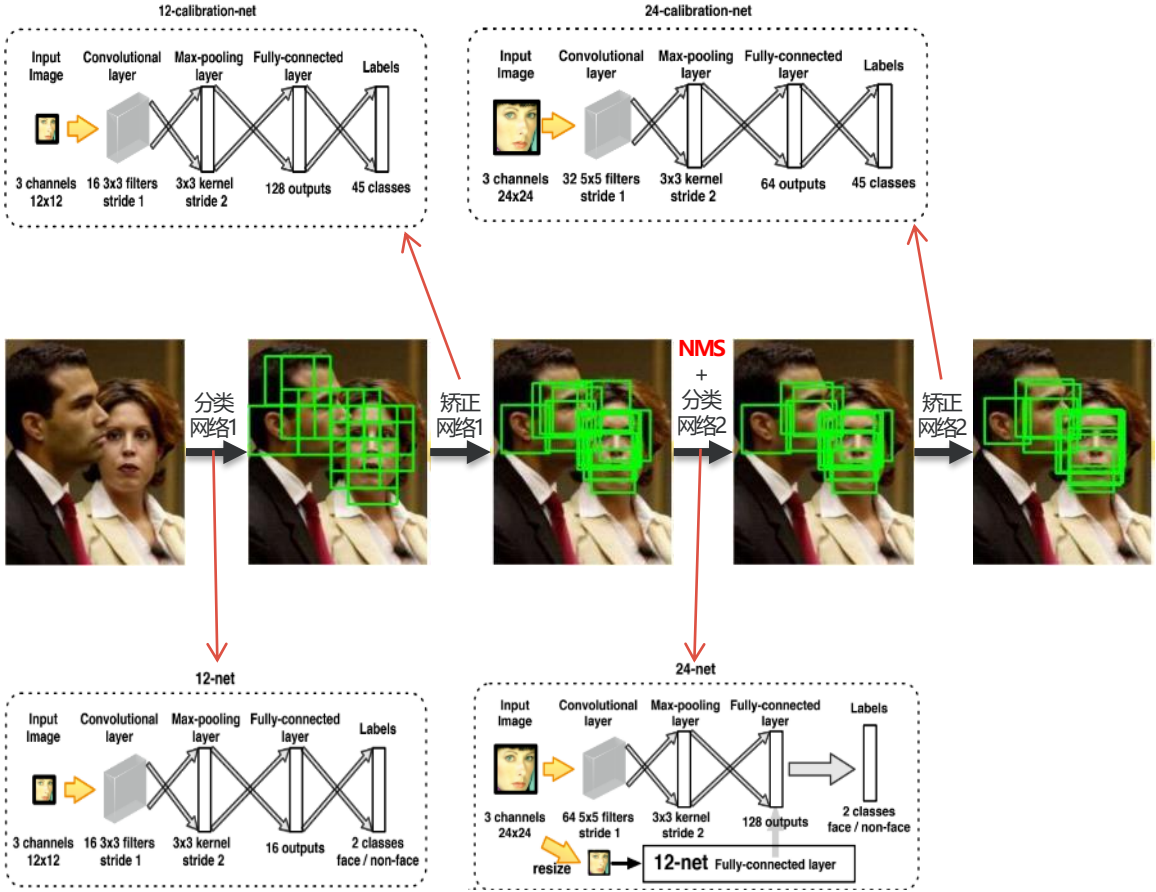
CascadeCNN人脸检测算法：整体框架



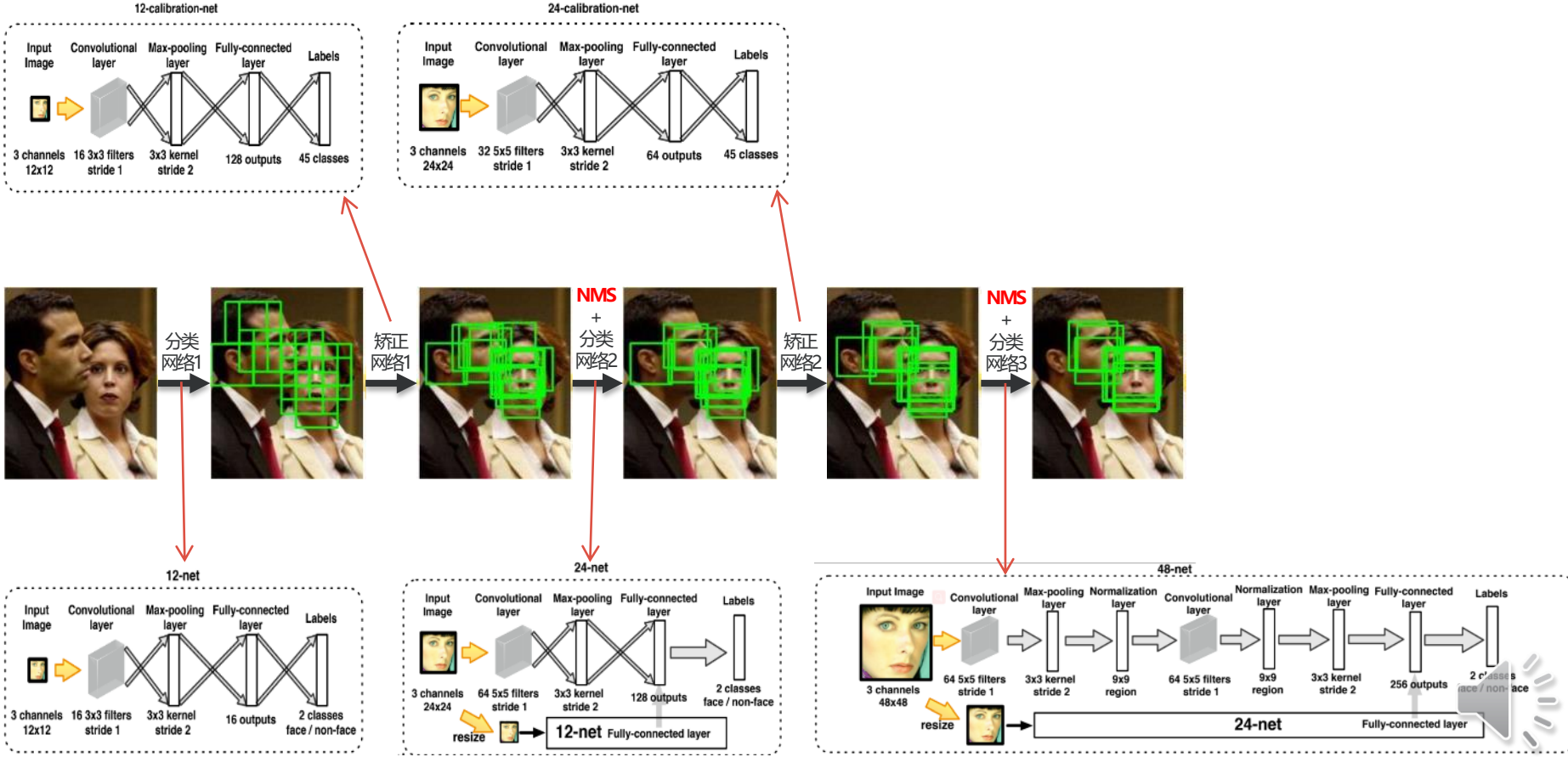
CascadeCNN人脸检测算法：整体框架



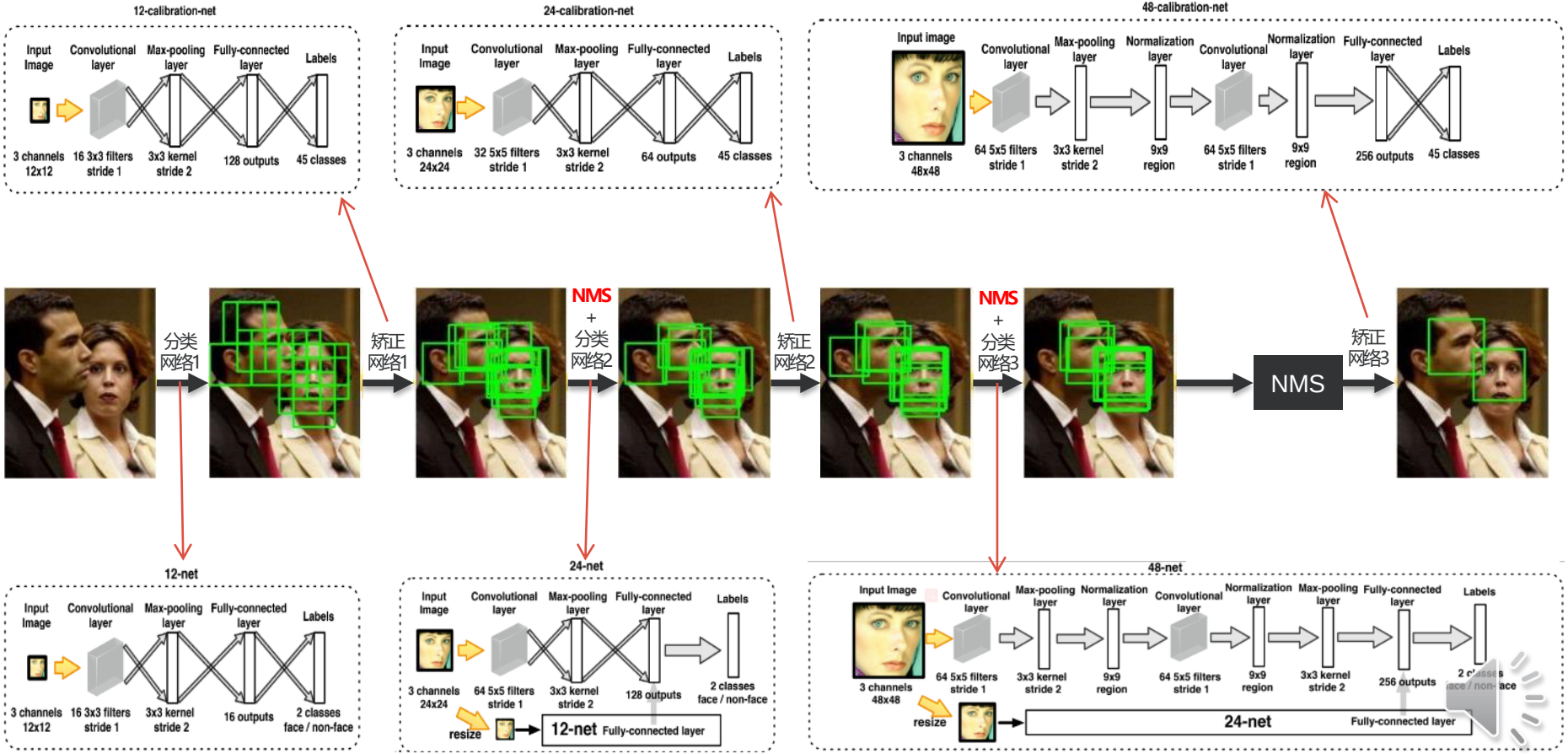
CascadeCNN人脸检测算法：整体框架



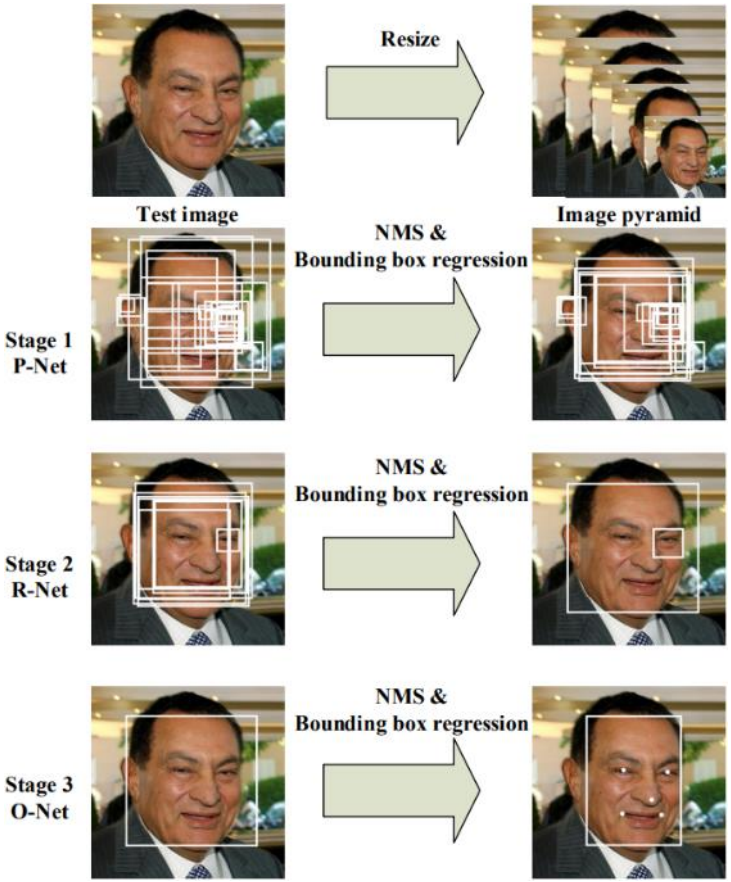
CascadeCNN人脸检测算法：整体框架



CascadeCNN人脸检测算法：整体框架



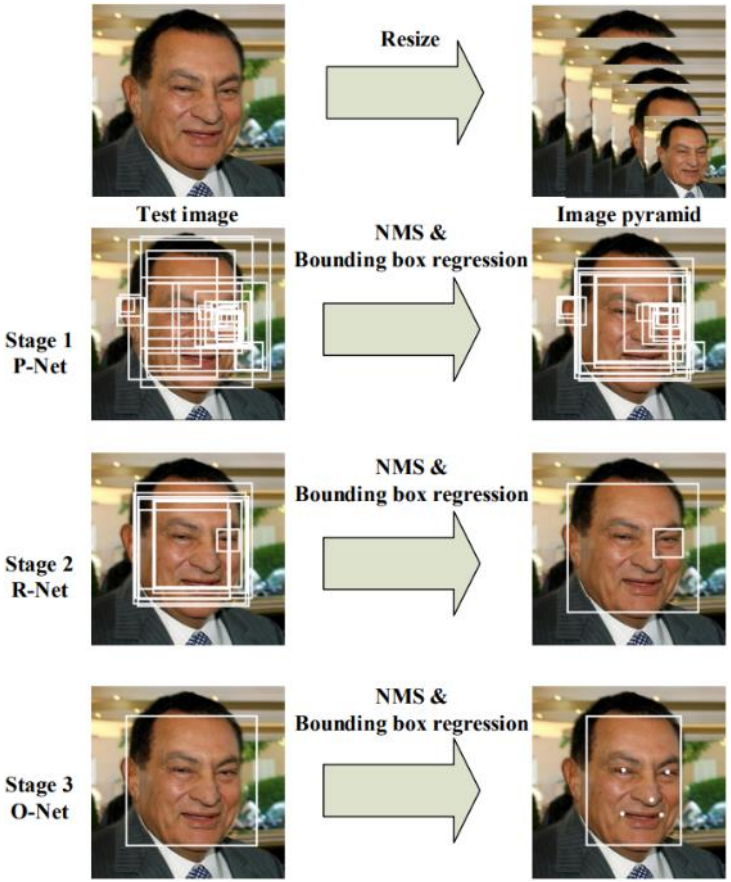
CascadeCNN人脸检测算法的改进：MTCNN



- MTCNN基于CascadeCNN，检测流程没有变化：
  - ① 滑窗操作：在图像上进行滑窗以遍历所有位置的人脸
  - ② 图像金字塔：对图像进行缩放以遍历所有大小的人脸
  - ③ 级联思想：MTCNN由P-Net, R-Net, O-Net级联构成

Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks, SPL

CascadeCNN人脸检测算法的改进：MTCNN

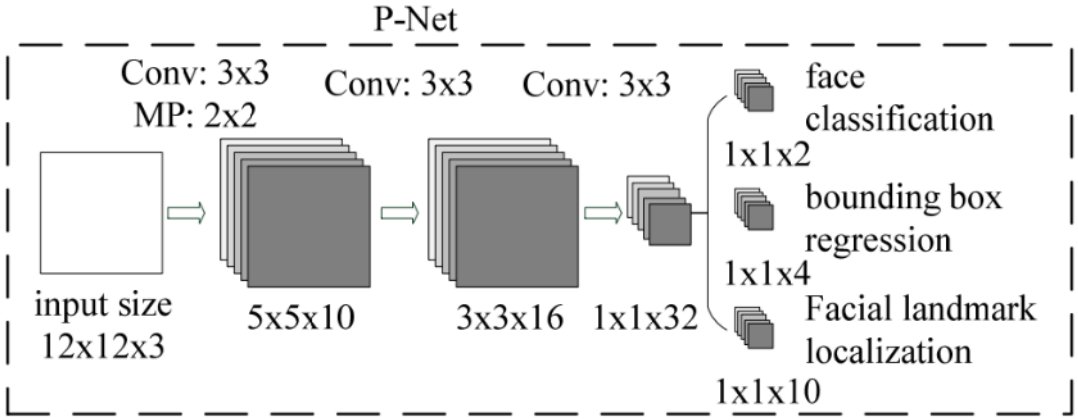


- MTCNN基于CascadeCNN，检测流程没有变化：
  - ① 滑窗操作：在图像上进行滑窗以遍历所有位置的人脸
  - ② 图像金字塔：对图像进行缩放以遍历所有大小的人脸
  - ③ 级联思想：MTCNN由P-Net, R-Net, O-Net级联构成

- MTCNN相对于CascadeCNN的主要变化：
  - ① MTCNN对网络进行改进（全连接层 -> 卷积层）
  - ② MTCNN把分类和矫正放在同一个网络中进行
  - ③ **MTCNN额外地输出5个人脸关键点**

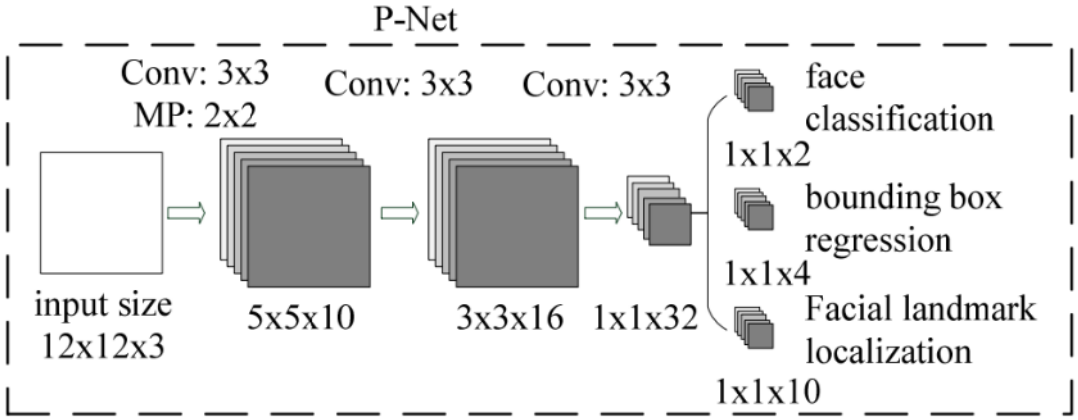
Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks, SPL

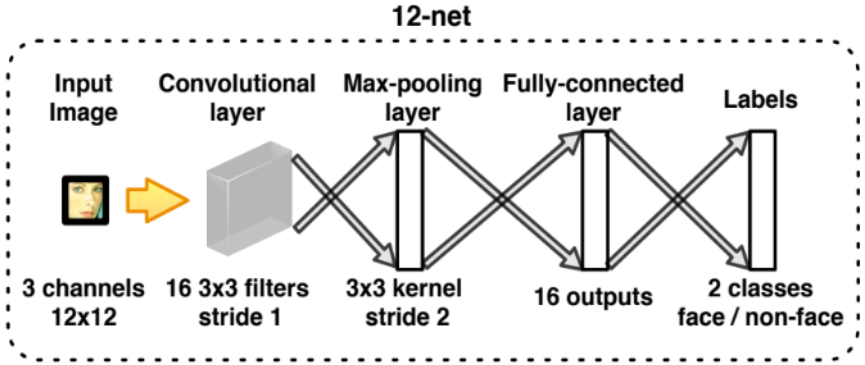
MTCNN人脸检测算法：P-Net

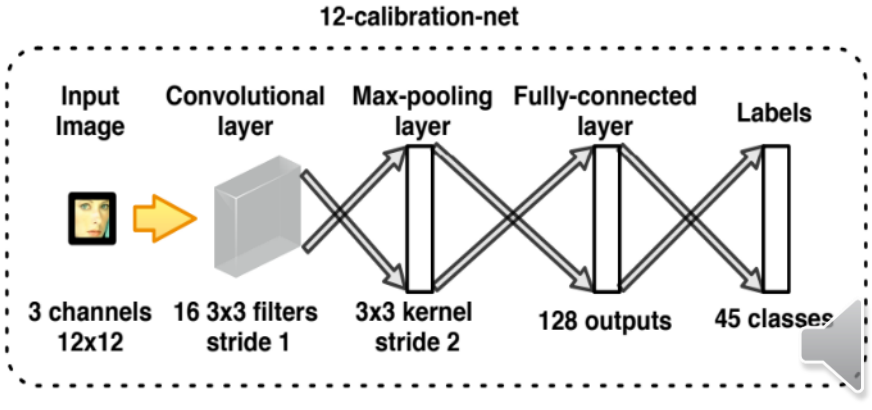


- 输入为12x12的RGB图像
- 网络由6个卷积层和1个池化层组成
- 全卷积网络，没有全连接层
- 3个分支：分类、矫正、关键点定位

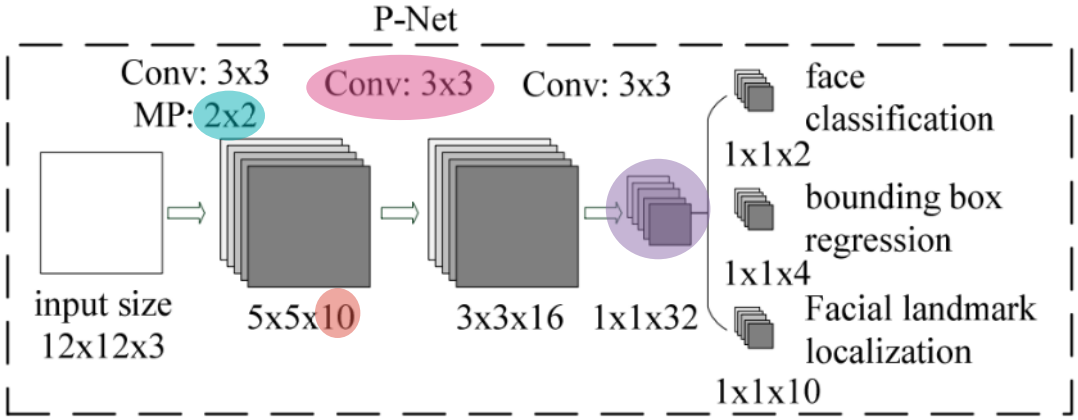
MTCNN人脸检测算法：P-Net





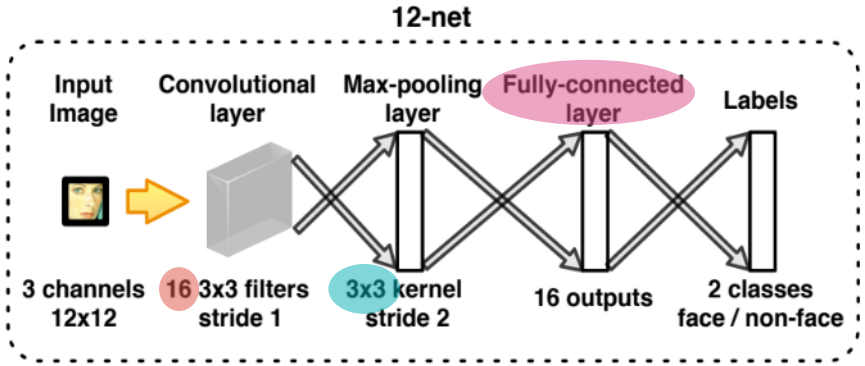


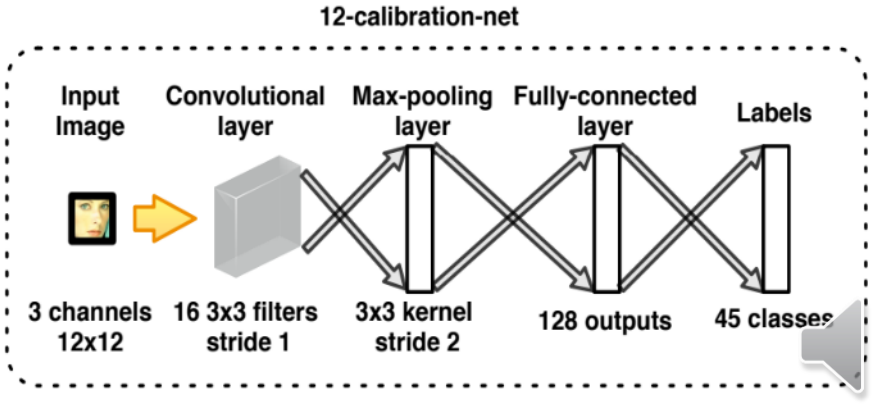
MTCNN人脸检测算法：P-Net



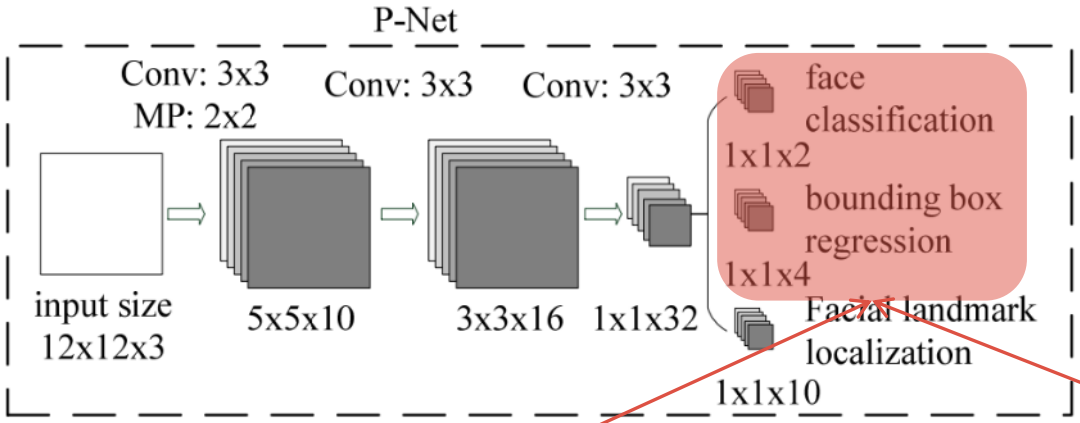
- MTCNN的主要改进点：

① 网络改进：通道数、卷积核、全连接层

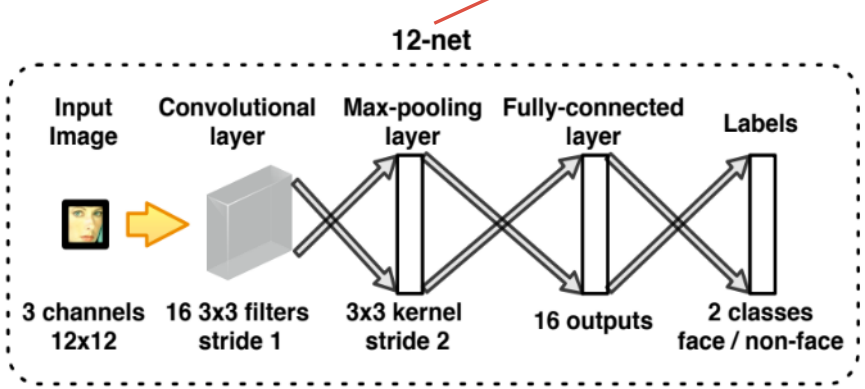


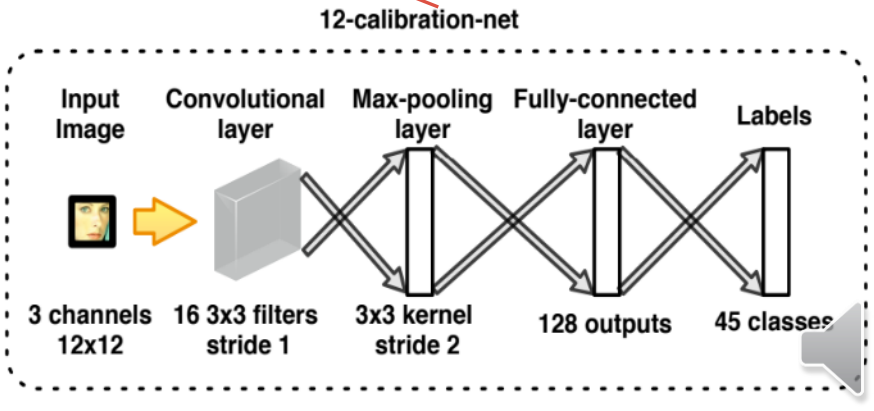


MTCNN人脸检测算法：P-Net

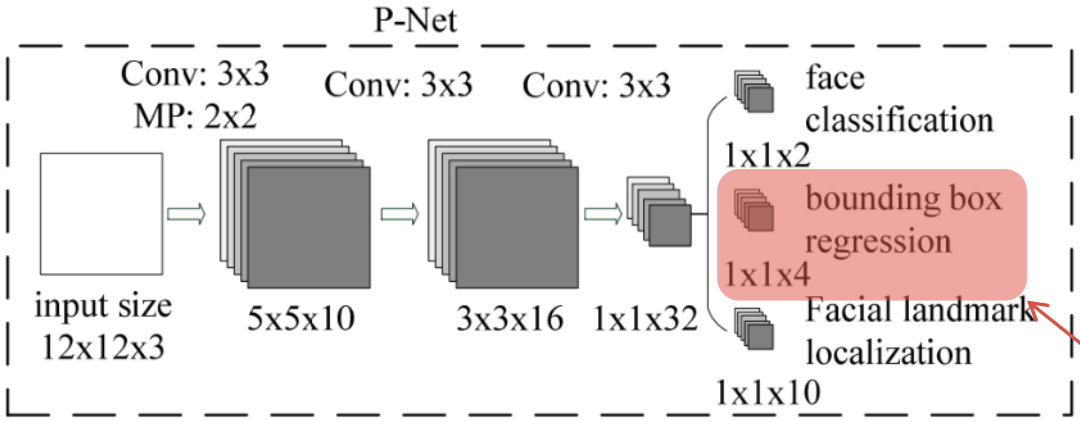


- MTCNN的主要改进点：
  - ① 网络改进：通道数、卷积核、全连接层
  - ② 把分类和矫正放在同一个网络中进行



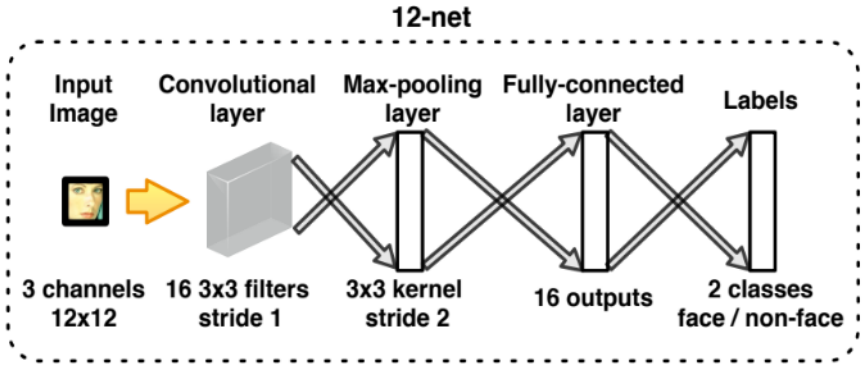


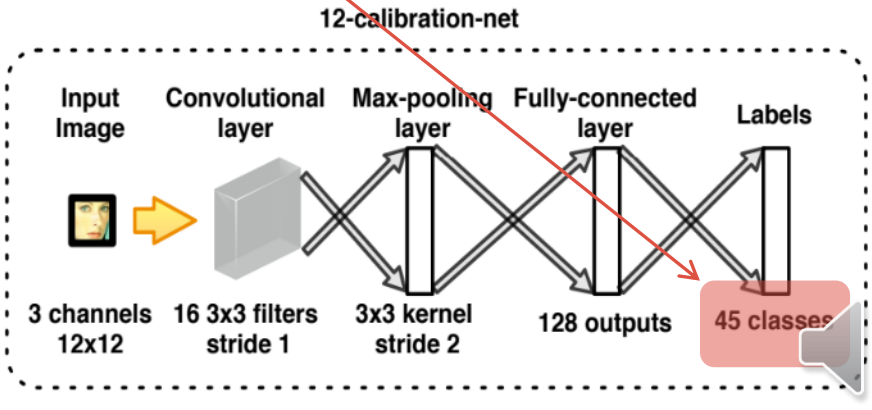
MTCNN人脸检测算法：P-Net



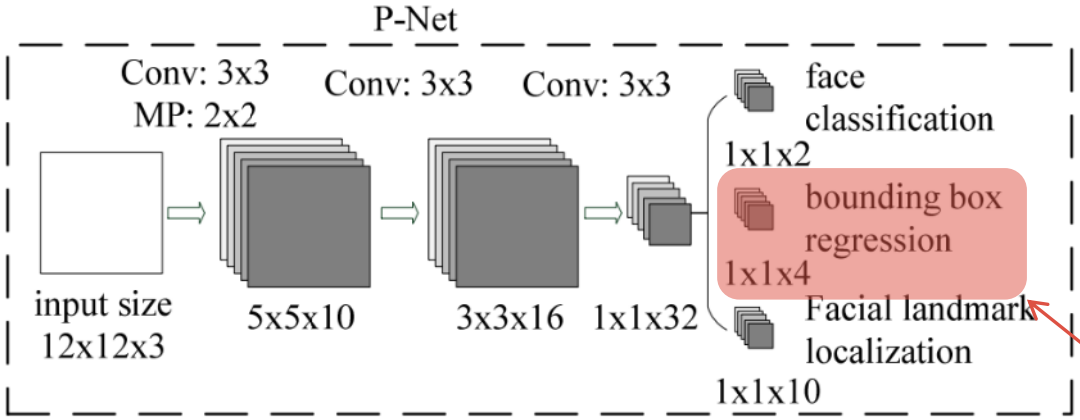
- MTCNN的主要改进点：
  - ① 网络改进：通道数、卷积核、全连接层
  - ② 把分类和矫正放在同一个网络中进行

位置矫正方式
分类 -> 回归



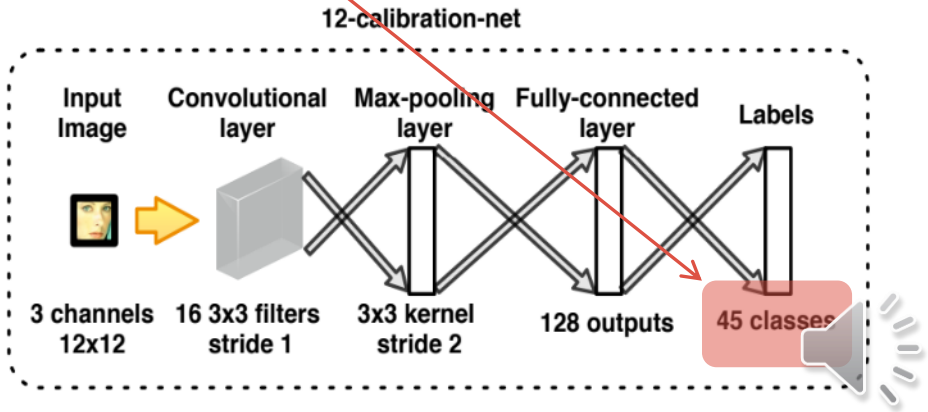


MTCNN人脸检测算法：P-Net



- MTCNN的主要改进点：
  - ① 网络改进：通道数、卷积核、全连接层
  - ② 把分类和矫正放在同一个网络中进行

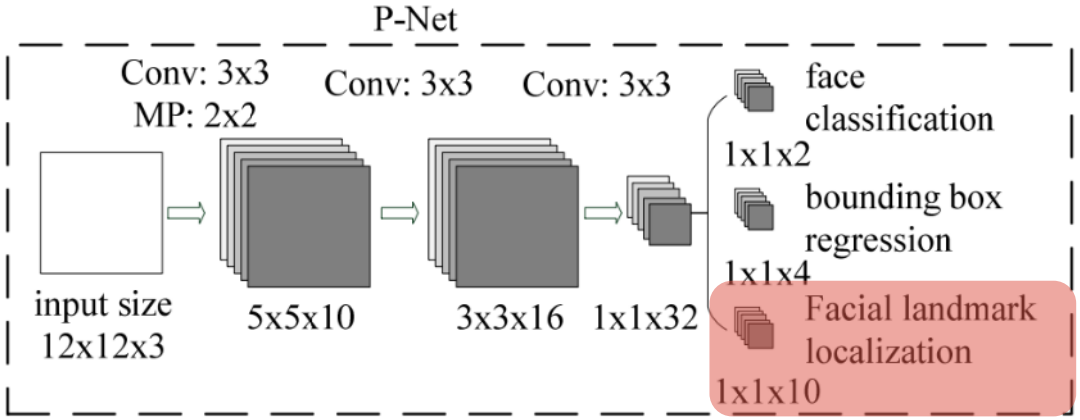
位置矫正方式
分类 -> 回归



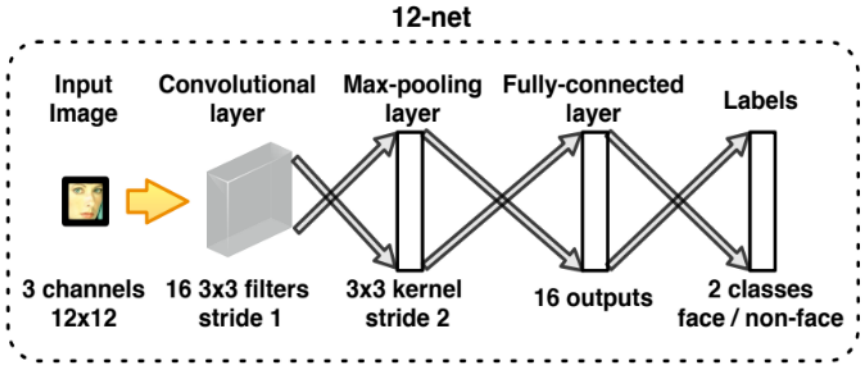
位置矫正方式

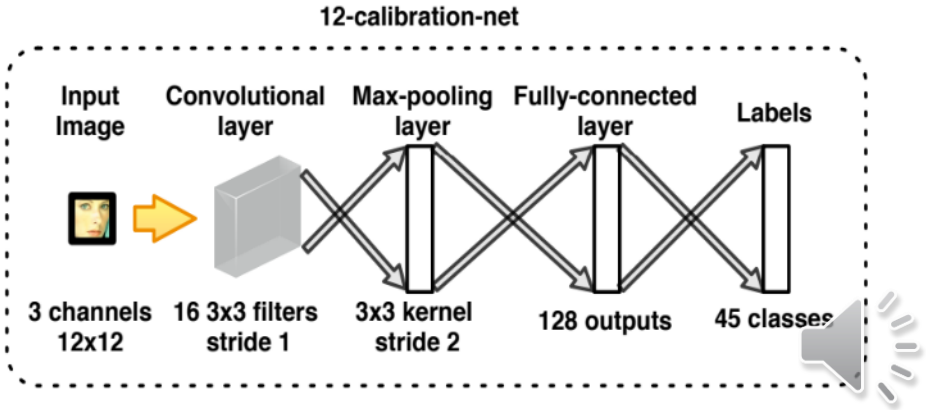
分类	回归
$(x-\frac{x_n w}{s_n}, y-\frac{y_n h}{s_n}, \frac{w}{s_n}, \frac{h}{s_n})$	$\Delta x_c^* = \frac{x_c - x_c^*}{w}$, $\Delta y_c^* = \frac{y_c - y_c^*}{h}$, $\Delta w^* = ln\frac{w^*}{w}$, $\Delta h^* = ln\frac{h^*}{h}$
$s_n \in \{0.83, 0.91, 1.0, 1.10, 1.21\}$ $x_n \in \{-0.17, 0, 0.17\}$ $y_n \in \{-0.17, 0, 0.17\}$.	

MTCNN人脸检测算法：P-Net

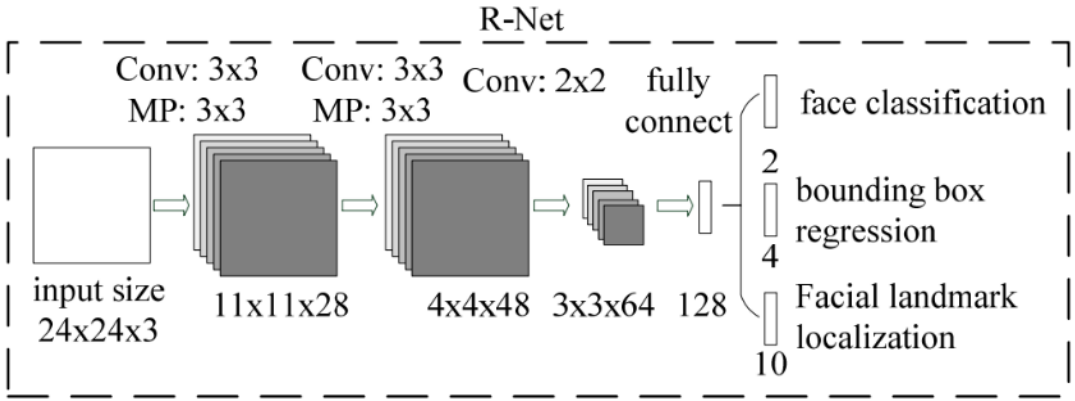


- MTCNN的主要改进点：
  - ① 网络改进：通道数、卷积核、全连接层
  - ② 把分类和矫正放在同一个网络中进行
  - ③ 额外地输出5个人脸关键点

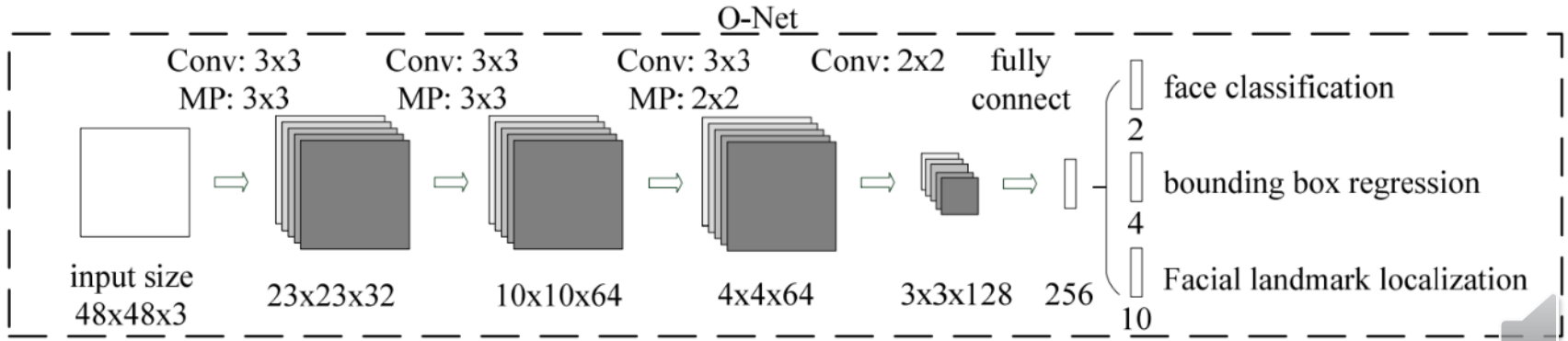




MTCNN人脸检测算法：R-Net和O-Net

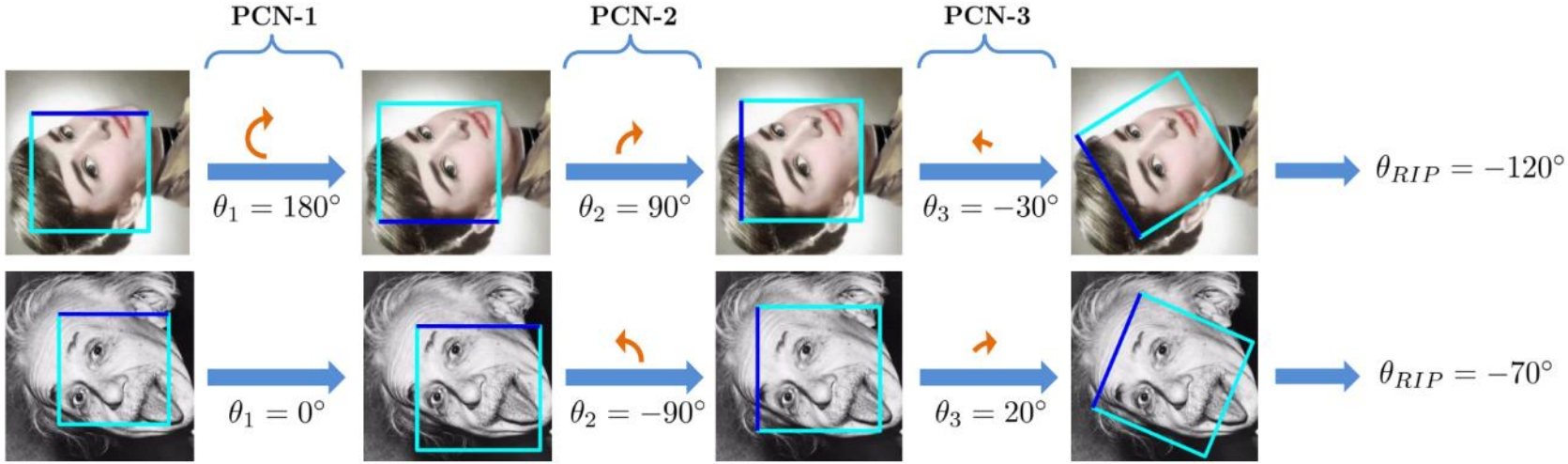


- 输入为24x24的RGB图像
- 网络由3个卷积层，4个全连接层和2个池化层组成
- 3个分支：分类、矫正、关键点定位



- 输入为48x48的RGB图像，4个卷积层+4个全连接层+3个池化层，3个分支 (分类/矫正/关键点定位)

MTCNN人脸检测算法的改进：PCN



- 相对于MTCNN, PCN的检测流程没变化：
  - ① 滑窗操作以遍历所有位置的人脸
  - ② 图像金字塔以遍历所有大小的人脸
  - ③ 级联思想：PCN-1, PCN-2, PCN-3

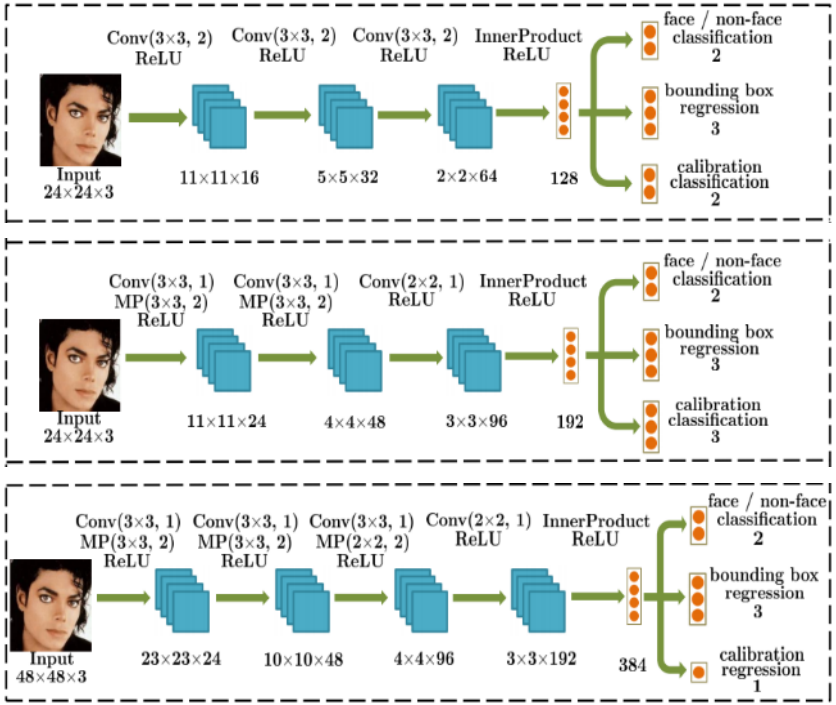
- 相对于MTCNN，PCN的主要变化：

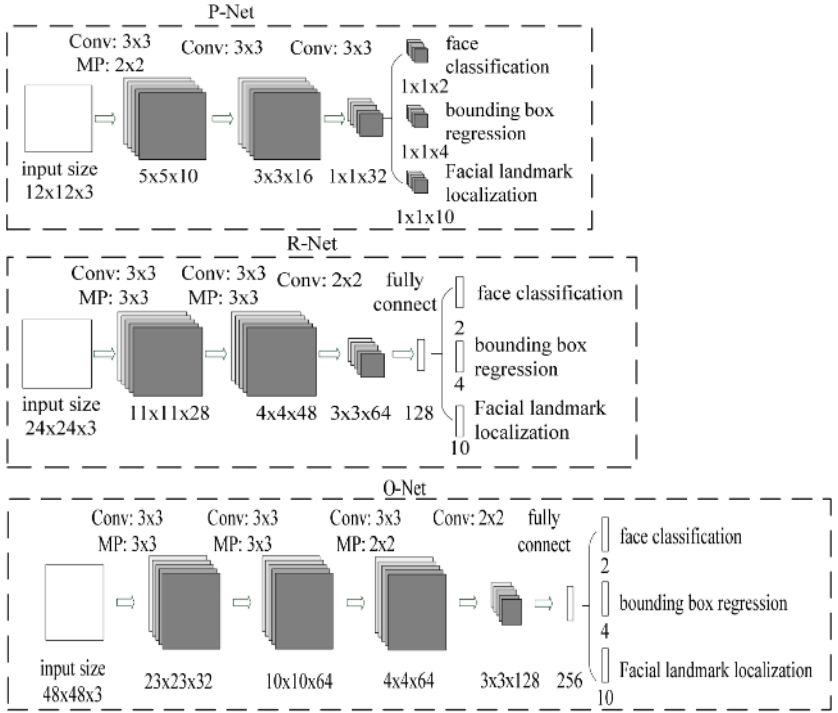
关键点定位分支

角度预测分支

Real-Time Rotation-Invariant Face Detection with Progressive Calibration Networks, CVPR 2018

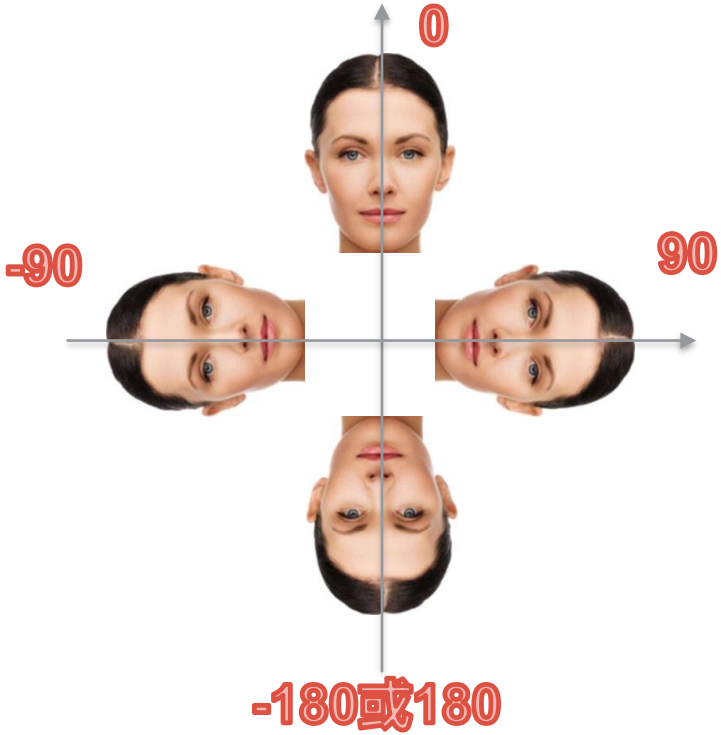
PCN人脸检测算法的改进



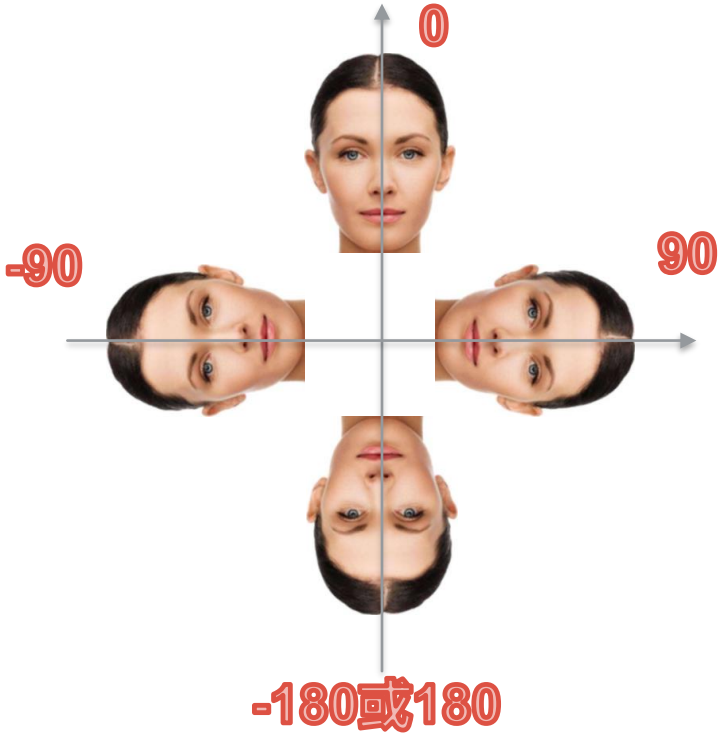


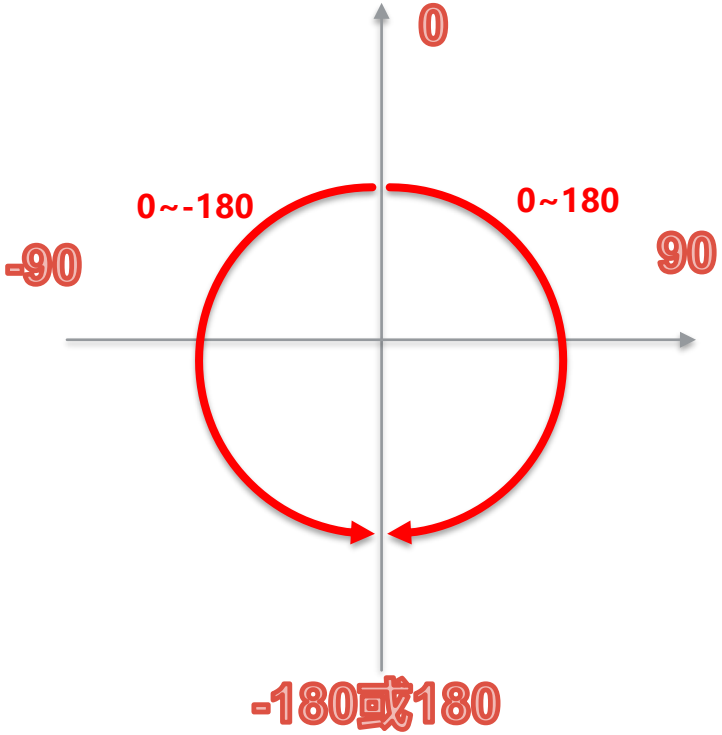
- 网络结构上有轻微的变化：网络的输入，池化层->带下采样的卷积层，通道数等
- 回归维度4->3：PCN的人脸检测结果是正方形
- 关键点定位分支->角度预测分支：由预测人脸的关键点变成预测人脸的角度

PCN人脸检测算法：检测流程

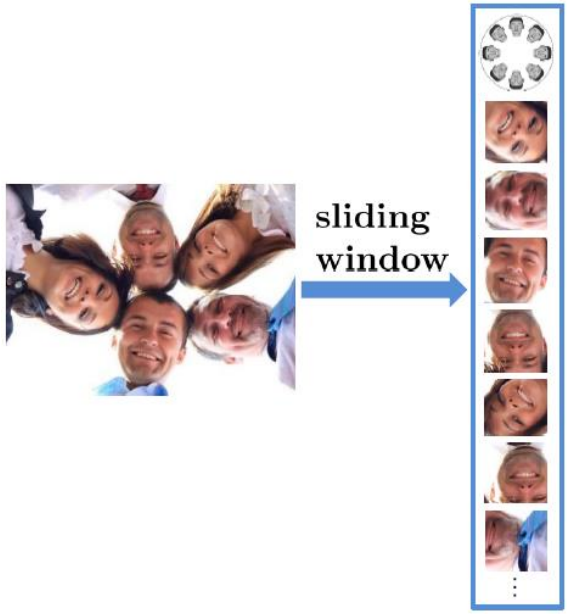


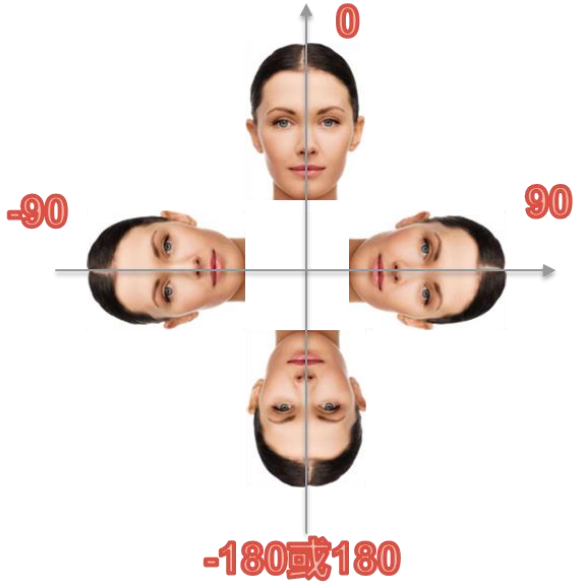
PCN人脸检测算法：检测流程





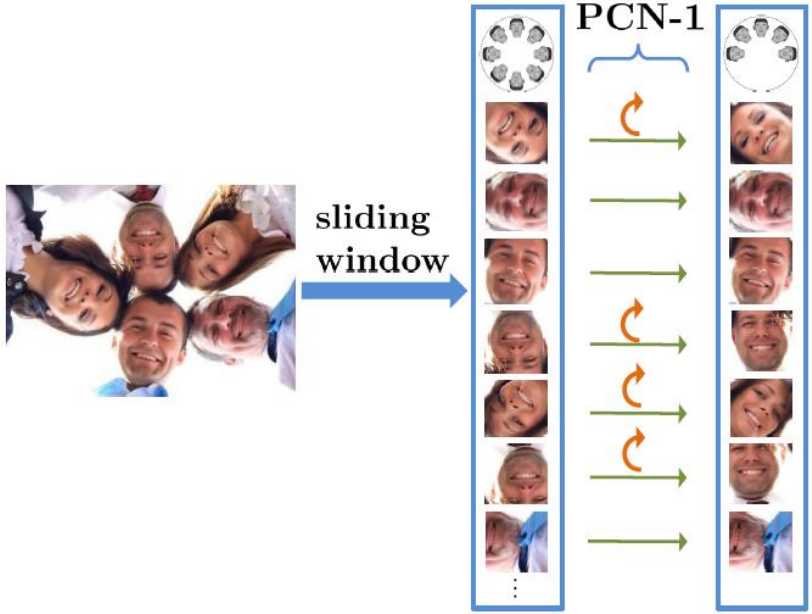
PCN人脸检测算法：检测流程

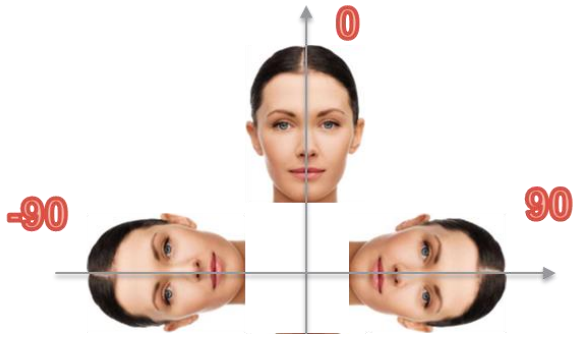




- 输入图像：人脸的角度在 -180~180

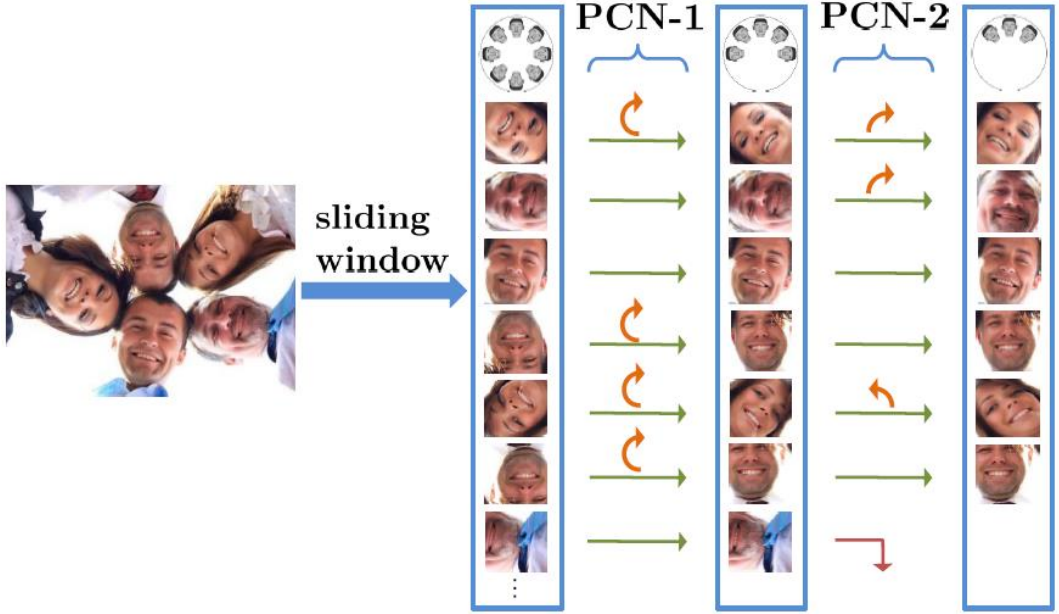
PCN人脸检测算法：检测流程

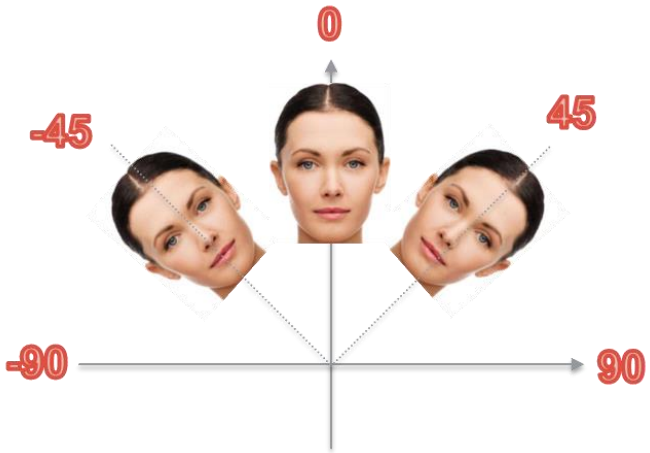




- 输入图像：人脸的角度在 -180~180
- PCN-1：角度预测分支为2分类任务，即人脸朝上还是朝下，并把朝下的人脸旋转180度，人脸的角度在-90~90

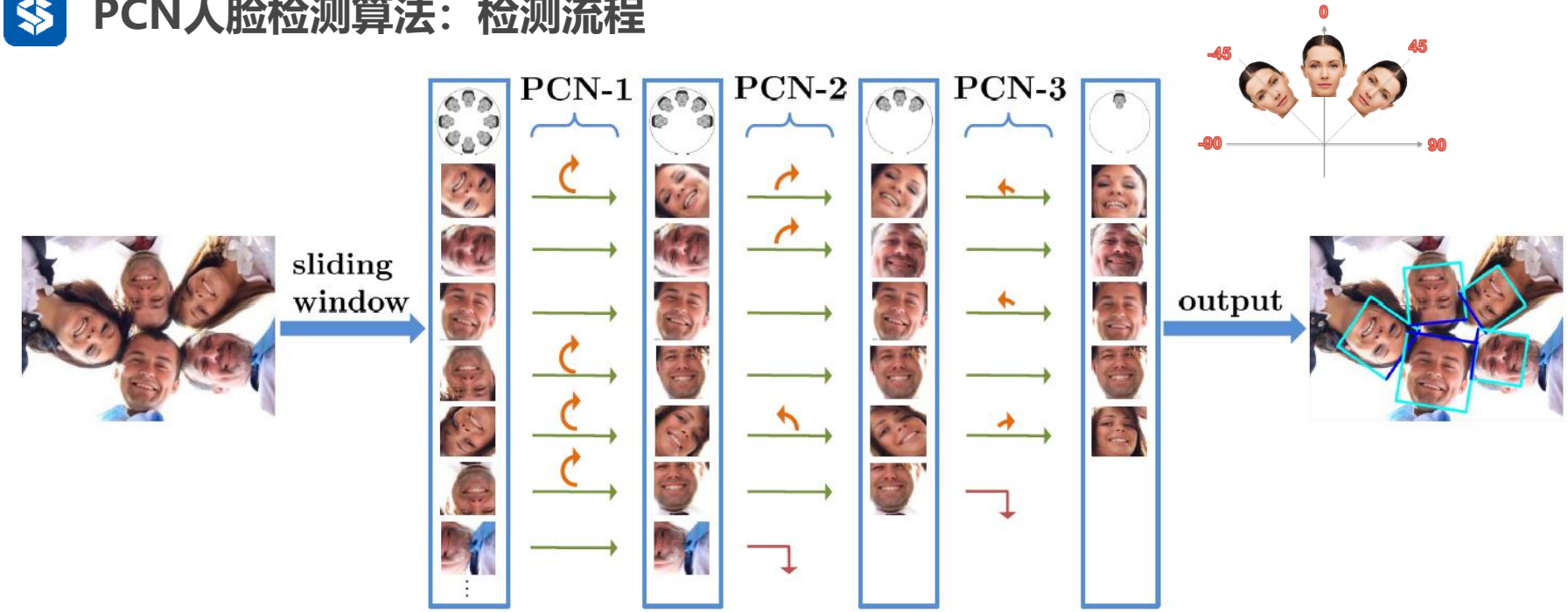
PCN人脸检测算法：检测流程





- 输入图像：人脸的角度在 -180~180
- PCN-1：角度预测分支为2分类任务，即人脸朝上还是朝下，并把朝下的人脸旋转180度，人脸的角度在-90~90
- PCN-2：角度预测分支为3分类任务，即人脸朝左、朝上、朝右，并把朝左/右的人脸旋转45度，人脸的角度在-45~45

PCN人脸检测算法：检测流程



- 输入图像：人脸的角度在 -180~180
- PCN-1：角度预测分支为2分类任务，即人脸朝上还是朝下，并把朝下的人脸旋转180度，人脸的角度在-90~90
- PCN-2：角度预测分支为3分类任务，即人脸朝左、朝上、朝右，并把朝左/右的人脸旋转45度，人脸的角度在-45~45
- PCN-3：角度预测分支为回归任务，在[-45, 45]范围内预测人脸的具体角度

深度学习早期人脸检测算法：总结

CascadeCNN → +关键点分支 → MTCNN → +角度预测分支 → PCN

滑窗	滑窗操作遍历所有的位置
金字塔	图像金字塔遍历所有的大小
级联	3个级联的阶段，人脸数量从多到少，人脸难度从易到难，网络结构从简单到复杂
深度学习	利用深度学习中的神经网络进行特征提取+分类器+矫正器+其他
优点	有着满足实际需求的精度，具备CPU实时的速度
缺点	检测过程仍然分为多个独立的阶段，训练过程比较繁琐
总结	深度学习早期人脸检测算法的代表，开创了深度学习时代下，人脸检测的一个派系，很多实际场景中都在使用该类型的算法

课程作业

- MTCNN人脸检测算法的测试报告

1. MTCNN参考代码链接（https://github.com/kuaikuaikim/DFace）
2. 按照安装教程，利用Anaconda配置好环境
3. （可选）按照教程中的步骤，训练MTCNN模型
4. 测试MTCNN算法的性能（上述代码提供了训好的模型，可跳过训练直接测试）
5. 撰写MTCNN的测试报告，包括CPU速度，GPU速度，与人脸个数的关系等

感谢各位聆听!

Thanks for Listening