BlazeFace

BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs

[BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs]

code

摘要

我们展示了 BlazeFace，这是一种轻量级且性能良好的人脸检测器(face detector)，专为移动 GPU 推理量身定制。它在旗舰设备上以 200–1000+ FPS 的速度运行。这种超实时性能使其能够应用于任何需要准确的面部感兴趣区域(facial region of interest)作为特定任务模型输入的增强现实管道(augmented reality)，例如 2D/3D 面部关键点(facial keypoint)或几何估计(geometry estimation)、面部特征(facial features)或表情分类(expression classification)和人脸区域分割(face region segmentation)。**我们的贡献包括一个受 MobileNetV1/V2 启发但与 MobileNetV1/V2 不同的轻量级特征提取网络(lightweight feature extraction network)、一个 从 Single Shot MultiBox Detector (SSD) 修改而来的GPU 友好的锚点方案(anchor scheme)，以及改进的联合分辨率策略(tie resolution strategy)替代非极大值抑制(non-maximum suppression)。**

1. Introcduction

近年来，通过对**深度神经网络(DNN)**中各种架构的改进，我们已经可以实现实时目标检测(real-time object detection)。在移动应用程序中，实时目标检测通常是视频处理流程中的第一步，接着是各种特定任务组件，例如分割，跟踪或几何推理。因此，目标检测模型推理必须尽可能快地运行，其性能最好能够达到远高于标准的实时基准。
我们提出了一种名为 BlazeFace 的新人脸检测框架，该框架是在**单镜头多盒检测器（Single Shot Multibox Detector, SSD）**框架上针对移动 GPU 推理进行的优化。我们的主要贡献包括：

- 关于推理速度(inference speed)
  - 一个专为轻量级目标检测而设计的在结构上与 MobileNetV1/V2 相关的**非常紧凑的特征提取器卷积神经网络**(CNN)。A very compact feature extractor convolutional neural network related in structure to MobileNetV1/V2, designed specifically for lightweight object detection.
  - 一种基于 SSD 的**新型GPU友好的锚定机制**，旨在提高 GPU 利用率。Anchors（锚点，SSD 术语中的先验）是预定义的静态边界框，作为网络预测调整和确定预测粒度的基础。A novel GPU-friendly anchor scheme modified from SSD, aimed at effective GPU utilization.Anchors, or priors in SSD terminology, are predefined static bounding boxes that serve as the basis for the adjustment by network predictions and determine the prediction granularity.
- 关于预测质量(prediction quality)

一种替代非最大抑制的**联合分辨率策略**，可在多预测之间实现更稳定、更平滑的联系分辨率。A tie resolution strategy alternative to non-maximum suppression that achieves stabler, smoother tie resolution between overlapping predictions.

2. Face detection for AR pipelines

虽然所提出的框架**适用于各种对象检测任务**，但在本文中，我们着重于在手机相机取景器中检测面部。由于不同的焦距和典型的被摄物体尺寸，我们分别为前置摄像头和后置摄像头构建了单独模型。
除了预测与轴对齐的脸部矩形(predicting axis-aligned face rectangles)外，我们的BlazeFace模型还生成**6个脸部关键点坐标**（用于眼睛中心，耳，嘴中心和鼻尖），使我们可以估计脸部旋转（滚动角度）(estimate face rotation (roll angle))。 这样可以将旋转的面部矩形传递到视频处理管道的后续任务特定阶段，从而减轻了后续处理步骤中显着平移(significant translation)和旋转不变性(rotation invariance)的要求（请参阅第5节）。

3. Model architecture and design

BlazeFace 模型架构主要是围绕下面讨论的四个重要设计考虑因素而构建的。

Enlarging the receptive field sizes

扩大感受野的大小。尽管大多数现代卷积神经网络(CNN)体系结构（包括MobileNet [3，9]版本）在模型图的各处都倾向于使用3×3卷积核(convolution kernels)，但我们注意到，**深度可分离卷积计算**(depthwise separable convolution computations)主要由它们的逐点部分(pointwise parts)决定。在一个$s \times s \times c$的输入张量上，应用可分离卷积操作，一个 $k \times k$ 的深度卷积涉及 s^2ck^2 次乘加运算，而后续的 1×1 卷积到 d 个输出通道由 s^2cd 次乘加运算组成，是深度阶段的 d/k^2 倍。

例如，实际上，在具有Metal Performance Shaders实现的Apple iPhone X上，对于16×56×128张量，在16位浮点算法中进行3×3深度卷积需要0.07毫秒，相比之下 128 到 128 通道的 1×1 卷积运算会慢 4.3 倍，即后续的点卷积操作需要 0.3 毫秒（由于固定成本和存储器访问因素导致的纯算术运算计数差）

该观察表明增加深度部分的核尺寸(increasing the kernel size of the depthwise part)性价比更高。
我们在模型架构中使用 5×5 内核，这样使得感受野达到指定大小所需的瓶颈(bottleneck)数量大大减少，得到的 BlazeBlock 有下图所示的两种结构：

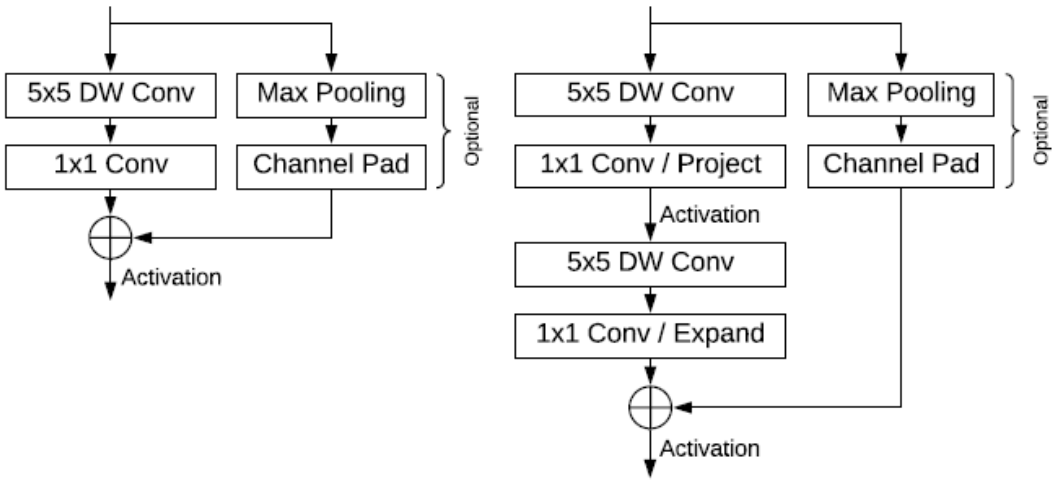


Figure 1. BlazeBlock (left) and double BlazeBlock

MobileNetV2瓶颈包含后续的**深度增加的扩展**(depth-increasing expansion)和**深度减少的投影逐点卷积**(depth-decreasing projection pointwise convolutions)，这些卷积被非线性分隔。为了适应中间张量中较少数量的通道，我们交换了这些级(stages)，以使瓶颈中的剩余连接以"扩展的"（提高的）通道分辨率(channel resolution)运行。

最后，深度卷积(depthwise convolution)的低开销使我们能够在这两个点式卷积(pointwise convolutions)之间引入另一个这样的层，从而进一步加快了接收场大小的进程(receptive field size progression)。这形成了**双BlazeBlock**的本质，该BlazeBlock用作BlazeFace的较高抽象级别层的选择瓶颈。

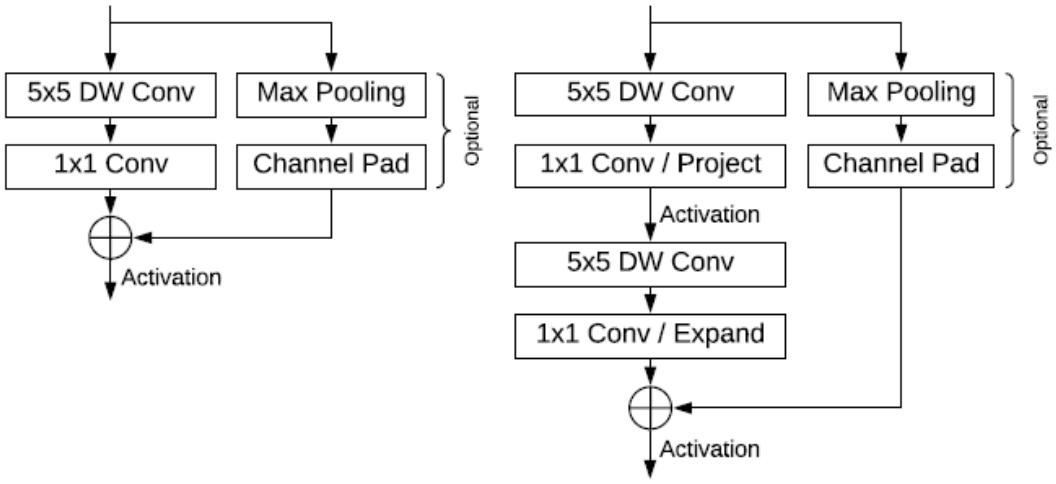


Figure 1. BlazeBlock (left) and double BlazeBlock

Feature extractor

特征提取器。一个具体的例子，我们专注于前置摄像头模型的特征提取器。该特征提取器必须考虑较小范围的目标尺度，因此它具有较低的计算需求。**提取器采用 128×128 像素的 RGB 输入，包括一个 2D 卷积和 5 个单 BlazeBlock 和 6 个双 BlazeBlock 组成，完整布局见下表。最大张量深度（通道分辨率 channel resolution）为 96，而最低空间分辨率(spatial resolution)是8×8（与 SSD 相比，它将分辨率一直降低到 1×1）。**

Appendix A. Feature extraction network architecture

Layer/block	Input size	Conv. kernel sizes
Convolution	128×128×3	5×5×3×24 (stride 2)
Single BlazeBlock	64×64×24	5×5×24×1 1×1×24×24
Single BlazeBlock	64×64×24	5×5×24×1 1×1×24×24
Single BlazeBlock	64×64×24	5×5×24×1 (stride 2) 1×1×24×48
Single BlazeBlock	32×32×48	5×5×48×1 1×1×48×48
Single BlazeBlock	32×32×48	5×5×48×1 1×1×48×48
Double BlazeBlock	32×32×48	5×5×48×1 (stride 2) 1×1×48×24 5×5×24×1 1×1×24×96
Double BlazeBlock	16×16×96	5×5×96×1 1×1×96×24 5×5×24×1 1×1×24×96
Double BlazeBlock	16×16×96	5×5×96×1 1×1×96×24 5×5×24×1 1×1×24×96
Double BlazeBlock	16×16×96	5×5×96×1 (stride 2) 1×1×96×24 5×5×24×1 1×1×24×96
Double BlazeBlock	8×8×96	5×5×96×1 1×1×96×24 5×5×24×1 1×1×24×96
Double BlazeBlock	8×8×96	5×5×96×1 1×1×96×24 5×5×24×1 1×1×24×96

		1×1×24×96

Table 4. BlazeFace feature extraction network architecture

Anchor scheme

锚定方案。类似于 SSD 的目标检测模型依赖于预定义的固定大小的基础边界框，称为先验机制，或 Faster-R-CNN 术语中的锚点(anchors)。为每个锚预测一组回归（可能还包括分类）参数，例如中心偏移量(center offset)和尺寸调整(dimension)。它们用于将预定义的锚位置调整为紧密的边界矩形。

通常的做法是根据目标比例范围在多个分辨率级别定义锚点(anchors)，同时**下采样**(Aggressive downsampling)也是计算资源优化的手段。典型的 SSD 模型使用 1×1,2×2,4×4,8×8 和 16×16 特征映射大小的预测。然而，**金字塔池化网络 (Pooling Pyramid Network, PPN)** 架构的成功意味着在特征图达到某个特征映射分辨率后，将产生大量额外的计算。
相比于 CPU 计算，GPU 独有的关键特性是调度特定层计算会有一个显著的固定成本(a noticeable fixed cost of dispatching a particular layer computation)，这对于流行的 CPU 定制架构固有的深度低分辨率层(deep low-resolution layers inherent)而言非常重要。例如，在一个实验中我们观察到 MobileNetV1 推理时间需要 4.9 毫秒，而在实际 GPU 计算中花费 3.9 毫秒。

考虑到这一点，我们采用了另一种锚定方案，该方案停留在 **8×8 特征图尺寸**(feature map dimensions)处而无需进一步**下采样(downsampling)**（图 2）。我们已经**将 8×8,4×4 和 2×2 分辨率中的每个像素的 2 个锚点替换为 8×8 的 6 个锚点。**由于人脸长宽比的变化有限，因此发现将锚固定为 1： 1 纵横比足以进行精确的面部检测。(Due to the limited variance in human face aspect ratios, limiting the anchors to the 1:1 aspect ratio was found sufficient for accurate face detection.

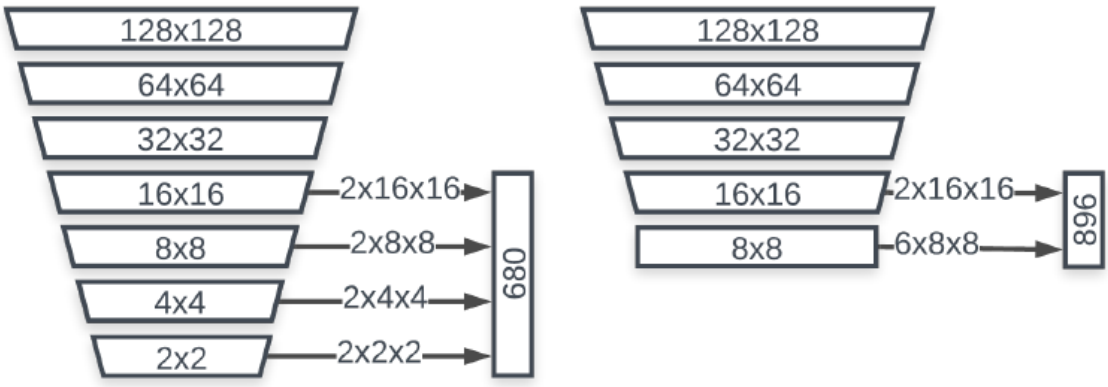


Figure 2. Anchor computation: SSD (left) vs. BlazeFace

Post-processing

后期处理。由于我们的特征提取器未将分辨率降低到 8×8 以下，因此给定目标重叠的锚点数量会随目标尺寸的增加而显著增加。在典型的非最大抑制(non-maximum suppression)方案中，只有一个锚点被选中作为算法的输出。这样的模型应用于后续视频人脸预测时，预测结果将在不同锚之间波动并且在时间序列上检测框上持续抖动（人类易感噪声）。(When such a model is applied to subsequent video frames, the predictions tend to fluctuate between different anchors and exhibit temporal jitter (humanperceptible noise).

为了最小化这种现象，我们用一种混合策略(blending strategy)代替抑制算法(suppression algorithm)，**该策略以重叠预测之间的加权平均值估计边界框的回归参数(estimates the regression parameters of a bounding box as a weighted mean between the overlapping predictions**)，它几乎不会产生给原来的 NMS 算法带来额外成本。对于人脸检测任务，此调整使准确度提高 10%。

我们**通过连续输入目标轻微偏移的图像来量化抖动量**(We quantify the amount of jitter by passing several slightly offset versions of the same input image into the network and observing how the model outcomes (adjusted to account for the translation) are affected.
)，并观察模型结果（受偏移量影响）如何受到影响。在联合分辨率策略(tie resolution strategy)修改之后，**抖动量**(jitter metric)（定义为原始输入和移位输入的预测之间的均方根差）在我们的前置摄像头数据集上下降了 40%，在包含较小人脸的后置摄像头数据集上下降了 30%。

4. Experiments

我们在 66K 图像的数据集上训练我们的模型。为了评估实验结果，我们使用了由 2K 图像组成的地理位置多样数据集。对于前置摄像头模型，它只考虑占据图像区域的 20%以上的面部，这是由预期的用例决定的（后置摄像头型号的阈值为 5%）。

回归参数误差(regression parameter errors)采用眼间距离（ inter-ocular distance ,IOD）进行尺度不变性(scale invariance)归一化，中值绝对误差为 IOD 的 7.4%。通过上述程序评估的抖动度量是 IOD 的 3%。

表1 显示了所提出的正面人脸检测网络的平均精度（ average precision, AP）度量（标准 0.5 交叉联合边界框匹配阈值）和移动 GPU 推理时间，并将其与基于 MobileNetV2 的目标检测器（MobileNetV2-SSD）进行了比较。我们在 16 位浮点模式下使用 TensorFlow Lite GPU 作为推理时间评估的框架。

Model	Average Precision	Inference Time, ms (iPhone XS)
MobileNetV2-SSD	97.95%	2.1
Ours	**98.61%**	**0.6**

Table 1. Frontal camera face detection performance

Table 2 gives a perspective on the GPU inference speed for the two network models across more flagship devices.

Device	MobileNetV2-SSD, ms	Ours, ms
Apple iPhone 7	4.2	**1.8**
Apple iPhone XS	2.1	**0.6**
Google Pixel 3	7.2	**3.4**
Huawei P20	21.3	**5.8**
Samsung Galaxy S9+ (SM-G965U1)	7.2	**3.7**

Table 2. Inference speed across several mobile devices

表3展示了由于模型尺寸较小引起的回归参数预测质量的退化程度(the amount of degradation)。如下一节所述，这不一定会导致整个 AR 管道质量的成比例降低。

Model	Regression error	Jitter metric
MobileNetV2-SSD	**7.4%**	**3.6%**
Ours	10.4%	5.3%

Table 3. Regression parameters prediction quality

5. Applications

上述模型可以在完整图像或视频帧上运行，并且可以作为几乎任何与人脸相关的计算机视觉应用的第一步，例如 2D / 3D 人脸关键点、轮廓或表面几何估计、面部特征或表情分类以及人脸区域分割(2D/3D facial keypoints, contour, or surface geometry estimation, facial features or expression classification, and face region segmentation)。因此，计算机视觉流程中的后续任务可以根据适当的面部剪裁(facial crop)来定义。结合 BlazeFace 提供的少量面部关键点估计，此结果也可以旋转，这样图像中的面部是居中的、标准化的并且滚动角接近于零。这从特定于任务的模型中消除了对显着平移(significant translation)和旋转不变性(rotation invariance)的要求，从而实现了更好的计算资源分配。

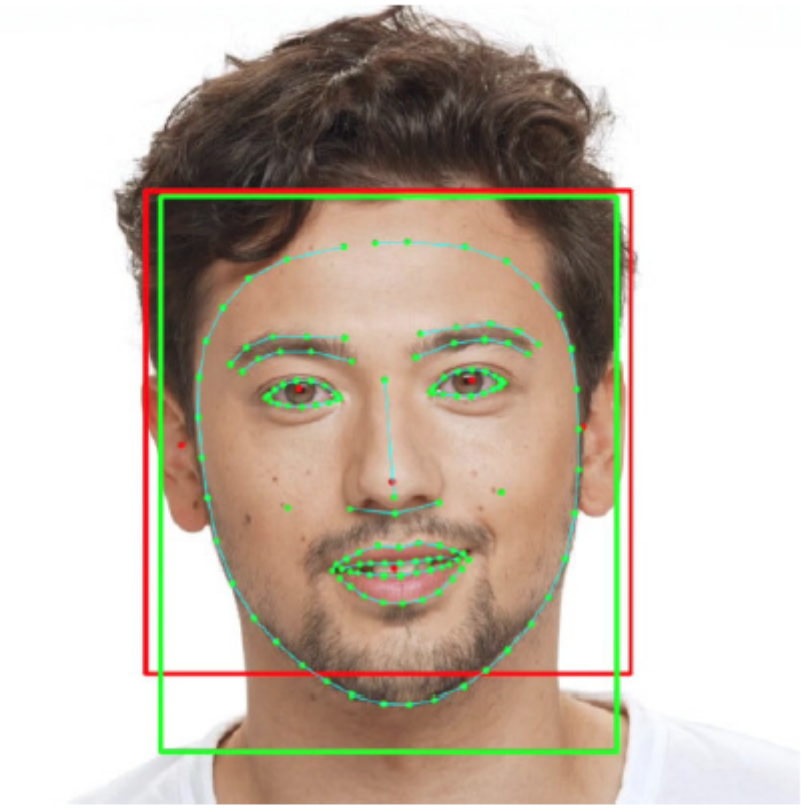


Figure 3. Pipeline example (best viewed in color). Red: BlazeFace output. Green: Task-specific model output.

我们通过一个具体的人脸轮廓估计示例来说明这种方法。在图3 中，我们展示了 BlazeFace 的输出，即**预测的边界框**和面部的 **6 个关键点**（红色）如何通过一个更复杂的人脸轮廓估计模型来进一步细化，并将其应用于扩展的结果。详细的关键点可以产生更精细的边界框估计（绿色），并在不运行人脸检测器的情况下重新用于后续帧中的跟踪。为了检测该计算节省策略的故障，该模型还可以检测面部是否存在所提供的矩形裁剪中合理地对齐。每当违反该条件时，BlazeFace 人脸检测器将再次在整个视频帧上运行。
本文描述的技术正在推动手机上主要的 AR 自我表达应用程序和 AR 开发人员 API。(The technology described in this paper is driving major AR self-expression applications and AR developer APIs on mobile phones.)

总结 --- 主要创新点

- 改进网络，增大感受野。基于Mobilenet v1/v2，改进网络，提出一种轻量级的特征提取网络(Single BlazeBlock and Double BlazeBlock)
- 特征提取网络设计
- 改进Anchor。基于SSD的Anchor设计，改进Anchor的梯度金字塔结构，使其更加适应于GPU运算，从而达到提速的目的。
- 使用blending策略（tie resolution）替换非极大抑制（NMS）。

设a,b两个矩形框为人脸框的重叠框，经由网络输出$a(bbox, score), b(bbox, score)$，并且$a_score > b_score$，$IoU(a, b) > iou_nms_threhold$

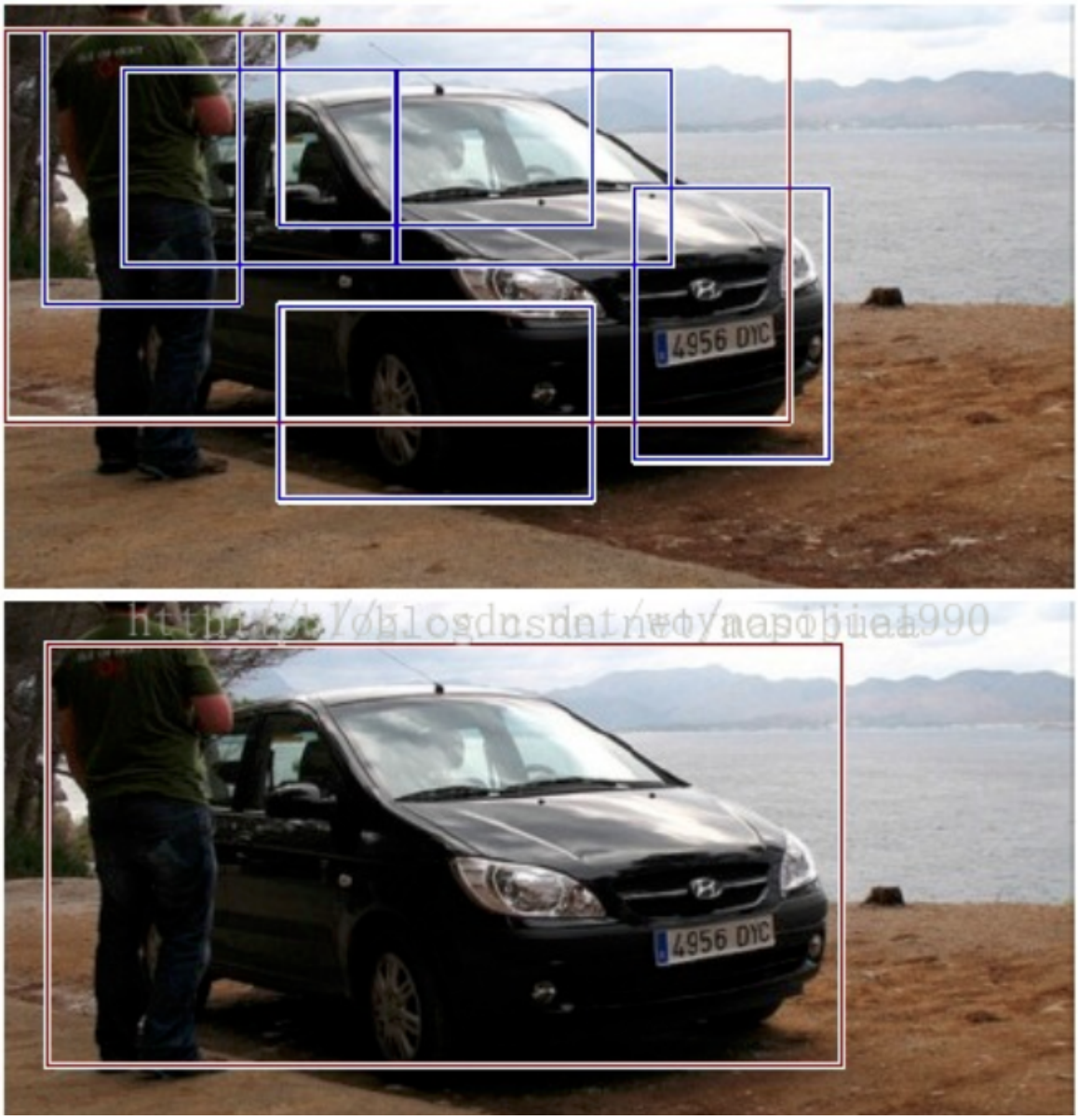
传统的NMS方法：$b(score)$ 设置为0，$bbox$直接选取$a(bbox)$

Blending: $(weigth_a * bbox_a) + (weight_b * bbox_b) = bbox$

其中 $weidth_a = a(score)/[a(score) + b(score)]$

扩展--- NMS

NMS即non maximum suppression即非极大抑制，就是抑制不是极大值的元素，搜索局部的极大值。在最近几年常见的物体检测算法（包括rcnn、sppnet、fast-rcnn、faster-rcnn等）中，最终都会从一张图片中找出很多个可能是物体的矩形框，然后为每个矩形框为做类别分类概率。



就像上面的图片一样，定位一个车辆，最后算法就找出了一堆的方框，我们需要判别哪些矩形框是没用的。
所谓非极大值抑制：先假设有6个矩形框，根据分类器类别分类概率做排序，从小到大分别属于车辆的概率分别为A<B<C<D<E<F。

(1) 从最大概率矩形框F开始，分别判断A、B、C、D、E与F的重叠度IOU是否大于某个设定的阈值;

(2) 假设B、D与F的重叠度超过阈值，那么就扔掉B、D；并标记第一个矩形框F，是我们保留下来的。

(3) 从剩下的矩形框A、C、E中，选择概率最大的E，然后判断A、C与E的重叠度，重叠度大于一定的阈值，那么就扔掉；并标记E是我们保留下来的第二个矩形框。

(4) 重复这个过程，找到所有被保留下来的矩形框。