PPT

2022年5月13日 9:15

ViDT

2022年5月13日 9:13

西安电子科技大学
XIDIAN UNIVERSITY

ViDT: An Efficient and Effective Fully Transformer-based Object Detector

Hwanjun Song[1], **Deqing Sun**[2], **Sanghyuk Chun**[1], **Varun Jampani**[2], **Dongyoon Han**[1], **Byeongho Heo**[1], **Wonjae Kim**[1], **Ming-Hsuan Yang**[2,3,4]
[1]NAVER AI Lab [2]Google Research [3]University of California at Merced [4]Yonsei University
{hwanjun.song, sanghyuk.c, dongyoon.han, bh.heo, wonjae.kim}@navercorp.com
{deqingsun, varunjampani}@google.com, mhyang@ucmerced.edu

ICLR 2022

李娜

DETR

第一个基于Transformer的端到端的目标检测模型

动机：

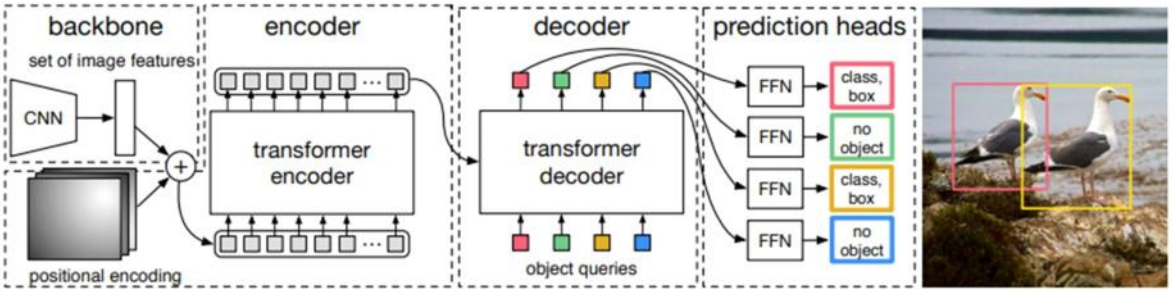
Anchor generation、NMS

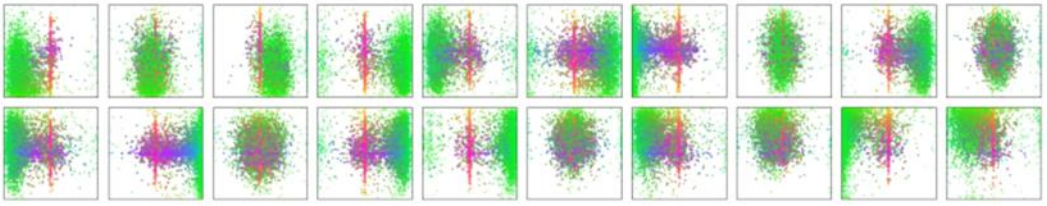
性能取决于特定的后处理步骤，难以进行完全端到端的训练

DETR实现了真正的端到端的训练：

使用Transformer的Encoder和Decoder,消除精心设计的组件,这种结构作为Neck组件，用于连接以CNN为主体的特征提取器和检测头

DETR





- 缺陷：
  - 收敛慢：传统37个epoch可以达到的效果，DETR需要500个epoch
  - 小目标检测效果差：用的是最后一层特征图生成的嵌入向量，小目标信息较少

西安电子科技大学
XIDIAN UNIVERSITY

Deformable DETR

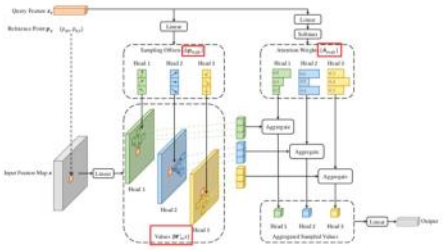
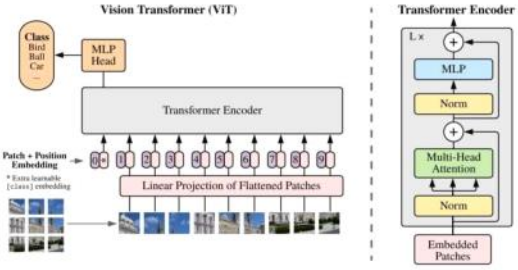


Figure 2: Illustration of the proposed deformable attention module.

- 改进的Deformable DETR
  - 可行变注意力+多尺度特征图
    - 可行变注意力
      - 每一个Q不需要于所有的K进行计算
      - 以当前Q为参考点，通过线性映射计算偏移量，选择当前Q附近的M个点为采样点，计算注意力
      - $$\mathrm{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m [\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk})]$$
    - 多尺度特征图
      - 选取ResNet多个Stage的输出特征图生成嵌入向量，每一层的嵌入向量都添加了对应层的【层】标记
      - 训练过程中随机初始化【层】标记，Encoder-Decoder协同训练

ViT

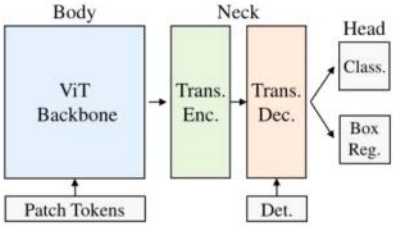
第一个完全基于Transformer的图像分类模型，实现了图像分类基准中最先进的分类结果



- VIT
  - 完全基于Transformer的图像分类模型
    - 额外添加了一个【CLS】Token负责类别预测
    - ViT也证明不需要额外的归纳偏置（先验知识：图像结构信息，卷积的平移不变新），模型依旧可以学习到有效的特征信息
  - 缺陷：复杂度较高，与图像大小呈二次增长

Swin Transformer引入了支持局部注意力的移动窗口操作，解决了计算复杂度高的问题

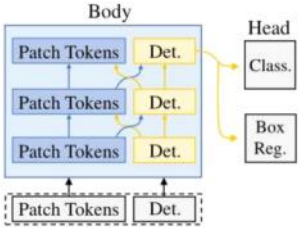
DERT+ViT



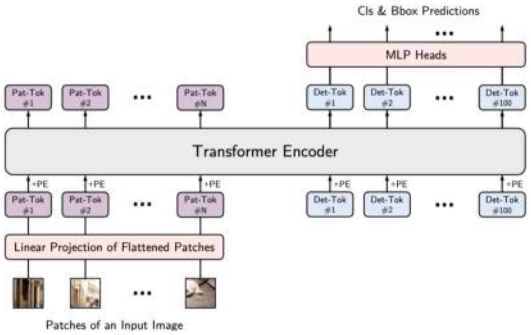
(a) DETR (ViT).

- DETR（ViT）
  - 利用ViT代替ResNet提取图像特征，利用Encoder-Decoder作为Neck组件进行编码解码，Object Queries对应的Tokens通过FFN进行一一对应的最终的检测
  - 缺陷：
    - ViT的缺陷
    - Neck组件中Encoder-Decoder的注意力计算增加检测器的计算开销
    - 难扩展

YOLOS

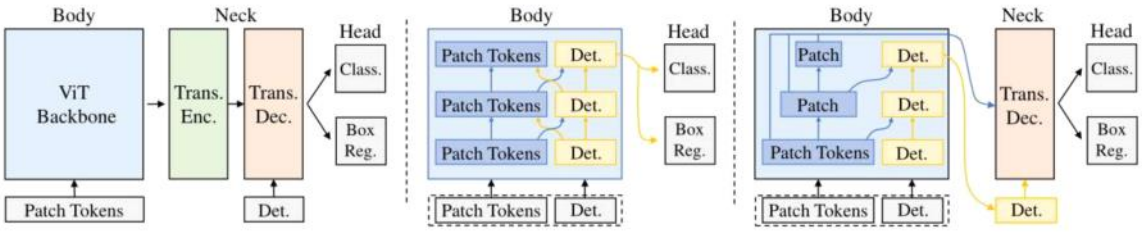


(b) YOLOS.



- YOLOS
  - 基于ViT的目标检测器
    - 在Encoder的输入中添加了【DET】Tokens，实现了无Neck组件的目标检测
      - 因为【DET】组件在ViT的Encoder中已经进行了全局注意力的交互，可直接用于检测，因此不再需要Decoder，从而实现无Neck组件的检测
    - YOLOS实现了高效的计算，同时也证明2D目标检测可以通过序列-to-序列的方式实现
    - 缺陷：
      - 继承了ViT的缺陷：高复杂度是由于全局注意力
      - 无Neck组件：无法使用多尺度特征提升检测性能

ViDT



(a) DETR (ViT). (b) YOLOS. **(c) ViDT (Ours).**

- 重新配置的注意力模块(RAM): 将 Swin Transformer 扩展为一个独立的目标检测器
- Encoder-Free Neck组件: 多尺度特征和辅助损失函数，在不增加计算负载的情况下提高检测性能。
- 基于标签匹配的知识蒸馏：降低计算成本

西安电子科技大学
XIDIAN UNIVERSITY

重新配置的注意力模块RAM

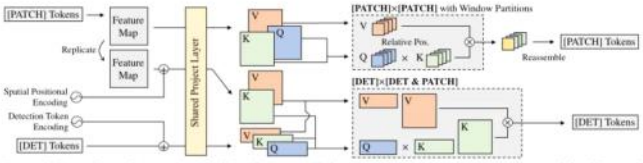
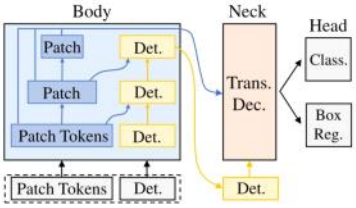


Figure 3. Reconfigured Attention Module (Q: query, K: key, V: value). The skip connection and feedforward networks following the attention operation is omitted just for ease of exposition.

- RAM将【Patch】和【DET】的全局注意力分解为三部分：
  - 【Patch】*【Patch】
    - 作用：聚合全局关键内容
    - 操作：
      - 窗口注意力：局部自注意力计算
      - 移动窗口注意力：提供窗口之间的交互，捕获全局信息
  - 【DET】*【DET】
    - 作用：每个【DET】通过计算全局注意力来捕获他们之间的关系，有助于定位不同的目标
    - 操作：自注意力（self-attention）
  - 【Patch】*【DET]
    - 作用：每个【DET】都表示不同的对象，因此需要为每一个【DET】生成不同的我们embedding，【Patch】中的关键内容被聚合到每个【DET】上
    - 操作：交叉注意力
  - 利用RAM替换Swin Transformer中的所有注意力模块
  - 位置编码：
    - 【Patch】：Swin Transformer的相对位置编码
    - 【DET】：可学习的位置编码（【DET】没有特定的顺序
  - 【Patch】*【DET】注意力只在Swin Transformer的最后一个Stage使用，避免增加额外的计算开销

Encoder-Free Neck



- Encoder-Free Neck组件
  - 无Encoder的原因：Swin Transformer已经提取了适用于目标检测的细粒度特征
  - Decoder：采用Deformable DETR的Decoder可以充分利用多尺度特征
    - 两个输入：每个Stage生成的【Patch】+最后一个Stage生成的【DET】
    - 使用多尺度形变注意力：用于生成新的【DET】，聚合从多尺度特征图中采样的一小部分关键内容
      - 【DET】*【DET】的自注意力
      - 【Patch】*【DET】的交叉注意力

$$\mathrm{MSDeformAttn}([\mathtt{DET}], \{\boldsymbol{x}^l\}_{l=1}^{L}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlk} \cdot \boldsymbol{W}'_m \boldsymbol{x}^l \big(\phi_l(\boldsymbol{p}) + \Delta \boldsymbol{p}_{mlk}\big) \Big], \quad (1)$$

      - where m indices the attention head and K is the total number of sampled keys for content aggregation. In addition, $\phi_l(\boldsymbol{p})$ is the reference point of the $[\mathtt{DET}]$ token re-scaled for the l-th level feature map, while $\Delta \boldsymbol{p}_{mlk}$ is the sampling offset for deformable attention; and A_{mlk} is the attention weights of the K sampled contents. $\boldsymbol{W}_m$ and $\boldsymbol{W}'_m$ are the projection matrices for multi-head attention.
  - 辅助技术
    - 辅助解码损失
    - 迭代的边界细化

基于标签匹配的知识蒸馏

- 目的：降低计算成本
- 操作：
  - 教师模型：预训练的大型ViDT
  - 学生模型：需要学习的小型ViDT
  - 将学生模型的【Patch】和【DET】与教师模型的【Patch】和【DET】匹配，将教师模型的知识转移到学生模型
  - $$\ell_{dis}(\mathcal{P}_s, \mathcal{D}_s, \mathcal{P}_t, \mathcal{D}_t) = \lambda_{dis}\Big(\frac{1}{|\mathcal{P}_s|}\sum_{i=1}^{|\mathcal{P}_s|}\Big\|\mathcal{P}_s[i] - \mathcal{P}_t[i]\Big\|_2 + \frac{1}{|\mathcal{D}_s|}\sum_{i=1}^{|\mathcal{D}_s|}\Big\|\mathcal{D}_s[i] - \mathcal{D}_t[i]\Big\|_2\Big)$$

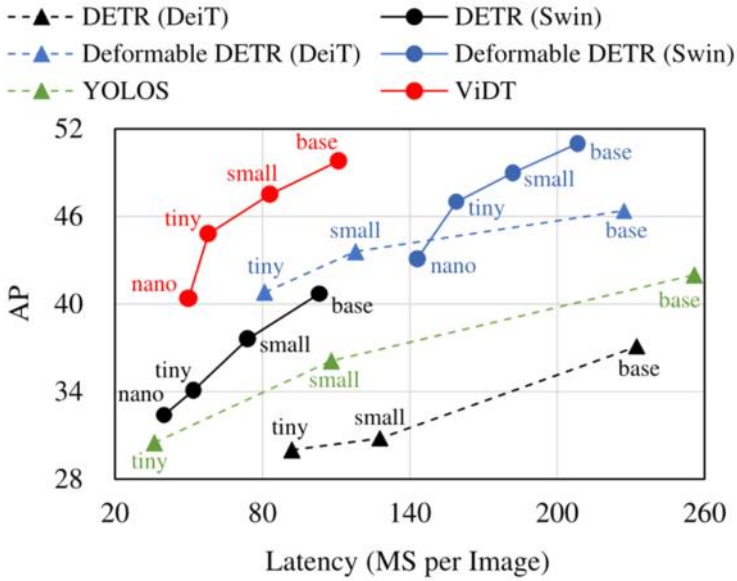


Figure 1. AP and latency (milliseconds) summarized in Table 2. The text in the plot indicates the backbone model size.

SSD及其变体

2022年5月13日 9:12

西安电子科技大学
XIDIAN UNIVERSITY

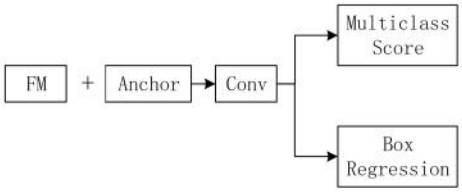
SSD算法原理

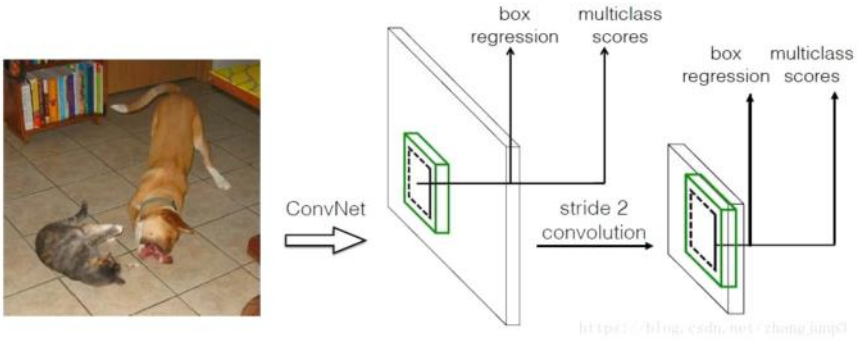
- DSSD
- DSOD
- FSSD
- RSSD

SSD算法基本流程

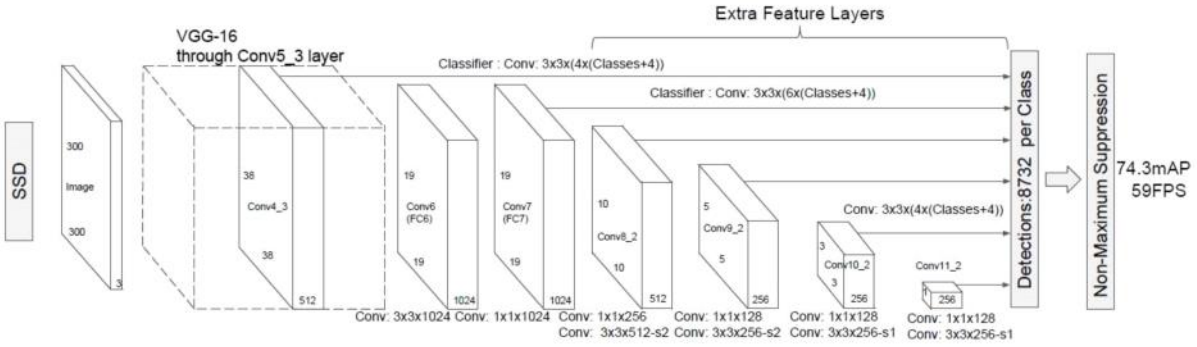
特征提取：

卷积检测：

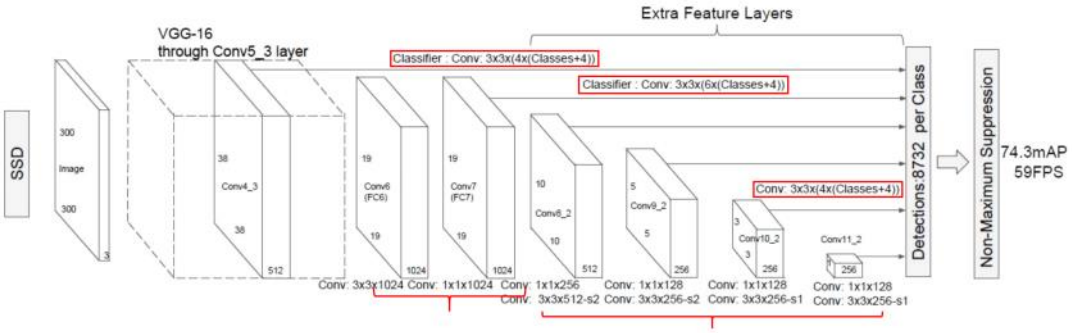




SSD网络结构


Extra Feature Layers
VGG-16
through Conv5_3 layer
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
SSD
300
Image
300
3
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1

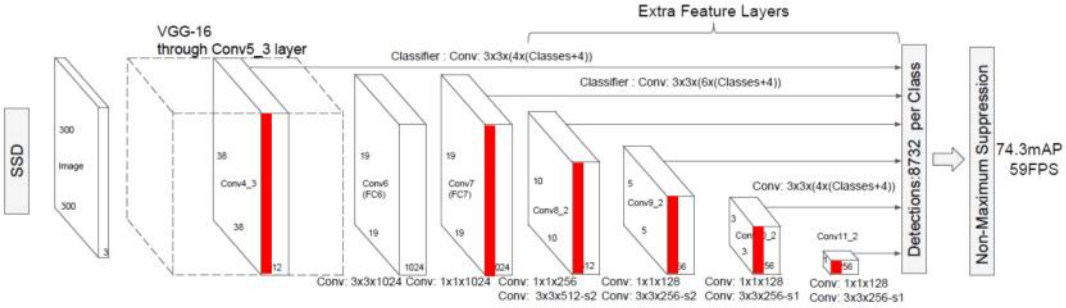
SSD网络结构详细说明



1.将VGG16的FC6+FC7 => Conv6:3*3 + Conv7:1*1

2.移除Dropout层和FC8，新增一系列卷积层.

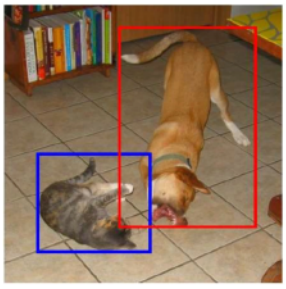
SSD网络结构详细说明



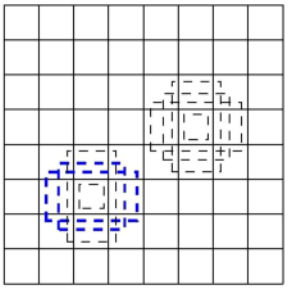
不同层提取特征图:

(38*38),(19*19),(10*10),(5*5),(3*3),(1*1)

先验框（anchor）设置



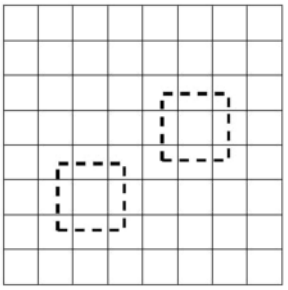
(a) Image with GT boxes



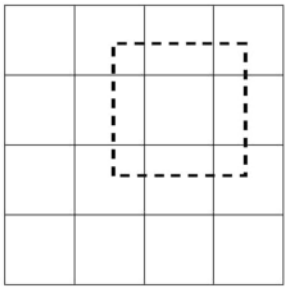
(b) 8×8 feature map

不同尺寸特征图设置的先验框数目是不同的

同一个特征图上每个单元设置的先验框是相同的



8×8 feature map



4×4 feature map

- 先验框(anchor)尺寸设置

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1, m]$$

Sk是先验框相对于原图的比例
Smin=0.2 Smax=0.9
m是待检测特征图的数量(m=5)

- 先验框的纵横比（aspect-ratio）

$$\{1, 2, 3, \frac{1}{2}, \frac{1}{3}, 1'\}$$

- anchor(Cx, Cy, w,h)

$$w_k^a = s_k\sqrt{a_r},\ h_k^a = s_k/\sqrt{a_r}$$

$$(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|}), i, j \in [0, |f_k|)$$

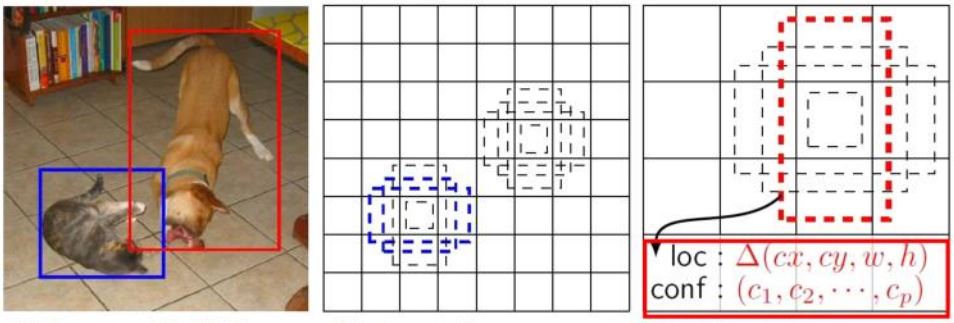
- 总先验框数量

38*38*4+19*19*6+10*10*6+5*5*6+3*3*4+1*1*4=8732 个先验框

卷积检测

- 对于m*n*p的特征图，用于检测的卷积核为3*3*p
- 对于一个m*n特征图 => m*n个单元
- 每个单元设置k个先验框 => (C+4)*k个预测值 (C=c+1)
- 采用(C+4)*k个卷积核检测

边界框检测值



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

边界框检测值包含两部分：

C个类别：预测(C+1)个置信度分数，置信度最高的类别即为边界框类别

Location:(Cx,Cy,w,h) (注意：预测值是边界框相对于先验框的offset)

西安电子科技大学
XIDIAN UNIVERSITY

边界框Location检测值

Encode(边界框预测值L是b相对于d的转换值)

$$l^{cx} = (b^{cx} - d^{cx})/d^{w},\ l^{cy} = (b^{cy} - d^{cy})/d^{h}$$

$$l^{w} = \log(b^{w}/d^{w}),\ l^{h} = \log(b^{h}/d^{h})$$

Anchor: $d = (d^{cx}, d^{cy}, d^{w}, d^{h})$

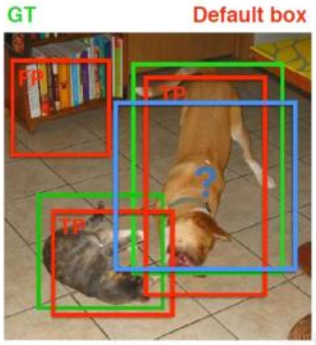
Bounding Box: $b = (b^{cx}, b^{cy}, b^{w}, b^{h})$

Decode(由预测值得到边界框真实位置)

$$b^{cx} = d^{w}l^{cx} + d^{cx},\ b^{cy} = d^{y}l^{cy} + d^{cy}$$

$$b^{w} = d^{w}\exp(l^{w}),\ b^{h} = d^{h}\exp(l^{h})$$

训练——先验框匹配



匹配原则：

1.对于图片中每个ground truth,找到与其IOU最大的先验框匹配，该先验框记为正样本

2.对于剩余的未匹配先验框,若与某个GT的 IOU大于指定阈值(0.5),那么该先验框也与这个ground truth进行匹配

训练——正负样本平衡

SSD采用了hard negative mining

1.对负样本按照置信度误差降序排列

2.选取误差较大的top-k作为训练的负样本

正负样本比例接近1:3

训练——损失函数

$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g))$$

N 是先验框的正样本数量。

$x_{ij}^{p}\in\{0,1\}$ 为一个指示参数

当其为1:表示第 i 个先验框与第 j 个ground truth匹配，类别为 p

c 为类别置信度预测值

l 为先验框的所对应边界框的位置预测值

g是ground truth的位置参数

训练——损失函数

Conf_Loss:

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

Loc_Loss:

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right)$$

位置误差只针对正样本计算

预测过程

- 对于每个预测框，根据类别置信度确定类别，过滤属于背景的边界框
- 根据置信度阈值过滤置信度值低的预测框
- 将保留的预测框解码，得到真实位置参数
- 根据置信度值（降序排列）保留top_k个预测框
- 利用NMS算法过滤掉重叠度较大的预测框，剩余预测框为检测结果

问题

- 重复检测

  不同层特征图分别送入检测网络进行检测 => 相同目标被多次检测

- 小目标检测效果差

  浅层特征图特征表征能力不够强

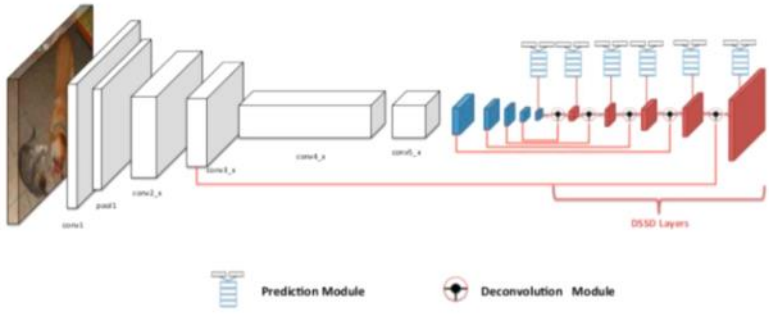
- 特征图之间缺少联系，不能充分将全局和局部特征结合

DSSD(Devolutional Single Shot Detector)

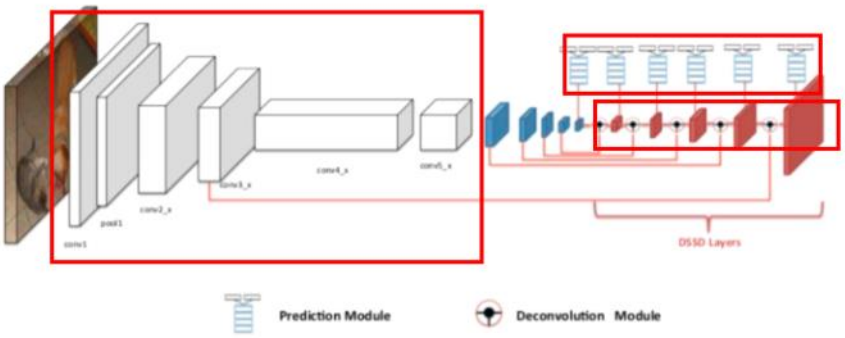
- SSD小目标检测效果差:

  特征图分别独立检测缺少上下文信息的融合

  负责检测小目标的浅层特征图表征能力弱

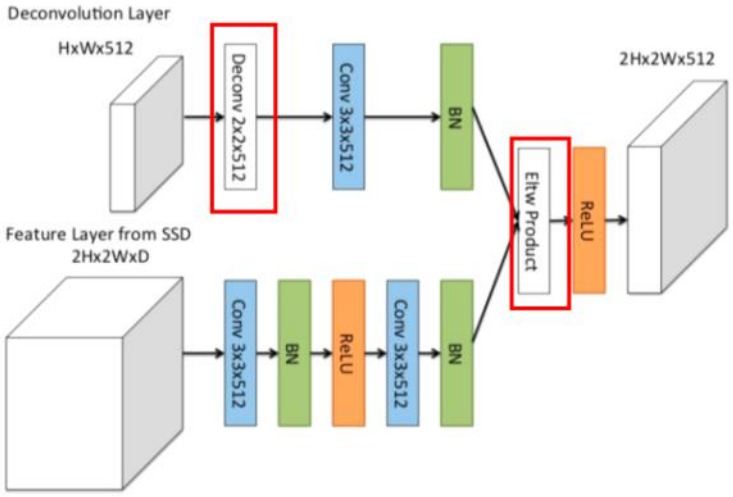


DSSD

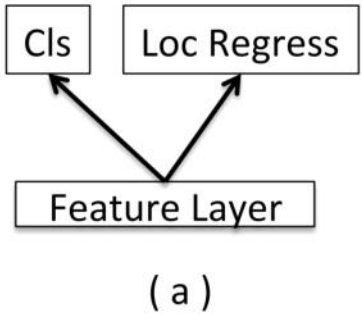


- ResNet101作为特征提取网络
- 反卷积特征融合模块
- 预测阶段引入残差单元

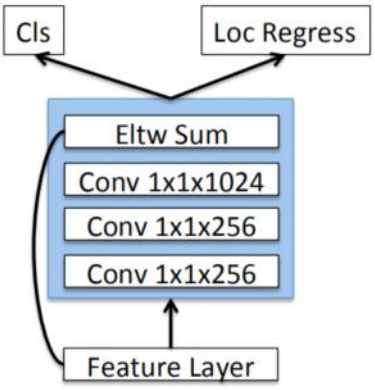
DSSD——Deconvolution Fusion Module



DSSD——Residual Prediction Module



(a)



(b)

DSOD(Learning Deeply Supervised Object Detectors from Scratch)

- Limited Structure Design Space:模型结构灵活性差难修改
- Learning Bias:分类网络与检测网络优化空间存在差异
- Domain Dismatch:目标域差异大时，微调结果不理想

DSOD结构

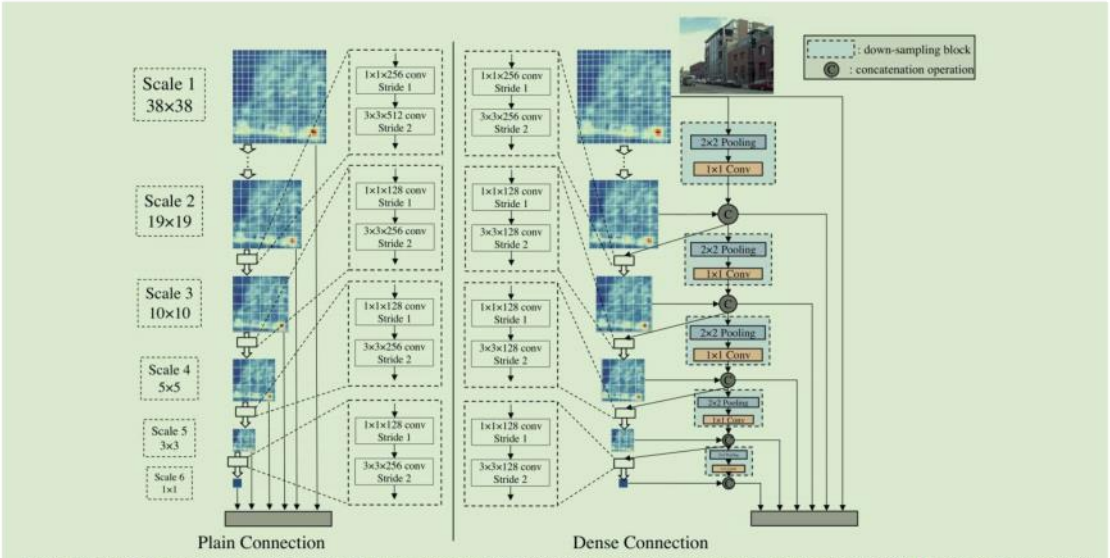
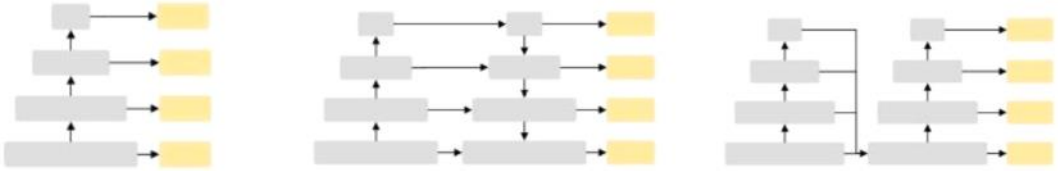


Figure 1: DSOD prediction layers with plain and dense structures (for 300×300 input). Plain structure is introduced by SSD [21] and dense structure is ours. See Section 3 for more details.

西安电子科技大学
XIDIAN UNIVERSITY

FSSD(Feature Fusion Single Shot MultiBox Detector



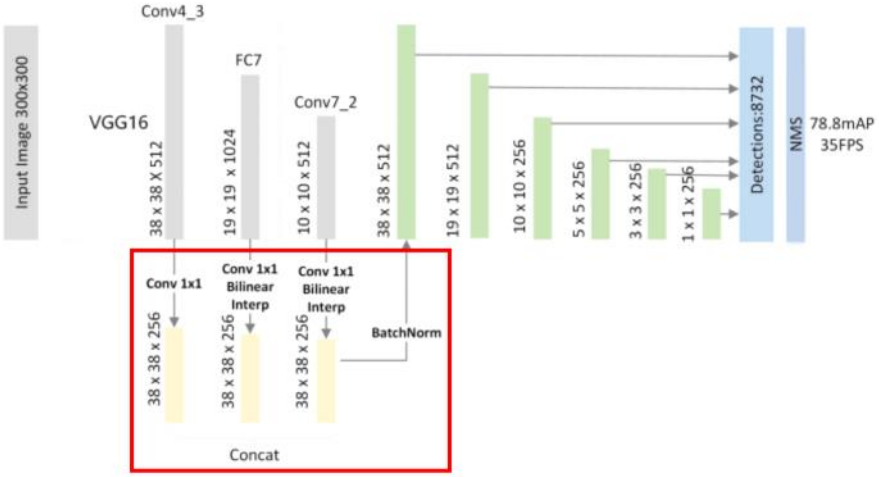
a　　b　　c

a.SSD独立地在feature map分支上做检测

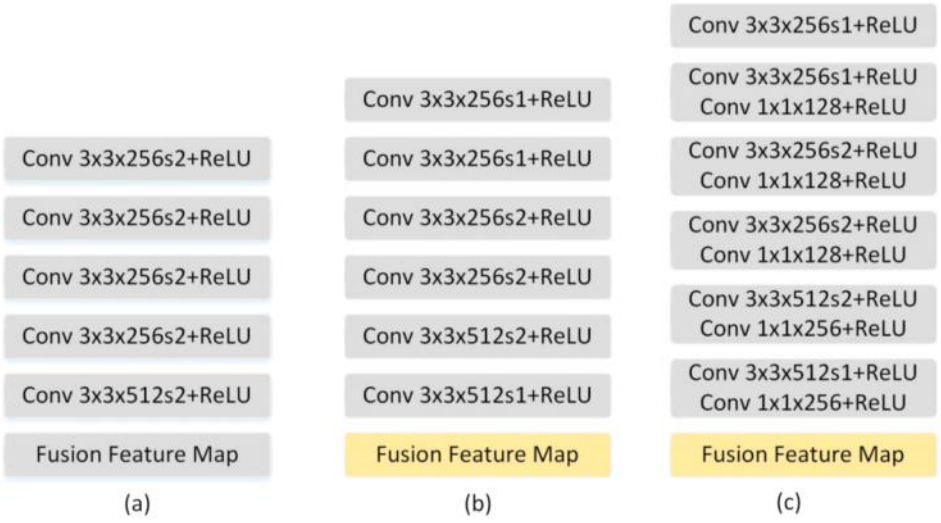
b.FPN通过top-down连接生成特征金字塔

c.FSSD特征融合

FSSD网络结构


Input Image 300x300
VGG16
Conv4_3
38 x 38 X 512
FC7
19 x 19 x 1024
Conv7_2
10 x 10 x 512
38 x 38 x 512
19 x 19 x 512
10 x 10 x 256
5 x 5 x 256
3 x 3 x 256
1 x 1 x 256
Detections:8732
NMS
78.8mAP
35FPS
Conv 1x1
Conv 1x1
Bilinear
Interp
Conv 1x1
Bilinear
Interp
BatchNorm
38 x 38 x 256
38 x 38 x 256
38 x 38 x 256
Concat

FSSD——fusion feature => feature pyramid



RSSD(Enhancement of SSD by concatenating feature maps for object detection)

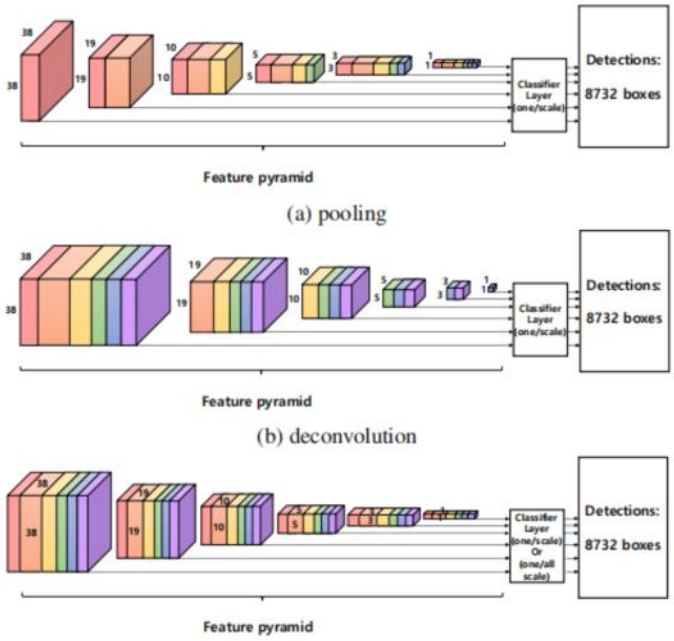
问题:

- 重复检测
- 对小尺寸物体的检测效果比较差

改进:

- 增加特征金字塔中feature map的个数
- 共享检测卷积层

RSSD


38
38
19
19
10
10
5
5
3
3
1
1
Classifier Layer (one/scale)
Detections:
8732 boxes
Feature pyramid
(a) pooling
Feature pyramid
(b) deconvolution
Classifier Layer (one/scale) Or (one/all scale)
Feature pyramid

思考

Scale_aware ?　　Scale_confused!

False Negative Problem + False Positive Problem

如何做到真正的Scale_aware,让浅层特征只检测小目标，深层特征只检测大目标?

上采样方法总结

2022年5月13日 9:14

基于线性插值的上采样

- 最近邻算法
- 双线性插值算法
- 双三次插值算法

基于深度学习的上采样

- 转置卷积
- PixelShuffle
- DUpsampling
- Meta-Upscale

最近邻插值算法

- 当图片放大时，缺少的像素值直接使用与之距离最近的原有颜色生成

34	56	18
21	34	56
90	59	31

⇨

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

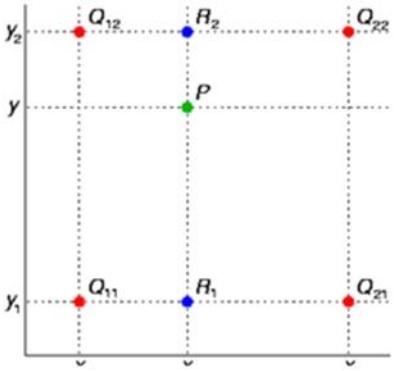
$$Src_y = Tar_y \times (Src_h / Tar_h)$$

$$Src_x = Tar_x \times (Src_w / Tar_w)$$

$$(Src_x, Src_y) = (0 \times (3/4), 1 \times (3/4)) = (0, 0.75) \Rightarrow (0, 1)$$

双线性插值算法

- 利用原图像中目标点四周的四个真实存在的像素值来共同决定目标图中的一个像素值



x方向上：

$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \text{ Where } R_1 = (x, y_1)$$

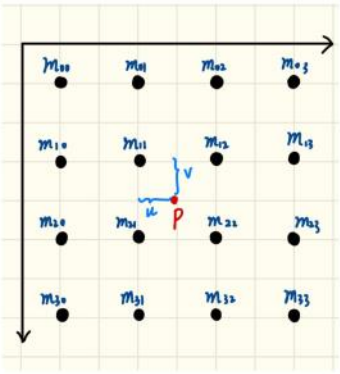
$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \text{ Where } R_2 = (x, y_2)$$

y方向上：

$$f(P) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

双三次插值

- 利用P点周围的16个点的像素贡献值的加权和作为P点处的像素值



$$W(x)=\begin{cases}(a+2)|x|^3-(a+3)|x|^2+1 & \text{for } |x|\le 1\\ a|x|^3-5a|x|^2+8a|x|-4a & \text{for } 1<|x|<2\\ 0 & \text{otherwise}\end{cases}$$

其中，a取-0.5.

m_{00} 与P点距离为(1+u，1+v)

m_{00} X方向权重: $W(1+u)$

m_{00} Y方向权重: $W(1+v)$

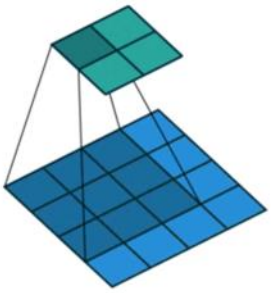
m_{00} 对P贡献: $m_{00}\times W(1+u)\times W(1+v)$

$$f(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}f(x_i,y_j)W(x-x_i)W(y-y_j)$$

西安电子科技大学
XIDIAN UNIVERSITY

转置卷积

卷积运算



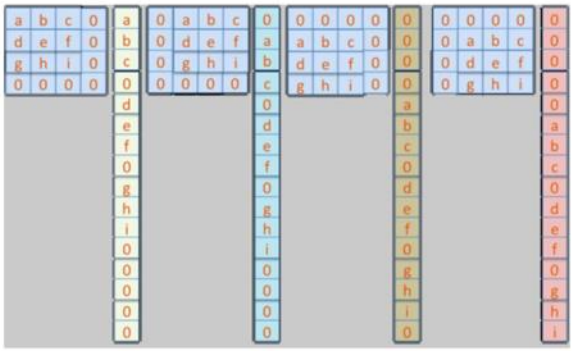
$$Y = CX$$

3*3卷积核表示如下：

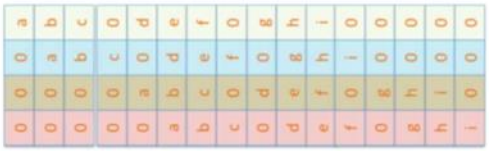
a	b	c
d	e	f
g	h	i

输入矩阵X: $[4,4] \Rightarrow [16,1]$

卷积核的4*4向量表示：



[4,16]

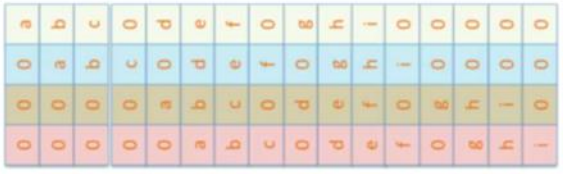


$$Y = CX \qquad [4,16]\times[16,1] \Rightarrow [4,1] \Rightarrow [2,2]$$

$$[2,2] \Rightarrow [4,4]$$

输入矩阵： $[2,2] \Rightarrow [4,1]$

[4,16]

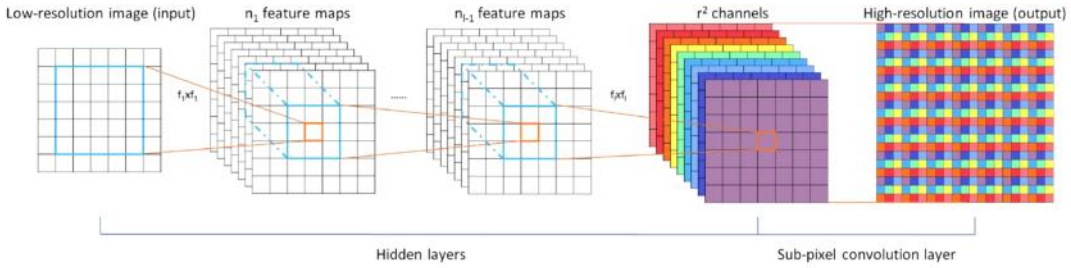


$[16,4]$

$$Y = CX \qquad [16,4] \times [4,1] \Rightarrow [16,1] \Rightarrow [4,4]$$

PixelShuffle

- 对于低分辨的特征图，通过卷积和多通道间的重组得到高分辨率的特征图



$$[N, Cr^2, H, W] \Rightarrow [N, C, rH, rW]$$

$$[N,64,64,64] \Rightarrow [N,256,64,64]$$

$$[N,4,64,64] \Rightarrow [N,64,2,64,2] \Rightarrow [N,128,128,1] \Rightarrow [N,64,128,128]$$

DUpsampling

- PixelShuffle类似，通过卷积学习亚像素，并最后重组来获得更大的图像。

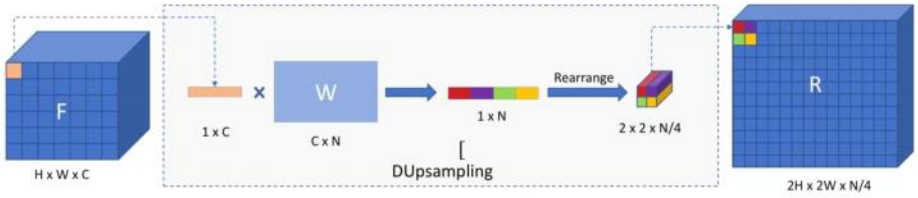


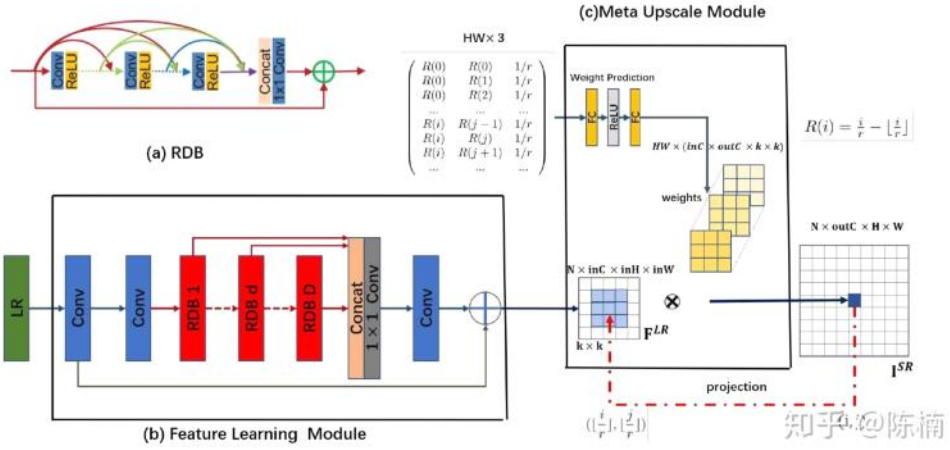
Figure 3: The proposed DUpsampling. In the figure, DUpsampling is used to upsample the CNNs outputs $\mathbf{F}$ by twice. $\mathbf{R}$ denotes the resulting maps. $\mathbf{W}$, computed with the method described in Sec. 3.1, is the inverse projection matrix of DUpsampling. In practice, the upsampling ratio is typically 16 or 32.

上采样过程可以看成是将特征分辨率扩大，
例如将H*W*C的特征表示上采样到2H*2W*C的表示。
也可以看成是针对特征图中的每一个像素，扩大成四个像素表示

$$[1, C]\times[C, N]\Rightarrow[1, N]\Rightarrow[2,2, N/4]$$

Meta-Upscale

- 动态预测上采样卷积所需的权重，通过输入尺度因子和权重能够对图像进行任意比例的放大



Meta-Upscale

- 目标图像是由原图像图和相关上采样卷积的权重决定的，上采样模块可以看作是原图像和目标图像之间的映射函数

$$I^{SR}(i,j) = \Phi(F^{LR}(i^{'},j^{'}),W(i,j))$$

Meta-Upscale

Location Projection：把目标图像坐标投射到 原图像上，找到像素之间的对应关系

Weight Prediction： 为 目标图像上每个像素预测对应滤波器的权重

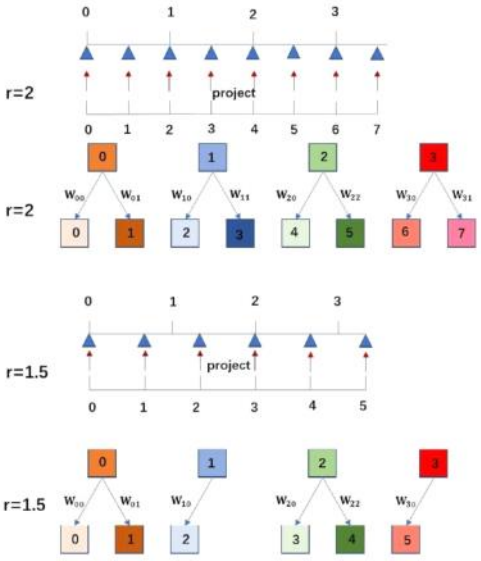
Feature Mapping： 利用预测得到的权重将原图像的特征映射到 目标图像空间以计算其像素值。

Meta-Upscale

Location Projection：把目标图像坐标投射到 原图像上，找到像素之间的对应关系

对于目标图像中的每个像素点，均需找到原图像中的一个像素点与之相对应

$$(i', j') = T(i, j) = (\left\lfloor \frac{i}{r} \right\rfloor, \left\lfloor \frac{j}{r} \right\rfloor)$$



Meta-Upscale

Weight Prediction：为目标图像上每个像素预测对应滤波器的权重

$$W(i,j)=\varphi(v_{ij};\theta)$$

$$v_{ij}=(\frac{i}{r}-\left\lfloor\frac{i}{r}\right\rfloor,\frac{j}{r}-\left\lfloor\frac{j}{r}\right\rfloor,\frac{1}{r})$$

$$v_{ij}=(\frac{i}{2}-\lfloor\frac{i}{2}\rfloor,\frac{j}{2}-\lfloor\frac{j}{2}\rfloor)=(\frac{2i}{4}-\lfloor\frac{2i}{4}\rfloor,\frac{2j}{4}-\lfloor\frac{2j}{4}\rfloor)=v_{2i2j}$$

Meta-Upscale

Feature Mapping： 利用预测得到的权重将原图像的特征映射到 目标图像空间以计算其像素值。

$$\Phi(F^{LR}(i^{'},j^{'}),W(i,j))=F^{LR}(i^{'},j^{'})\cdot W(i,j)$$