CCF BDCI CCF BIG DATA & COMPUTING INTELLIGENCE CONTEST

2023 CCF大数据与计算智能大赛 11th

基于TPU平台实现超分辨率重建模型部署

作品名：基于预训练ESPCN的轻量化图像超分辨率模型TPU部署方案

队伍名：Absofastlutely

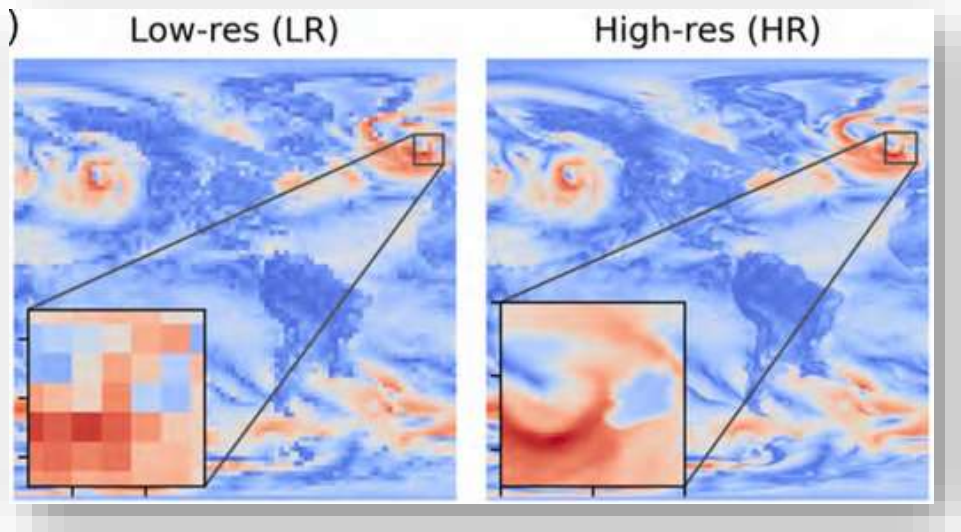
目录

CONTENT

问题描述

- 图像超分辨率
  - 尺寸放大：x2, x3, x4…
  - 细节增加：deblur, denoise, anti-alias…
- TPU硬件开发
  - sophon BM1684X
    - CPU: 8 x ARM A53 2.3GHz
    - Mem: 16G 4266Mbps

dtype	TFLOPS
FP32	2
FP16/BF16	16
INT8	32



模型选择

- 基准模型
  - Bilinear / Lanczos
  - Real-ESRGAN
- 轻量化模型
  - NinaSR
  - Carn / Carn-m
  - FSRCNN
  - ESPCN

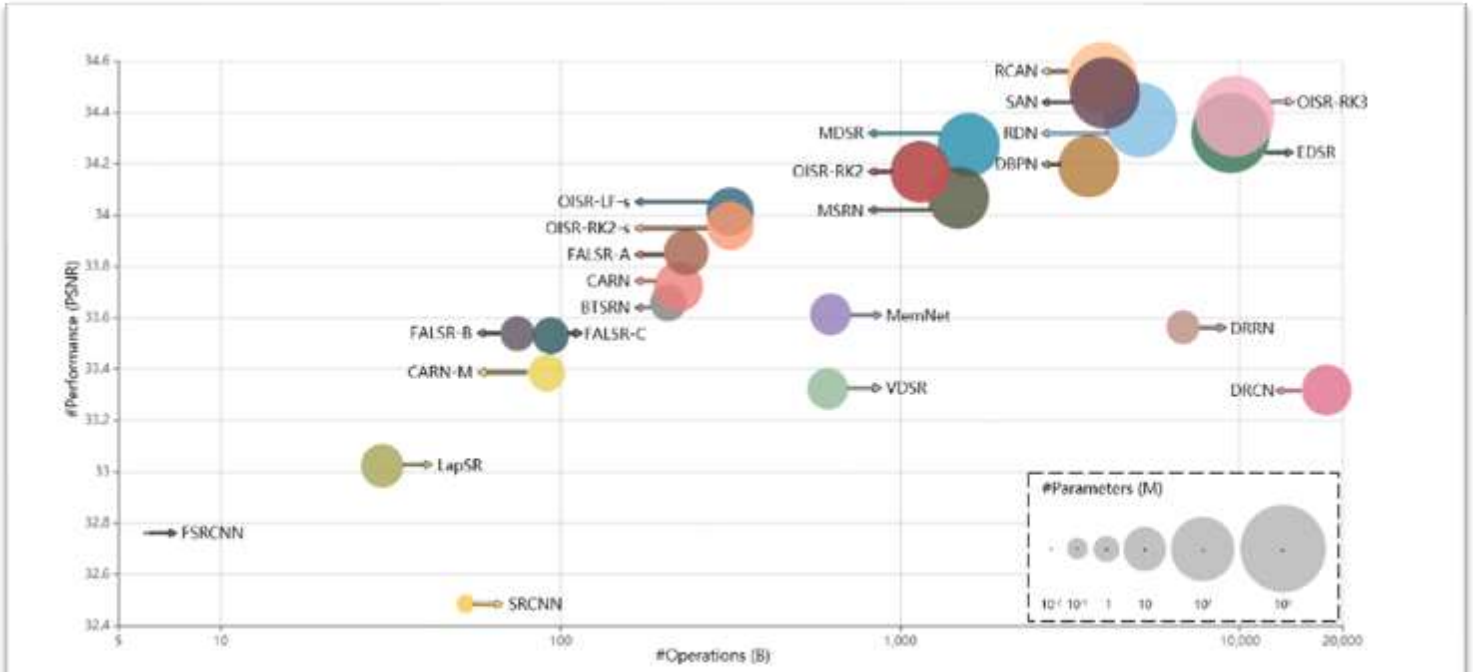
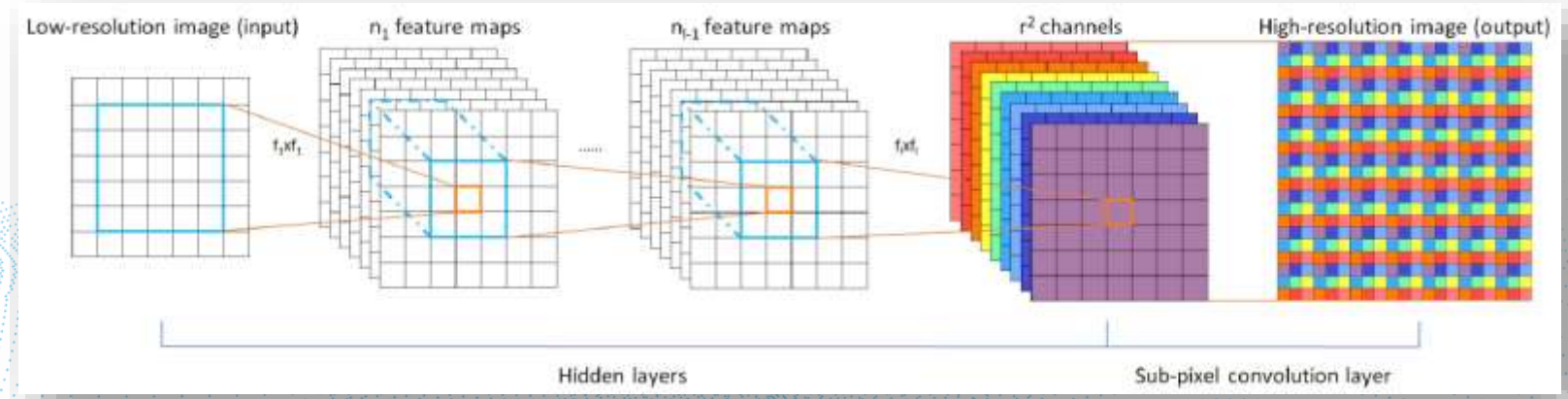


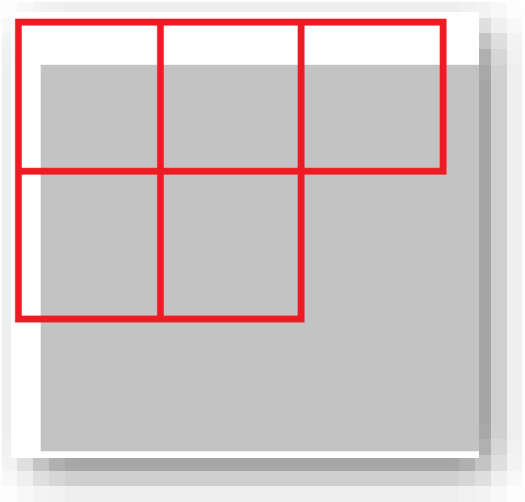
Fig. 8. Super-resolution benchmarking. The x-axis and the y-axis denote the Multi-Adds and PSNR, respectively, and the circle size represents the number of parameters.



分块策略

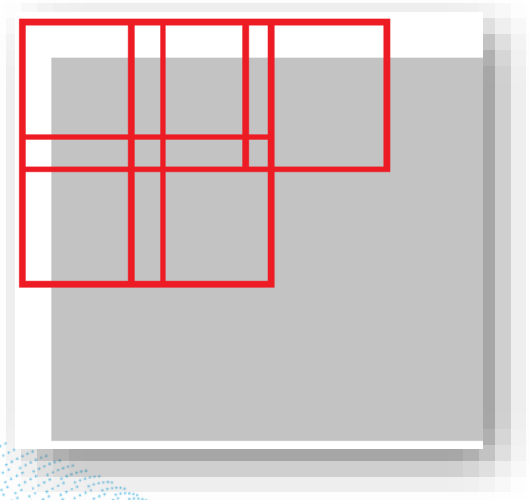
- 朴素分块Tile
- 重叠缝合Overlap
  - 朴素平均、高斯羽化
- 边缘裁剪Crop

- Conv2d设置pad="replicate"

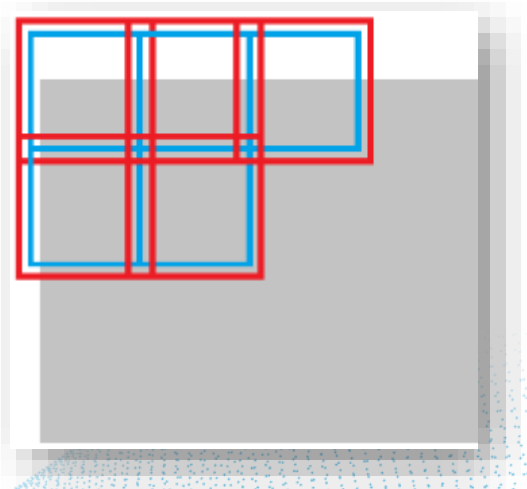


Tile

⇒ 有接缝！



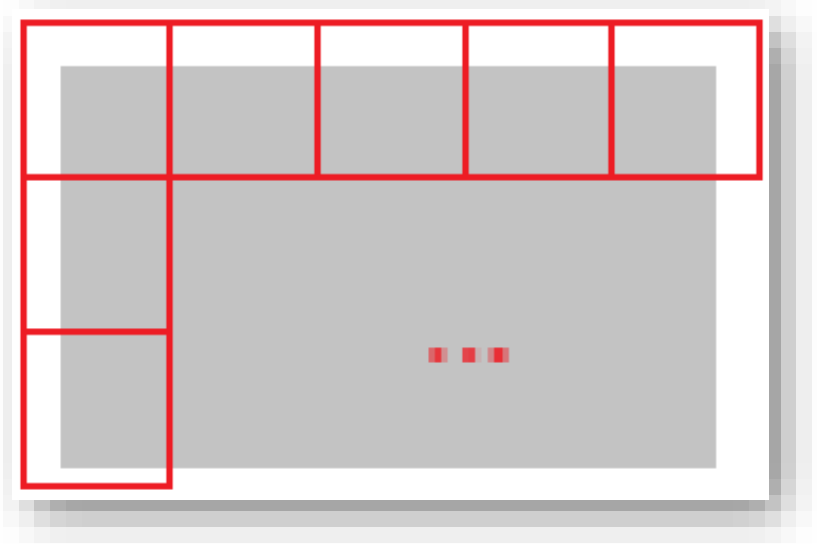
Overlap



Crop

分块最优大小

- 最小化冗余计算量
  - $cost = \min_{tile\_size} \left(1 - \frac{raw\_area}{tiled\_area}\right) * (tile\_count + \mathrm{C})$
  - tile_size = (192, 256)



- 最小化单位计算费率
  - tile_size = (128, 128)

- TPU不能并行化batch

```
# ts_avg

(0.00386, 1x3x128x128)
(0.00768, 2x3x128x128)
(0.01180, 1x3x192x256)
(0.01543, 4x3x128x128)
(0.01616, 1x3x256x256)
(0.02342, 2x3x192x256)
(0.02457, 1x3x256x384)
(0.05305, 4x3x192x256)
(0.06987, 1x3x512x512)
(0.13496, 2x3x512x512)
(0.26931, 4x3x512x512)
```

```
# price

(7.81733-08, 2x3x128x128)
(7.85208-08, 4x3x128x128)
(7.86957-08, 1x3x128x128)
(7.94329-08, 2x3x192x256)
(8.00720-08, 1x3x192x256)
(8.22264-08, 1x3x256x256)
(8.33321-08, 1x3x256x384)
(8.56115-08, 4x3x512x512)
(8.58066-08, 2x3x512x512)
(8.88498-08, 1x3x512x512)
(8.99449-08, 4x3x192x256)
```

其他trick

- 多线程加速
- 后处理滤波
  - EDGE_ENHANCE
  - UnsharpMask (SHARPEN)

```
-1, -1, -1,
-1, 10, -1,
-1, -1, -1,
```

EDGE_ENHANCE

```
 0, -1,  0
-1,  5, -1
 0, -1,  0
```

UnsharpMask

filter	multi-thread	tile size	runtime avg.	niqe avg.	score
-	-	(192,256)	0.1852	5.5697	1291.6236
EDGE	-	(192,256)	0.1615	4.5679	1930.9247
EDGE	4	(192,256)	0.1140	4.5242	2761.0642
EDGE	4	(128,128)	0.0983	4.3761	3296.2499
EDGE	4	(128,128)	0.1195	4.4613	2666.0168
SHARPEN	4	(128,128)	0.1164	4.2143	2867.1234

模型评估

$$score = \frac{\sqrt{7 - niqe}}{runtime} * 200$$

	runtime avg.	niqe avg.	score	rank
A榜	0.1924	4.4465	1661.0888	9
B榜	0.0983	4.3761	3295.7186	4

总结 & 反思

- 贡献
  - 在TPU平台上部署轻量化超分模型的方法
  - 三种分块策略、两种最优分块大小定义
  - 多线程加速和后处理滤波trick

- 提升空间
  - 预训练的ESPCN权重为Y通道分布，微调后artifacts会更少
  - 区分Anime/Photo/Painting等不同风格分布的图像，集成多个模型
  - 在TPU上实现分块-合并过程，减少数据传输

参考引用

- 论文/文档
  - https://arxiv.org/abs/2107.10833
  - https://arxiv.org/abs/1609.05158v2
  - https://github.com/Coloquinte/torchSR/blob/main/doc/NinaSR.md
  - https://sophon-file.sophon.cn/sophon-prod-s3/drive/23/03/02/20/BM1684X%20Introduction%20V1.7.pdf
- 代码仓库
  - https://github.com/sophgo/TPU-Coder-Cup/tree/main/CCF2023
  - https://github.com/Lornatang/ESPCN-PyTorch
  - https://github.com/Coloquinte/torchSR

谢谢观看