

# ACCV2022 | 大规模细粒度图像分类冠军方案

极市平台 2022-12-13 22:01:00 发表于广东 手机阅读 罍

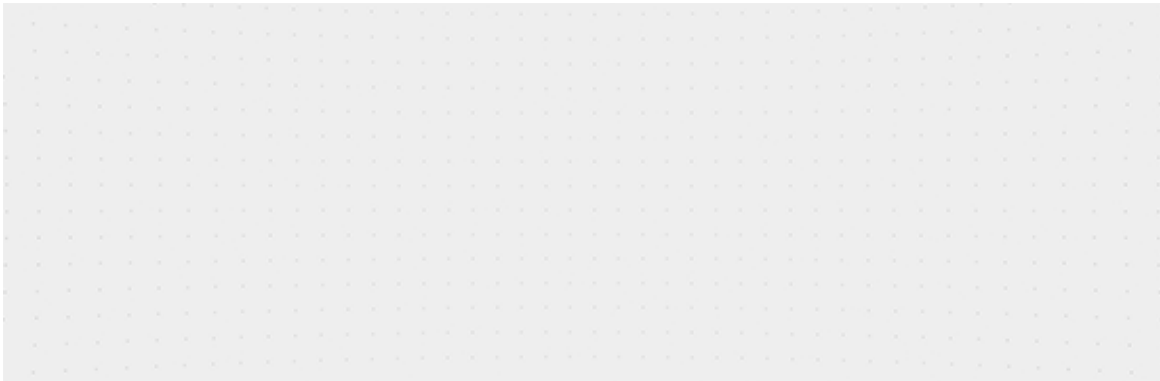
以下文章来源于OpenMMLab，作者带来新知识的



OpenMMLab

构建国际领先的人工智能开源算法平台

↑ 点击蓝字 关注极市平台



作者 | ABCDEFG

编辑 | 极市平台

## 极市导读

由南京理工大学和澳大利亚 University of Wollongong 主办、极市平台提供技术支持的 ACCV2022国际细粒度网络图像识别大赛已经圆满结束。本文是本次大赛大规模细粒度图像分类赛套的冠军团队的解决方案解读，包括了数据处理、模型选择等并开源了所有预训练，微调以及推理的代码和文件。 >>加入极市CV技术交流群，走在计算机视觉的最前沿

在刚刚结束的 ACCV 2022 国际细粒度图像分析挑战赛中，我们团队在 **133** 支参赛队伍中脱颖而出，在 **Leadboard-B** 上以 2.5 的绝对优势取得冠军。

排名	团队名	score	排名	团队名	score
1	ABCDEFG	0.7732	1	ABCDEFG	0.7894
2	...	0.7609	2	...	0.764
3	...	0.7555	3	...	0.7611
4	...	0.7534	4	...	0.7607
5	...	0.7529	5	...	0.76
6	...	0.7492	6	...	0.7594

A 榜

B 榜

比赛成绩截图

在比赛过程中，我们使用了一些对细粒度分类十分有效的解决方案。

- 例如，在数据清洗方面我们去掉二义性的图片；
- 模型选择和方法方面我们使用了 ViT-L和 Swin-v2；
- 为了提高实验效率，方便整个实验过程的管理，我们基于 **MMSelfSup** 和 **MMClassification** 两个开源算法框架开展比赛。

所有的模型权重和全部代码都已经开源，欢迎大家使用。

本次赛事相关的所有预训练，微调以及推理的代码和配置文件分别放在 MMSelfSup 以及 MMClassification 的 projects 中, 欢迎大家 star 和使用。

自监督预训练部分: [https://github.com/open-mmlab/mmselfsup/tree/dev-1.x/projects/fgia\\_accv2022\\_1st](https://github.com/open-mmlab/mmselfsup/tree/dev-1.x/projects/fgia_accv2022_1st)

微调与模型推理: [https://github.com/open-mmlab/miclassification/tree/dev-1.x/projects/fgia\\_accv2022\\_1st](https://github.com/open-mmlab/miclassification/tree/dev-1.x/projects/fgia_accv2022_1st)

今天这篇文章我们将整体介绍此次夺冠的方案，希望为大家提供一些经验以供参考。

## 比赛介绍

ACCV 2022 细粒度图像分析挑战赛是由南京理工大学和澳大利亚 University of Wollongong 主办、极市平台提供技术支持的国际性赛事。

本赛事涉及的细粒度图像分析是计算机视觉和模式识别中的热门研究课题，其目标是对细粒度级别图像中的物体子类（如不同种类的“狗”：“哈士奇”、“阿拉斯加”、“萨摩耶”）进行定位、识

别及检索等视觉分析任务的研究，具有真实场景下广泛的应用价值。然而因细粒度级别子类别间较小的类间差异和较大的类内差异，使其区别于传统图像分析问题成为更具挑战的任务。

此次我们参加的赛题是：**网络监督的细粒度识别**，接下来我们将从**数据清洗**、**模型选择**、**训练技巧**（Bag-of-tricks）、**模型集成和后处理**等角度介绍我们的方案。

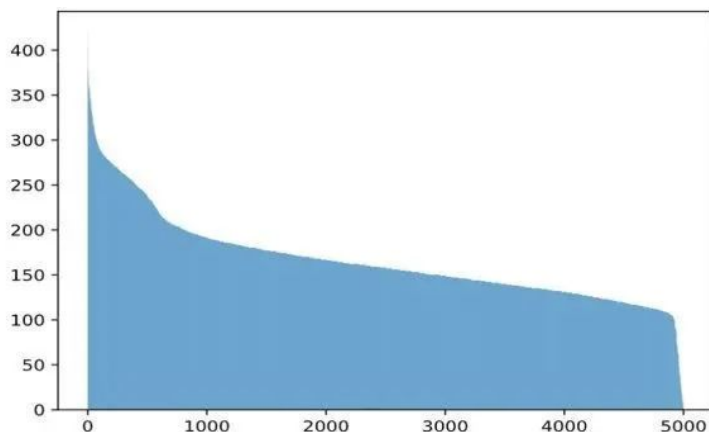
## 数据清洗

官方提供的数据包含：Train、Test-A 和 Test-B。Train 是用于训练的数据，这部分数据官方提供了 label。同时因为这项比赛分为 A/B 榜，所以 Test A 和 Test B 分别是用于评测的数据，这两部分数据是可以获得的，但是他们的 label 是不提供的。

这几部分的数据的详细信息如下：

- Train: 835K
- Test-A: 60K
- Test-B: 90K

对于用于训练的数据 Train, 其分布如下：



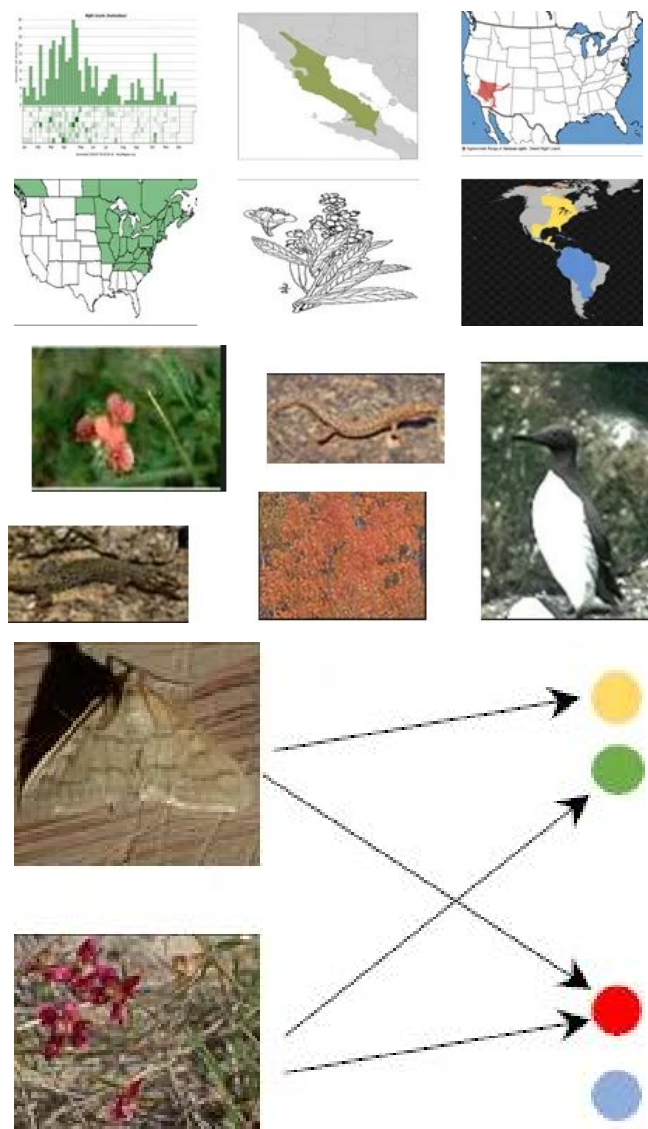
可以发现该数据为长尾分布，其中最多类的数据有 422 张图，最少类数据却只有 3 张图，均值为 166 张。

除了以上的分布问题，训练数据还存在以下几个问题：

1. 有噪声的图片，数据集中存在大量的图表图片（如下面这些图片所示）
2. 有些图片的分辨率非常小

3. 有一部分图片有多个 label

4. 有些数据不可读取



针对以上问题，我们尝试了以下几种方案：

成功的方案：

- 1. 我们过滤掉这些不可读取的图片，一共 4687 张
- 2. 根据哈希文件名，过滤掉这些具有多个 label 的图片
- 3. 我们不创建额外的 validation set，所有数据用于训练

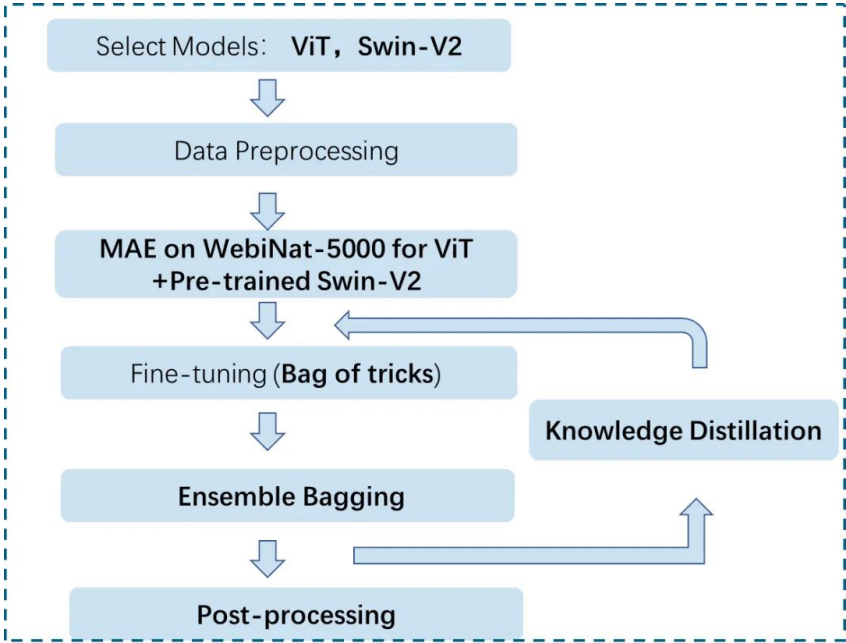
无明显效果的方案：

- 1. 由于数据集是一个长尾分布，我们尝试了 class-balanced sampler

2. 采用聚类方法，将这些离群的异常值去除掉

## 模型选择和方法

考虑到模型性能和速度，我们所有实验均采用 ViT-L 和 Swin-V2。方法总览如下图所示：



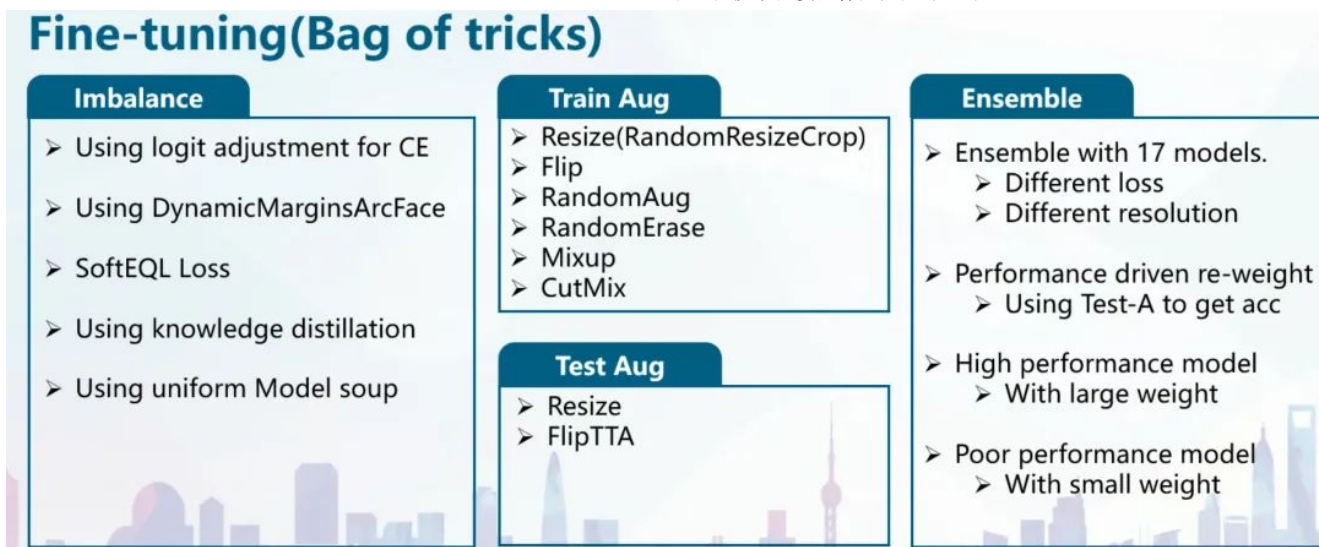
总结起来就以下几步：

1. 对 ViT-L [1] 和 Swin-V2 [2] 进行自监督预训练
2. 使用预训练的权重进行微调（fine-tunine）
3. 然后对各个模型进行集成
4. 推理时候进行数据的后处理

其中第 2 步和第 3 步通过自蒸馏的形式进行迭代，自蒸馏是指本次训练得到的集成模型充当 teacher 对下一轮模型的训练进行指导。

### 训练技巧

在分类微调（fine-tuning）的过程中我们使用了以下几种策略：

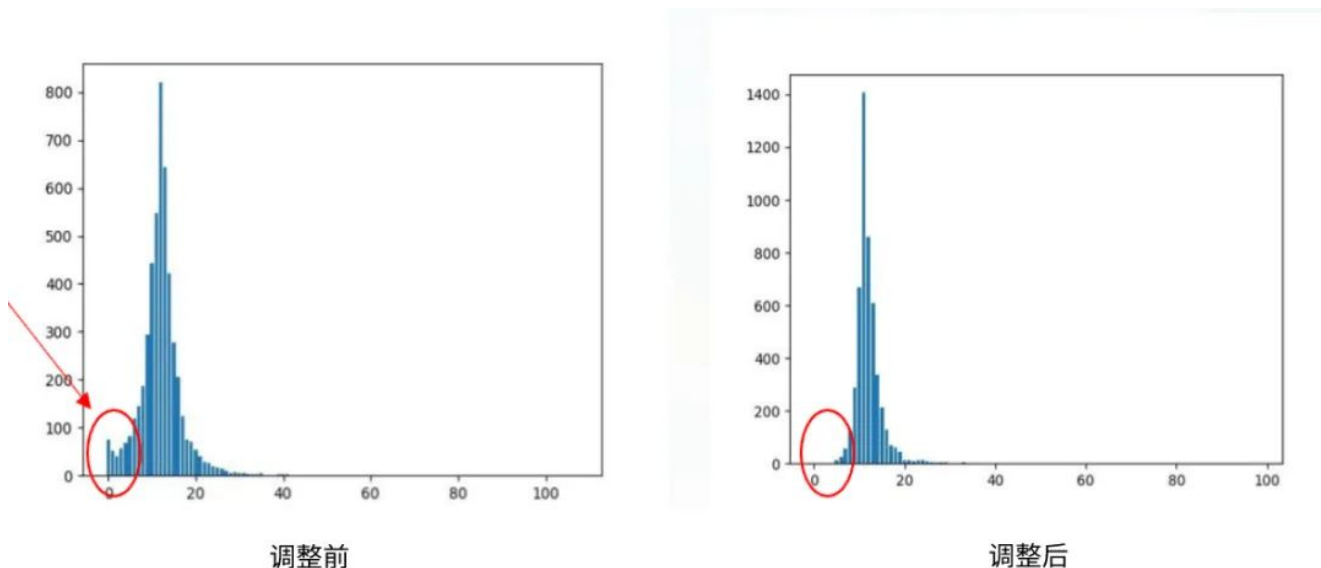


## 模型集成

我们最终使用了 17 个模型进行集成，其中包含 10 个 ViT-L 模型和 7 个 Swin-v2 模型，集成的权重根据模型在 Test A 上的精度，按比例得到。总体来说就是，高精度模型权重更大，低精度模型权重更小。

## 后处理

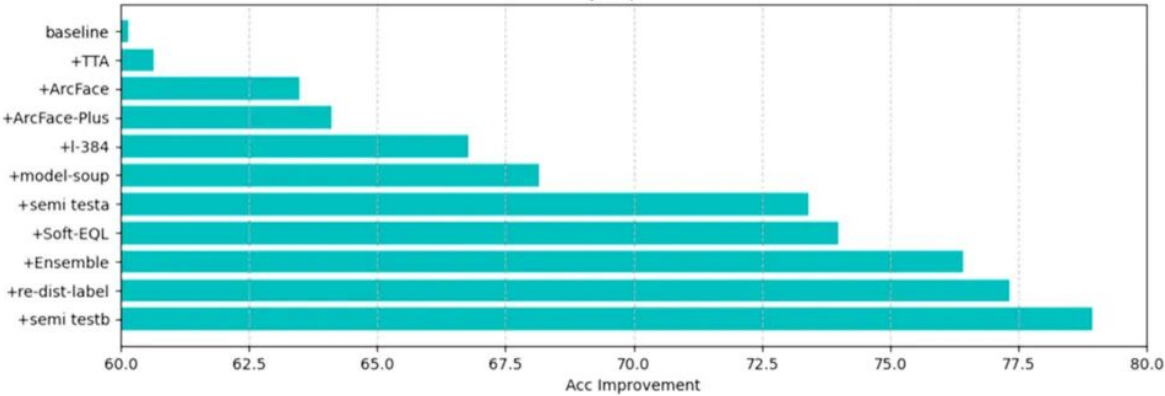
因为比赛中有说明，Train 集中的类别不平衡，但 Test 集中类别是按均匀分布的，所以对最后的预测结果，我们微调了标签的分布，调整前后的标签分布如下图所示：



## 总结

最终，我们模型的消融效果如下图所示：





## 代码仓库

考虑到效率和实验管理，我们选择了 **MMSelfSup** 和 **MMClassification**，这两个库均来自于 OpenMMLab 开源框架。

此次赛事相关的所有预训练，微调以及推理的代码和配置文件分别放在 **MMSelfSup** 和 **MMClassification** 的 projects 中, 欢迎大家 star 和使用。

自监督预训练部分: [https://github.com/open-mmlab/mmselfsup/tree/dev-1.x/projects/fgia\\_accv2022\\_1st](https://github.com/open-mmlab/mmselfsup/tree/dev-1.x/projects/fgia_accv2022_1st)

微调与模型推理: [https://github.com/open-mmlab/miclassification/tree/dev-1.x/projects/fgia\\_accv2022\\_1st](https://github.com/open-mmlab/miclassification/tree/dev-1.x/projects/fgia_accv2022_1st)

```
(base) ➜ final git:(master) x ls
l-384-roundb1      efficientv2-b-ce-384  l-384-arc-round2      swin-b-arc-subencter      swin-l-384-arc-round3-eql-7400
efficientv2-b-arc-384  efficientv2-l-arc-448  l-384-arc-roundb2-eql  swin-b-arc-subencter-advmargin  swin-l-384-arc-roundb1-eql
(base) ➜ final git:(master) x ls swin-l-384-arc-round3-eql-7400
20221103_220226  20221103_220445  20221103_100202  epoch_20.pth  l-384-arc_round3_eql.py  last_checkpoint  modules_statistic_results.json
(base) ➜ final git:(master) x tail swin-l-384-arc-round3-eql-7400/20221103_100202/20221103_100202.log
2022/11/03 23:21:11 - mmengine - INFO - Epoch(train) [20][2700/3216]  lr: 1.0684e-05  eta: 0:06:10  time: 0.7191  data_time: 0.0019  memory: 58940  loss: 0.8060
2022/11/03 23:22:23 - mmengine - INFO - Epoch(train) [20][2800/3216]  lr: 1.0684e-05  eta: 0:04:58  time: 0.7196  data_time: 0.0019  memory: 58940  loss: 0.6707
2022/11/03 23:23:32 - mmengine - INFO - Exp name: l-384-arc_round3_eql_20221103_100202
2022/11/03 23:23:35 - mmengine - INFO - Epoch(train) [20][2900/3216]  lr: 1.0684e-05  eta: 0:03:46  time: 0.7206  data_time: 0.0019  memory: 58940  loss: 0.9351
2022/11/03 23:24:47 - mmengine - INFO - Epoch(train) [20][3000/3216]  lr: 1.0684e-05  eta: 0:02:35  time: 0.7184  data_time: 0.0019  memory: 58940  loss: 0.8458
2022/11/03 23:25:58 - mmengine - INFO - Epoch(train) [20][3100/3216]  lr: 1.0684e-05  eta: 0:01:23  time: 0.7185  data_time: 0.0020  memory: 58940  loss: 0.9283
2022/11/03 23:27:10 - mmengine - INFO - Epoch(train) [20][3200/3216]  lr: 1.0684e-05  eta: 0:00:11  time: 0.7170  data_time: 0.0023  memory: 58940  loss: 0.8469
2022/11/03 23:27:21 - mmengine - INFO - Exp name: l-384-arc_round3_eql_20221103_100202
2022/11/03 23:27:21 - mmengine - INFO - Saving checkpoint at 20 epochs
2022/11/03 23:29:08 - mmengine - INFO - Epoch(val) [20][74/74]  accuracy/top1: 92.8553  accuracy/top5: 96.8044
```

实验日志信息截图

作者们也将于本周四晚在 **OpenMMLab 社区开放麦**开展直播，介绍冠军方案更多的比赛细节，分享比赛经验，欢迎预约观看！



OpenMMLab浦视

已结束直播，可观看回放

观看回放

ACCV 2022 大规模细粒度图像分类冠军方案

视频号

引用:

- [1] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
- [2] Swin Transformer V2: Scaling Up Capacity and Resolution
- [3] ArcFace: Additive Angular Margin Loss for Deep Face Recognition
- [4] Masked Autoencoders Are Scalable Vision Learners
- [5] SimMIM: A Simple Framework for Masked Image Modeling
- [6] Sub-center ArcFace: Boosting Face Recognition by Large-scale Noisy Web Face  
s
- [7] AdaptiveMarginArcFace: Google Landmark Recognition 2020 Competition Third  
Place Solution
- [8] The Equalization Losses: Gradient-Driven Training for Long-tailed Object Recog  
nition
- [9] Learning with not Enough Data Part 1: Semi-Supervised Learning
- [10] Model soups: averaging weights of multiple fine-tuned models improves accur  
acy without increasing inference time

公众号后台回复“**极市直播**”，获取极市直播第**106**期【阿里达摩院：兼顾速度与精度的高效目标检测框架**DAMO-YOLO**】回放以及**PPT**



**极市平台**

为计算机视觉开发者提供全流程算法开发训练平台，以及大咖技术分享、社区交流、竞...  
848篇原创内容

公众号

## 极市干货

**技术干货：**数据可视化必须注意的30个小技巧总结 | 如何高效实现矩阵乘？万文长字带你从CUDA初学者的角度入门

**实操教程：**Nvidia Jetson TX2使用TensorRT部署yolov5s模型 | 基于YOLOV5的数据集标注 & 训练，Windows/Linux/Jetson Nano多平台部署全流程



# CV技术社群邀请函 #

△长按添加极市小助手

添加极市小助手微信 (ID : cvmart2)

备注：姓名-学校/公司-研究方向-城市（如：小极-北大-目标检测-深圳）

即可申请加入极市目标检测/图像分割/工业检测/人脸/医学影像/3D/SLAM/自动驾驶/超分辨率/姿态估计/ReID/GAN/图像增强/OCR/视频理解等技术交流群

极市&深大CV技术交流群已创建，欢迎深大校友加入，在群内自由交流学术心得，分享学术讯息，共建良好的技术交流氛围。

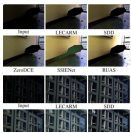
点击阅读原文进入CV社区

获取更多技术干货

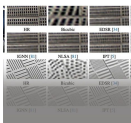
阅读原文

喜欢此内容的人还喜欢

ICCV23 | 将隐式神经表征用于低光增强，北大张健团队提出NeRCo  
极市平台



ICCV 2023 | 南开程明明团队提出适用于SR任务的新颖注意力机制（已开源）  
极市平台



YOLOv5帮助母猪产仔？南京农业大学研发母猪产仔检测模型并部署到Jetson Nano开发板  
极市平台



