PP-ShiTuV2

2023 年 11 月 25 日

模型及论文解析:

参考论文：https://arxiv.org/abs/2111.00775

摘要

PaddleClas 是一个提供了从数据处理、模型准备、模型优化、到预测部署全流程工具的图像分类开发套件。其中的轻量级图像识别模型 PP-ShiTu 更是综合了目标检测、图像分类、度量学习、图像检索等多重技术。轻量级图像识别系统 PP-ShiTu，由以下 3 个模块、主体检测、特征提取和向量搜索等模块组成。引入了流行的策略，包括度量学习、深度散列、知识蒸馏和模型量化，以提高精度和推理速度。PP-ShiTuV2 是基于 PP-ShiTuV1 改进的一个实用轻量级通用图像识别系统，由主体检测、特征提取、向量检索三个模块构成，相比 PP-ShiTuV1 具有更高的识别精度、更强的泛化能力以及相近的推理速度。主要针对训练数据集、特征提取两个部分进行优化，使用了更优的骨干网络、损失函数与训练策略，使得 PP-ShiTuV2 在多个实际应用场景上的检索性能有显著提升。

1 模型介绍

图 1 展示了 PP-ShiTu 的框架。PP-ShiTu 包含主体检测，即特征提取、向量搜索三个模块。pp-shitu 的管道非常简单。当获取一个图像时，我们首先检测到一个或几个主体区域来找到图像的主要区域。然后，使用 CNN 模型从这些区域中提取特征。特征是一个浮点向量或一个二进制向量。根据度量学习理论，特征意味着两个物体的相似性。两个特征之间的距离越短，原始的两个对象就越相似。最后，使用向量搜索算法找到图库中最接近从图像中提取的特征的特征，并使用相应的标签作为识别结果。

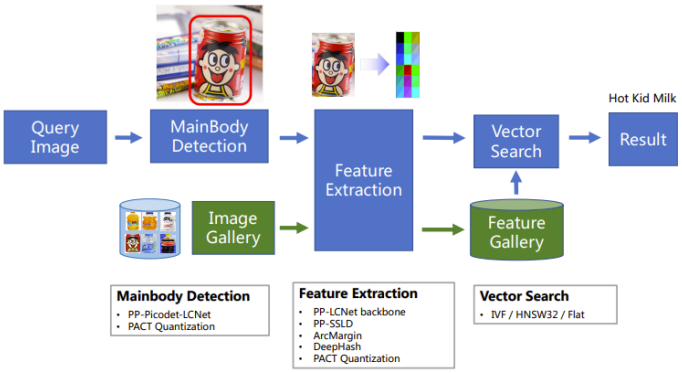

图 1: 算法框架

2 模型与策略

根基于 PP-ShiTuV1,PP-ShiTuV2 进行了一些优化, 从以下三个方面提高预测的精度与速度。

- 基于 ReID Strong Baseline 等方法，对特征提取模块进一步优化，精度提升 8 个点
- 轻量骨干网络 PP-LCNet v2，配合 SSLD 蒸馏算法，模型精度大幅提升。
- 超轻量主体检测算法 PP-PicoDet，快速检测出图像中的目标物体。

PP-ShiTuV2 整体结构如下所示：

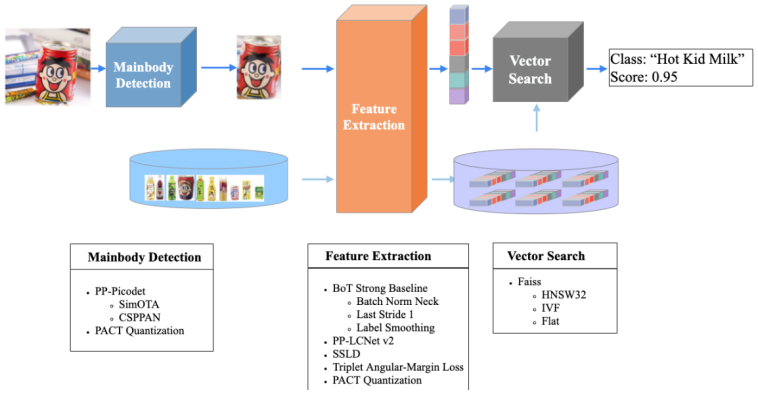

图 2: PP-ShiTuV2 网络结构

2.1 主体检测

主体检测是目前应用非常广泛的一种检测技术，它指的是检测出图片中一个或者多个主体的坐标位置，然后将图像中的对应区域裁剪下来进行识别。主体检测是识别任务的前序步骤，输入图像经过主体检测后再进行识别，可以过滤复杂背景，有效提升识别精度。

PP-PicoDet 以 PP-LCNet 为其骨干，并结合了许多其他的检测器训练技巧，如 PAN FPN(Liu 2018)，csp-net(Wang 2020)，SimOTA(Ge 2021)，最终成为第一个在 coco 数据集上 mAP（0.5：0.95）超过 0.30+ 以上的目标检测器。

一个轻量级的图像识别系统需要一个轻量级的主要主体检测模型。考虑到检测速度、模型大小、检测精度等因素, 最终 PP-ShiTu 中的主体检测使用的是 PP-PicoDet。

整个网络的结构如图 3 所示：

对 LCNet 的一些改进措施：

- Better activation function。为了提高 BaseNet 的拟合能力，将网络中的激活函数替换为原 ReLU 中的 H-Swish 函数，可以显著提高精度，但仅略微提高推理时间。

- SE modules at appropriate positions。SE(Hu、Shen，Sun2018) 模块自被提出以来已被大量网络使用。这是一个很好的方式来加权网络通道，以获得更好的功能，并被用于许多轻量级网络，如 MobileNetV3(Howard

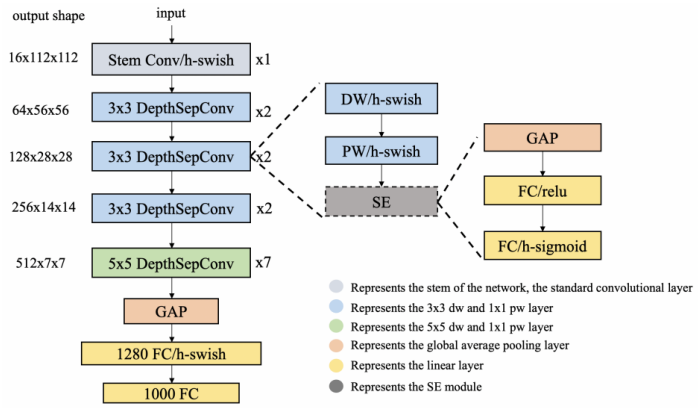

图 3: PP-LCNet 网络结构

2019)。但是，在 Intelcpu 上，SE 模块增加了推理时间，因此不能将其用于整个网络。事实上，通过大量的实验，发现 SE 模块越靠近网络的尾部，效果就越有效。所以只将 SE 模块添加到网络尾部的块中，从而得到更好的精度-速度平衡。SE 模块中两层的激活函数为 ReLU 和 H-Sigmoid。

- Larger convolution kernels。卷积核的大小往往会影响网络的最终性能。在 mixnet(Tan，Le2019) 中，作者分析了不同大小的卷积核对网络性能的影响，最终在网络的同一层中混合了不同大小的卷积核。然而，这种混合物减慢了模型的推理速度，因此试图增加卷积核的大小，但最小化推理时间的增加。最后，将网络尾部的卷积核的大小设为 5×5。

- Larger dimensional 1×1 conv layer after GAP。在 PPLCNet 中，GAP 后的网络输出维度较小，直接连接最终的分类层将失去特征的组合。为了使网络具有更强的拟合能力，在 GAP 层和最终分类层之间插入了一个 1280 维尺寸的 1x1conv，这为模型提供了更多的存储空间，而不会增加推理时间

PP-PicoDet 是由 paddleDetection 开发的一系列新的目标检测模型。具体来说，对于骨干模型，利用了上述的 PPLCNet，这有助于提高检测器的推理速度和 mAP。对于颈部，结合了 CSP-Net(Wang 2020a) 与 PANFPN(Liu 2018) 开发一种新的 CSP-PANFPN 结构，有助于提高特征图提取能力

2.2 特征提取

特征提取是图像识别中的关键一环，它的作用是将输入的图片转化为固定维度的特征向量，用于后续的向量检索。一个好的特征需要具备“相似度保持性”，即相似度高的图片对，其特征的相似度也比较高（特征空间中的距离比较近），相似度低的图片对，其特征相似度要比较低（特征空间中的距离比较远）。

图像识别的主要问题是如何从模型中提取更好的特征。因此，特征提取的能力直接影响了图像识别的性能。在特征提取的训练阶段，使用度量学习方法来学习图像的特征。下面将简要描述度量学习。

为了图像识别任务的灵活定制，我们将整个网络分为 Backbone、Neck、Head 以及 Loss 部分，整体结构如下图所示:

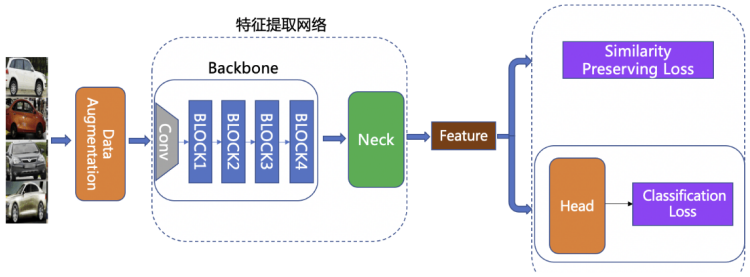

图 4: 特征提取结构

- Backbone 部分采用了 PP-LCNetV2_ base，其在 PPLCNet_V1 的基础上，加入了包括 Rep 策略、PW 卷积、Shortcut、激活函数改进、SE 模块改进等多个优化点，使得最终分类精度与 PPLCNet_x2_5 相近，且推理延时减少了 40 % 在实验过程中我们对 PPLCNetV2 _ base 进行了适当的改进，在保持速度基本不变的情况下，让其在识别任务中得到更高的性能，包括：去掉 PPLCNetV2_base 末尾的 ReLU 和 FC、将最后一个 stage(RepDepthwiseSeparable) 的 stride 改为 1。
- Neck 部分采用了 BN Neck，对 Backbone 抽取得到的特征的每个维度进行标准化操作，减少了同时优化度量学习损失函数和分类损失函数的难度，加快收敛速度。
- Head 部分选用 FC Layer，也就是全连接层。使用分类头将 feature 转换成 logits 供后续计算分类损失。

- Loss 部分选用 Cross entropy loss 和 TripletAngularMarginLoss，在训练时以分类损失和基于角度的三元组损失来指导网络进行优化。我们基于原始的 TripletLoss (困难三元组损失) 进行了改进，将优化目标从 L2 欧几里得空间更换成余弦空间，并加入了 anchor 与 positive/negtive 之间的硬性距离约束，让训练与测试的目标更加接近，提升模型的泛化能力

- Data Augmentation 考虑到实际相机拍摄时目标主体可能出现一定的旋转而不一定能保持正立状态，因此在数据增强中加入了适当的随机旋转增强，以提升模型在真实场景中的检索能力。

2.2.1 BNNeck

Batch Norm Nack。考虑网络的任意隐藏层。上一层的 activations 只是这一层的输入。例如，从下图中的第 2 层的角度来看，如果“清除”所有前面的层，则第 1 层的 activations 与原始输入没有区别。要求对第一层的输入进行归一化的相同逻辑也适用于这些隐藏层中的每一个。每个隐藏层

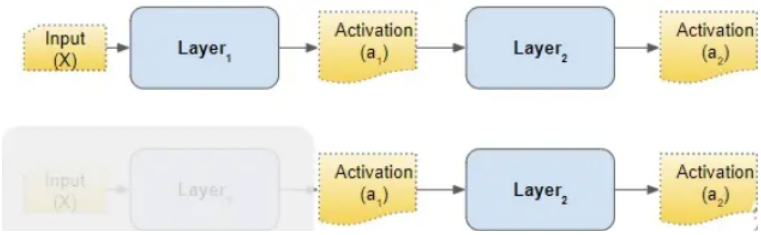

图 5: 层结构

的输入都是来自前一层的 activations，且必须归一化。换句话说，如果能够以某种方式对来自每个前一层的 activations 进行归一化，那么梯度下降会在训练期间更好收敛。这正是 Batch Norm 层所做的。

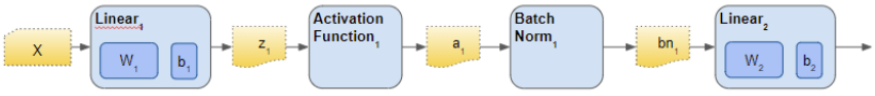

图 6: Batch Norm

Batch Norm 只是插入在隐藏层和下一个隐藏层之间的另一个网络层。它的工作是从第一个隐藏层获取输出并在将它们作为下一个隐藏层的输入传递之前对其进行标准化。

如上图，Batch Norm 层在第 1 层的 activations 到达第 2 层之前对其进行标准化

```
class BNNeck(nn.Layer):
    def __init__(self, num_features, **kwargs):
        super().__init__()
        weight_attr = paddle.ParamAttr(
             initializer =paddle.nn. initializer .Constant(value=1.0))
        bias_attr = paddle.ParamAttr(
             initializer =paddle.nn. initializer .Constant(value=0.0),
            trainable =False)

        if 'weight_attr' in kwargs:
            weight_attr = get_param_attr_dict(kwargs['weight_attr'])

        bias_attr = None
        if 'bias_attr' in kwargs:
            bias_attr = get_param_attr_dict(kwargs['bias_attr'])

        self.feat_bn = nn.BatchNorm1D(
            num_features,
            momentum=0.9,
            epsilon =1e-05,
            weight_attr=weight_attr,
            bias_attr=bias_attr)

        self.flatten = nn.Flatten()
```

对于 BNNeck 进行了定义：初始化了权重参数与偏置参数；设置了特征维度，移动平均的动量值，为数值稳定性而加到分母上的值以及指定权重参数和偏置参数的属性。

2.2.2 Loss

PP-ShiTuV2 在损失函数方面的优化主要是使用了 TripletAngularMarginLoss。1. 训练集中随机选取一个样本：Anchor（a）

2. 再随机选取一个和 Anchor 属于同一类的样本：Positive（p）

3. 再随机选取一个和 Anchor 属于不同类的样本：Negative（n）这样 <a, p, n> 就构成了一个三元组：学习目标是让 Positive 和 Anchor 之间的

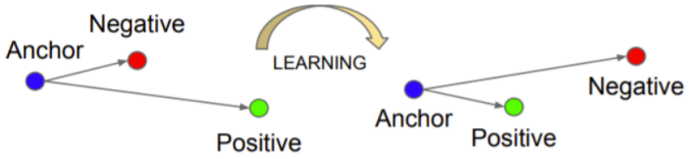

图 7: 学习过程

距离 D (a,p) 尽可能的小，Negative 和 Anchor 之间的距离 D (a,n) 尽可能的大：

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T}$$

图 8: 目标公式

优化目标 距离用欧式距离度量，+ 表示括号内公式的值大于零的时候，

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

图 9: 优化目标

取该值为损失值，而小于零的时候，损失值则为零。也可以这么表示：

其中 α 迫使 positive pairs (a, p) 和 negative pairs (a, n) 之间有一个 margin(α)（也就是外边距属性）。$\mathcal{T}$ 是训练集中所有可能的三元组的集合。

$$L = \max(D(a, p) - D(a, n) + \alpha, 0)$$

图 10: 优化目标

2.2.3 Label Smoothing

标签平滑应用于分类任务。传统的分类任务用的是交叉熵损失，而监督 label 用的是 one-hot 向量。因为交叉熵是相对熵在 one-hot 向量前提下的一种特例。但是 one-hot 是一种很强的监督约束。为了缓和 label 对于网络的约束，LS 对标签做了一个平滑：

$$q_i = \begin{cases} 1 - \frac{N-1}{N}\varepsilon & \text{if } i = y \\ \varepsilon / N & \text{otherwise} \end{cases}$$

图 11: 标签平滑

举个例子，加入原始的 label 是 [0，0，1，0，0，0]，平滑参数设置为 0.1，则平滑之后的 label 就会变成 [0.02，0.02，0.9，0.02，0.02，0.02]，计算损失时由交叉熵换回原始的相对熵。经过标签平滑之后，网络的过拟合程度也会被抑制一点。

2.2.4 PP-LCNetV2

骨干网络对计算机视觉下游任务的影响不言而喻，不仅对下游模型的性能影响很大，而且模型效率也极大地受此影响，但现有的大多骨干网络在真实应用中的效率并不理想，特别是缺乏针对 Intel CPU 平台所优化的骨干网络，我们测试了现有的主流轻量级模型，发现在 Intel CPU 平台上的效率并不理想，然而目前 Intel CPU 平台在工业界仍有大量使用场景，因此我们提出了 PP-LCNet 系列模型，PPLCNetV2 是在 PPLCNetV1 基础上所改进的。PP-LCNetV2 模型的网络整体结构如上图所示。PP-LCNetV2 模型是在 PP-LCNetV1 的基础上优化而来，主要使用重参数化策略组合了不同大小卷积核的深度卷积，并优化了点卷积、Shortcut 等。

PP-LCNetV2 的优化部分主要从 Rep 策略，PW 卷积和 ShortCut 三个方面来讲

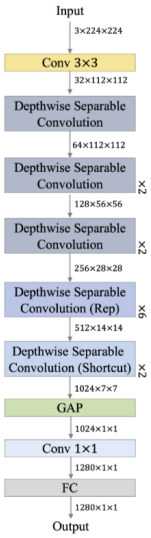

图 12: LCNetV2 网络框架

- Rep 策略：卷积核的大小决定了卷积层感受野的大小，通过组合使用不同大小的卷积核，能够获取不同尺度的特征，因此 PPLCNetV2 在 Stage4、Stage5 中，在同一层组合使用 kernel size 分别为 5、3、1 的 DW 卷积，同时为了避免对模型效率的影响，使用重参数化（Re parameterization，Rep）策略对同层的 DW 卷积进行融合, 如图 13 所示

- PW 卷积：深度可分离卷积通常由一层 DW 卷积和一层 PW 卷积组成，用以替换标准卷积，为了使深度可分离卷积具有更强的拟合能力，我们尝试使用两层 PW 卷积，同时为了控制模型效率不受影响，两层 PW 卷积设置为：第一个在通道维度对特征图压缩，第二个再通过放大还原特征图通道，如下图所示。通过实验发现，该策略能够显著提高模型性能，同时为了平衡对模型效率带来的影响。流程如图 14 所示

- ShortCut：残差结构（residual）自提出以来，被诸多模型广泛使用，但在轻量级卷积神经网络中，由于残差结构所带来的元素级（element-wise）加法操作，会对模型的速度造成影响，我们在 PP-LCNetV2 中，以 Stage 为单位实验了残差结构对模型的影响，发现残差结构的使用

并非一定会带来性能的提高，因此 PPLCNetV2 仅在最后一个 Stage 中的使用了残差结构：在 Block 中增加 Shortcut

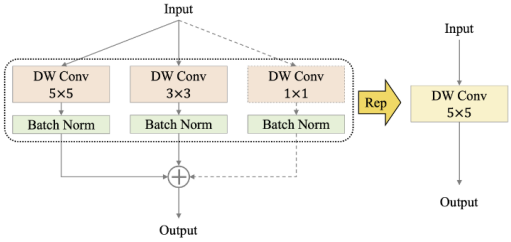

图 13: Rep 策略

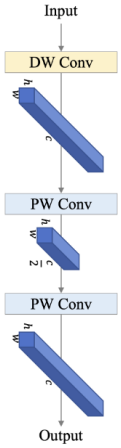

图 14: DW 卷积

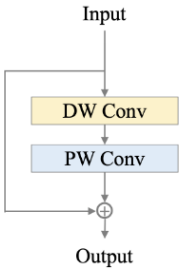

图 15: ShortCut

2.3 向量检索

向量检索技术在图像识别、图像检索中应用比较广泛。其主要目标是对于给定的查询向量，在已经建立好的向量库中进行特征向量的相似度或距离计算，返回候选向量的相似度排序结果。

在 PP-ShiTuV2 识别系统中，我们使用了 Faiss 向量检索开源库对此部分进行支持，其具有适配性好、安装方便、算法丰富、同时支持 CPU 与 GPU 的优点。在 PP-ShiTu 中，选择了三种算法：

- HNSW32(Malkov,ashunin 2018)
- IVF(Babenko, Lempisky 2012)
- FLAT

以满足不同场景的需求。值得注意的是，在向量搜索之前，需要建立一个特性库。特征库由从标记图像中提取的特征组成。在 PP-ShiTu 管道中，通过相似性搜索，从特征库中获得查询的标签。

3 代码讲解

PP-ShiTuV2 模型支持 PaddleX，其训练可以通过 UAPI 接口，可以借用其中的类模块进行参数配置，在接受模型名称和配置文件后可以快速的进行训练并进行预测。根据飞桨提供的文件，对 model_ config.yaml 进行分析如下:

```
1   LCNet:
```

```
    feature_maps:
    - 3
    - 4
    - 5
    scale: 2.5

    PicoDet:
        backbone: LCNet
        head: PicoHead
        neck: CSPPAN
    PicoHead:
        assigner:
          candidate_topk: 10
          iou_weight: 6
          name: SimOTAAssigner
        cell_offset: 0.5
        conv_feat:
          feat_in: 128
          feat_out: 128
          name: PicoFeat
          norm_type: bn
          num_convs: 4
          num_fpn_stride: 4
          share_cls_reg: true
        feat_in_chan: 128
        fpn_stride:
        - 8
        - 16
        - 32
        - 64
        loss_bbox:
          loss_weight: 2.0
          name: GIoULoss
```

```
35          loss_class :
36            iou_weighted: true
37            loss_weight: 1.0
38            name: VarifocalLoss
39            use_sigmoid: true
40          loss_dfl :
41            loss_weight: 0.25
42            name: DistributionFocalLoss
43        nms:
44          keep_top_k: 100
45          name: MultiClassNMS
46          nms_threshold: 0.6
47          nms_top_k: 1000
48          score_threshold: 0.025
49        prior_prob: 0.01
50        reg_max: 7
```

这一部分是对于 PP-ShiTuV2 里依赖的两个模型：PicoDet 与 LCNet 的设置, 包含 PicoDet 的卷积个数与卷积大小，步长；类别损失函数，bbox 损失函数，坐标损失函数以及他们的权重。

```
1  LearningRate:
2    base_lr: 0.4
3    schedulers :
4    - max_epochs: 100
5      name: CosineDecay
6    - name: LinearWarmup
7      start_factor : 0.1
8      steps: 300
```

这一部分是对于学习率的设置，使用 cosine decay 的衰减方式进行学习率调整，并且提供一种学习率优化策略-线性学习率热身 (warm up) 对学习率进行初步调整。在正常调整学习率之前，先逐步增大学习率。

```
1      CSPPAN:
2      num_csp_blocks: 1
```

```
    num_features: 4
    out_channels: 128
    use_depthwise: true
```

这个是 PicoDet 的 neck 部分，设置了 csp 块的数目，特征数等。

4 模型训练以及调参

4.1 模型训练

PaddleX 是基于飞桨技术生态的全流程深度学习模型开发工具。pp-ShituV2 可以利用 paddlex 进行开发训练。借助于工具箱模式，可以简单快速的上手。

需注意，在提交数据集时需要保证格式与要求的格式一致。

4.2 模型调参

图 16: 模型训练界面

里面涵盖的涉及影响模型训练因素的参数如下：

- 轮次：模型训练伦次。训练轮次越大，耗时越久，最终精度通常越高，但是过大也会容易过拟合。
- 批大小：模型训练批处理大小。Batch Size 越大，显存占用越高，训练耗时也越短（轮次单位为 Epochs 时）。

- 学习率：模型优化器的学习率。如果减小 Batch Size，一般需要等比例减小学习率。
- 热启动步数：在训练初始阶段以较小学习率训练的轮次，有助于训练稳定收敛。

4.2.1 调参过程

对于模型的好坏，主要依据是在 iou 从 0.5 到 0.95 之间的 ap 均值的大小来判定。在调参过程中，我主要针对于学习率和训练论数进行调节：

第一次训练：设置训练轮数为 30. 学习率为 0.08，模型评估为 0.461；

第二次训练进行了改进：设置训练轮数为 15. 学习率为 0.01，模型评估为 0.593，模型精度具有显著提升；

第三次训练增大了训练轮数：设置训练轮数为 30. 学习率为 0.01，模型评估为 0.662，模型精度具有显著提升；但是在修改模型权重，进行评估时，发现有权重的 map 可以达到 0.795，可能是学习率太小，训练轮次不够，没有达到最优解。

第四次训练与第五次训练进行了对比: 同样设轮次为 20，学习率一个为 0.01，一个为 0.02，模型评估分别为 0.622 与 0.65。

第六次训练主要修改学习率，增大轮数：设置训练轮数为 40. 学习率为 0.02，模型评估为 0.725.

图 17: 第六次训练结果

图 18: 第六次训练后的预测结果