

---

# 神经网络和深度学习 期末作业

---

钱皓然

18307110289

18307110289@fudan.edu.cn

朱笑一

18307130047

18307130047@fudan.edu.cn

梁敬聰

18307110286

18307110286@fudan.edu.cn

## 1 Introduction

在本次期末作业中，我们主要完成了三个任务。首先，我们运用了 Cityscapes 数据集上开源的语义分割模型对于测试视频进行测试并可视化；其次，我们考察了使用不同的方法初始化 backbone 后训练 Faster R-CNN 得到的不同训练结果并进行比较。最后我们设计与期中作业中模型相同参数量的 Transformer 网络模型，进行 CIFAR-100 的训练，并与期中作业 1 的模型结果进行比较。

在接下来的部分中，我们就将展示我们关于这些任务的成果。具体介绍顺序如下：在第二部分我们将会介绍我们使用的模型，测试数据以及测试结果，包含对于我们视频分割后的可视化以及两种不同语音分割结果直接的对比；在第三部分，我们会在简要介绍 VOC 数据集和 Faster R-CNN 后，展示我们使用不同 backbone 初始化方法后的实验结果并进行比较；在第四部分，我们会介绍我们针对 CIFAR-100 使用的数据增强方法，以及用于比较的卷积神经网络模型、图像 Transformer 模型和最后的实验结果。

12

## 2 The First Problem

在这一问题中，我们的目标是使用在 Cityscapes 数据集上开源的任意一个语义分割模型对于测试视频的每一帧进行测试并可视化。我们接下来会首先介绍我们使用的模型和测试视频数据，然后介绍我们的实验结果，并与其他模型进行简单的对比。

### 2.1 Model

我们最终选择的模型基于高通 AI 研究院的研究Borse et al. (2021)。这一模型的表现于 Cityscapes 数据集上所有模型中排名第八，所有开源模型中排名第一。他们的研究发现大部分的语义分割模型在边界检测的时候往往使用加权熵损失作为损失函数，但是加权熵损失往往忽略了像素距离目标边界的空间距离，并不能有效地测量预测边界和目标边界的局部空间变化，如平移、旋转或缩放等。为了解决这

<sup>1</sup> 实验代码请参考 <https://github.com/ljcleo/NeuNetTwo>。

<sup>2</sup> 模型可以从[https://drive.google.com/drive/folders/1D0E7L-8ovsyvZ\\_g0tpizyXt\\_K6u80zCi?usp=sharing](https://drive.google.com/drive/folders/1D0E7L-8ovsyvZ_g0tpizyXt_K6u80zCi?usp=sharing)下载。

一问题，作者提出了 Inverse transformation network，如图 1所示，这个网络使用两个边界映射图作为输入，并输出单应性矩阵的系数，作者再使用不同的度量来处理得到的系数。

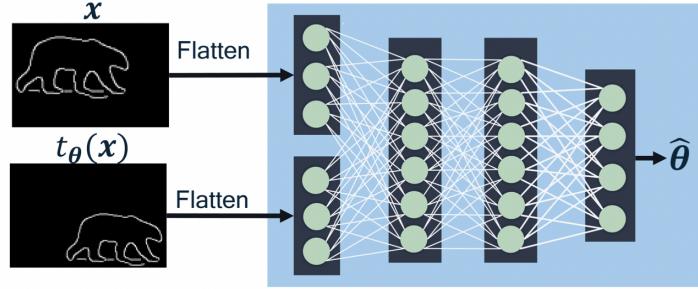


图 1: Inverse transformation network

作者将 Inverse transformation network 融入现有的语义检测模型中并有效的提升了效果，完整的框架如图 2所示。注意到这里的框架可以使用任意的现有语义分割模型，在我们的实验中，为了得到的最好的效果，我们使用了Tao et al. (2020) 的研究结果。

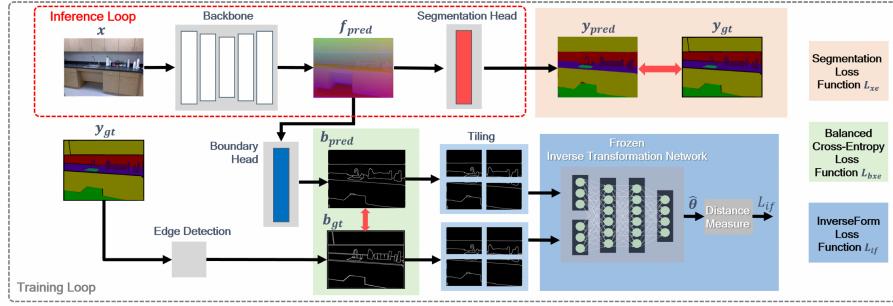


图 2: Overall framework for proposed boundary-aware segmentation

## 2.2 Test data

为了对于我们的模型进行测试，我们使用了 CamVid (Cambridge-Driving Labeled Video Database) 数据集。CamVid 是第一个具有目标类别语义标签的视频集合，提供了多个从驾驶汽车的角度拍摄的高质量视频。由于整个数据集比较大，我们只在 CamVid 测试数据集上进行测试和可视化，来验证我们的模型性能。

## 2.3 Experiments

在使用上述的模型和测试数据，我们得到了如下的实验结果，由于整个视频图片众多，我们抽选了部分图像进行展示，如图 3所示，完整视频详见我们最终提供的结果中。其中，左边为原图，右边为我们的模型的分割结果。可以看到，我们的模型基本上可以清楚的识别图片中不同的语义信息，并进行较好的分割。

我们也对比了其他的语义分割模型，这里以Zhao et al. (2017) 的 PSP net 为例，对比他们的结果和我们的结果如图 4所示，其中左图为 PSP net 的结果，右图为我们的结果。可以看到 PSP net 在图像较



图 3: Results of our model on the test data-set

暗的时候难以进行准确的分割，且对边界的划分较为模糊，相比较而言，我们的模型结果不但可以在阴暗的环境下继续进行，还可以对边界有更清晰明显的划分。

### 3 The Second Problem

在这一部分，我们会对 Faster R-CNN 模型进行不同的训练。在接下来的部分，我们首先对 VOC 数据集和 Faster R-CNN 模型进行简单的介绍，接下来介绍我们分别用不同的方法初始化 backbone 后得到的结果。

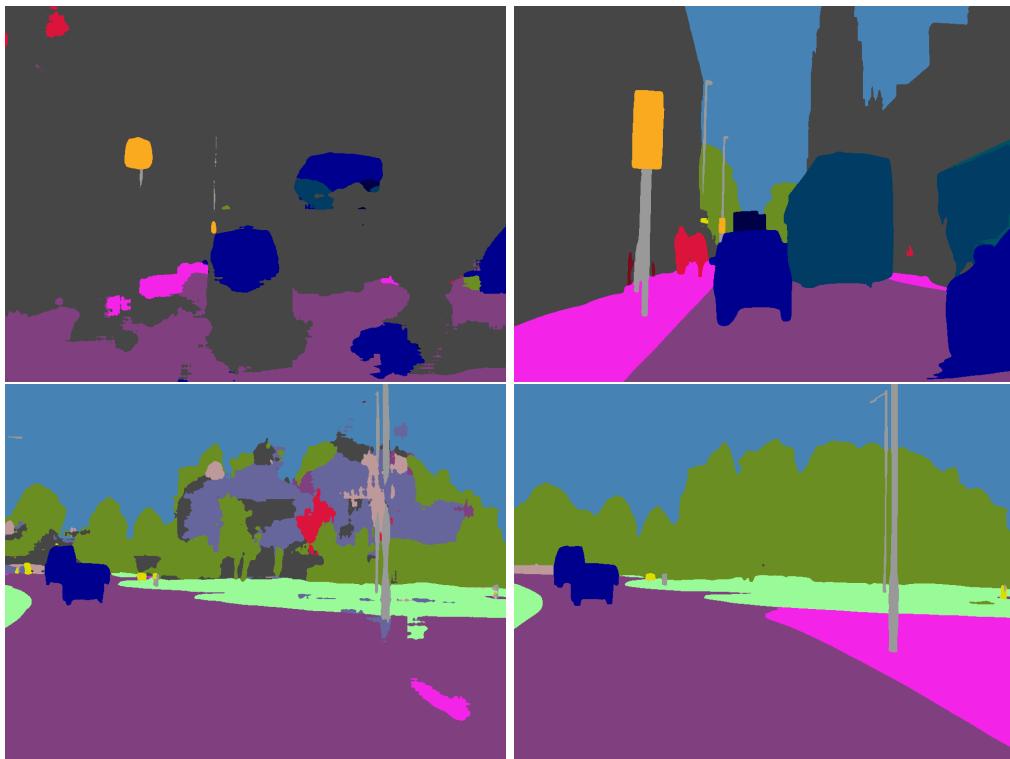


图 4: Comparison of our model with PSP net (left: PSP net; right: Ours)

### 3.1 VOC Dataset

PASCAL VOC 挑战赛 (The PASCAL Visual Object Classes) 是一个世界级的计算机视觉挑战赛，关注于分类，定位，检测，分割，动作识别等多个任务。VOC 数据集即为该挑战赛提供的数据集，包含一共 20 个类别。

在我们接下来的训练中，我们使用的是 VOC2007 数据集，我们把其中的训练集和验证集用以模型的训练，而测试集用以模型的测试。

### 3.2 Faster R-CNN

在使用 Faster R-CNN 对 VOC 数据集进行训练和测试之前，我们首先对 Faster R-CNN 进行大致的介绍。注意，这里我们使用的 Faster R-CNN 代码为 bubbliliing 提供的版本。Ross B. Girshick 在 2015 年发表的论文《Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks》Ren et al. (2015) 中提出了 Faster R-CNN 模型。该模型解决了 SPPnet 和 Fast R-CNN 模型中建议区域计算的瓶颈，从而使该模型计算更加迅速。

Faster R-CNN 算法的关键部分有三个：

1. 共享基础卷积层。该层用基础的卷积层 +RELU 激活函数 + 池化层提取图片的特征，这些特征被共享用于后续的 Region Proposal Networks 层。

2. Region Proposal Networks 层。该层用于生成 region proposals，并判断 anchors 为正例或负例，然后用 bounding box regression 对 anchors 进行修正，从而得到精确的 anchors。
3. Roi 池化层。利用第一步得到的特征和第二步得到的 Proposals，综合后提取出针对特定 proposals 的特征。

最终利用第三步得到的新特征，进行分类，并且再次微调 anchors。

为了实现这个网络，Faster R-CNN 使用了如图 5 所示的网络结构。该图的上半部分则为刚刚提到的共享基础卷积层，左下角的网络则为 Region Proposal Networks 层，右下角的网络则为 Roi 池化层与最终的分类器。

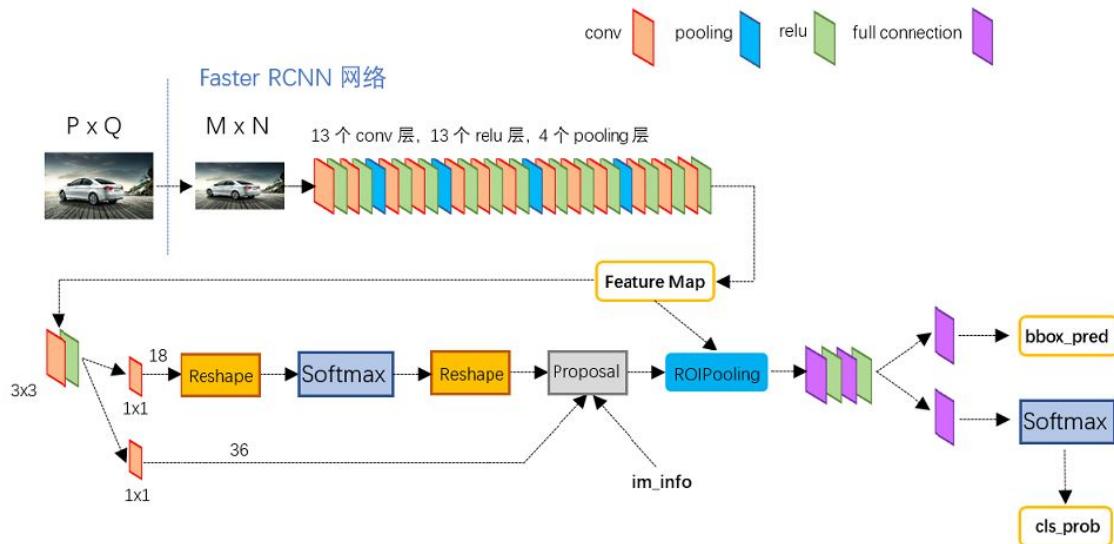


图 5: Structure of Faster R-CNN.

而其中的共享基础卷积层，又称 backbone 层，我们将采用 ResNet50 这一经典网络。

### 3.3 Experiments

在这一部分，我们介绍我们分别使用随机初始化，使用 ImageNet 预训练 backbone 网络和使用 coco 训练的 Mask R-CNN 的 backbone 网络参数，再使用 VOC 进行 fine tune 后得到的实验结果。

我们以 VOC2007 作为数据集进行训练与测试。由于模型参数量较大，加上硬件设备算力限制，我们设置 Batch Size 为 2。

我们设置初始学习率为 0.01，使用余弦退火的方式减少学习率，优化器为随机梯度下降（SGD），总共训练 100 个 epoch。训练集与测试集的损失值与测试集的 mAP 如图 6 所示。

可以看到，一方面，无论是哪种方法初始化 backbone，我们的模型的 loss 都在不断的下降，map 不断上升，证明了模型训练的有效性；另一方面，当我们对几个方法横向进行比较可以发现，使用随机初始化的方法得到的训练效果最差，最好的是使用 ImageNet 预训练 backbone 网络后得到的结果。这样的实验结果也证明了预训练 backbone 的有效性和重要性，缺乏有效的预训练可能导致权值太过随机而困在局部最优解，需要更长的训练用以得到较好的结果。

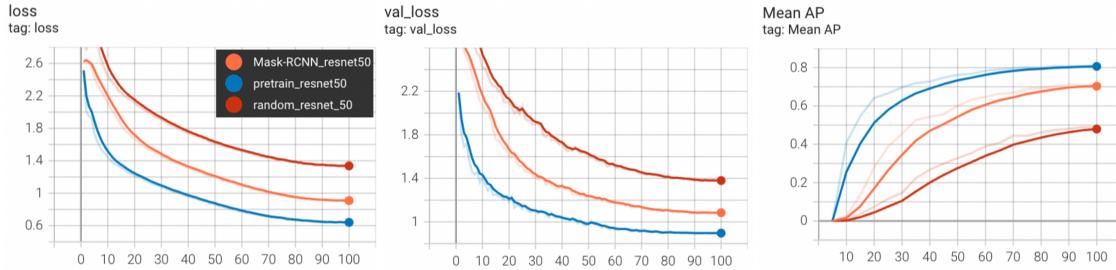


图 6: Loss and mAP of the Three Methods [the red line uses the random weights; the orange line uses the backbone weights of Mask R-CNN pretrained on COCO data-set; the blue line uses the backbone weight pretrained on ImageNet]

## 4 The Third Problem

在这一部分，我们会重温 CIFAR-100 数据集，并比较传统的卷积神经网络和图像 Transformer 模型在图像分类任务上的性能差异。首先我们会介绍训练过程中采用的数据集（即 CIFAR-100）和数据增强方法，然后简要地展示两种模型之间的异同，接下来列出训练采用的方法与配置，最后分析两种模型在图像分类实验中的表现和特点。

### 4.1 Dataset and Augmentation

本节的实验仍然采用 CIFAR-100 数据集 (Krizhevsky et al., 2009)，包括一个有 50,000 张图像的训练集和一个包括 10,000 张图像的测试集。每张图像的大小为  $32 \times 32$ ，并分别属于 100 个类别中的一个（这里不考虑粗分类标签）。不过，由于训练过程中需要对超参数进行调整，因此我们从训练集中划分出 10%，即 5,000 个样本作为验证集，并在训练过程中记录模型在验证集（而非测试集）上的损失函数与分类准确率，其余的 45,000 个样本则作为真正的训练集供模型学习；最终模型的分类效果则以测试集为准。

新近的图像分类模型，无论是卷积神经网络还是图像 Transformer，都依赖于庞大的训练样本以优化模型参数，如 ImageNet 等；甚至在此基础上，许多模型还需要在训练过程中引入基于图像的数据增强技术，才能满足训练数据的多样性。由于本实验只能采用 CIFAR-100 数据集训练模型，其图像分辨率与样本数量均远小于 ImageNet 数据集，因此数据增强的重要性不言而喻。

下面，我们会介绍训练过程采用的图像和批次数据增强方法。整个流程主要参考了 DeiT (Touvron et al., 2021) 的训练配置；具体而言，我们首先将图像数据增强应用到样本图像上，然后对一个样本批次应用批次增强，产生最终的图像和目标标签（100 维向量）。

#### 4.1.1 Image Data Augmentation

我们采用如下的顺序对图像进行处理，增加图像的多样性：

**随机裁剪并缩放** 我们为每张图像会随机选取一个矩形区域裁剪，裁剪后的图像重新缩放（保持长宽比）并填充至  $32 \times 32$ 。其中，矩形区域的长宽比在  $3:4$  到  $4:3$  之间随机选取，面积则至少大于某个阈值（作为超参数）。

**随机水平翻转** 由于水平翻转通常不会改变图像语义，因此裁剪、缩放后的图像会有 50% 的比例水平翻转。然而，我们不会随机垂直翻转图像。

**朴素随机增强 (TrivialAugment)** 除了上述最基础的图像数据增强方法外，还有数十种可用的图像变换手段可以用于数据增强，例如添加噪声、调节色系等。因此，许多工作致力于寻找最佳的变换选取规则，以获得良好的训练效果，如 AutoAugment、RandAugment 等。这些方法通常需要利用训练数据搜索出合适的图像变换种类和幅度等，而且增加了许多超参数。然而朴素随机增强 (TrivialAugment) (Müller and Hutter, 2021) 表明，只要随机选取变换和幅度（如图 7 所示），无须任何搜索和超参数，就能够帮助模型取得训练效果上的突破。这里我们采用 PyTorch 提供的开源实现，其中参数取默认值。

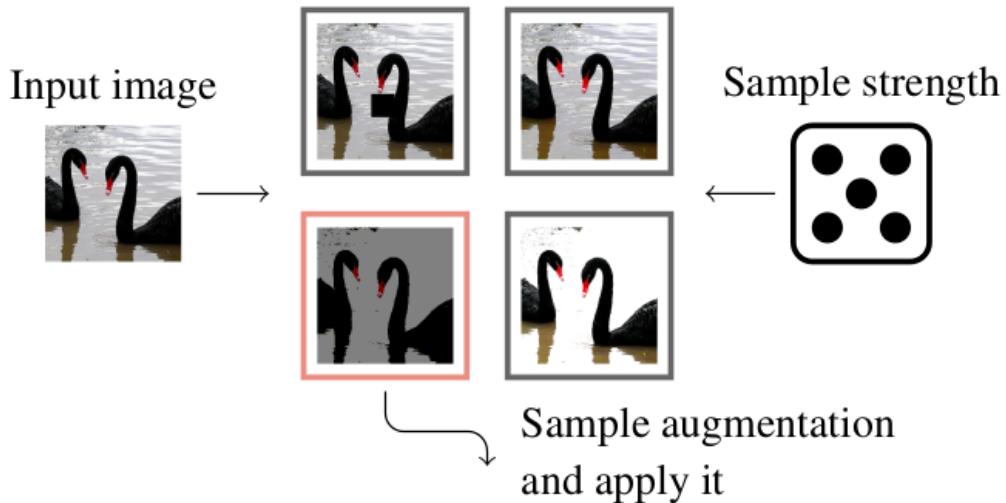


图 7: 朴素随机增强示例 (Müller and Hutter, 2021)

**通道标准化** 经过朴素随机增强后，图像的 RGB 三个通道会分别标准化，使均值和标准差均为 0.5，从而令不同图像有相近的颜色分布。这也是模型在预测阶段（即在验证集和测试集上）需要应用到输入图像的唯一变换。

**随机擦除 (RandomErasing)** 最后，每张图像会有一定的概率（作为超参数）擦除掉一个矩形区域，其长宽比在  $3:10$  到  $10:3$  之间随机选取，面积则介于原图像的  $1/50$  到  $1/3$ 。由于擦除的区域是用 0 填充的，所以该方法等价于 CutOut 方法 (Zhong et al., 2020; DeVries and Taylor, 2017)。

#### 4.1.2 Batch Data Augmentation

在完成图像的增强工作后，对于一个小批量数据，我们还会应用以下批次增强方法，进一步提高数据多样性：

**MixUp 和 CutMix** 这两种方法都是将另一个样本插入当前样本中，其中 Mixup (Zhang et al., 2017) 采用通道叠加，而 CutMix (Yun et al., 2019) 则采用裁剪覆盖，并且二者都会按照两个样本在新图像中的比例生成混合标签，表示为一个 100 维向量。具体而言，批次内的相邻样本会按 1 : 1 的概率随机挑选 MixUp 或 CutMix 方法，其中 Mixup 从 Beta(0.8, 0.8) 中随机选取叠加比例，CutMix 从 Beta(1, 1) 中随机选取裁剪比例。这里我们采用 `timm` (Wightman, 2019) 的统一开源实现。

**标签平滑** 在 MixUp 和 CutMix 的基础上，我们进一步通过标签平滑得到最终的软分类标签向量。`timm` 的 MixUp、CutMix 统一开源实现中已经提供了标签平滑功能，这里幅度固定为 0.1。

**重复增强 (Repeated Augmentation)** 与一般的数据增强方法不同，重复增强 (Berman et al., 2019) 的目的是在批次中加入同一样本的不同增强结果，帮助模型学习图像的增强不变特征。在本实验中，每个样本会重复 3 次，并分别施加不同的图像和批次增强技术；相应地，每个迭代只会利用训练集中 1/3 的样本，以保证训练批量总数不变。

## 4.2 Models

由于 CIFAR-100 数据集的图像分辨率为  $32 \times 32$ ，而为了避免图像严重失真，我们没有将图像放缩至 ImageNet 数据集的  $224 \times 224$  分辨率，因此实验选取的模型都是专门输入  $32 \times 32$  图像的模型。所有模型均利用 `timm` 提供的开源实现构建，其中部分组件根据实际设计有所调整。

### 4.2.1 Convolutional Network

实验中训练的传统卷积神经网络模型是 ResNeXt-29 (Xie et al., 2017)。该模型共有 4,868,004 个参数，包括输入阶段的  $3 \times 3$  卷积层（后续不接最大池化，与输入  $224 \times 224$  图像的 ResNeXt 不同），三个卷积阶段和最后的全局平均池化以及全连接层。其中，三个卷积阶段各包含 3 个瓶颈块，内部采用 32 组分组卷积，输出通道分别为 256、512 和 1024。具体的模型结构可以参考 Xie et al. (2017)。

### 4.2.2 Vision Transformer

与 ResNeXt-29 对比的是 ViT 模型 (Dosovitskiy et al., 2020)，属于图像 Transformer 模型。ViT 模型将图像分为若干块，并将每一块子图像编码，得到图像的序列表达（包括编码序列和位置序列等）。这个序列通过若干 Transformer 层后，序列末尾的输出就可以作为图像的特征编码，之后接一个全连接层就可以实现图像分类任务，如图 8 所示。

由于 CIFAR-100 数据集的图像分辨率较小，因此我们设定子图大小为  $2 \times 2$ （对应长度为  $16 \times 16 = 256$  的序列）。同时，为使模型参数量尽可能与 ResNeXt-29 相近，我们提出两种 ViT 结构：ViT-13 和 ViT-6（包含编码层和分类层的总层数分别为 13 和 6）；两种模型的配置与参数量如表 1 所示，它们的参数量都与 ResNeXt-29 非常接近。

## 4.3 Training Method and Configuration

在本实验中，ResNeXt-29 模型和两种 ViT 模型均从随机的初始化参数出发，使用 AdamW 优化预测标签的交叉熵损失。两种模型均在 CIFAR-100 的训练集（不含验证集）上直接训练 200 轮迭代，不采用任何预训练结果。由于三种模型都包含堆叠结构，因此训练过程中我们采用随机深度 (Stochastic

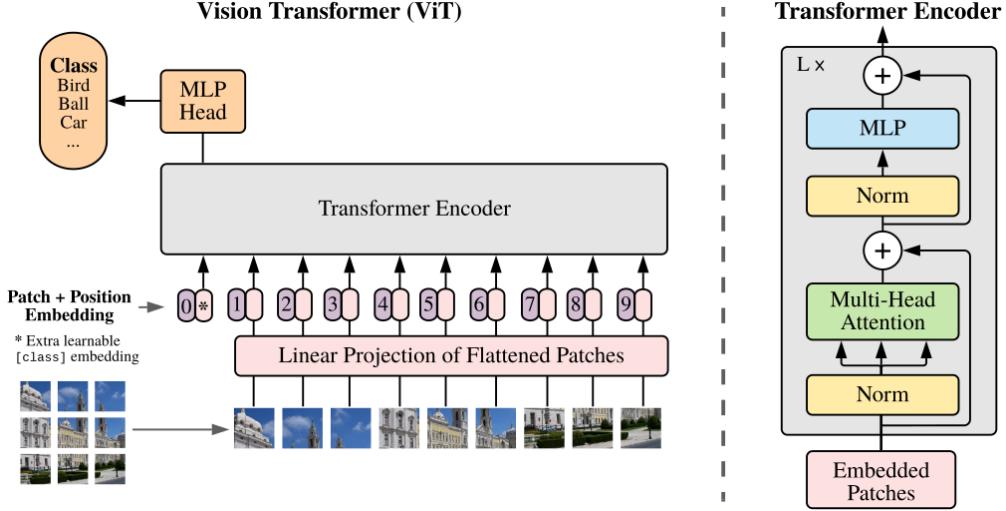


图 8: Vision Transformer 架构 (Dosovitskiy et al., 2020)

模型配置	ViT-13	ViT-6
编码维度	192	384
隐藏层维度	768	768
注意力头数	3	6
堆叠层数	11	4
参数量	4,915,684	4,878,820

表 1: ViT 模型配置与参数量

Depth) (Huang et al., 2016) 进行正则化，每次随机丢弃一部分卷积层或 Transformer 层，丢弃的概率则作为超参数调节。

模型在训练的不同阶段往往需要不同的超参数组合才能达到最佳的优化效果。为此，我们使用群体训练 (Population Based Training) 技巧 (Jaderberg et al., 2017)，同时用多组超参数训练多个模型，并定期用验证集分类准确率最高的模型参数替换最差的；同时，我们也替换较差模型的超参数，并允许一定概率的突变，期望在当前训练阶段实现模型性能的突破。图 9 展示了群体训练方法中各个模型的训练流程。

具体而言，我们同时维护 8 组超参数，用它们同步训练 8 个模型，并且每 10 轮迭代进行一次参数覆盖与超参数调整；最终用于测试的模型是在最后一轮迭代后，在验证集上分类准确率最高的模型。实验中纳入搜索目标的超参数与采样范围如表 2 所示。

#### 4.4 Results

表 3 显示了三种模型在验证集与测试集上的分类准确率。可以看出，两种 ViT 模型的表现都明显不如 ResNeXt-29，而 ViT-13 和 ViT-6 虽然一个更深一个更宽，但二者的性能差距并不明显，其中 ViT-13

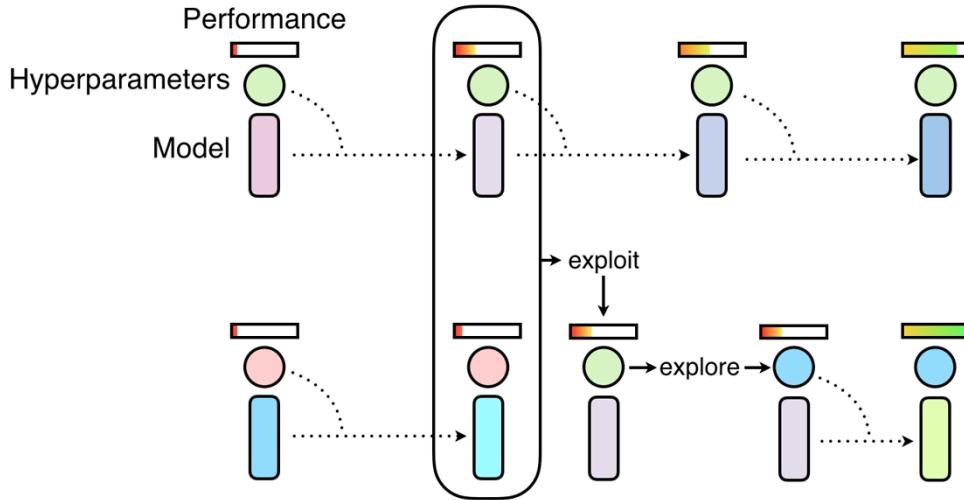


图 9: 群体训练流程 (Jaderberg et al., 2017)

超参数	采样分布
随机裁剪比例阈值	(0.1, 0.9)
随机擦除概率	(0, 0.5)
随机深度丢弃概率	(0, 0.3)
学习率	$(10^{-6}, 10^{-3})$
权值衰减	$(10^{-5}, 10^{-2})$
批量大小	$\{32, 64, 128, 256, 384\}$

表 2: 超参数与搜索范围

略优于 ViT-6。这一结果表明，至少在  $5 \times 10^6$  的参数量下，即使用各种手段提高了数据多样性，但朴素的图像 Transformer 模型仍然不能从  $32 \times 32$  这样小的图像分辨率下学习到更多的信息，与已有的、采用数据量更多且图像分辨率更大的 ImageNet 数据集预训练的结果差别较大。

模型	验证集		测试集	
	Top-1	Top-5	Top-1	Top-5
ResNeXt-29	<b>63.18</b>	<b>86.30</b>	<b>63.10</b>	<b>85.78</b>
ViT-13	42.44	71.30	42.93	71.18
ViT-6	39.52	66.90	38.79	65.21

表 3: 各模型在验证集与测试集的分类准确率

图 10 展示了上述结果对应的三个模型的训练和测试（在验证集上）曲线，可见在采用相同数据增强方法的情况下，ResNeXt-29 从训练初期就一直领先于 ViT-13 和 ViT-6，并且随着迭代次数的增加，两类模型之间的差距不断增大。不过值得注意的是，在训练后期 ResNeXt-29 频频出现过拟合的现象，需要依靠群体训练中其它模型的参数帮助纠正；而无论是 ViT-13 还是 ViT-6 都没有出现这一现象。这可能是因为图像 Transformer 模型有更强的泛化能力，也可能是由于模型仍然需要更多更好的数据继续学习。

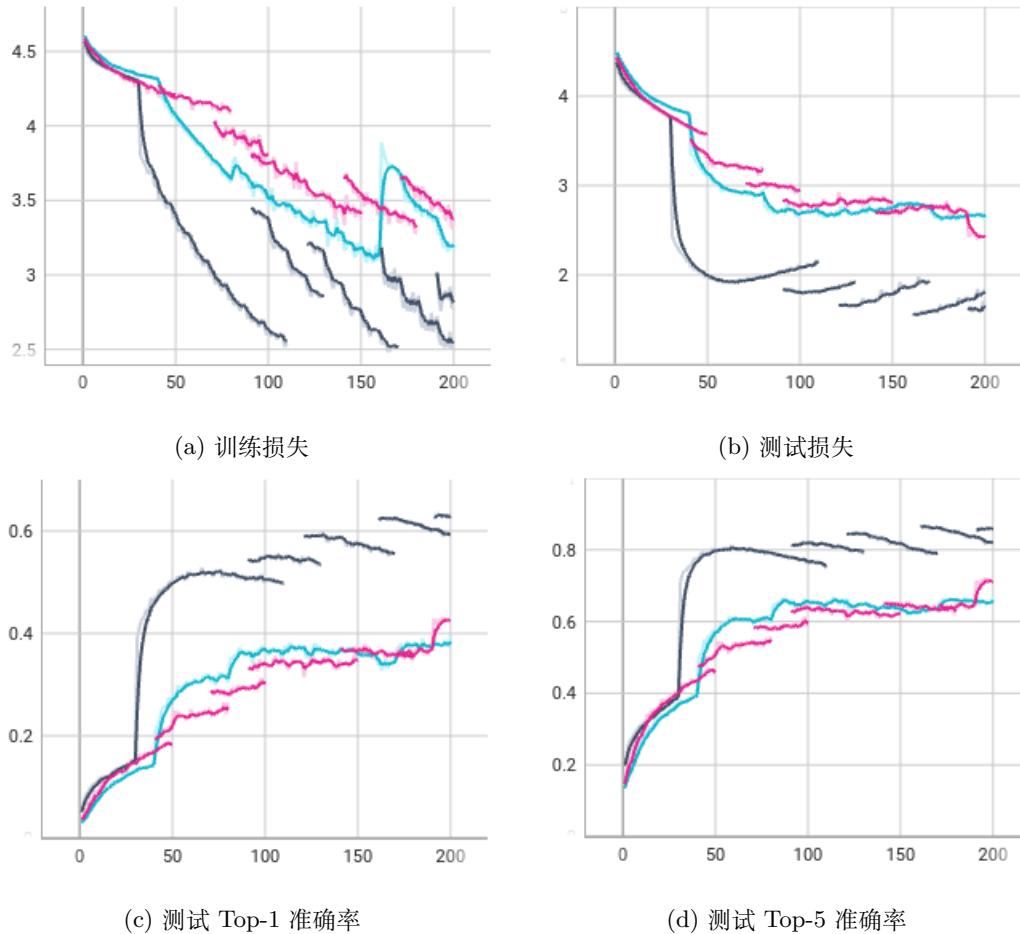


图 10：最优模型训练、测试曲线，曲线的断点代表模型参数被覆盖并重启训练。灰色：ResNeXt-29；粉色：ViT-13；青色：ViT-6

最后，图 11 展示了三类模型的所有群体训练模型的训练和测试曲线，由此可见群体训练同时实现了超参数的搜索与调整，同时在过拟合与欠拟合之间取得了不错的平衡。

## 5 Conclusion

在第一个任务中，我们的模型基本上可以清楚的识别测试数据中不同的语义信息，并进行较好的分割，相较于其他语义分割模型而言，我们的模型可以在阴暗的环境下有更好的识别，且对边界也有更好的处理。在第二个任务中，我们的模型在三种 backbone 初始化下都得到了不错的效果，并且我们发现，

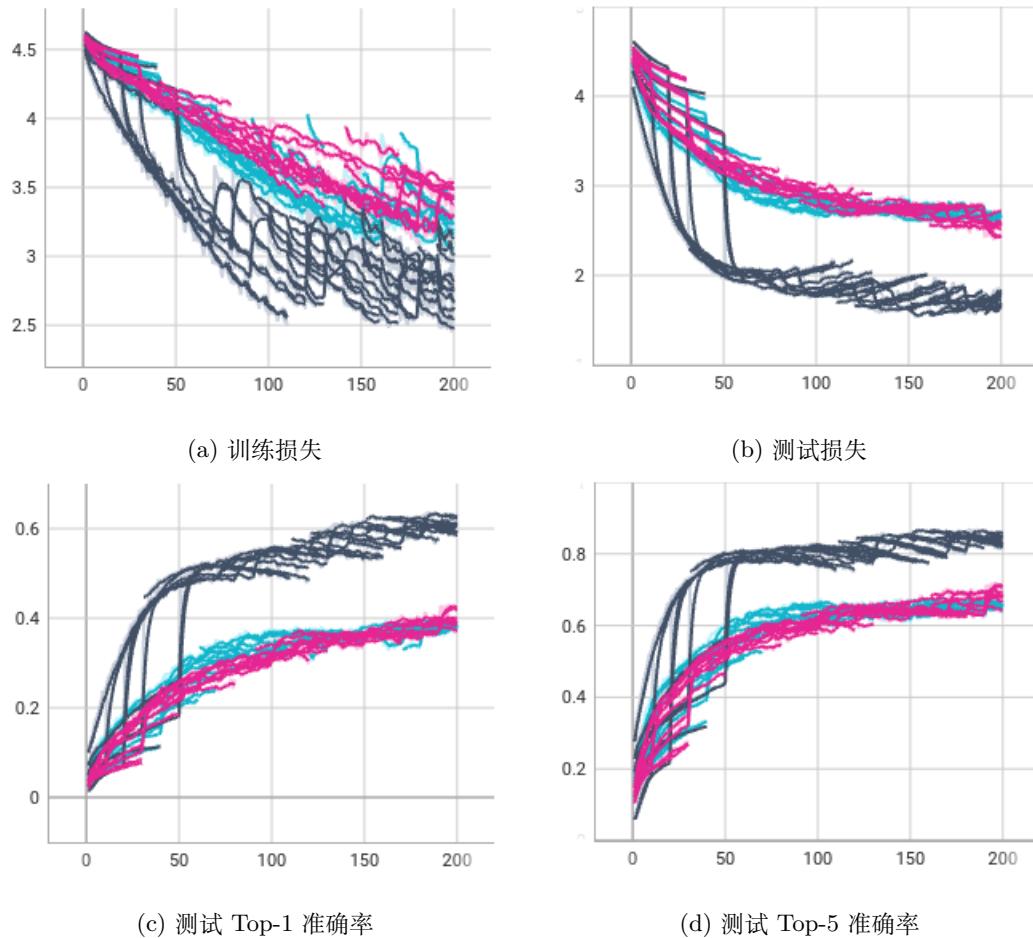


图 11: 全部模型训练、测试曲线, 曲线的断点代表模型参数被覆盖并重启训练。灰色: ResNeXt-29; 粉色: ViT-13; 青色: ViT-6

对 backbone 进行良好的初始化会对最终结果产生重要影响。在第三个任务中, 图像 Transformer 模型在参数量水平一致的前提下并未能够超越传统的卷积神经网络模型, 但仍然显示出学习稳定、不易过拟合等特性。

## 参考文献

Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5911, 2021.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 774–782, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.