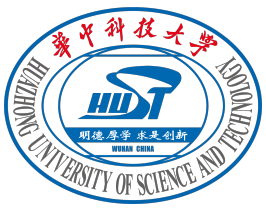
华中科技大学
人工智能与自动化学院

视觉认知工程 - 课程设计
基于深度学习的语义分割方法研究

孟繁鹏 (U201914689)
自动化 1905 班
fanpengmeng@hust.edu.cn
指导老师：曹治国、肖阳、陆昊

2022 年 8 月 23 日



摘要

语义分割（Semantic Segmentation）是指将图像中的每个像素进行类标签的过程，这里面的语义信息即为每个像素点的分类信息。详细地说，我们可以将语义分割视为像素级别的图像分类。在过去的十年中，深度学习已经成为解决通用视觉问题的一大主流模型，语义分割问题也包括其中。近年，随着注意力机制的推出，注意力机制也被应用在了语义分割领域。本次考察中，笔者使用 Weizmann Horse 数据集进行图像语义分割，分类为两类：马和非马。实现 UNet，UNet++，EMANet 进行了尝试，其中 UNet 和 UNet++ 的 MIoU 结果和 BIoU 结果均达到标准。EMANet 没有达到标准，笔者分析了可能的原因。最后，笔者对基于深度学习的语义分割方法进行了总结和展望。

考察作业说明

本次课程考察中。笔者选取了**题目**三，深度语义分割这一课题进行了调研和探索，并选用了 **UNet**、**UNet++** 和 **EMANet** 三种方法，在 Weizmann Horse 数据集上进行实验。在本考察作业中，笔者主要完成了以下工作：

- 复现了 EMANet，但是没有达到要求指标。分析了其失败的原因。
- 复现了 UNet、UNet++，并在数据集上达到了要求指标。
- 归纳总结了基于深度学习的语义分割的研究现状和发展方向。

实验代码已经开源在Github上，模型参数和训练记录上传到Google Drive。

1 概述

近年来，语义分割应用于静止的二维图像、视频，甚至三维或体素数据，是计算机视觉领域的关键问题之一。总体上看，语义分割是为完成场景理解铺平道路的高级任务之一，强调了场景理解作为一个核心的计算机视觉问题，越来越多的从图像推断知识中寻求解决方案。其中一些应用程序包括自动驾驶 [1] [2]，人机交互 [3]，计算摄影 [4]，图像搜索引擎 [5]，和增强现实技术等。这个问题在过去已经通过各种传统的计算机视觉和机器学习技术来解决。尽管这类方法很受欢迎，但深度学习革命已经扭转了局面，使得许多计算机视觉问题，尤其是语义分割问题，正在使用深度结构来解决 [6]，通常是卷积神经网络 [7] [8] [9]，它在准确性、有时甚至是效率方面都远远超过了其他方法。

在本次考察作业中，我们选用了 Weizemann Horse 数据集作为实验数据，该数据集中有 238 张图片，只有两个标签：马和非马。我们选用了如下三个算法来解决这一问题：

- UNet [10]：使用编码器-解码器的结构进行解决，是目前深度学习方法的语义分割模型的**模板结构**。
- Unet++ [11]：基于 UNet 结构进行了改进，效率更高，效果更好。
- EMANet [12]：在解码缓解引入 EM 注意力机制，进一步提升表现。

笔者在 UNet 上达到了 91.4%MIoU，74.5%BIoU；Unet++ 达到 88.9%MIoU，70.1%BIoU，均达到了要求，但是 UNet++ 并没有取得预期的好于 UNet 的效果，主要认为可能是数据集规模的问题。而 EMANet 表现为 80.6%MIoU，49.3%BIoU，没有达到预期效果，主要认为可能是域差异和数据集规模差异导致，抑或是笔者复现方法或参数调整出现错误。

2 实验方法

2.1 基础框架

语义分割主要有两大主流框架：全卷积神经网络（FCN）和编码器-解码器结构（Encoder-Decoder）

全卷积神经网络最早在 CVPR2015 中被提出 [13]。如图1，受卷积神经网络的启发得到，相较于卷积神经网络，去掉了最后用于分类的 Softmax 层，转而将特征图转化为语义分割的结果。这样做的缺点就是图像由于分辨率的丢失，精度会大幅度降低；但是这一思路构造了完善的特征提取过程，并为今后语义分割的奠定了结构基础。

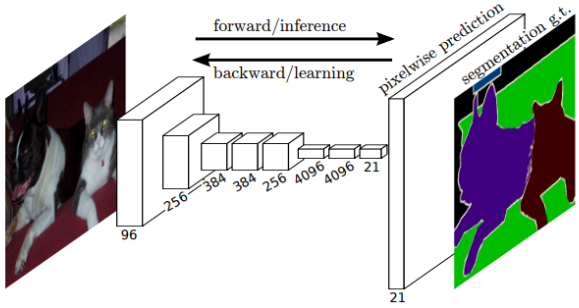


图 1: 全卷积神经网络 [13]

编码器-解码器结构是现在大多数分类语义分割工作的主流框架，最早在 CVPR2015 的 SegNet [14] 中被提出。如图2，在进行特征提取的编码器基础上，提出了解码器结构，旨在将特征图在保留图像上下文信息和位置信息的条件下还原到原分辨率。解码器采用了上采样策略。

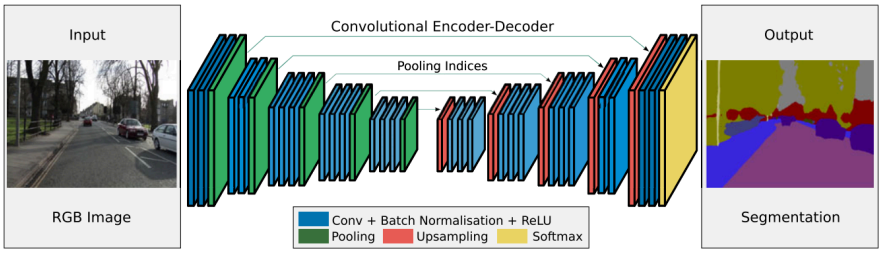


图 2: SegNet [14] 中的编码器-解码器结构

如今，也可以认为编码器-解码器结构是语义分割问题的一个方法论级别的解释。首先利用编码器降维、提取图像特征，如今很多工作利用了 ResNet [?] 或者 VGG [15] 的预训练模型。再利用解码器还原图像，经典的解码器会使用上采样 [14]、条件随机场 [16] 等方式；而如今的解码器很多引入了注意力机制 [12] [17]，通过和编码器中的特征图进行更高精度的对比和补充，实现更好的还原效果。

2.2 UNet

UNet [10] 最早在解决医学问题的过程中被提出。本文基于编码器-解码器结构 [14]，提出了一种 U 型的网络结构来获取上下文信息和位置信息。该工作在 2015 年的 ISBI Cell Tracking 比赛中获得了多个第一，是一项为了解决细胞层面的分割任务的挑战。发表于 MICCAI2015。

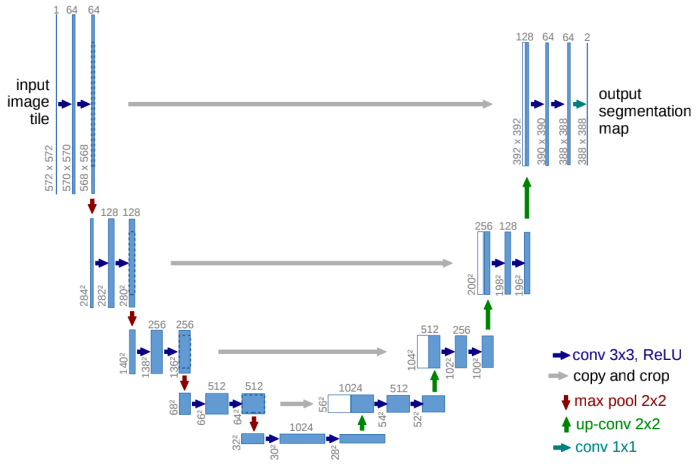


图 3: U-Net [10]

如图3，UNet 采用全卷积神经网络。左侧为**特征提取网络**，使用卷积和池化层，图片的尺寸减小，通道数增加。右边网络为**特征融合网络**，使用上采样产生的特征图与左侧特征图进行拼接操作。因为池化层会丢失图像信息、降低图像分辨率，且是永久性的，所以我们需要上采样让包含高级抽象特征低分辨率图片在保留高级抽象特征的同时变为高分辨率，然后再与左边低级表层特征高分辨率图片进行拼接操作。最后还原到原尺寸并输出特征图。

UNet 由于其出色的特征表达能力和轻量化结构，成为了小样本语义分割的范式，但在特征复杂的场景可能表现欠佳。至今，医学图像分割仍以 UNet 为主流，主要在于该问题的两大主要特点 [18]：1）数据少，样本小；2）区别于自动驾驶等数据集，目标场景简单。而本数据集也有着相似的特点，这也是我们选用 UNet 作为基础网络的原因。

2.3 UNet++

UNet++ [11] 基于 UNet 提出，相于 UNet 的跳跃链接，UNet++ 在解码器阶段提出了一种更稠密的连接方式，以强化原图像的特征。发表于 MICCAI2018。

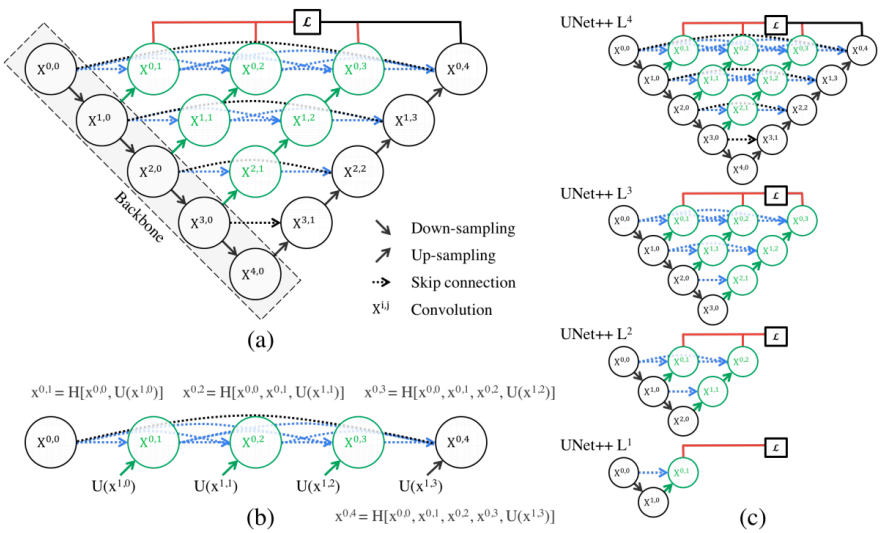


图 4: UNet++ 详解 [11]

在图4(a) 中，UNet++ 由一个编码器和解码器组成，它们通过一系列嵌套的密集卷积块进行连接。相比于 UNet 的长连接，新结构同时使用了浅层和深层的特征。同时使用深监督的方案，在每个链接后面加一个 1x1 的卷积核，相当于去监督每个层级的特征图，或者每个分支的 U-Net 的输出。图4(b) 详细分析了 UNet++ 的第一个跳跃通路。图4(c) 如果经过深度监督训练，UNet++ 可以在推理时进行修剪。这一策略大量减少了参数，使得本该指数级增加的参数只增加了有限的比例。

UNet++ 是一篇技巧性比较强的工作，作者在其中加入了许多小改进来提升网络的表现，笔者选取了该工作以锻炼自己对于基础框架实现一些技巧性地改变的代码能力。

2.4 EMANet: Expectation-Maximization Attention Networks

EMANet [12] 在解码器模块中引入了一个 EMA 单元，即最大化期望注意力单元，以提高解码器质量。该工作在 CVPR2019 中提出。

EMANet 仍采用编码器-解码器结构，主要在解码器中插入了一个**期望最大化注意力（EMA）单元**。EMA 单元提出解决了自注意力机制解决语义分割的一大瓶颈 [19]。自注意力机制需要生成一个巨大的注意力图，其空间复杂度和时间复杂度巨大。EMA 摒弃了在全图上计算注意力图的流程，转而通过期望最大化算法迭代出一组紧凑的基，在这组基上运行注意力机制，从而大大降低了复杂度。其中，E 步更新注意力图，M 步更新这组基。E、M 交替执行，收敛之后用来重建特征图。本文把这一机制嵌入网络中，构造出轻量且易实现的 EMA 单元。

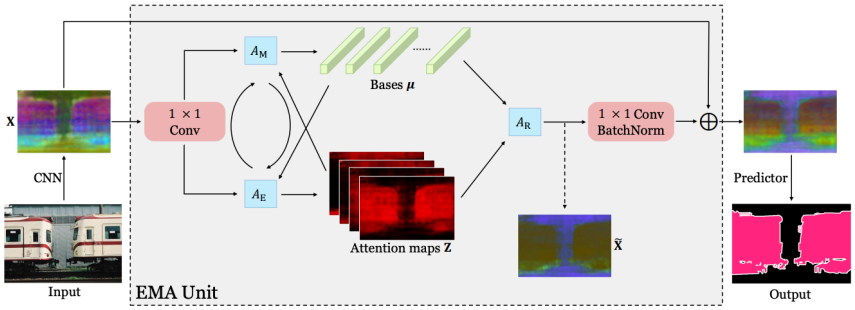


图 5: EMANet: 期望最大化注意力网络 [12]

期望最大化注意力模块（EMA）的结构如图5所示。除了核心的 EMA 之外，两个 1×1 卷积分别放置于 EMA 前后。前者将输入的值域从 R^+ 映射到 R；后者将 $\bar{X}$ 映射到 X 的残差空间。在训练过程中，迭代初值维护参考 BN 中 $running_{mean}$ 和 $running_{std}$ 的滑动平均更新方式，即：

$$\mu^{(0)} = \alpha\mu^{(0)} + (1-\alpha)\bar{\mu}^{(T)}$$

其中，α 表示动量，其中 $\alpha \in [0,1]$；$\bar{\mu}^{(T)}$ 表示 $\mu^{(T)}$ 在一个 batch 上的平均。

EMA 的迭代过程可以展开为一个 RNN，其反向传播也会面临梯度爆炸或消失等问题。公式也要求 $\mu^{(0)}$ 和 $\bar{\mu}^{(T)}$ 的差异不宜过大，不然初值 $\mu^{(0)}$ 的更新也会出现不稳定。RNN 中采取 $LayerNorm(LN)$ 来进行归一化是一个合理的选择。但在 EMA 中，LN 会改变基的方向，进而影响其语义。因为，本文选择 $L2NORM$ 来对基进行归一化，由此 $\mu^{(0)}$ 的更新轨迹便处在一个高维球面上。

近年随着注意力机制的兴起，很多工作都引入了注意力机制 [?]。语义分割的几个 state-of-the-art 算法都是基于 Transformer 实现，图6是 ADE20K 数据集语义分割的排行榜，也佐证了这一趋势。笔者选取了其中较易理解和实现的一个网络尝试了注意力机制在该问题上的表现。

Rank	Model	Validation↑ mIoU	Test Score	Params (M)	GFLOPs (512 x 512)	Extra Training Data	Paper	Code	Result	Year	Tags
1	**FD-SwinV2-G**	61.4				✓	Contrastive Learning Rivals Masked Image Modeling in Fine-tuning via Feature Distillation			2022	
2	**Mask DINO** (SwinL, multi-scale)	60.8		223		✓	Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation			2022	
3	**ViT-Adapter-L** (Mask2Former, BEiT pretrain)	60.5		571		✓	Vision Transformer Adapter for Dense Predictions			2022	
4	**SwinV2-G** (UperNet)	59.9				✓	Swin Transformer V2: Scaling Up Capacity and Resolution			2021	Swin-Transformer
5	**ViT-Adapter-L** (UperNet, BEiT pretrain)	58.4		451		✓	Vision Transformer Adapter for Dense Predictions			2022	
6	**SeMask** (SeMask Swin-L FaPN-Mask2Former)	58.2				✓	SeMask: Semantically Masked Transformers for Semantic Segmentation			2021	Swin-Transformer

图 6: ADE20K 语义分割数据集排行榜

3 评价指标

3.1 MIoU

Intersection over Union (IoU)，是交和并的比值。在语义分割的问题中，单类的交并比即为该类的真实标签和预测值的交和并的比值。全局 MIoU 对每一个类的 MIoU 取平均。计算公式如下：

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{i=0}^{k}p_{ij}+\sum_{i=0}^{k}p_{ji}-p_{ii}}$$

其中，i 为真实值，j 为预测值，p_{ij} 表示将 i 预测为 j。

在网络完成预测后，可以通过一个矩阵去展示出最终的预测效果，即为混淆矩阵 [20]。混淆矩阵对角线上的值是该类的交。混淆矩阵的每一行再加上每一列，最后减去对角线上的值就是该类的并。在本作业中，只有马和非马两种类别，MIoU 即等于 IoU。

3.2 Boundary IoU

MIoU 对于大物体边界的分割质量不敏感。随着物体尺寸的增加，物体内部像素数量以指数形式增加，而物体边界像素数量以线性形式增加，导致尺寸越大的物体，边界像素占总像素的比重越小。当物体内部像素被正确分割时，即使边界像素分割质量不好，MIoU 的值也比较高。所以我们引入了带有边界信息的 Iou 评价指标，Boundary IoU 来进行评价。

Boundary IoU 的定义为 [21]：

$$BoundaryIoU = \frac{(G_d \cap G) \cap (P_d \cap P)}{(G_d \cap G) \cup (P_d \cap P)}$$

上式中的 G_d 表示与 ground truth mask 的轮廓距离不大于的像素集合 P_d 表示与预测 mask 的轮廓距离不大于 d 的像素集合。距离 d 控制着 BIoU 的敏感程度，当 d 足够大时，BIoU 等效于 MIoU；当 d 比较小时，BIoU 会忽略 mask 中远离边界的像素，使即使尺寸较大的物体，BIoU 也会更关注物体边界附近的分割质量。

除了 BIoU 之外，其他的评价指标的对比也例举如下：

Name	Type	Definition	Symmetric	Preference	Insensitivity
Pixel accuracy	mask-based	$\frac{\lvert G \cap P \rvert}{\lvert G \rvert}$	✗	larger prediction	–
Mask IoU	mask-based	$\frac{\lvert G \cap P \rvert}{\lvert G \cup P \rvert}$	✓	–	boundary errors
Trimap IoU	boundary-based	$\frac{\lvert G_d \cap (G \cap P) \rvert}{\lvert G_d \cap (G \cup P) \rvert} = \frac{\lvert (G_d \cap G) \cap P \rvert}{\lvert (G_d \cap G) \cup (G_d \cap P) \rvert}$	✗	larger prediction	errors far from ground truth boundary
F-measure	boundary-based	$\frac{2 \cdot \tilde{p} \cdot \tilde{r}}{\tilde{p} + \tilde{r}},\ \tilde{p} = \frac{\lvert P_1 \cap G_d \rvert}{\lvert P_1 \rvert},\ \tilde{r} = \frac{\lvert G_1 \cap P_d \rvert}{\lvert G_1 \rvert}$	✓	–	errors on small objects
Boundary IoU	boundary-based	$\frac{\lvert (G_d \cap G) \cap (P_d \cap P) \rvert}{\lvert (G_d \cap G) \cup (P_d \cap P) \rvert}$	✓	–	errors far from predicted and ground truth boundaries

图 7: 各种评价指标对比 [21]

4 实验设置及结果

4.1 数据及其预处理

使用 Weizmann Horse 数据集，该数据集是一个比较简单的数据集，只有马或非马两个标签，同时场景比较单一，信噪比比较高。该数据集的场景和医学图像分割的场景比较相似，相比自动驾驶数据集，户外数据集等有一定的差异，在模型的选择上也应该给予注意。

训练验证集按 0.85:0.15 的比例随机划分。将 horse 图片与 mask 图片同时重新更改为 224*224 的大小，并将灰度值 0-255 映射到 0-1 之间。

4.2 实验参数

所有实验都训练 250 个 Epoch，没 50 次记录一次模型权重，并保留指标最好的权重作为最佳权重。

UNet 使用交叉熵损失。使用 RMSprop 优化器，学习率为 $1e-5$，权重衰减为 $1e-8$，动量值 0.9。

UNet++ 使用二分类交叉熵损失，同时联合了几个层次链接的损失。使用 Adam 优化器，学习率为 $1e-3$，权重衰减为 0，ϵ 值为 0.9，β 取值范围为 $[0.99, 0.999]$。

EMANet 使用交叉熵损失。使用 SGD 优化器，学习率为 $1e-2$，权重衰减为 $11e-4$，动量值 0.9。

4.3 实验结果

4.3.1 结果展示

指标	算法	结果/epoch		
		1	20	best
MIoU	UNet [10]	0.576	**0.904**	**0.914**
	UNet++ [11]	0.588	0.817	0.889
	EMANet [12]	**0.617**	0.791	0.806
BIoU	UNet [10]	**0.310**	**0.717**	**0.745**
	UNet++ [11]	0.164	0.536	0.701
	EMANet [12]	0.198	0.459	0.493

训练过程中的各项指标记录如下：

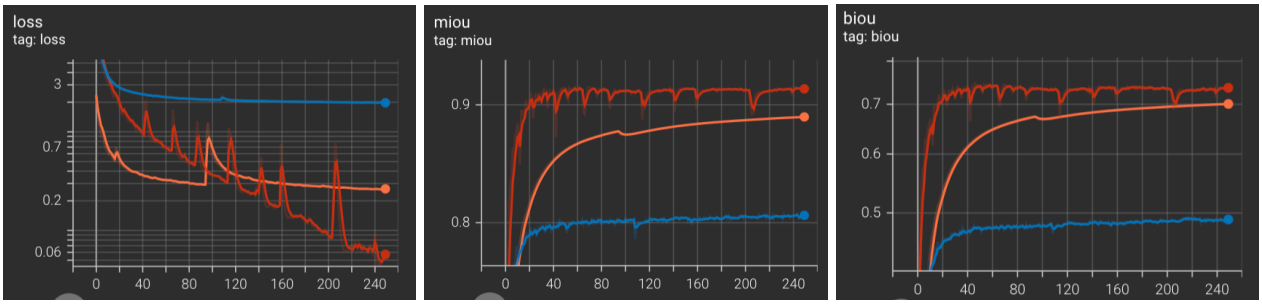


图 8: 训练过程中的指标记录

图11从左到右依次记录测试集损失、MIoU、BIoU。红色为 UNet，橙色为 UNet++，蓝色为 EMANet。可以看出训练大概在 40epoch 即可收敛，并且 UNet 表现最好，UNet++ 次之，EMANet 最差，甚至没能达到基本预期。

实验代码已经开源在Github上，模型参数和训练记录已经上传到Google Drive。

4.3.2 结果分析

Unet 的最佳表现 MIoU 达到 0.914，BIoU 达到 0.745，达到了指标要求。再次展示了 UNet 在小样本，简单数据集上的强悍表现。

```
epoch: 1  loss: 15.649153709411621  miou: 0.5767097473144531  biou: 0.3088115964617048
epoch: 11  loss: 3.0519306659698486  miou: 0.8837000983101981  biou: 0.6688653400966099
epoch: 21  loss: 1.7741763591766357  miou: 0.9041416304452079  biou: 0.7167538234165737
epoch: 31  loss: 1.2437993288040161  miou: 0.8999975749424526  biou: 0.7103876386369977
epoch: 41  loss: 0.9341796636581421  miou: 0.9142068454197475  biou: 0.7426883152553013
epoch: 51  loss: 0.7941555976867676  miou: 0.914757251739502  biou: 0.7419345038277763
epoch: 61  loss: 0.701221764087677  miou: 0.9126858030046735  biou: 0.7376757349286761
epoch: 71  loss: 0.5912718176841736  miou: 0.9156338146754673  biou: 0.7406317847115653
epoch: 81  loss: 0.478328555822372 44  miou: 0.9126417977469308  biou: 0.7354930468967983
epoch: 91  loss: 0.472358226776123 05  miou: 0.9119762693132673  biou: 0.7348955018179757
epoch: 101  loss: 0.35914430022239685  miou: 0.9126322610037667  biou: 0.7331997326442173
epoch: 111  loss: 0.3186754584312439  miou: 0.913494382585798  biou: 0.7354003361293248
epoch: 121  loss: 0.3002082407474518  miou: 0.9113785198756627  biou: 0.731762341090611
epoch: 131  loss: 0.2416183352470398  miou: 0.9124900272914341  biou: 0.7337113789149693
epoch: 141  loss: 0.23142491281032562  miou: 0.9093212400163923  biou: 0.7251384598868233
epoch: 151  loss: 0.1785852462053299  miou: 0.9143077305385044  biou: 0.7358013561793736
epoch: 161  loss: 0.5108697414398193  miou: 0.9046269825526646  biou: 0.7224701472691127
epoch: 171  loss: 0.13506238162517548  miou: 0.913853577205113  biou: 0.7385075432913644
epoch: 181  loss: 0.1211107149720192  miou: 0.9139528274536133  biou: 0.7368479456220355
epoch: 191  loss: 0.10548476129770279  miou: 0.9141295296805245  biou: 0.7333615847996303
epoch: 201  loss: 0.08410201221704483  miou: 0.9145603179931641  biou: 0.735565321786063
epoch: 211  loss: 0.10446136444807053  miou: 0.9107953480311802  biou: 0.7322735105242048
epoch: 221  loss: 0.06277231872081757  miou: 0.9131105286734444  biou: 0.7297664369855609
epoch: 231  loss: 0.06922727078199387  miou: 0.913639886038644  biou: 0.7328055245535714
epoch: 241  loss: 0.09410318732261658  miou: 0.9139423370361328  biou: 0.7336339269365583
======== best performance ========
epoch: 47  miou: 0.9144716262817383  biou: 0.7457083293369838
```

图 9: UNet 训练记录

Unet++ 的最佳表现 MIoU 达到 0889，BIoU 达到 0.701，达到了指标要求，但略逊于 UNet。分析

```
epoch: 1  train loss: 2.356012968456044  tar: 0.36984366572955074  miou: 0.588104520525251 1  biou: 0.16415749277387345
epoch: 11  train loss: 0.6190517299315509  tar: 0.07135189554708846  miou: 0.7731554105684355  biou: 0.44502074997146407
epoch: 21  train loss: 0.4835357648484847  tar: 0.04683216332512743 5  miou: 0.8166047699597417  biou: 0.5355863506291189
epoch: 31  train loss: 0.3918464332818985  tar: 0.03231794622671954 6  miou: 0.8393399286929364  biou: 0.5861195507137457
epoch: 41  train loss: 0.3593320189153446 6  tar: 0.02711501071120009 7  miou: 0.8525144464047528  biou: 0.6162871782788001
epoch: 51  train loss: 0.3482123858788434  tar: 0.02525743321679971  miou: 0.8607637635132178  biou: 0.6356328542159051
epoch: 61  train loss: 0.32352409730939 3  tar: 0.02151989290381179  miou: 0.8664455726498463  biou: 0.6489415045923594
epoch: 71  train loss: 0.3085742066882485  tar: 0.01941475042087 19  miou: 0.8702558981820612  biou: 0.6584144147109218
epoch: 81  train loss: 0.2978680449373582  tar: 0.01754412556286243 8  miou: 0.8733870356591711  biou: 0.6659812321738591
epoch: 91  train loss: 0.2938768456087393  tar: 0.01679695687968941 3  miou: 0.8760146974955848  biou: 0.6721982851133241
epoch: 101  train loss: 0.5952063851496753  tar: 0.06126425491974634 6  miou: 0.8734774016254863  biou: 0.6684734339086811
epoch: 111  train loss: 0.4182892906315186 5  tar: 0.03474520015365937 4  miou: 0.8753067468277429  biou: 0.6719439689410392
epoch: 121  train loss: 0.3596293517965117  tar: 0.02642006251741858  miou: 0.8773737415372 27  biou: 0.6760697077691484
epoch: 131  train loss: 0.3424098561791813  tar: 0.02418965692905819  miou: 0.8792097617911668  biou: 0.6797771557856802
epoch: 141  train loss: 0.3154606512364219  tar: 0.02010275691966 8  miou: 0.8807894350185 22  biou: 0.6829207554655838
epoch: 151  train loss: 0.3097309698076809  tar: 0.01955145307104377  miou: 0.8821087767746275  biou: 0.6855597518588872
epoch: 161  train loss: 0.2966077888713163 4  tar: 0.01737339211189571 6  miou: 0.8832162481720885  biou: 0.6878400530133929
epoch: 171  train loss: 0.2888905519948286  tar: 0.01608841595075586  miou: 0.8841936179968 15  biou: 0.6898618106953582
epoch: 181  train loss: 0.2866780735226238  tar: 0.01572948412093169 7  miou: 0.8850118852357932  biou: 0.6915983683146211
epoch: 191  train loss: 0.2800691373207989 7  tar: 0.01501043584636029  miou: 0.8857781603520007  biou: 0.6932470283137387
epoch: 201  train loss: 0.2798830840517493  tar: 0.01496416900087805 4  miou: 0.8864787773820629  biou: 0.6947723800778474
epoch: 211  train loss: 0.2722550972419627  tar: 0.01386046615045736 9  miou: 0.8871243011329977  biou: 0.6961369085085 9
epoch: 221  train loss: 0.2661507261150023 5  tar: 0.01284230324127083  miou: 0.8877811641483516  biou: 0.6975145000909018
epoch: 231  train loss: 0.2651442092131166  tar: 0.01255406244822284 7  miou: 0.8883463541666666  biou: 0.6987485113008272
epoch: 241  train loss: 0.2642131066497634 4  tar: 0.01252507941578241  miou: 0.8888607330412714  biou: 0.6999165111710655
======== best performance ========
epoch: 250  miou: 0.8893198939732143  biou: 0.7008771623883928
```

图 10: UNet++ 训练记录

原因可能在于：UNet++ 相较于 UNet 建立了许多额外的链接，而对于本实验只有 300 张左右图片、二分类的简单小数据集，这些链接可能会造成适得其反的效果。当然，也有可能是参数调整不够精细。

EMANet 的最佳表现 MIoU 只有 0807，BIoU 达到 0.493，远远未达到指标要求，笔者也在调整该模型上花费了较多的时间精力。

```
epoch: 1  loss: 12.860124588012695  miou: 0.617607729775565  biou: 0.1980192150388445
epoch: 11  loss: 3.384084939956665  miou: 0.7721413884844098  biou: 0.424071686608450 74
epoch: 21  loss: 2.7641239166259766  miou: 0.7906928062438965  biou: 0.4586740221296038
epoch: 31  loss: 2.521751642227173  miou: 0.7961071559361049  biou: 0.4697836126599993
epoch: 41  loss: 2.4040400981903076  miou: 0.7983580998011998  biou: 0.4729281152997698
epoch: 51  loss: 2.31534481048584  miou: 0.7973216601780483  biou: 0.4772425379071917
epoch: 61  loss: 2.2644922733306885  miou: 0.7999600682939801  biou: 0.4761856624058315
epoch: 71  loss: 2.1941521167755127  miou: 0.7994112287248883  biou: 0.4756346770695278
epoch: 81  loss: 2.165890693664551  miou: 0.79807220186506  biou: 0.4756431579589844
epoch: 91  loss: 2.1233530044555664  miou: 0.8014591080801827  biou: 0.48188791956220356
epoch: 101  loss: 2.1118860244750977  miou: 0.8034360068184989  biou: 0.4833174433026995
epoch: 111  loss: 2.157757043838501  miou: 0.7986091886247907  biou: 0.47844784600394114
epoch: 121  loss: 2.082195997238159  miou: 0.8036140033176967  biou: 0.4843803814479283
epoch: 131  loss: 2.0643324851989746  miou: 0.80333525793893  biou: 0.4830927848815918
epoch: 141  loss: 2.058267116546631  miou: 0.8032475880214146  biou: 0.4836804526192801
epoch: 151  loss: 2.0430195331573486  miou: 0.8041958127702985  biou: 0.4838927132742746
epoch: 161  loss: 2.039308547973633  miou: 0.8018719809395927  biou: 0.4817920071738107
epoch: 171  loss: 2.0196285247802734  miou: 0.8050702639988491  biou: 0.48643016815185547
epoch: 181  loss: 2.0130860805511475  miou: 0.8022776331220355  biou: 0.48591467312404085
epoch: 191  loss: 2.0127806663513184  miou: 0.8046022142682757  biou: 0.4867818696158273
epoch: 201  loss: 1.9875726699829102  miou: 0.804760183606829  biou: 0.48722076416015625
epoch: 211  loss: 1.988792896270752  miou: 0.8037101200648716  biou: 0.488100528717041
epoch: 221  loss: 1.9951585531234741  miou: 0.8035488128662109  biou: 0.4894940171922956
epoch: 231  loss: 1.9773972034454346  miou: 0.8056602478027344  biou: 0.49038692883082796
epoch: 241  loss: 1.9776828289031982  miou: 0.8060215541294643  biou: 0.4876015526907785
======== best performance ========
epoch: 217  miou: 0.8062949180603027  biou: 0.4932307175227574
```

图 11: EMANet 训练记录

分析原因可能在于：

- **实验设置存在问题**。或许可以通过调整损失函数、优化器和优化器参数解决这一问题，但是笔者已经在这方面花费了大量时间进行了尽可能多的尝试，认为是该原因的可能性较小。
- **数据集之间存在的较大的域差异 (domain gap)**。本实验采用 Weizmann Horse 数据集进行实验，该数据集有两大特点：1）数据集及小，只有约 300 张图片。2）场景单一，只有马和背景两个标签。在这种情况下，我们导入一些真实场景下的复杂数据集，如 Cityscap 等，中表现较好的模型，有可能会因为数据集的域差异导致结果不尽人意。
- **注意力机制对解决小样本、简单问题效用有限**。我们都知道，注意力机制需要大量的数据来训练 [22]。所以在本数据集的小样本条件下，注意力机制不一定能取得预想的效果。但是这一假设是否成立，可能还需要引入更多其他基于注意力机制设计的算法 [23] [17] [24]，如图12，来进行实验、对比结果。

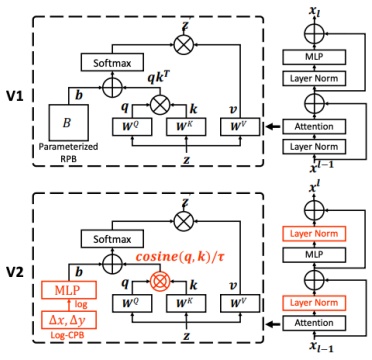


图 12: Swin-Transformer V2 [23]

4.4 后续实验构想

由于算力限制，笔者的实验做的并不算完善，如果算力允许，笔者还将尝试进行下列实验。

- 对于 EMANet，笔者采用了文章给定的损失函数、优化器和相似的参数进行训练，如果算力和时间允许应尝试更多实验设置进行试验以尝试获得更好的精度和结果。

- 选取更多的基于注意力设计的算法进行实验。验证注意力机制算法在小数据集上的表现。
- 二分类图像分割问题和复杂场景图像分割问题存在比较大的区别，可以分别尝试在两种问题下表现好的模型解决另一问题，或者对解决复杂环境的模型进行 finetune，以更好地解决二分类问题。

5 结论及感想

在本综述中，笔者回顾了现有的基于深度学习的语义分割方法，包括 FCN，Encoder-Decoder，Unet，基于注意力的语义分割等等。实现了 UNet、UNet++、EMANet 并在 Weizmann Horse Dataset 上训练测试，其中 UNet 和 UNet++ 的结果达到标准。EMANet 没有达到标准，笔者分析了可能的原因，同时也尝试分析了现有各种方法的使用范围和局限。笔者也再次意识到，在根据场景选择算法时，一定要考虑场景的特点，一些综合表现很优秀的模型，在一些特定的数据集上，不一定能取得预期的效果。

笔者之前虽然接触了一些计算机视觉与图像处理的实践和科研项目，但是一直没有系统性地学习整个人类视觉认知过程和计算机构建视觉认知的过程。在三位老师的讲解下，笔者第一次完整地构建了视觉认知的知识体系，从人类的视觉认知过程到现有的各种模拟人类认知机制创造的算法，都有了更深刻全面的了解。同时，三位老师课程中讲授知识的先进性，对于学术研究的前瞻性，是我在自动化学院前所未闻的。在本课程结束后，笔者将持续在计算机视觉和深度学习领域深耕，具体在 3D 视觉和具身智能的方向工作，并希望能做出一些有意义的工作。再次感谢三位老师在本课程中的讲解和指导，也欢迎老师持续指导和关注我的工作。

此外，笔者本是自动化方向的同学，但是对本课程很感兴趣故额外选择了这门课程，也再次感谢老师对于我这个来“蹭课”的同学耐心的指导和包容！

参考文献

[1] ESS A, MÜLLER T, GRABNER H, et al. Segmentation-Based Urban Traffic Scene Understanding.[C] // BMVC: Vol 1. 2009: 2.

[2] CORDTS M, OMRAN M, RAMOS S, et al. The cityscapes dataset for semantic urban scene understanding[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213–3223.

[3] OBERWEGER M, WOHLHART P, LEPETIT V. Hands deep in deep learning for hand pose estimation[J]. arXiv preprint arXiv:1502.06807, 2015.

[4] YOON Y, JEON H-G, YOO D, et al. Learning a deep convolutional network for light-field image super-resolution[C] // Proceedings of the IEEE international conference on computer vision workshops. 2015: 24–32.

[5] WAN J, WANG D, HOI S C H, et al. Deep learning for content-based image retrieval: A comprehensive study[C] // Proceedings of the 22nd ACM international conference on Multimedia. 2014: 157–166.

[6] VOULODIMOS A, DOULAMIS N, DOULAMIS A, et al. Deep learning for computer vision: A brief review[J]. Computational intelligence and neuroscience, 2018, 2018.

[7] CIRESAN D, GIUSTI A, GAMBARDELLA L, et al. Deep neural networks segment neuronal membranes in electron microscopy images[J]. Advances in neural information processing systems, 2012, 25.

[8] FARABET C, COUPRIE C, NAJMAN L, et al. Learning hierarchical features for scene labeling[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(8): 1915–1929.

[9] HARIHARAN B, ARBELÁEZ P, GIRSHICK R, et al. Simultaneous detection and segmentation[C] // European conference on computer vision. 2014: 297–312.

[10] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C] // International Conference on Medical image computing and computer-assisted intervention. 2015: 234–241.

[11] ZHOU Z, RAHMAN SIDDIQUEE M M, TAJBAKHSH N, et al. Unet++: A nested u-net architecture for medical image segmentation[G] // Deep learning in medical image analysis and multimodal learning for clinical decision support. [S.l.]: Springer, 2018: 3–11.

[12] LI X, ZHONG Z, WU J, et al. Expectation-maximization attention networks for semantic segmentation[C] // Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9167–9176.

[13] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431–3440.

[14] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481–2495.

[15] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[16] CHEN L-C, PAPANDREOU G, KOKKINOS I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834–848.

[17] CHEN Z, DUAN Y, WANG W, et al. Vision Transformer Adapter for Dense Predictions[J]. arXiv preprint arXiv:2205.08534, 2022.

[18] PHAM D L, XU C, PRINCE J L. A survey of current methods in medical image segmentation[J]. Annual review of biomedical engineering, 2000, 2(3): 315–337.

[19] WANG X, GIRSHICK R, GUPTA A, et al. Non-local neural networks[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794–7803.

[20] TOWNSEND J T. Theoretical analysis of an alphabetic confusion matrix[J]. Perception & Psychophysics, 1971, 9(1): 40–50.

[21] CHENG B, GIRSHICK R, DOLLÁR P, et al. Boundary IoU: Improving object-centric image segmentation evaluation[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 15334–15342.

[22] HAN K, WANG Y, CHEN H, et al. A survey on vision transformer[J]. IEEE transactions on pattern analysis and machine intelligence, 2022.

[23] LIU Z, HU H, LIN Y, et al. Swin transformer v2: Scaling up capacity and resolution[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12009–12019.

[24] WEI Y, HU H, XIE Z, et al. Contrastive Learning Rivals Masked Image Modeling in Fine-tuning via Feature Distillation[J]. arXiv preprint arXiv:2205.14141, 2022.