長安大學

二〇二一届毕业设计

高分辨遥感影像的精细化语义分割

学　　院：地质工程与测绘学院

专　　业：地理信息科学

姓　　名：李元宸

学　　号：2017901016

指导教师：席江波

完成时间：2021 年 5 月

二〇二一年六月

摘 要

近年来，卫星平台遥感技术不断发展完善，遥感技术在生产生活中如资源调查、农业病虫害监测、土地测绘、灾害监控以及军事等各个方面中都发挥了举足轻重的作用。遥感影像的解译是遥感领域的重要任务，而现在高分辨率遥感影像由于技术的发展正在更多地被应用，因而高分辨率遥感影像的语义分割成为了遥感领域的研究热点之一。高分辨率遥感影像的信息丰富，地物复杂，尺度多变，这使得高分遥感影像的语义分割更具挑战性。传统的图像分割方法更多基于图像的底层特征，而传统遥感影像语义分割方法需根据不同目标对象的特征，人工设计相应的特征提取器。这对于专业知识要求很高，且不能适应复杂的大规模应用场景，准确性不够高且计算时间较长，泛化能力有限。

随着深度学习在计算机视觉领域的持续发展，对它在遥感领域的探索也在持续深入。本文根据深度学习的卷积神经网络理论基础，引入了一种基于残差网络 ResNet 的 U-Net 网络遥感图像语义分割模型，并利用 BatchNormalization 优化模型训练速度。将 ResNet-18/50 作为 U-Net 的 encoder 部分，进行特征提取，并在每一卷积阶段输出特征图，以作为 decoder 部分的特征融合图。受到将注意力机制应用在计算机视觉领域的启发，尝试将基于通道注意力机制的网络模块 SE-Network 嵌入网络的 encoder 部分提高模型精度。除此之外，针对分割模型训练过拟合的现象，利用深度学习领域常用的 Dropout 方法来减轻此问题。

本文在 ISPRS 2D 的 Vaihingen 数据集上进行实验结果对比分析后，发现 ResNet-18-UNet 模型的总体分割精度到达了 86.5，且对建筑物的分割精度最高，f1 分数达到了 92.3%；更深的 ResNet-50-UNet 的表现更加出色，总体分割度接近 90%，并且汽车类别的分割精度相比 ResNet-18-UNet 提升了近 11%。另外，在网络中加入 BN 层确实可以加快训练速度。本文还尝试引入 SE 模块提高精度，但实验结果显示最终性能未能得到有效提升，后续工作中将尝试进一步优化超参数，提升引入 SE 模块后的模型精度。

关键词：卷积神经网络，语义分割，高分辨率遥感影像，残差网络

ABSTRSACT

In recent years, the remote sensing technology on satellite platform has been developing and improving consistently. Remote sensing technology plays an important role in production and normal life, such as resource survey, agricultural pest monitoring, land mapping, disaster monitoring and military. The interpretation of remote sensing images is an important task in the field of remote sensing. Nowadays, the high-resolution remote sensing images are being increasingly applied because of the development of technology, so the semantic segmentation of high-resolution remote sensing images has become one of the hot research topics in the field of remote sensing. The high-resolution remote sensing images have rich information, complex and changeable-scaled ground objects, which makes the semantic segmentation of high-resolution remote sensing images more challenging. Traditional images segmentation method is mostly based on the underlying features of the image, while the traditional semantic segmentation method of remote sensing image needs to design the corresponding feature extractor manually according to the characteristics of different target objects, which is a kind of high requirement for professional knowledge and can not adapt to complex large-scale application scenarios. It is not accurate enough and has a long calculation time and limited generalization.

With the continuous development of deep learning in the field of computer vision, in the field of remote sensing the exploration of it is also in-depth. Based on the convolutional neural network theory of deep learning, this paper introduces a remote sensing images semantic segmentation model U-Net network based on residual network ResNet, and optimizes the training speed of the model by batch-normalization. ResNet-18/50 is used as the encoder of U-net, and the feature is extracted, and the feature maps are output from each convolution stage to be the feature fusion mpaps of the decoder part. Inspired by the application of attention mechanism in computer vision, this paper attempts to embed SE-network module based on channel attention mechanism into the encoder part of the network to improve the model accuracy. In addition, the dropout method, a popular method preventing model from over-fiiting in the filed of deep learning, is used to mitigate the phenomenon of over-fitting.

After comparing the experimental results on the vaihingen data set of ISPRS 2D, it is turn out that the segmentation overall accuracy of ResNet-18-Unet model reaches 86.5%, and the segmentation accuracy of the building is the highest, its f1-scores reaching 92.3%; The deeper resnet-50-unet is more outstanding, with the segmentation overall accuracy approaching 90%, and the segmentation accuracy of car category is nearly 11% higher than ResNet-18-Unet. In addition, adding BN layer to the network can speed up the training speed. The paper also tries to introduce SE module to improve the accuracy, but the experimental results show that the final performance has not been effectively improved. In the subsequent work, further optimization of the superparameters will be tried to improve the model precision after the introduction of SE module

KEY WORDS: convolution neural network, semantic segmentation, high-resolution remote sensing image, residual network

.

目 录

装 订 线

第一章 绪论

1.1 研究背景及意义

遥感即遥远感知，是在不接触的情况下，对需要探索的目标进行远距离探测感知的一种技术[1]。遥感技术可以在高空的平台上，利用传感器来获取与地表特征有关的数据，经过处理得到有用的信息，然后对地物空间位置、性质、特征变化等进行研究。近年来，遥感技术不断发展，随着各种遥感平台的增多，遥感影像数据的来源呈现多样化，高分辨率的趋势。尤其是卫星遥感平台的高速发展， 使得航天遥感无论在空间、光谱还是时间分辨率方面都有很大的进步，这样的航天遥感具备了高光谱、高空间分辨率、全天候、准实时的对地观测能力，一系列卫星如 Landsat、SPOT、IRS 以及我国不断发射更新的高分系列卫星，无不说明着卫星遥感的重要作用。

高分辨率遥感是一种对遥感的数据质量要求很高的遥感技术。高分辨率遥感影像是指具有高空间分辨率的遥感影像，如空间分辨率 10m，5m，2m 甚至 1m 以下的空间分辨率。显著提高的遥感影像分辨率,使得遥感影像呈现出更加丰富多样的地物细节信息,几何特征和纹理等也更加明显。高分辨率遥感影像中具备极其丰富的地物信息，正因为这些丰富的地物信息，使高分辨率遥感影像也成为了主要的地理空间信息来源之一。它在地理数据更新、资源调查、城市建设、环境监测、土地测绘、灾害监控以及军事应用等方面都有着极其重要的应用。

由此看来，对遥感图像信息的提取占据了遥感研究方向的一部分重要地位。遥感图像解译即对遥感图像进行信息提取。传统的解译方法是靠人工信息提取，这对解译员的经验、专业知识和资料搜集能力要求较高，因此也会导致培训成本较高。同时人工解译自动化程度低下，当面临大规模或实时处理时，人工解译就难以满足需求而显得捉襟见肘了。而随着计算机技术的发展，现在许多图像的解译都是依靠计算机来完成。而近年来人工智能技术也因为数据爆炸而获得了绝佳的发展机会，其中与深度学习相关的技术飞速发展，因此遥感图像也可以利用各种神经网络来进行解译。

图像语义是对像素信息的进一步组织和抽象，使其符合人类活动的语义和逻辑[2]。1978 年，Ohta 等人提出,图像语义分割是指在图像中为每个像素分配一个预先定义的表示其语义知识的类别标签。当每个像素有了计算机能识别的标签，就能应用于目标识别乃至场景理解等应用场景。当前，越来越多的应用场景需要利用图像中的语义知识，包括自动驾驶中的场景识别与理解、无人机控制与应用[3]、医学影像辅助分析[4]、图像搜索[5]、增强现实等。近年来，深度学习在计算机视觉方面取得了令人震惊的进步，从许多计算机视觉问题的解决方法上来看，使用深度神经网络来解决计算机视觉领域如图像分类、语义分割和目标检测等问题已经成为主流，而基于卷积神经网络(CNN)的各种方法，事实上远远超过了传统方法的准确率和效率。

图像的语义分割是在像素级别上的分类，属于同一类的像素都要被归为一类，因此语义分割是从像素级别来理解图像的。遥感图像的语义分割需要将分类图像中所有的像素都进行分类，这种分类会基于一定的语义准则，最终将遥感图像分割为若干的对象区域，如低植被区、高植被区、水体、建筑、道路、车辆等。遥感图像语义分割是遥感图像解译的非常重要的组成之一,而精细化的语义分割可以帮助人们更加准确的获取更多有价值的地理信息，从而在各个相关领域如土地利用、城市建设、空间分析、自然灾害监测和农业病虫害监测等很多情况下起到非常重要的作用。

因此，本文通过对遥感图像语义分割的探索研究，尝试高效且精度更高的遥感图像语义分割模型。

1.2 遥感图像语义分割研究现状

传统的图像分割方法主要利用灰度、颜色、纹理和形状等特征将图像进行区域划分，让区域间的差异性显现出来，而使区域内呈现某种相似性。由于图像的语义分割最早应用于医学领域，而医学影像属于背景和目标区分明显的影像，所以当时传统的语义分割方法大都是基于阈值的方法[6]。这种方法通过对不同灰度阈值的设定，对图像的灰度直方图进行处理，即认为在某一灰度范围内的像素属于同一类目标或具有相似性。之后随着在更多分割场景的应用，出现了许多方法。如基于边缘的分割方法[7]，同样利用像素点与相邻像素点的灰度值差异，提取出边界，而将图像划分为不同的区域。基于区域的图像分割方法[8]，根据图像的空间信息进行分割，利用像素的相似性质形成不同的图像区域。还有基于图论的分割方法，随机森林法[9]，马尔可夫随机场[10]，而这类传统方法都仅仅利用图像的表层特征信息来进行分割，并且只能表明像素之间的差异，无法为图像赋予语义信息，即指明此像素所属何种类别。

近年来，深度学习在计算机视觉领域取得了巨大的突破与进步，利用卷积神经网络参数共享和稀疏连接的特点高效率、高精度地实现了图像分割任务。2012 年，Alex 等人[11]提出了 AlexNet 网络，此网络在 ImageNet 图像识别竞赛的图像分类任务中以年度最佳成绩胜出。VGGNet 系列网络[12]在 2014 年被 Oxford 的 Visual Geometry Group 提出，它证明了在一定程度上，将网络的深度增加是可以做到影响网络的性能。并且由于其规整的网络结构以及简洁和实用性，被作为许多著名网络的基础。2014 年，GoogLeNet 由 Google 研究院提出[13]，它是一种基于 Inception 模块提出的深度神经网络模型，在当年的 ImageNet 竞赛中夺得了冠军，在随后的两年中一直在改进，形成了 Inception V2、V3、V4 等版本。2014 年，Girshick 等提出 R-CNN （Region-based Convolutional Neural Networks)模型[14]，目标检测和语义分割都可使用该模型。该模型首先从图像上采集足够数量的候选区域，特征提取部分利用 CNN 来完成，最终利用支持向量机 SVM 实现各区域的分类，从而实现语义分割。2015 年，Long[15]等人提出的全卷积神经网络（FCN）在 Pascal VOC 视觉识别比赛的图像分割任务中取得了傲人成绩，FCN 实现了端到端的像素级分类，这也标志着图像语义分割的研究迈入了下一个阶段。FCN 由于其特性，在此后众多研究中也被作为改进基础。2016 年，Badrinarayanan 等提出的 SegNet[16]，也是一种全卷积神经网络，并且它还具有对称的编码-解码器的结构，也实现了端到端（end-to-end）的像素级图像分割。Olaf Ronneberger 等人针对医学影像的分割提出了 U-Net 网络结构[17]，其采用对称的且可视化形状呈 U 型的编码器-解码器网络结构，并且集成了高分辨率浅层特征以及低分辨率深层特征，在样本数较少的情况下较好地帮助模型提高分割精度。Deeplab[18]系列网络模型利用空洞卷积（或称扩张卷积）扩大感受野，采用 ASPP（atrous spatial pyramid pooling）对给定输入以不同采样率的空洞卷积进行采样，以多比例捕捉图像上下文从而解决多尺度问题，利用条件随机场（CRF）增强了模型捕捉细节的能力。

遥感影像的解译即信息提取是遥感领域的重要任务，而遥感影像的语义分割是一种很好的解决方案，于是近年来遥感影像的语义分割也成为了遥感领域的研究热点之一。而在早期，遥感影像的自动识别分类分割方法主要采用决策理论或统计方法[1]。这些方法会对象中提取特征，这些特征其实是反映模式属性的一组测量值，它们被定义在一个特征空间中，进而利用决策原理对特

征空间进行划分。遥感影像的特征主要表现为光谱特征和纹理特征，利用这两类特征的传统方法，比如说有经典的机器学习方法，svm 支持向量机进行遥感图像地物分类，或采用非监督学习聚类等方法。

由于传统遥感方法的语义分割需要根据不同目标对象的特征，人工设计相应的特征提取器。人工设计的特征提取器对专业知识要求很高，不能适应复杂的应用场景，泛化能力有限。面对遥感影像的大数据量，情况更为糟糕，因为影像环境变化很大，图像在短时间内变化很大。传统的图像分割方法难以胜任复杂的含有大量语义信息的任务，随着深度学习技术及其在计算机视觉领域应用的飞速进步，以卷积神经网络为基础的深度学习方法被广泛应用于遥感影像语义分割任务中。深度学习理论目前完全可以作为一种从训练集中自动学习有效特征的替代方法，这使得人们可以利用非常大的原始图像数据集中进行无监督的特征学习。深度学习可以对多层次结构的信息进行表达与组织，这使数据之间的复杂关系得以展现。事实上，深度学习技术可以从图像中映射出不同层次的抽象，并将它们从底层特征到高层特征进行融合[19]，因此利用基于深度学习的语义分割方法，我们可以有效地克服传统遥感方法的局限性。事实上，深度学习已经被证明是一个新的和令人兴奋的工具，它非常可能是遥感图像处理发展的下一个趋势。

深度神经网络学习不同目标的特征信息，从而实现像素级的图像分类，该方法具有较强的泛化能力，但在遥感领域应用深度学习也存在一定的困难。首先，计算机视觉领域中的图像一般都是 RGB 三通道图像。然而，遥感图像是由多波段数据组成的。还有一些遥感数据，如归一化植被指数（NDVI）和数字地表模型（DSM），这些数据不是由光学传感器获得的，具有不同于普通光学图像的特征。因此，特征的有效融合仍然是一个开放的研究方向。在卷积神经网络中，卷积层用于提取特征，池层用于聚合特征。网络越深，提取的信息就越抽象。然而，在池化层会有大量的空间信息丢失。网络的浅层部分不能充分提取抽象信息，但空间信息保持完整。语义分割必须既能提取抽象信息，又能保留更准确的位置信息，才能实现正确的像素级图像分类。低层次空间特征与高层次抽象特征的有效融合是一个有待进一步优化的问题。2017 年，Fu 等人[20]提出了 RA-CNN(Recurrent Attention Convolutional Neural Network)网络，这是一种循环注意力卷积神经网络，可以用于微观尺度或者说细粒度的图像分类。它思想就是不断地对图像进行聚焦，得到图像中有助于分类的细微区域的特征，再利用它们进行更细化的聚焦来分类。Li[21]等人于 2018 年提出了一种新的卷积神经网络 DeepUNet。与 U-Net 一样，它的结构也有一条收缩路径和一条扩展路径来获得高分辨率的输出。但 DeepUNet 在收缩路径中使用下行块（downblock）而不是卷积层，在扩展路径中使用上行块（UpBlock），这两个新的模块使用了两个新的连接，即 U 连接和 Plus 连接。最近，Yang[22]等人提出了一种提出了一种新的卷积神经网络结构，即注意力融合网络（AFNet），超高分辨率遥感影像多源数据融合可以增加网络的可学习信息，融合高层抽象特征和底层空间特征，可以高目标边界的分类精度。因此它利用一种多路径编码结构来提取多路径输入的特征，一个多路径注意融合块模块来融合多路径特征，一个精细注意融合块模块来融合高层抽象特征和底层空间特征。

注意力机制最早是应用于自然语言处理（Natural Language Processing）的领域，它可以指出两种语言代表的序列，每个单词之间对应的关系，或者说是这个单词和哪个单词有着密切的联系还是联系不大。2017 年，Google 提出经典网络结构 Transformer[23]，它打破了传统，即 encoder-decoder 框架一般都与卷积神经网络或循环神经网络进行结合的传统它是全部采用 Attention 注意力结构的方法来构建 Transformer 网络。这种创新性的方法在两项机器翻译任务

中体现出了它的强大，任务的结果很好。Transformer 在 NLP 上取得的成就具有划时代的意义，因为它后来被应用于计算机视觉领域，甚至被一些学者称为统一融合两个领域的方法。2020 年，Alexey Dosovitskiy[24]等人提出了 Vision Transformer（ViT），它尝试将 Transformer 应用于图像，模拟 Transformer 中对语言序列的处理方式，将图像分割为小块以序列形式作为 ViT 的输入进行处理。2021 年 3 月，Liu[25]等人提出了 Swin-Transformer，这是一种利用滑动窗口的层级 Vision Transformer，它相当于一种基于图像二维空间上的注意力机制，利用层级结构将计算复杂度从$O(n^2)$减低到了$O(n)$，并用基于多头注意力机制的滑动窗口帮助建立局部区域像素内的相关关系。Li[26]等人在 2020 年提出了一种多阶段注意力残差 U-Net 网络（MAResU-Net），由于点积注意力机制的记忆和计算成本随着输入的时空大小而二次增加，阻碍了注意力机制在具有大规模输入的应用场景中的使用。因此它利用一种线性注意机制（LAM），与点积注意近似等价，增加计算效率，使得注意力机制和深层网络的结合更加灵活多样。在此基础上，设计出了 MAResU-Net。

基于以上遥感图像语义分割的现状，本文希望能探索一点遥感领域语义分割和将注意力机制融合进来的网络结构。受限于本人的研究水平，经过和指导老师的讨论后，决定从编码器-解码器结构的 U-Net 入手，将 ResNet 作为 U-Net 网络的 encoder，以期利用残差网络的特性获得更好的结果。同时尝试将最近在计算机视觉领域比较火热的研究方向注意力机制融入网络，而考虑到自身能力和时间限制，选用了较为容易实现的 SE-Network 通道注意力机制嵌入网络，进行实验。

1.3 本文相关工作

（1）数据预处理：研究对原始影像及其标签数据的标准化、数据裁剪和数据扩充等处理，从而保证数据在训练过程中的有效性。

（2） 基于 ResNet-18/50 的 U-net 网络架构搭建：针对高分遥感数据集较小且深度学习网络训练繁琐等问题，使用 python 语言搭配 TensorFlow 和 Keras 深度学习框架搭建基于 ResNet-18/50 的 U-net 网络架构。

（3） 加入 BatchNormalization 和 Dropout 来防止过拟合：通过在 encoder 的 ResNet 部分加入 Dropout 和 decoder 部分加入 Batch Normalization 和 dropout 减轻过拟合现象。

（4） 网络中实现通道注意力机制：利用 SE-Network 通道注意力机制提升网络性能，将 SE-Network 嵌入 ResNet 网络，使得特征提取部分具备注意力特性。

第二章 遥感图像语义分割基本相关知识及理论

2.1 深度学习神经网络基本理论

神经网络的雏形最早在上世纪就已出现。上世纪 40 年代 Warren McCulloch 和 Walter Pitts 两人作为心理学和逻辑学方面的专家，受到人类大脑中神经元结构的启发，提出了人工神经网络的概念，及人工神经元的数学模型，从这时起神经网络就开始萌芽。在此之后，在对人工神经网络的研究中，模拟人类感知系统的机器“感知机”被提出。

多个神经元可以形成神经网络，而多层堆叠则可以形成深度神经网络（DNN）。深度神经网络一般由三大层构成，第一层为输入层，最后一层为输出层，而中间的多层统称为中间层或隐藏层。

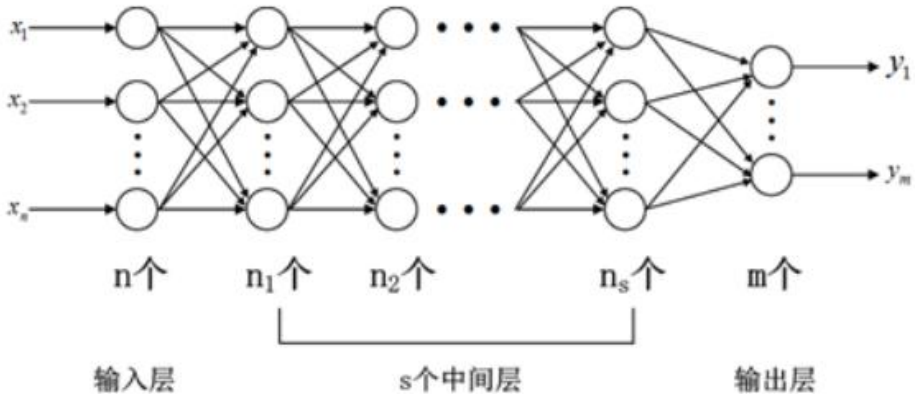


图 2-1 深度神经网络示意图

层与层之间都是每个神经元互相完全接的，也就是说，某一层的任意一个神经元一定与它下一层的所有神经元都相连。虽然 DNN 看起来很复杂，但是如果把一个小的局部模型单独拎出来看，其实还是和感知机一样，即一个线性关系，再加上一个激活函数得到输出。

（1）前向传播

为了得到最终的输出，我们需要通过多层的神经网络，将输入进行处理，一步步地传入输出层，最后得到输出。我们需要将每一层的输出作为下一层的输入，因此，若第$l-1$层有 m 个神经元，则对于第l层来说的第 j 个神经元的输出a_j^l有：

$$a_j^l = \sigma(\sum_{k=1}^{m} w_{jk}^l a_k^{l-1} + b_j^l) \tag{2-1}$$

为了方便表示与计算，我们需要用矩阵来表示第l层的输出。假设第l-1 层共有 m 个神经元，而第l层共有 n 个神经元，则第l层的线性系数 w 组成了一个 n×m 的矩阵 W^l。第l层的偏差 b 组成了一个$n\times1$的向量 ，第l-1 层的输出a组成了一个$m\times1$的向量，l层的激活函数前的线性输出 z 组成了一个$n\times1$的向量 ，第l层的输出a组成了一个$n\times1$的向量。因此第l层输出为：

$$a^l=\sigma(z^l)=\sigma(W^la^{l-1}+b^l) \tag{2-2}$$

（2）反向传播

我们最终的目的是为了利用这样一个神经网络模型得到我们想要的输出，因此我们需要它去“学习”使自己的参数适应这个任务而得到最终结果。为了使模型更好地学习，我们需要利用损失函数来帮助我们训练。损失函数即表明了我们的输出和我们想要的结果或者是样本之间的差异性，损失函数越大代表差异越大，反之亦然。

各种损失函数的介绍放在后面，这里用$J(w,b)$表示。利用梯度下降法或其他优化算法来更新我们的参数w和b。

$$w=w-\alpha\frac{\partial J(w,b)}{\partial w} \tag{2-3}$$

$$b=b-\alpha\frac{\partial J(w,b)}{\partial b} \tag{2-4}$$

式 2-4 中w为权重，b为偏差，α为学习率，通过多次迭代，即前向传播再反向传播，更新权重与偏差，直至损失函数优化为全局最优，即神经网络模型的输出与我们所期望的一致。

2.2 卷积神经网络及经典模型

2.2.1 卷积神经网络

卷积神经网络（Convolutional Neural Network，CNN）与普通的深度神经网络不太一样，一般用于计算机视觉处理的任务中。由于一张图像包含许多的像素，且每张图片一般有三个通道，如果依旧采用普通深度神经网络的结构的话（进行全连接），会导致计算量指数级增长，并且当图片数量增多时由于参数过多，网络会难以训练并容易过拟合。因此利用卷积核构成的卷积神经网络大幅降低了参数和计算量，并可以达到更好的效果。

卷积神经网络有两个最大的特点：

(1) 参数共享（权值共享）：由于特征检测器（或称卷积核、滤波器）在训练时会逐步平移与图片中某一片区域进行卷积，并最终覆盖整张图片（一般情况下），因此其可以保证在图像不同的区域中使用同样的参数。这样一来先当与减少了求解所需的参数，不同的各种卷积核可以去检测图像中不同的特征，还可以保证检测整张图像中不同位置的同一类型特征。

(2) 局部感受野（稀疏连接）：图像的像素一般都是与周围的像素有一定的关系，因此每个卷积核秩序对局部的特征进行卷积，检测局部特征。同时在每层中，每个输出值都只依赖与一定较小数量的输入。这样同样可以显著减少网络中需要计算的参数从而提升效率。

基本卷积神经网络的结构一般为：

输入层->[[卷积层->激活函数]*M->池化层]*N->[全连接层->激活函数]*K->全连接层->输出层

我们来一一介绍卷积神经网络中的各层：

（1）输入层：与深度卷积神经网络的输入层功能一致，将数据输入进网络，一般以图像的高、宽、通道数为维度输入即 H*W*C。

（2） 卷积层：这是卷积神经网络中最重要的结构，它利用卷积核的窗口滑动对全图进行卷积操作。CNN 中的卷积操作和数学中严格意义上的卷积有些不同，二维卷积中在局部范围内，实际上就是对卷积核与图片像素点相对应的位置进行相乘再将所有乘积求和。

在卷积神经网络中由于输入的图片通道数不止一个，因此卷积核也不止一个，实际上是三维卷积，二维卷积推广到三维空间也就是将原先二维卷积输出值按通道（第三维）相加。下图为 CNN 中卷积示意图：

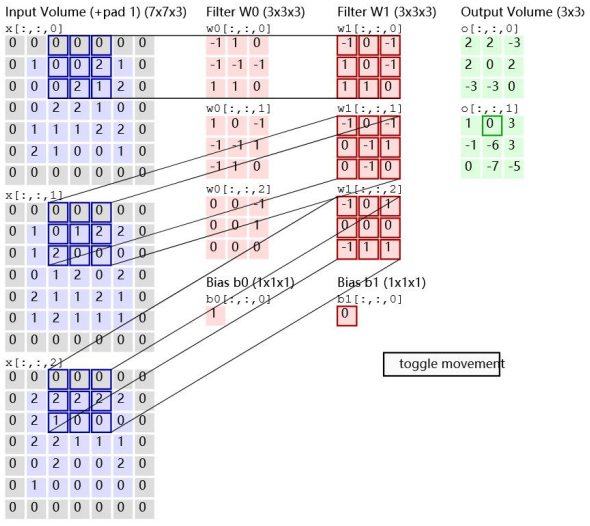


图 2-2 三维卷积示意图

此处为了便于理解，还需要介绍卷积中的 padding 即填充操作，填充操作实际上就是在图像边缘补 0。一般情况下如果不进行填充操作，图像分辨率会由于卷积的特性而逐渐降低，而且对于图像边缘的像素来说，图像内部的像素被卷积处理的次数会多得多。因此可以通过 padding 填充操作来解决上述问题。除此之外，卷积神经网络中卷积核的窗口滑动距离并不固定，此滑动距离被称之为 stride 步长，图 2-4 中的卷积步长就为 2。

因此卷积的输出维度实际上与输入图片尺寸、卷积核的大小、填充大小以及步长相关。具体公式为：

$$D=\frac{n+2p-f}{s}+1 \qquad (2\text{-}5)$$

最后结果 D 需要向下取整。式中 D 代表卷积输出的维度，n 为输入图片尺寸，p 为填充大小，f 为卷积核大小，s 为步长大小。

（3）池化层：又称之为下采样层，池化层是对卷积层提取的特征图进行局部取均值（平均池化）或去最大值（最大池化）等来得到新的特征图。一般来讲，池化的最大作用就是压缩图像，减少参数和计算量。除此之外，池化层还有两个特点，其一是特征不变性，也就是说，池化过后的图像依旧能保留原图像之前的特征，这包括平移、旋转和尺度不变性。二是特征降维，由于一幅图像含有大量信息，然而其中许多信息是重复或作用较小，通过池化我们可以去除这些冗余信息而保留主要信息。池化层的输入输出维度变化与卷积层公式一致，即式 2-5，下图即用 2*2 卷积核进行步长为 2 的最大池化操作示意图：

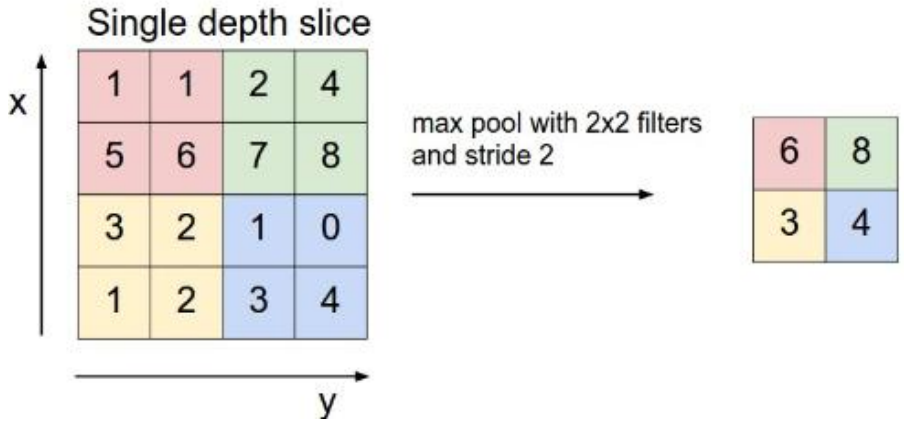


图 2-3 池化层示意图

（4）激活函数：激活函数同样是卷积神经网络中的重要一环，但其实它在卷积神经网络中起到的作用和普通深度神经网络的作用大致一样。激活函数是将前面经过卷积的输出的线性模型

转换为非线性模型。此处非线性的变化非常重要，因为假设一个神经网络只进行线性计算的话，根据线性代数相关知识，会导致最终结果同样为线性。这样我们的模型就无法完成非线性的任务，因此激活函数可以很好地帮助网络提升性能和表达。

接下来介绍两个常用的激活函数：

1）Sigmoid 函数，在机器学习早期使用，公式及函数图像如下：

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{2-6}$$

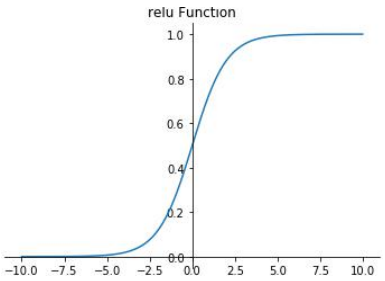


图 2-4 sigmoid 函数

可以看出，Sigmoid 函数将输入值映射到[0,1]区间内，最大值为 1 而最小值为 0。Sigmoid 函数有两个缺点，一个是会比较容易发生梯度消失的问题，因为 Sigmoid 在输入值较大时，导数趋近于 0，也就是梯度趋近 0，网络训练过慢或无法收敛。另外，Sigmoid 在进行反向传播时，因为求导涉及除法，会导致计算量增大，因而降低训练速度。鉴于上述问题，现在已经较少使用 sigmoid 作为一般卷积神经网络的激活函数了。

2）ReLU 函数（Rectified Linear Unit，线性修正单元），公式及函数图像如下：

$$f(x)=\begin{cases}x & (x>0)\\ 0 & (x\le 0)\end{cases} \tag{2-8}$$

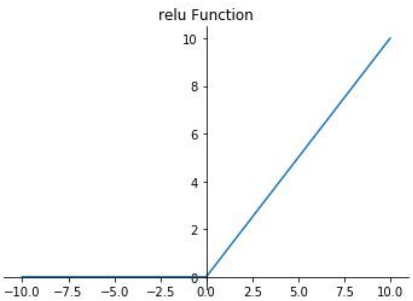


图 2-5 ReLU 函数示意图

ReLU 梯度为 1，只有一边又饱和的问题，因此其梯度能在深度网络中传递。Relu 会使坐标轴

左侧即输入值为负值的神经元的输出为 0，这样就相当于对网络进行稀疏，减少了参数的相互依赖关系，成功地缓解了过拟合问题的严重性。

（5）全连接层：全连接层在卷积神经网络中的作用同样类似于普通神经网络，卷积层，池化层等是将原始数据映射到隐层特征空间，而全连接层相当于将网络前面学习到的特征表示映射到样本标记空间。所以全连接层可以看作是对输出层的铺垫，方便输出。

（6）输出层：全连接层之后一般接输出层，输出层和全连接层之间也进行完全连接，而输出层的作用就是将网络最后的输出分类，常常采用多分类器 Softmax。

2.2.2 AlexNet

2012 年由 Alex Krizhevsky、Geoffrey Hinton 等人提出 Alex Net 一举摘得了当年 ILSVRC(ImageNet 大规模视觉识别挑战赛)的桂冠。它 top5 错误率低至 15.315%，与之形成鲜明对比的是使用传统计算机视觉的第二名小组的 top5 错误率为 26.172%。这在当时震惊了整个计算机视觉界，而卷积神经网络也从那时起崭露头角。Alex Net 的网络架构如下：

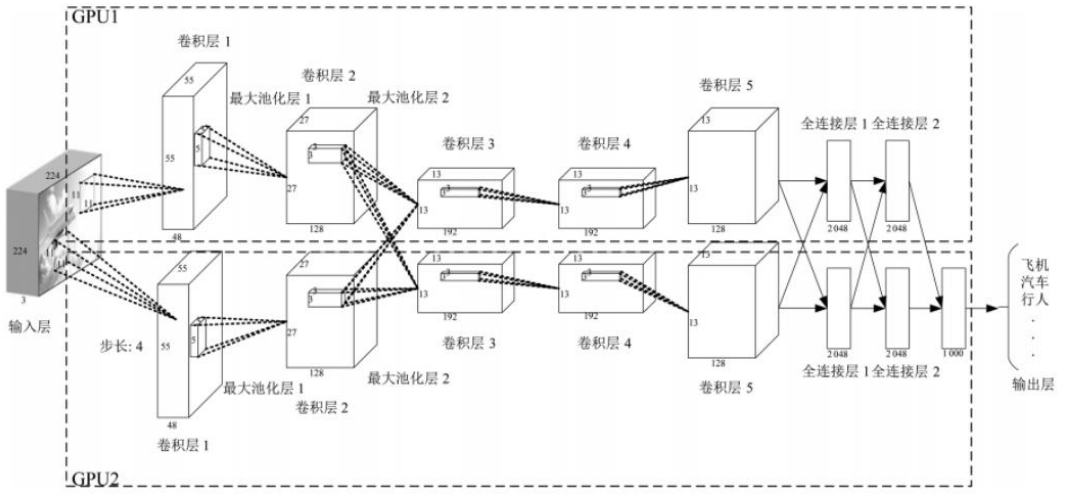


图 2-6 Alex Net 网络结构示意图

由于 Alex Net 采用两个 GPU（计算量太大只能分开），因此会看到两条流程图路线，实际上基本上下对称。输入层输入图像大小为 227×227×3。卷积层 1：卷积核大小为 11×11,共 48 个卷积核，步长为 4，卷积后 ReLU 激活，然后用步长为 2 的 3×3 最大池化得到 27×27×96featuremap。卷积层 2 中使用 128 个 5×5 卷积核，输出两组 13×13×128 feature map；卷积层 3：用 192 个 3×3 卷积核，输出为 13×13 ×384；卷积层 4 处理过程与层 3 相同，输出为 13×13 ×384；卷积层 5：卷积核大小为 3×3，128 个，池化后输出为 6×6×256；全连接层 6：每边 2048 个神经元，输出 4096×1 向量即 4096 个神经元，Dropout 层随机不激活某些神经元；全连接层 7，过程与全连接层 6 一样；输出层即最后一个全连接层，1000×1 的向量来输出预测结果。

总的来说，其学习过程为边缘区域到局部区域的特征，最后接近全连接层分类器的部分可以学习到整体的形状特征，再到整体的全局特征。AlexNet 在训练技巧方面也作出了一些贡献：（1）

采用两个 GPU 增加算力，但这某种程度上也是缺点，因为卷积神经网络的大参数量导致必须这样做。（2）利用 dropout 方法来减少过拟合现象，原理本文后面会介绍。（3）还采用了最大池化的方式，避免了平均池化的缺点即模糊化。（4）采用了 ReLU 激活函数，此函数的优点已经在之前介绍过了。它还引入了 LRN（Local Response Normalization）局部响应归一化处理，利用临近的数据做归一化，实验表明该方法贡献了 1.2%的准确率。

2.2.3 VGGNet

VGGNet 是牛津大学几何学视觉组（Visual Geometry Group）提出的一种深度卷积神经网络。它在 2014 年 ILSVRC 竞赛定位项目种夺得第一名，而在分类竞赛中也有第二名的好成绩，分类竞赛的第一名是 GoogLeNet，这个我们随后会介绍。虽然没有包揽第一名，但是 VGG 还非常适合迁移学习，而且其规整的网络结构使得其之后被作为许多其他网络的基础。VGGNet 网络结构如下，其中 D、E 即为 VGG-16 和 VGG-19。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

图 2-7 VGGNet 网络结构示意图

VGGNet 为卷积神经网络的发展作出了许多贡献。首先它成功探索了卷积神经网络的性能随深度增加的可能，其次它使用了尺寸更小的卷积核。

VGG 最大的特点就在于卷积核的使用，它采用连续的 3×3 的卷积核来替代 AlexNet 中较大的如 5×5 的卷积核。实际上两个 3×3 卷积核的感受野相当于一个 5×5 的卷积核，而三个则相当于 7×7 的卷积核。为什么可以这样来替代呢，5×5 卷积可以堪称一个在 5×5 区域内滑动的小型全连接网络，因此我们可以先用 3×3 的卷积核进行卷积，再用一个全连接层将这个 3×3 卷积输出，当这个全连接层也被看作 3×3 卷积时，就等于用两个叠加的 3×3 卷积核代替 5×5 卷积核。用三个 3×3 的卷积核代替 7×7 的卷积核也是一样的道理。

这样做的好处有几个：首先包含更多的 ReLU 激活函数，增加了决策函数的非线性，使得模型拟合性更好。其次还大大减少了参数量，我们假设输入通道数为 C，输出通道数为 M，则用 3×3 的 2 个卷积核需要 9×C×M 个参数，5×5 的卷积核则为 25×C×M 个参数，相当于减少了三分之一左右的参数量。而且，VGG 使用的 1×1 的卷积层，实际上可以增加函数的非线性，同时做到不影响卷积层的感受野。但是其缺点就在于三个全连接层导致其总参数过多，需要更长的训练时间。

2.2.4 GoogLeNet

GoogLeNet 是 google 在 2014 年提出的基于 Inception 模块的深度卷积神经网络模型，其在当年的 ImageNet 分类竞赛中夺得了冠军。之后又提出了许多改进的 v2、v3 等版本，为方便理解，此处介绍最初版本。

为什么要提出 GoogLeNet 呢，一般来说，增加网络的深度是提升网络性能最直接的办法，但是着同时意味着巨大的计算量。同时巨大的参数量也会容易产生过拟合。如果网络结构能被不失性能地稀疏，就可以在保持网络结构稀疏性地同时利用密集矩阵的高计算性。

Inception 模块就是构成 GoogLeNet 的基本模块，模块中并列地含有多个不同的卷积层和池化层，然后通过连接每个卷积或池化操作的输出来得到最终的输出。从一般程度上来说，卷积核越大所能笼罩的范围越大，感受野也就越大，所以不同大小的卷积核意味着它们所的感受野也不同，最后的特征图拼接也就意味着不同尺度特征的融合。采用 1×1、3×3 和 5×5 的卷积核同时也是为了方便使输出特征图尺寸一致。Inception 模块使我们不需要去决定在网络的哪一层中使用哪一种卷积核或池化层，这将由网络在训练时自行学习确定各卷积核和池化核的操作。这是 Inception 的基本思想，但是实际中，3×3 和 5×5 的卷积核依旧会带来巨大的计算量，所以 google 的研究人员采用了 1×1 卷积核来降维，以减少计算量。

为了说明此处 1×1 卷积的作用，我们假定输入层维度为 28×28×192，在简单 Inception 模块中，采用 32 个 5×5×192 的卷积核时，得到输出为 28×28×32，所以总共需要作（5×5×192）×（28×28×32）次乘法运算，也就是计算每个输出值所需的乘法次数乘以输出值的个数，大概为 1.2 亿次乘法运算。而当采用 1×1 卷积核来降维时，从输入层经过 16 个 1×1×192 卷积构成的卷积层时所需运算的乘法次数为（1×1×192）×（28×28×16），大约为 240 万次。然后再经过上述相同的 5×5 卷积层，乘法运算次数为（5×5×16）×（28×28×32），大约为 1000 万次。它们加起来为 1240 万次，几乎为 1.2 亿次的十分之一。可以看出，这个方法对计算量的减少是非常有用的。在实际操作中，通过合理设置 bottleneck 瓶颈层（1×1 降维）可以做到减少计算量的同时不影响网络性能。从上述的解释中还可以发现，利用 bottleneck 还可以帮助池化层降维，减少参数量。因此改进 Inception 模块如下所示：

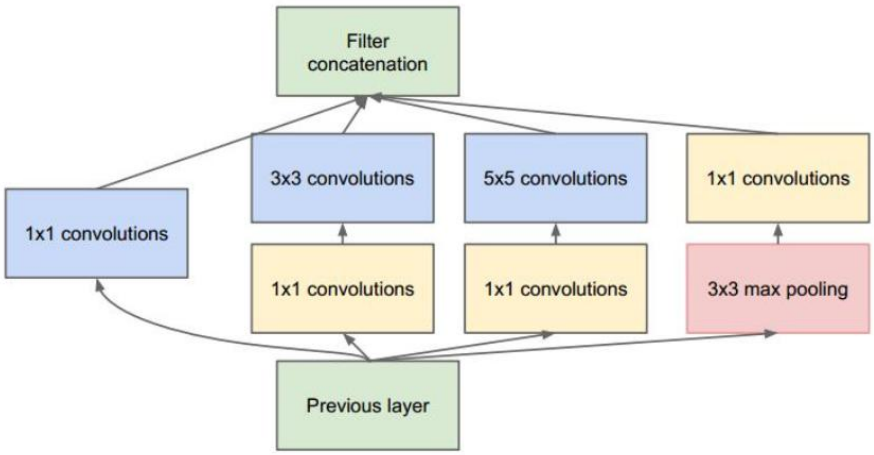


图 2-8 改进 Inception 模块

最后是 GoogLeNet 的整体结构，由众多的 Inception 模块堆叠而成，这方便后人在此基础上改进。同时可以发现，其结构处还有两处分支，在网络结构的最后输出部分，实际上分支和最后输出部分的结构基本一样。通过隐藏层作为 softmax 输出，用于向前传导梯度防止梯度消失，还可以防止过拟合。

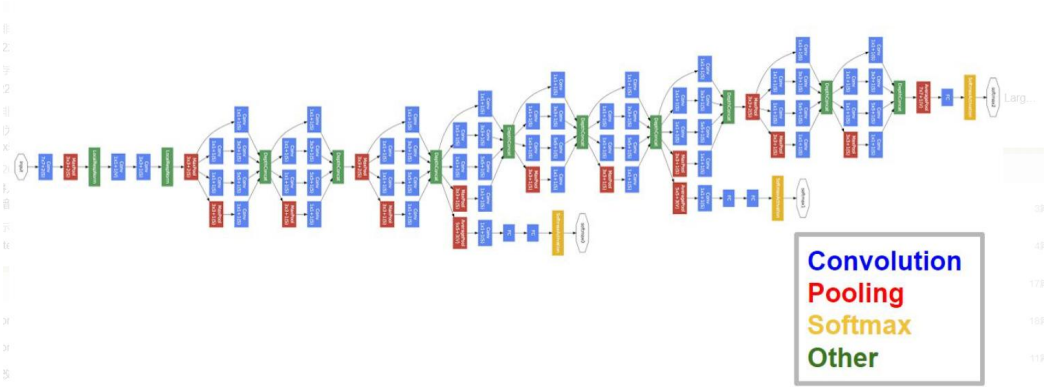


图 2-9 GoogLeNet 整体结构

2.3 深度学习的图像语义分割模型

2.3.1 全卷积神经网络（FCN）

正如我们前面所介绍的，在传统的卷积神经网络中，它会利用全连接层来讲二维的特征图映射为一维的向量，然后对应的就是取最大概率的类别为分类标签输出。这样的卷积神经网络无法完成语义级别的图像分割，而全卷积神经网络 FCN 将卷积神经网络中最后的全连接层换为卷积层，然后通过上采样的操作，将图像恢复到与输入图像相同的尺寸，因而可以对每个像素进行预测，即像素级别的预测。

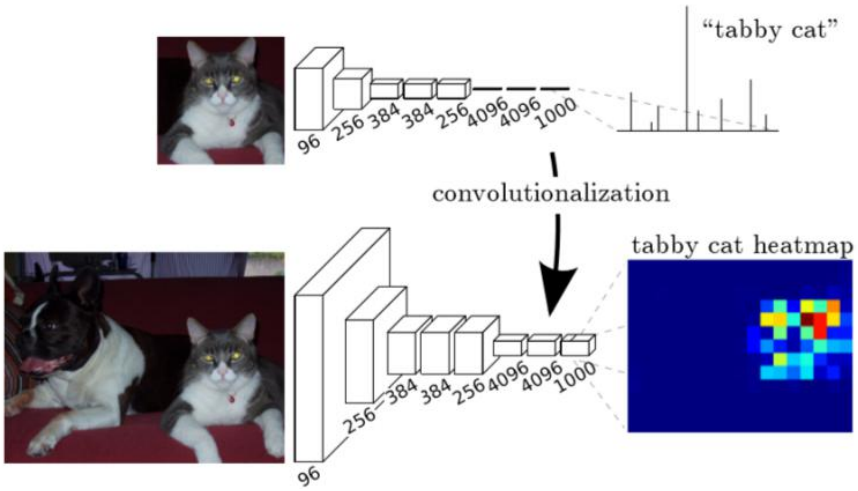


图 2-10 FCN 与普通 CNN 区别示意图

全连接层与卷积层之间完全可以相互转化，它们唯一的不同在于卷积层中的神经元，或者说卷积核，每次只对整个图像也就是输入数据中的一个局部区域进行卷积，而全连接层则相当于对上一层的数据每个神经元都连接。两者可以这样转化，比如，一个输入数据尺寸为 5×5×512，全连接层有 4096 个神经元，则此全连接层可以被看成一个卷积核尺寸为 5×5×4096 的卷积层，这样它的输出为 1×1×4096，也就相当于上述的全连接层了。

在全卷积神经网络的最后经过多次卷积与池化之后，分辨率最低的图像称之为 heatmap 热图。它是最重要的高位特征图，在此之后进行上采样操作，通过逐个像素地求其在所有分类类别中该像素属于某一分类地概率，取最大概率作为其分类，就可以最终产生被分割的图像。

FCN 还有一个结构叫做 skip layers，因为如果对最后一层的特征图进行上采样为原图大小，由于最后一层其尺寸国小，我们可能会损失很多细节。因此，利用这个跳跃结构，将富有全局信息的最后一层的预测与更浅层而富有局部细节的预测结合起来。

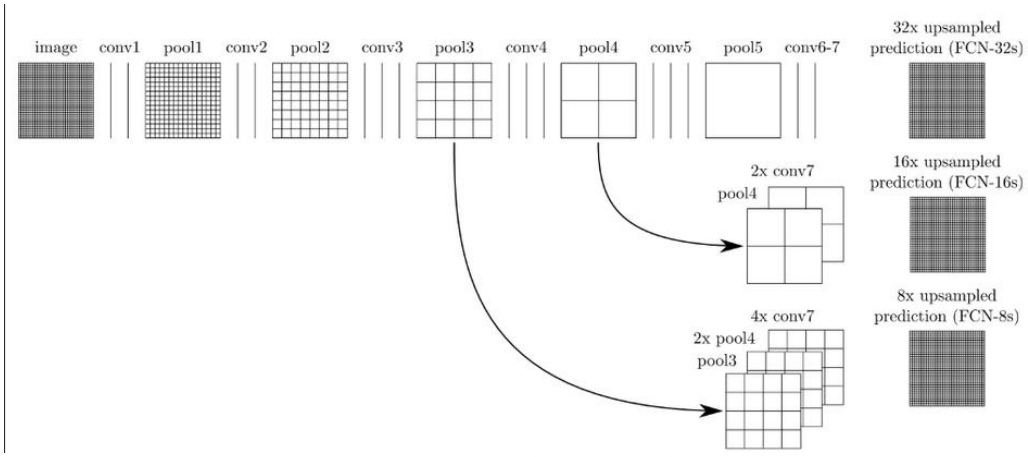


图 2-11 特征融合示意

如上图所示，我们将底层的预测进行 2 倍上采样之后得到的图像与 pool4 融合（相加），再进行 16 倍上采样后的图像为 FCN-16s。以此类推，4 倍的 conv7 核 2 倍上采样的 pool4 和 pool3 相融合，经过 8 倍上采样的图像成为 FCN-8s。下图可以看出这样的 skip 结构确实有利于特征图

的预测。

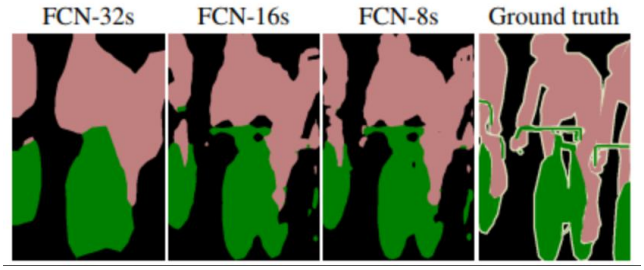


图 2-12 各种倍数上采样融合结果图

2.3.2 U-Net 网络结构介绍

U-Net 也是一种常用于图像语义分割的网络，它是由 Olaf Ronneberger 等人针对医学影像的分割而提出。同时它也是一种编码器-解码器结构，并且整体结构很像字母 U，因此被命名为 U-Net。

该网络左侧即为编码过程，论文中称为压缩路径（contracting path），右侧为解码过程为扩张路径（expansive path）。网络结构图中，每个蓝色框表示一个多通道特征图 feature map，而其通道数被表在顶部，左下角的数字即特征图的尺寸，白色框表示经过复制和裁剪后的特征图，不同箭头表示不同的操作，图中已经进行说明。

编码过程或收缩路径用于获取上下文信息（context），而解码扩张路径用于精确定位（localization）实际上就是表明不同位置的像素属于哪一类。

编码：输入图像维度为 572×572×3 的图像，在论文中医学图像为灰度值图。先用 64 个 3×3 卷积核进行两次卷积，每次都为 valid 填充，即不进行填充，所以得到 568×568×64 的 feature map 尺寸缩小了一点，整个网络中的激活函数都采用 ReLU 激活函数。再进行 2×2 的最大池化操作，得到 248×248×64feature map。重复该操作三次，即(两次 3×3 卷积+ReLU+2×2 最大池化)×3。由于采用 2×2 的池化核并以步长为 2 进行最大池化，因此每一次最大池化特征图尺寸缩小为原来的一半，3×3 卷积核数量为原来双倍。第 3 次也是最后一次最大池化后，输出 32×32×512 特征图，再用 1024 个 3×3 卷积核重复两次卷积，得到最底层的特征图大小为 28×28×1024。

解码与编码对称，只是将最大池化（下采样操作）改变为上卷积（上采样）过程如下：上面编码得到最底层的特征图 feature map 为 28×28×1024，然后先进行 2×2 上卷积，尺寸变为原来的两倍，通道为原来二分之一得到 56×56×512 的 feature map，然后将此时的特征图和编码中对应位置的最大池化层之前的卷积得到的特征图进行连接，但在此之前要先对特征图进行剪裁以保证尺寸一致，最后得到 56×56×1024 特征图，再用 512 个 3×3 卷积核进行两次卷积，得 52×52×512 的特征图。重复上述操作，即进行（2×2 反卷积+ReLU+裁剪并拼接 + 两次 3×3 卷积）操作 3 次。与解码过程刚好相反，每一次拼接后 3×3 的卷积核数量缩减至原来一半。最后以此操作完成后，得到 388×388×64 的特征图，最后需要再进行一次 1×1 的卷积，这是为了将 64 个通道的特征图映射到所需分类类别数量上，然后就可以进行输出。注意这里输出图像通道为 2，多分类问题中，输出图像的通道数对应所需分类的类别数。

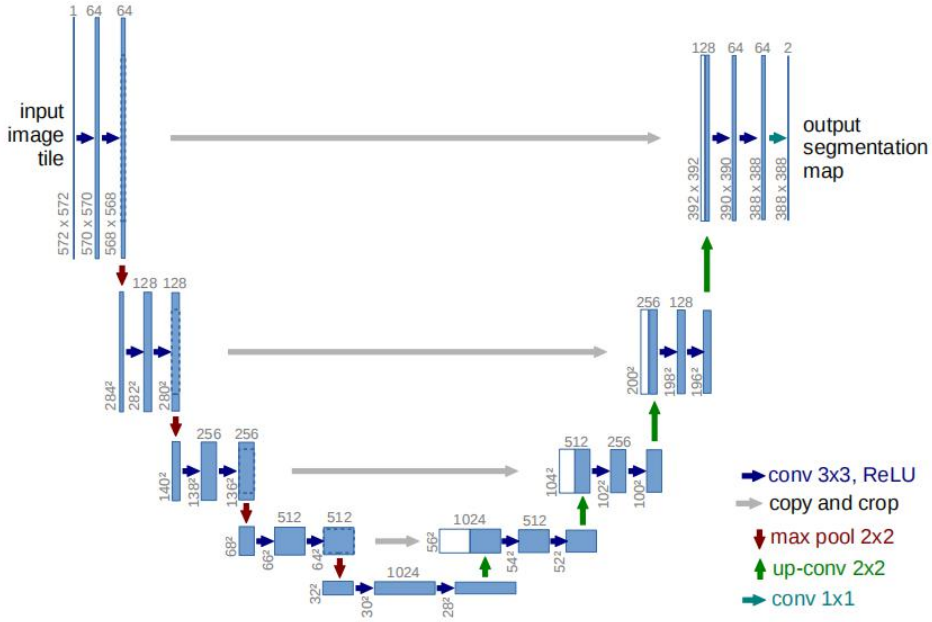


图 2-13 U-Net 网络结构图

U-Net 网络结构清晰，容易理解，最早被用于医学影像的分割中，后来也各种改进版本也被应用于其他任务的图像分割，后来还有 U-Net 生成对抗网络等。本文中采用 U-Net 作为基础结构也是考虑到其结构特点，容易编程实现。

第三章 基于残差网络 ResNet 的 U-Net 网络遥感图像语义分割模型

3.1 ResNet 介绍

2015 年，由来自 Microsoft Research 的 Kaiming He 等人提出的 ResNet 在 ImageNet 图像分类和物体识别竞赛中取得了佳绩。ResNet 又称残差网络，它最大的特点是通过跳跃连接的残差块缓解了深度神经网络中梯度消失的问题。

如我们之前介绍的那样，一般来说增加网络深度是提升网络性能最直接的方法，愈深愈复杂的网络愈有着强大的表达能力。从 AlexNet 到 VGGNet 再到 GoogLeNet 基本都是如此，然而后来人们发现，卷积神经网络似乎并不能无限地增加深度，增加到一定地深度后继续增加地话，网络收敛速度变慢，训练和验证集的误差反而会下降。好在，ResNet 的残差模块应运而生，接下来我会来介绍基本的残差模块以及残差网络为什么有用。

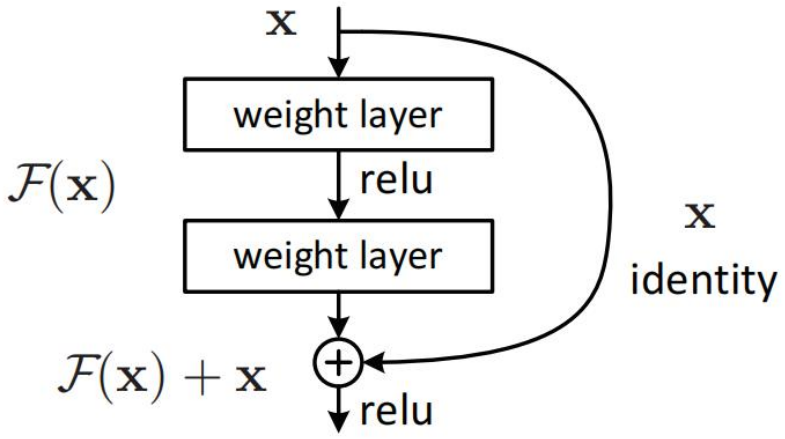


图 3-1 残差网络块模型

残差模块通过 Identity 恒等映射，在输入与输出之间建立了一个跳远连接。我们将输入记为 X，所需的基础映射（普通卷积层）输出记为 H(X)，将叠加的非线性层记为 F(X)，F(X):= H(X)-X，网络要学习的就是输入输出之间的残差 F(X)，而最终的输出被重铸为 F(X)+X。

为了说明残差模块为什么有用，我们以其在神经网络中的公式来举例：

$$a^{l+2} = g(z^{l+2} + a^{l}) = g(w^{l+2}a^{l+1} + b^{l+2} + a^{l}) \qquad (3\text{-}1)$$

式 3-1 中 a^{l} 代表第 l 层的输出，在这里是残差模块的输入。a^{l+1} 即 l +1 层的输出，也就是残差模块第一层的输出。a^{l+2} 即残差模块的输出，那么 z^{l+2} 也就是第 l +2 层的线性输出，w^{l+2} 和 b^{l+2} 代表 l +2 层的权重和偏差。函数 g(x)代表了 ReLU 激活函数。由于 ReLU 函数的输出值是具有非负性的，因此当 w^{l+2} 和 b^{l+2} 为 0 时，我们就将得到下式：

$$g(a^l)=a^l \tag{3-2}$$

这时我们发现经过残差模块后，第l+2 层的输出与第l层相等。也就是说，这两层网络相当于不存在，而这就保证了至少更深层的网络并不会比浅层的网络效果更差，因为它可以做到和浅层网络相同的输出。实验中也证明了残差快学习这个恒等映射函数并不难，而其他的不含有残差模块的普通深度神经网络却很难学习，也就是说无法做到第l+2 层的输出与第l层相等。同时，如果这两层网络确实学习到了一些有助于提升网络性能的特征，那么网络也就不会跳过这两层，这时的w^{l+2}和b^{l+2}则不为 0。

$$\mathbf{y}=\mathcal{F}(\mathbf{x},\{W_i\})+\mathbf{x}. \tag{3-3}$$

论文中，用式 3-3 来表示构建的残差模块，函数 F(x,{Wi})表示需要学习的残差映射，x 和 y 分别代表输入和输出。在输入和输出维度相等时，上式自然成立，但是当它们的维度并不相等时，就需要用 Ws 矩阵来与 x 相乘使得维度匹配。

$$\mathbf{y}=\mathcal{F}(\mathbf{x},\{W_i\})+W_s\mathbf{x}. \tag{3-4}$$

下图是残差网络的两种常用构建模块，左侧的模块一般用在较浅层网络（ResNet-18/34）中。右侧的模块也使用了 1×1 卷积层来帮助压缩特征图的维度，减少参数和计算量。

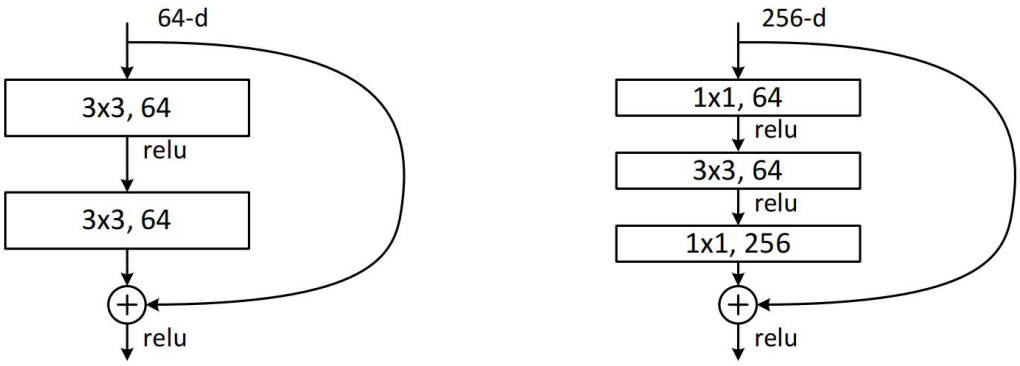


图 3-2 残差网络常用构建块

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

图 3-3 各种残差网络的结构

3.2 基于 Res-Net 的 U-Net 网络结构介绍

基于 ResNet 的 U-net 网络结构如图 3-4 所示，实际上它是基于 U-Net 整体架构进行改进的。它主要是将 U-Net 中的压缩路径（编码器）利用 ResNet 进行了替换。图中左侧即 ResNet-18 的简化架构，只是将最后用于输出的全连接层去掉了，因为这里 ResNet 是充当编码器的作用，并不需要输出。然后，参考 ResNet 的网络结构，可以发现，ResNet 是以网络输出的大小或者是池化层的位置为分界线，可以分为四个卷积阶段，因此我们可以在每一次池化后将输出作为特征图保存下来，为之后扩张路径（解码器）中的特征融合作准备。

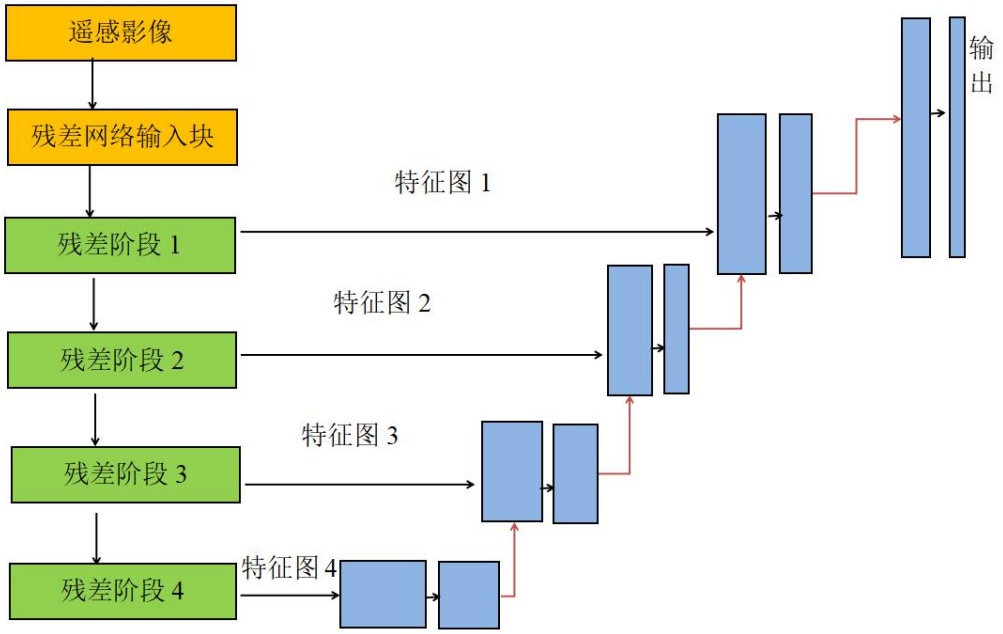


图 3-4 基于 ResNet 的 U-net 网络结构

网络右侧即为U-Net的扩张路径部分，其f1、f2、f3、f4即为从编码器中得到的特征图，向右的黑色箭头代表了连接特征图进行特征融合的操作，蓝色的方块代表了如U-Net一样代表连接后特征图和卷积层，棕红色的箭头代表了上采样的操作。在四个蓝色方块中每进行一次卷积，就接一层BatchNormalization批正则化。图中每次连接后卷积的操作会将通道压缩，然后再进行上采样，前三次都为2×2上采样，即将特征图的高宽放大两倍。最后一次比较特殊，由于ResNet部分实际上要比U-Net的编码器少一张连接特征图，因此最后一步采用了4×4的上采样，最后一次卷积将通道数输出为类别数，维度为H×W×6。

表3-1 ResNet-18-UNet结构

	遥感影像输入		
Conv1	BatchNorm Conv1 7×7,64,stride2		最终输出
Max Pool	max pool 3×3,stride2		Conv 1×1,n_classes
Conv2-x	[3×3,64->3×3,64] ×2	------>f1----->	Up Sample 4×4 Conv 3×3，64
Conv3-x	[3×3,128->3×3,128] ×2	------>f2----->	Up Sample 2×2 Conv 3×3，128
Conv4-x	[3×3,256->3×3,256] ×2	------>f3----->	Up Sample 2×2 Conv 3×3，256
Conv5-x	[3×3,512->3×3,512] ×2	------>f4----->	Up Sample 2×2 Conv 3×3，512

表3-2 ResNet-50-UNet结构

	遥感影像输入		
Conv1	BatchNorm Conv1 7×7,64,stride2		最终输出
Max Pool	max pool 3×3,stride2		Conv 1×1,n_classes
Conv2-x	[1×1，64->3×3,64 -1×1,256] ×3	------>f1----->	Up Sample 4×4 Conv 3×3，64
Conv3-x	[1×1，128->3×3,128 ->1×1,512] ×4	------>f2----->	Up Sample 2×2 Conv 3×3，128
Conv4-x	[1×1，256->3×3,256 ->1×1,1024] ×6	------>f3----->	Up Sample 2×2 Conv 3×3，256
Conv5-x	[1×1，512->3×3,512 ->1×1,2048] ×3	------>f4----->	Up Sample 2×2 Conv 3×3，512

表3-1和3-2呈现了网络结构内部的细节，实际上表格第一、二列和ResNet网络一致，只

是去除了最后的全连接层，改为将特征图直接输入 decoder 部分。同时左侧 ResNet 部分省略了每次卷积层部分的 BatchNorm 层，还省略了卷积阶段完成后池化层的操作，实际上每次池化层后图像尺寸都会缩小为原来的一半，因此输入的特征图才能和 decoder 部分的上采样特征图连接。

3.3 尝试嵌入 SE-Network 通道注意力机制网络

3.3.1 注意力机制在计算机视觉方面的应用

近年来，注意力机制在深度学习领域是个热点话题，它已经被证明在自然语言处理、语音识别和计算机视觉等领域都能起到相当大的帮助。事实上，注意力机制并不是什么深奥难懂的东西，它就是字面意思，就是对某些重要的或含有实用信息的部分集中注意力，而对剩下并不那么重要的部分减少关注甚至忽略。从图像的角度来看，图像中信息的容量要远远超过自然语言所蕴含的信息。因为我们的大脑无法在同一时间内处理如此大量的信息，必须作出取舍，而事实也证明，同一时间内我们视觉所感受到的信息绝大部分是对我们的判断没有帮助的。而且更重要的是，注意力机制除了提升我们处理信息的效率外，还提升了准确性。将注意力机制应用到深度学习中，可以说非常合理，因为深度学习本身也是在模仿人类学习的基础上提出并改进的。

2020 年，Alexey Dosovitskiy 等人提出了 Vision Transformer（ViT），它尝试将 Transformer 应用于图像，模拟 Transformer 中对语言序列的处理方式，将图像分割为小块以序列形式作为 ViT 的输入进行处理。

ViT 模型也是一个用于视觉任务的纯 Transformer 经典技术方案。它将输入图片切分为若干个图像块（patch），然后将每一个 patch 用向量来表示，再用 Tansformer 来处理图像 patch 序列，最终的输出做图像识别。但是 ViT 的缺点也十分明显，它将图像切块输入 Transformer，然后把图像块拉直成向量进行处理，从某种程度来说比较暴力，而且参数量较多。因此，图像块内部结构信息被破坏，还忽略了图像的特有性质。下图是 ViT 的基本架构:

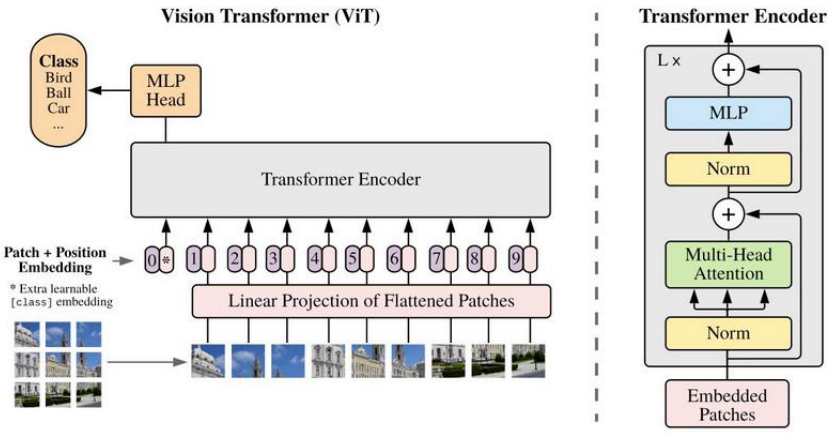


图 3-5 Vision Transformer 基本架构

2021 年 3 月，Liu[23]等人提出了 Swin-Transformer，这是一种利用滑动窗口的层级 Vision Transformer，可以作为其他网络骨干框架的 Transformer 骨干。Transformer 在应用到计算机视

觉领域时目前面临两大挑战，一是视觉目标尺度变化大，在不同的场景下视觉 Transformer（如 ViT）的性能未必好；二是，图像分辨率高，像素点多，Transformer 基于全局自注意力的计算导致计算量比较大。

针对上述问题 Swin-Transformer 架构被提出，它相当于一种基于图像二维空间上的注意力机制，利用层级结构将计算复杂度降低，并用基于多头注意力机制的滑动窗口帮助建立局部区域像素内的相关关系。这种架构包含滑窗操作，具有层级设计，其中滑窗操作包括不重叠的局部窗口 local window，和重叠的交叉窗口 cross-window。在窗口中计算各自的注意力，这样做的好处是既能引入 CNN 卷积操作的局部相关性，另一方面能节省计算量。

下图是 Swin Transformer 降低计算复杂度和滑动窗口的示意图：

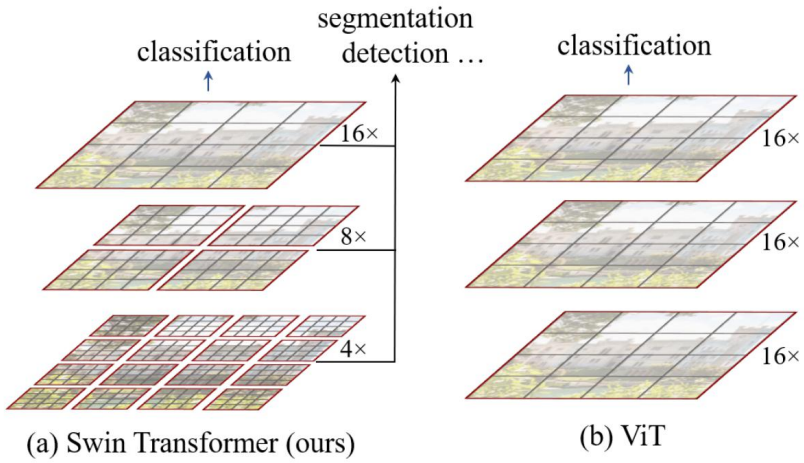


图 3-6 Swin Transformer 层级结构

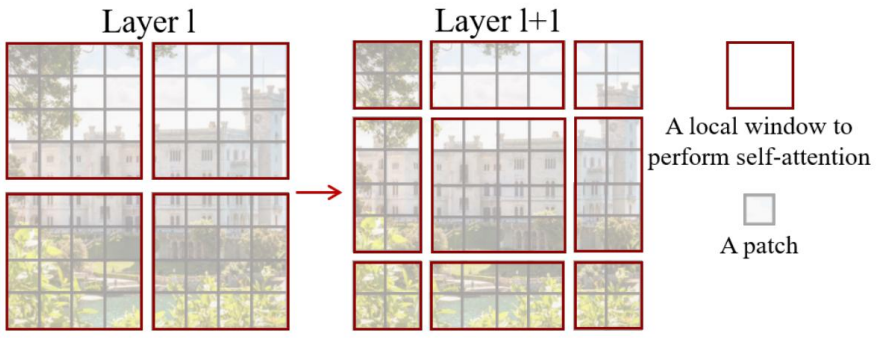


图 3-7 Swin Transformer 滑动窗口

现在从结果上看，Swin-Transformer 具备令人激动的潜力，目前在计算机视觉领域许多任务中排名第一，目标检测达到 58.7 AP，实例分割刷到 51.1 Mask AP，语义分割在 ADE20K 数据集上达到 53.5 mIoU。将 Transformer 应用到计算机视觉领域如 Swin-Transformer 的成功，从某种程度上来说为计算机视觉领域新的探索作出了贡献。

3.3.2 嵌入 SE-Network 注意力机制的优化网络

上一节中，将注意力机制应用到计算机视觉上的两种方法都是基于图像像素的空间位置关系，而 SE-Network 则是较早之前提出的一种基于卷积神经网络中通道嵌入注意力机制的方法。SE-Network 全称为 Squeeze-and-Excitation Network，它由 Hu[27]等人于 2017 年提出，这本质上就是一种 channel-wise 的注意力机制，也就是说它是在卷积神经网络的通道维度上来嵌入注意力机制。SE-Network 是通过对卷积神经网络通道赋予不同的权重来代表对每个通道不同的注意力。一个 SE-block 即 Squeeze-and-Excitation 模块结构如下：

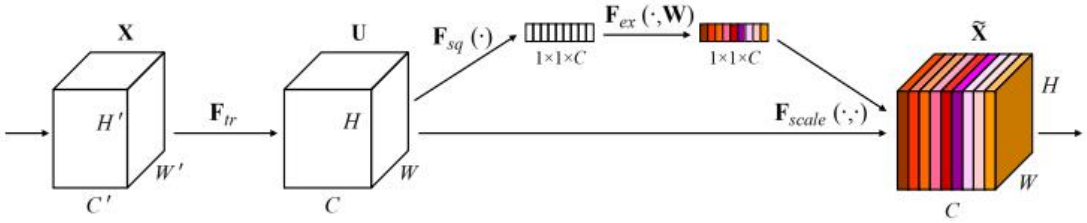


图 3-8 Squeeze-and-Excitation 模块结构示意图

我们可以看到，图中 $\mathbf{F}_{tr}$ 实际上就是普通网络中的卷积变换，X 为输入而 U 即经过卷积操作的特征图 feature map。接下来进行的才是 SE-Network 的操作，首先它通过 $\mathbf{F}_{sq}$ 即 Squeeze 操作，也就是压缩整个特征图，将整个特征图压缩为一个 1×1×C 的特征向量。然后我们要进行一个 $\mathbf{F}_{ex}$ Excitation 操作，这个 Excitation 的就是为了对每个通道赋予不同的权重，起到注意力机制的作用。当 Excitation 完成后，每个通道的权重被重新分配了，对于网络来说就是某些通道变得更重要而其他的一些通道则变得没那么重要了。最后我们需要用一个 $\mathbf{F}_{scale}$ 操作，将这个特征向量和整个特征图相乘，得到我们最终的输出 $\tilde{\mathbf{X}}$，而这个 $\tilde{\mathbf{X}}$ 就相当于是被放大的特征向量，也就是通道有着不同权重或者说重要性的特征图。

（1）Squeeze：全局信息嵌入

论文中提到，由于每个学习后的滤波器和一个局部感受野一起工作，因此输出 U 的每一个单元无法利用超出这个区域的上下文信息，利用 squeeze 模块可以减轻这个问题。正如我们上面所说的，Squeeze 部分最主要的作用就是压缩特征图，而这个压缩的过程同样可以起到全局信息嵌入的作用，因为特征向量中的每一个值都对应了它所在通道的全局信息，这是一种空间维度上的特征压缩。

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{3-5}$$

上式即为 squeeze 模块的公式，实际上 squeeze 模块采用的是全局平均池化（global average pooling），所以上式也就是全局平均池化的公式，其中 z_c 就是被压缩后的特征向量的第 c 个通道的值，而 u_c 为特征图的第 c 个通道值，i 和 j 分别代表特征图高和宽的第 i 或第 j 个像素，H 和 W 分别代表高和宽。

（2）Excitation：自适应再校准

我们完成 squeeze 操作后就需要利用 excitation 操作为通道间关系赋值了。为了充分利用 squeeze 中的信息并完全捕获通道上的依赖性，需要满足两个标准：其一，它必须较为灵活的（特别是它必须能够学习通道之间的非线性相互作用），其二，它必须学习非互斥关系，因为我们希望确保多个通道都能被重视。因此，论文中利用了一个简单的有 sigmoid 函数的门机制，下面是 excitation 模块的表示：

$$s = \mathrm{F}_{ex}(\mathrm{z}, \mathrm{W}) = \sigma(g(\mathrm{z}, \mathrm{W})) = \sigma(\mathrm{W}_2\delta(\mathrm{W}_1\mathrm{z})) \tag{3-6}$$

式中 δ 指代 ReLU 激活函数，σ 为 sigmoid 激活函数，而 W_1 和 W_2 代表了两个全连接层的权重矩阵，维度分别为 $\frac{C}{r}\times C$ 和 $C\times\frac{C}{r}$。r 是一个缩小比率，为的是将通道维度再压缩，是一个超参数，这个我们后面讨论。所以，可以看出 Excitation 部分其实是通过一个小型的神经网络来实现的，这个神经网络中先是一个 FC 全连接层，然后是 ReLU 激活函数，再接一个全连接层，最后利用 sigmoid 激活函数进行归一化。第一个全连接层可以说是一个维度压缩层，然后第二个全连接层是一个维度扩张层，最后为了使空间维度回到原来卷积后的输出 U 的维度，我们要将 U 与 s 相乘，公式如下：

$$\widetilde{\mathrm{x}}_{\mathrm{c}} = \mathrm{F}_{scale}(s_c, u_c) = s_c u_c \tag{3-7}$$

这样我们就得到了最后的输出，每个通道叠加起来得到 $\widetilde{\mathrm{X}}$ 特征图，维度为 H×W。这样我们就介绍完了整个 SE-Network 的架构，可以发现，SE-Network 网络其实可以非常容易的嵌入到一些主流的卷积神经网络中，作者在论文中给出了 SE 网络嵌入到 Inception 和 ResNet 中的结构图，下图为 SE-ResNet 结构：

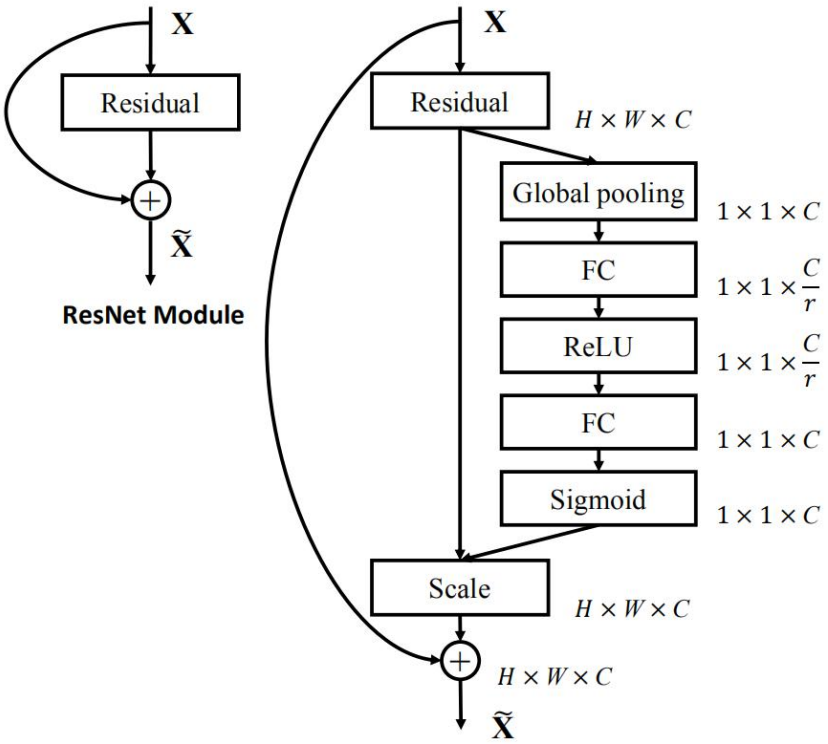


图 3-9 SE 模块嵌入 ResNet 网络示意图

作为一个基于通道注意力机制的网络，SE-Network 的架构非常简单，而且根据论文的实验，其做到了在只增加很少的参数量和计算资源的前提下，将分类精度提高一定程度。

关于上文中提到得超参数压缩比率 r，论文中也作出了讨论，压缩比率对于网络精度的影响并不是单调的，作者使 r 的取值区间为 2-32，取偶数，在平衡了参数量和网络精度的情况下，选择了 16 作为最优取值。本文中涉及到的遥感影像语义分割和论文中的任务不太一样，因此有必要尝试对 r 的取值重新进行实验后选择，这一块的讨论将在第五章中提到。本文实验中嵌入的 SE-Network 模块和上图中一样，没有对其 squeeze 或 excitation 操作部分进行修改。

第四章 遥感图像语义分割模型实验结果与讨论

4.1 实验方案设计

本文基于深度学习的卷积神经网络，设计出基于 ResNet-18/50 的 U-net 网络架构用于遥感图像的语义分割。考虑到遥感影像的高分辨率特性，可能需要用更深的网络去学习，因此本文利用 ResNet-18/50 作为经典语义分割网络 U-Net 的编码器，保证网络可以在足够深的条件下不出现梯度消失或爆炸的情况。接下来，通过在网络 encoder 的 ResNet 部分加入 Dropout 和 decoder 部分加入 Batch Normalization 和 dropout 层，以此来减轻过拟合现象。本文还尝试了利用 SE-Network 通道注意力机制网络性能，将 SE-Network 嵌入 ResNet 网络，使得特征提取部分具备注意力特性。

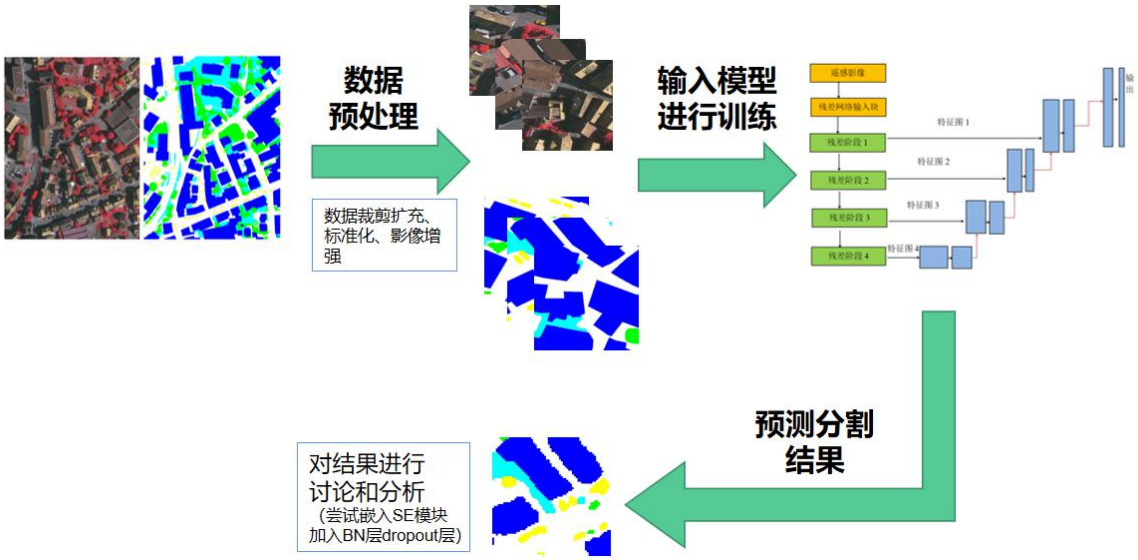


图 4-1 实验设计图

4.2 实验数据集介绍

本文实验中使用的是 ISPRS 2D 语义标签竞赛的公开数据集 Vaihingen，即 Vaihingen 地区的航空影像数据。数据集中包含了 33 幅不同大小高分辨率的正射影像，空间分辨率为 9cm。数据集中 16 幅影像包含了真实地面标签（ground truth），并且影像中的所有地物被分为如下六类：

不透水表面（路面）Impervious surfaces (RGB: 255, 255, 255) 白色

建筑物 Building (RGB: 0, 0, 255) 蓝色

低植被 Low vegetation (RGB: 0, 255, 255) 青色

树木 Tree (RGB: 0, 255, 0) 绿色

汽车 Car (RGB: 255, 255, 0) 黄色

杂物/背景 Clutter/background (RGB: 255, 0, 0) 红色

下图是 Vaihingen 数据集 33 幅影像在原始影像上的位置和第一块区域影像的正射影像图及真实地面标签图。

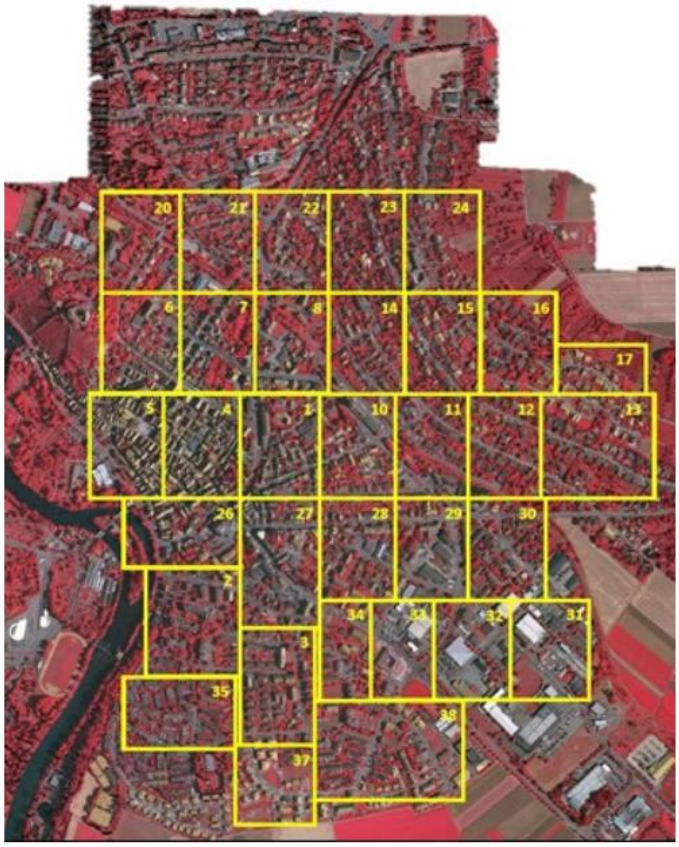


图 4-2 Vaihingen 数据集 33 幅影像与原始影像

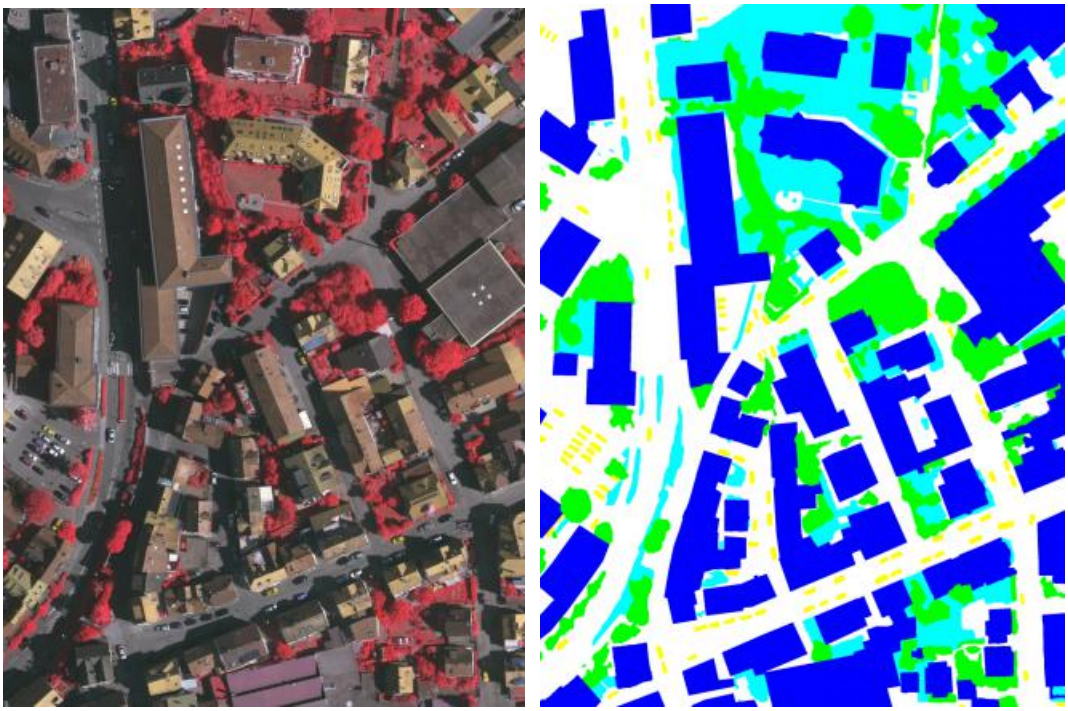


图 4-3 编号 1 切片正射影像图和地表真实标签

4.3 实验评估方法介绍

为了验证我们模型完成任务的能力，即模型的准确性，我们需要定义一些方法来帮助我们评估模型。而在图像语义分割的问题中，常用的经典评估准则有：混淆矩阵、准确率、召回率、F1 分数、整体准确率。

由于后面四个指标实际上都与混淆矩阵相关，我们先从混淆矩阵介绍。

(1) 混淆矩阵

混淆矩阵一般是一个 n 行 n 列的矩阵，n 行代表图像分割的类别所对应的真实值，n 列代表预测值类别。对角线上的数值代表该类别在分割结果中被正确分类的数量，而该行中其他列则代表被错误分割为其他类别时，被误分到该类中的数量。二分类的混淆矩阵如下：

表 4-1 二分类混淆矩阵

混淆矩阵		预测值	
		Positive	Negative
真实值	Positive	TP	FN
	Negative	FP	TN

表中 TP 为真阳性，即预测值和真实值都为真；FP 为假阳性，即真实值为真但预测值为假；

FN 为假阴性，即真实值为假而预测值为真；TN 时真阴性，即真实值和预测值都为假。

(2) 总体准确率（Overall Accuracy，OA）：在图像分割中，即预测正确的像素与图像含有总像素的比值。

$$OA = \frac{TP + FN}{TP + TN + FP + FN} \tag{4-1}$$

(3) 精确率（Precision，P）：它表示预测出的值中占样本中真实值的比例。

$$P = \frac{TP}{TP + FP} \tag{4-2}$$

(4) 召回率（Recall，R）：它表示样本中的真实值被正确预测的比例。

$$R = \frac{TP}{TP + FN} \tag{4-3}$$

(5) F1 分数（F1-score）：它同时兼顾了分类模型的准确率和召回率。F1 分数可以看作是模型准确率和召回率的一种加权平均，它的最大值是 1，最小值是 0，值越大意味着模型越好。

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4-4}$$

以上指标都为值越大，模型越精确。

4.4 实验环境及实验数据预处理

4.4.1 实验环境

本文实验的环境为 Windows 10 64 位操作系统，并在 Python 3.7.4 下运行并配合开源深度学习框架 Tensorflow-GPU 2.3.0(较新版本)和其官方高级 API(应用程序编程接口)Keras 2.4.3。

Keras 最早在 2015 年诞生，是一个构建深度学习模型时高的级易于使用的 API，由 Python 编写而成。Tensorflow，可简称 TF。这是一个用于机器学习和神经网络训练的符号数学库，它包含我们开发要求的许多功能，它就如同我们盖房子时所需的各种材料和工具，并且可以直接使用它的一些封装架构。如 C++/Python 的语言支持；支持多 GPU；包含网络训练的低级、高级接口等。TF 提供的 API 对搭建神经网络有了足够的自由度且能自动计算任意构建的损失函数的导数。并且他不仅支持深度学习，还包含支持强化学习和其他算法的工具。

Keras 最早是一个独立的包，而它的后端最开始也不是 tensorflow，但随着 tensorflow 的普及，tensorflow 渐渐成为了 keras 的默认后端。从 2.3.0 版本开始，Keras 与 tensorflow 中的 tf.keras 同步，由此意味着 tensorflow 的软件包 tf.keras 某种程度上替代了 Keras。

4.4.2 实验数据预处理

（1）数据标准化

原始影像数据中每个特征之间都是有差异的，这样我们训练的模型在处理对待不同的特征时也是有一定差异的，大概意思就是模型无法公平地学习每一个特征或者说模型可能会学某些特征比学另一些特征好学。而进行过数据标准化处理后，各特征会处于同一数量级，这时候就可以进行公平的综合对比评价了。

在本实验中，我们使用的是(0,1)标准化方法，他的公式为：x = (x - Min) / (Max - Min)，就是将影像的灰度值都在 0-255 之间的原始遥感影像数据缩放到灰度值 0-1 之间。

（2）影像增强

为了能凸显并方便学习遥感影像数据中的某些特征，本实验中还采用图像增强方法。在遥感图像的解译过程中，图像增强也非常常用，主要是为了突出遥感图像中的某些信息，且削弱或去除某些不需要的信息，使图像更易判读。本实验中将遥感影像映射到 HSV 色彩空间中，并利用随机值对遥感影像的亮度、色相、饱和度进行随机的调整，以满足模型训练的需要。

(3)数据裁剪与扩充

一般的深度学习都是需要巨大的数据量进行支撑，这样网络才有机会学习完善。然而在遥感数据集中，由于遥感影像的尺寸过大，高分辨率遥感影像公开数据集中的影像数量较少，如果直接一次性输入整幅遥感影像，由于硬件的限制，在很多情况下会导致显存无法使用。因此我们需要对影像进行裁剪。

除此之外，对影像进行裁剪还可以在某种程度上可以减轻对小目标分割训练的不精确。对裁剪完成之后的数据我们还可以采用一定的方法进行扩充，扩充我们整个数据集的大小，以满足深度学习训练时所需的大量数据。本实验中将数据集裁剪为 512*512 或 256*256 的大小，并对图像进行逆时针 90 度、270 度的旋转，还进行竖直和水平翻转。

(4)One-Hot 编码

One-Hot 编码，又称为一位有效编码（或独热编码），主要是采用 N 位状态寄存器来对 N 个状态进行编码。实际上，One-Hot 编码就是将分类值映射到整数值的一种方法，它将分类变量表示为二进制向量。下图为 One-Hot 编码示意：

图 4-4 One-hot 编码示意图

在图像语义分割的多分类问题中，由于有多个类别，为了使神经网络能顺利向后传播，则需要对标签数据进行 One-Hot 编码。因为是像素级别的分割，因此前向传播输出的就是此像素属于哪一类（利用 softmax 分类器）的一个一维向量合成的整张图像，维度为：H*W*类别数目，而此时就需要利用 One-Hot 编码来帮助计算损失函数然后进行反向传播。下表展示了本实验中具体类别的 One-hot 编码：

表 4-2 本文实验中 One-hot 编码

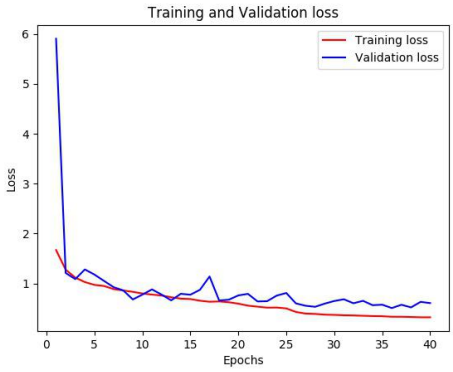
不透水表面	1	0	0	0	0	0
建 筑 物	0	1	0	0	0	0
低 植 被	0	0	1	0	0	0
树　　木	0	0	0	1	0	0
汽　　车	0	0	0	0	1	0
背　　景	0	0	0	0	0	1

装 订 线

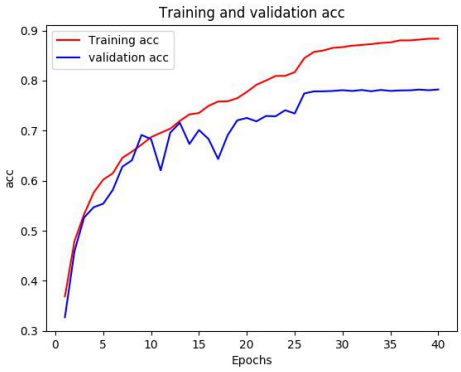
4.5 实验结果与讨论

4.5.1 基于 ResNet-18/50-UNet 网络实验结果与分析

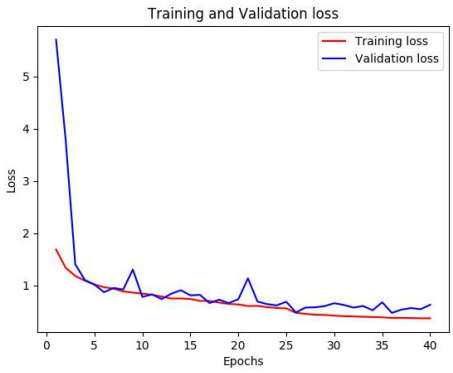
实验一首先使用基于 ResNet-18 的 U-Net 网络架构，如图 3-1 所示。超参数部分 batch_size 设置为 4/8/16，epoch 为 40，使用 Adam 优化器进行梯度下降，初始学习率为10^{-3}，使用 LearningRateScheduler 函数帮助调整学习率，当 epoch 达到 25 时，学习率下降为10^{-4}。损失函数采用交叉熵函数。使用 ISPRS 2D 语义标签数据集中有真实标签的 15 张影像进行训练和验证，剩余 1 张有真实标签的影像进行测试。依据显存容量，使用实验部分预处理中的方法，将训练集中的原始影像全部裁剪为 256×256 的图像块，并扩充训练数据集为 2400 张，最终进行模型训练的图片有 1920 张，验证的图片有 480 张。训练精度及损失函数结果如下图所示：



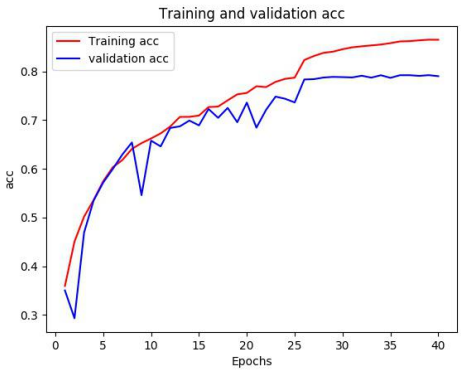
（a-1）



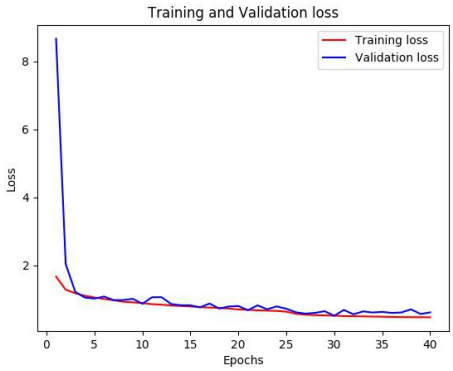
(a-2)



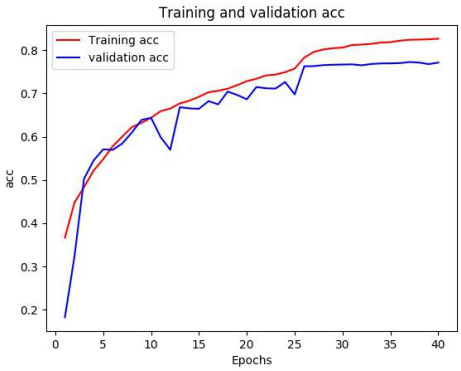
（b-1）



(b-2)



（c-1）



(c-2)

图 4-5 不同 batch_size 对 ResNet-18-UNet 训练影响 16_50

由于个人笔记本电脑内存的限制，最开始采用进行训练，后来在学校高性能计算平台上尝试了 batch_size 为 8 和 16 进行训练。图 4- 5 中子图 a-1 代表 batch_size 为 4 损失函数曲线，a-2 代表训练精度曲线，b 代表 batch_size 为 8，c 代表 batch_size 为 16。可以看出，各个不同的 batch_size 验证集的精度都较为一致，但训练精度不一，batch_size 为 4 时训练精度可以达到 88-90%，而 batch_size 为 8 或 16 时训练精度在 82-85%左右。同时还可以发现当在 epoch25 时训练和验证集的精度有一个较为明显的提升，此时是因为将学习率下降了，在此之后训练精度依然在上升，而验证集精度却基本保持平稳。除去 epoch 为 40 外，还进行了 epoch 为 60-80 的训练，

装 订 线

结果显示训练集精度依旧可以上升，最终在 epoch60 左右达到 90%后几乎不再上升，而验证精度依旧保持平缓甚至有一点下降，可能出现了过拟合。

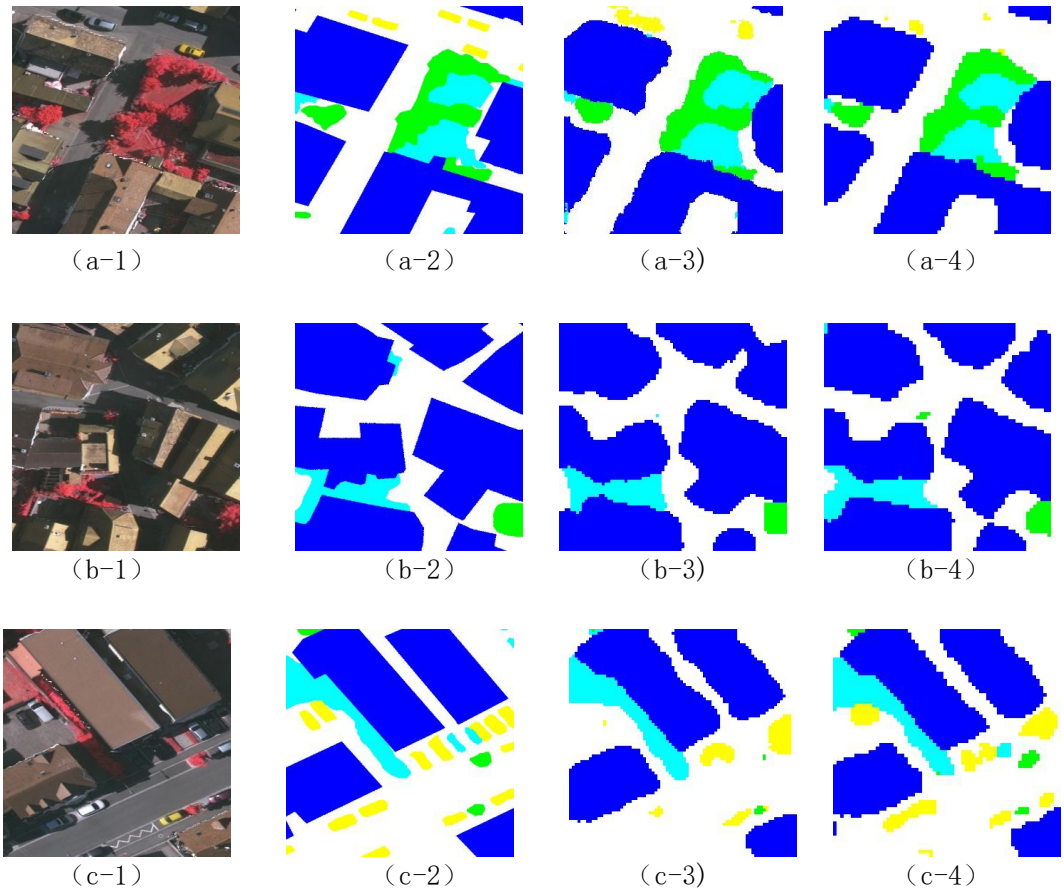


图 4-6 实验一分割结果图

图 4-6 中的第一列即遥感影像，第二列为遥感影像对应的真实地表标签。第三列则为 ResNet-18-UNet 的分割结果图，第四列即 ResNet-50-UNet 的分割结果图。

我们从结果图中直观地可以感受到 ResNet-50 为编码器骨架的结果要优于 ResNet-18。首先是建筑物的棱角，ResNet-50 棱角更加明显，直线弯曲较少，更贴近真实标签，如图 a-3 与 a-4。而且在 ResNet-18 中出现了建筑物粘连的情况，这种情况一般发生在距离较近且中间有阴影的建筑物之间，如图 b-3，而这样的情况在 ResNet-50 中情况得到了缓解，如 b-4。同时，ResNet-50 对小尺度且相对独立的物体分割效果更好，如图 c-3 和 c-4。

表 4-3 实验一分割精度结果

		不透水面（路面）	建筑物	低植被	树木	汽车
ResNet-18 encoder	F1 分数（%）	87.3	92.3	82.6	86.3	44.5
	总体精度 Overall Accuracy(%)	86.5				
ResNet-50 encoder	F1 分数（%）	90.1	94.0	85.3	88.3	55.2
	总体精度 Overall Accuracy(%)	89.3				

从表 4-3 中可以看出，ResNet-50 的分割精度会全方面领先 ResNet-18，在汽车的分割精度上提升最多，有接近 11%的提升。并且建筑物和不透水面（路面）的精度最高，且由于这两种地物在影像中本就占比较大，拉高了整体的精度；同时分割精度最差的是汽车，但同样因为汽车的样本数量很少，因此较低的精度对整体精度影像不大。

从混淆矩阵种发现，车辆与路面混淆分类的数量较多，主要可能有两个原因：其一，汽车基本只可能被包含在路面分割区域以内，很少与其它类别靠近或连接；其二，当汽车停靠在一起时，特别时当汽车之间有阴影时，容易出现两种情况，要么汽车都被分为了路面，要么汽车之间的路面被分为汽车，两辆甚至几辆车都连为一体。

4.5.2 尝试嵌入 SE 模块的网络实验讨论

在上一小节实验的基础上，本实验尝试基于 SE-Network 通道注意力机制对网络进行优化测试，利用 BatchNormalization 提高训练速度，并且针对上一节中疑似过拟合的现象（epoch 在 60 之前训练精度持续上升而验证精度平缓甚至下降）加入 dropout 层进行实验。超参数部分 batch_size 设置为 4，epoch 为 40，使用 Adam 优化器进行梯度下降，初始学习率为10^{-3}，使用 LearningRateScheduler 函数帮助调整学习率，当 epoch 达到 25 时，学习率下降为10^{-4}。在 SE-Network 的超参数 reduce_ratio 方面，原论文中平衡了性能与参数量之后选择了 16 作为 reduce_ratio 的取值。而由于遥感影像与原论文中数据集出入较大，有必要对不同的 reduce_ratio 值进行比较。根据原论文中的实验，选取了 2、8、16 三个值进行测试，因为这三个值得出的精度较其它值相对较高。除此之外，在 ResNet 每个卷积层后都加入了 dropout 层，随机失活概率为 0.3，还在 U-Net 的 decoder 部分加入了 BatchNormalization 层和 dropout 层，失活概率同样为 0.3。

Batch Normalization 和 dropout 是深度学习中常用的网络优化技巧，下文用一点篇幅介绍。

（1）BatchNormalization（BN）：BN 又称批量正则化，是一种对神经网络中层间输出值的归一化。在机器学习领域中，对原始输入值进行归一化是一种很常见的帮助网络更快收敛的方法，

因为这样可以使得不同的特征输入值处于一个较一致的区间内，或者说是数量级。BN 的作用就是对每一层的神经元进行归一化，通过将上一隐藏层的神经元的值归一化处理，限定在一定范围内，也就是使它们的均值和方差不变，来帮助加快训练过程。因为每层的均值和方差不变，这样 BN 等于减少了这些分布变化的隐藏层值的数量，同时减弱了靠前层的参数的作用与靠后层参数作用之间的联系。这样可以使每层的神经元自己学习，稍稍独立于其它层，这样来加速整个网络的学习。

（2）Dropout：Dropout 方法是深度学习中常用的防止过拟合的方法之一。它的基本思想就是在深度学习的训练过程中，对于神经网络中的神经元，一般是隐藏层中的，进行随机丢弃，或者以一定的概率将其移除。从另一种角度看，就是在每一次训练的过程中只有随机的一部分起到了作用，其它的神经元等于‘死亡’。这也意味着网络不会去过分依赖某些神经元学到的特征，这样就可以帮助网络泛化。

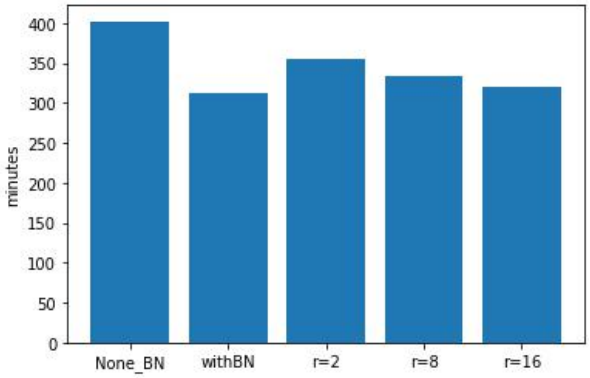


图 4-7 训练耗时

上图是加入 BN 层后的耗时情况，选取了 ResNet-18-UNet 作为测试网络，其余超参数设置与前面介绍一致。第一条柱状 None_BN 代表未在 decoder 部分添加 BN 层时训练的耗时情况(单位为分钟），而 withBN 则代表加入 BN 层后的训练耗时。r 代表 reduce_ratio 的值，并且此时的网络都已添加 BN 层。从图中可以看出，BN 层确实加快了网络的训练速度。

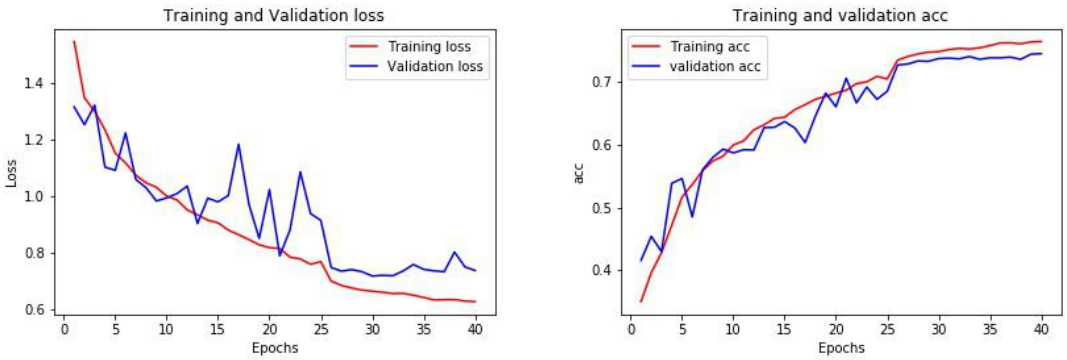


图 4-8 加入 dropout 层后网络训练情况

上图为在 ResNet-18-UNet（无 SE 模块）网络中加入 dropout 层后的训练图，实际上验证精度与未加入 dropout 层并无太大区别，但是与其它相同迭代次数的情况相比，其训练精度有所下降。Dropout 层在卷积神经网络中并不会都起作用，特别当数据集不同时，精度甚至可能不升反

降。训练精度下降证明网络对数据集的特征学习并不够好，猜测这有可能是因为 dropout 失活神经元导致网络无法更好地学习。换句话说，由于遥感影像的复杂性，数量不足的神经元难以将网络学习到较高的精度。

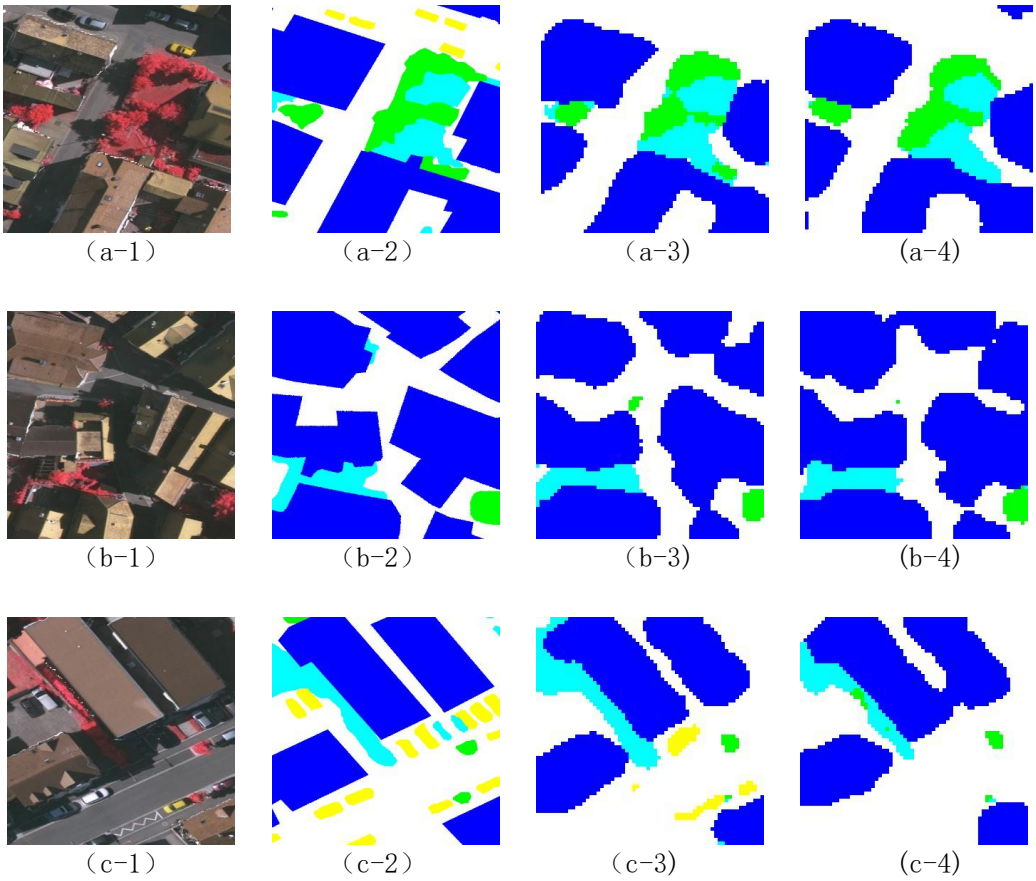


图 4-9 嵌入 SE 模块结果图

上图子图中，n=a/b/c，n-1/2 代表遥感影像和地表真实标签，所有网络 encoder 都为 ResNet-18；n-3 结果图对应以 reduce_ratio 为 8 嵌入 SE-Network 模块的网络；n-4 结果图对应 reduce_ratio 为 16。从目前的实验结果图我们可以发现，嵌入 SE-Network 的网络效果并不好。最明显的差距是汽车的分割，相比于无 SE 模块的网络，嵌入它之后网络将汽车几乎都分为了不透水面，或者说没有将汽车成功分割出来，如图 c-3 和 c-4。同时建筑物之间粘连的现象也严重了，特别是建筑物之间有阴影的情况，如图 b-3 和 b-4。由于时间有限，这里没有展示更多的结果图。以下是实验的精度结果表：

表 4-4 实验结果精度

	ResNet-18 b=4	ResNet-18 r=2	ResNet-18 r=8	ResNet-18 r=16	ResNet-18 r=8 dropout	ResNet-50 r=8 dropout
	F1 分数(%)	F1 分数(%)	F1 分数(%)	F1 分数(%)	F1 分数(%)	F1 分数(%)
不透水面（路面）	87.3	87.0	87.2	84.5	89.0	88.9
建筑物	92.3	91.5	91.6	90.3	89.1	91.1
低植被	82.6	80.0	81.5	77.5	73.6	79.1
树木	86.3	84.1	85.4	81.7	79.8	83.4
汽车	44.5	26.8	15.5	1.4	6.2	3.6
总体精度 Overall Accuracy(%)	**86.5**	**85.6**	**86.2**	**83.2**	**82.5**	**84.7**

表 5-4 中，表第一行写明了网络的种类，其中 ResNet-18 代表以 ResNet-18 作为 encoder，b=4 即 batchsize 为 4，同时之后的网络都设置 batchsize 为 4，然后 r=2 代表嵌入 SE 模块的网络中超参数 reduce_ratio 为 2，最后 dropout 表示网络中加入了 dropout 层。从表 5-4 中我们可以发现一些规律，首先是在当前实验下，同等条件加入 SE 模块的网络可能没有起到更好的效果，精度反而有所下降；其次，我们可以发现，当 reduce_ratio 为 8 时，引入 SE 模块网络的总体精度相对来说最高；除此之外，我们可以发现汽车类别的分割精度在加入 SE 模块后下降，并且还随着 r 的增大而减小，在 r 为 16 时达到极低的水平。在 SE 模块中进行 Excitation 操作时，r 是对 squeeze 之后的通道维度再次进行压缩，因此 r 越大则意味着通道被压缩的越多，虽然能减少网络的参数量，但可能在这里会损失更多的上下文信息，最终导致汽车类别的分割精度下降。目前的实验显示加入 SE 模块后总体精度有所下降，有可能是由于超参数调整不到位或其它原因造成的，因此，后续工作中将尝试进一步优化超参数，提升引入 SE 模块后的模型精度。

结论与展望

高分辨率遥感影像中存在大量复杂因素（例如目标空间分布分散，地物尺度不一变化多样，地物附近存在阴影复杂等），传统的图像分割方法（如阈值法、区域法，边缘检测法，基于图论的方法，马尔可夫随机场）准确性低且计算时间长，难以满足大规模的智能化的应用需求。为了适应高分辨率遥感影像的特点，本文参考经典的深度学习语义分割模型，设计出了基于深度学习的高分遥感影像语义分割的模型，并引入目前计算机视觉领域的热点注意力机制，将其嵌入网络中。

本文的主要工作如下：

（1）高分辨率遥感影像的信息丰富，地物复杂，因此高分辨率遥感影像的语义分割可能需要更深层更强大的网络来对进行分割。同时考虑到时间及自身能力，设计了一种基于残差网络 ResNet 的 U-Net 遥感图像语义分割模型。将 ResNet-18/50 作为 U-Net 的 encoder 部分，进行特征提取，并在每一卷积阶段输出特征图，以作为 decoder 部分的特征融合图，同时为加快网络模型的训练速度，还在 decoder 部分加入 BatchNormalization 层。在 ISPRS 2D 的 Vaihingen 数据集上进行实验，验证模型的有效性与精确度。

（2）本文尝试将基于通道注意力机制的网络模块 SE-Network 嵌入上述网络的 encoder 也就是 ResNet 中，并进行实验；尝试验证注意力机制在本文设计的网络中的有效性。针对模型过拟合的现象，本文利用深度学习领域常用的 dropout 方法来减轻此问题。在 ResNet 和 decoder 部分中加入 dropout 层，然后继续进行实验验证。

最终对上述方法进行实验结果对比后，首先 ResNet-18-UNet 模型的总体分割精度到达了 86.5，且对建筑物的分割精度最高，f1 分数达到了 92.3%；更深的 ResNet-50-UNet 的表现更加出色，总体分割度接近 90%，并且汽车类别的分割精度相比 ResNet-18-UNet 提升了近 11%，并且发现在网络中加入 Batch Normalization 层确实可以加快训练速度。其次，目前的实验显示加入 SE 模块后总体精度有所下降，并且尺度小样本量少的汽车类别分割精度极低，可能是由于超参数调整不到位或其它原因造成的，因此，后续工作中将尝试进一步优化超参数，提升引入 SE 模块后的模型精度。除此之外，实验结果表明加入 dropout 层并没有起到作用甚至使训练精度下降。

高分遥感影像的语义分割是一个非常值得深究的领域，但是由于时间和本人研究能力的限制，本次毕业设计有许多不足之处，需要进一步改善：

（1）U-Net 网络是语义分割中的经典网络，但是在多分类精度效果上并不如二分类理想，因此许多学者提出了很多 U-Net 网络的变体，同时语义分割还有很多更加复杂的网络模型，都需要进一步学习参考。残差网络 ResNet 也是一种非常著名的网络结构，经常在计算机视觉领域，本文仅仅对比了残差块有一定差异的 ResNet-18 和 ResNet-50 网络，对更深的网络没有进行，实验，而从结果看更深的网络很有可能更好地学习到遥感图像更多的特征，因此应该进一步尝试 ResNet 其他网络，甚至其变体如 ResNeXt

（2）除去网络结构的进一步尝试优化外，对于网络中卷积层的优化也是一个方向，目前有许多能一定程度增加精度而又不会增加过多参数量的方法，如深度可分离卷积，空洞卷积等。

（3）对于超参数的调整方面，本文还存在较大的问题。由于网络模型是从智能驾驶的道路识别二分类问题上迁移过来，并且加入 SE 模块后超参数组合增多，利用网格搜索或随机搜索调

参比较困难且耗时。可以尝试利用智能优化学习，对超参数进行优化，比如利用模拟退火算法，贝叶斯优化等方法。

（4）在注意力机制的优化方面，本文中采用的通道注意力机制比较简单，其主要的优势是可以轻易的嵌入到各种网络中并仅增加很少的计算量，因此可能对于复杂的高分遥感影像效果也不好。可以利用一些基于图像平面二维空间的注意力模型，比如 Swin Transformer 模型。

（5）对于小尺度低样本地物分割精度较低的问题，首先可以尝试针对性地扩充数据集，也就是将样本量较小地地物类别多进行扩充，使得其样本量占比上升。可以尝试优化损失函数，将交叉熵损失函数替换为加权交叉熵损失函数，也就是对地物占比权重的影响也计算进去。

参考文献

[1] 孙家抦. 遥感原理与应用（第三版）[M]. 武汉：武汉大学出版社，2013.

[2] 郑凯, 李建胜. 基于深度神经网络的图像语义分割综述[J].测绘与空间地理信息, 2020, 43 (10) :119-125.

[3] 侯永宏，叶秀峰，张亮，等. 基于深度学习的无人机人交互系统[J]. 天津大学学报: 自然科学与工程技术版，2017，50(9) : 967－974.

[4] 马乐乐，李照洋，董嘉蓉，等. 基于计算机视觉及深度学习的无人机手势控制系统［J］. 计算机工程与科学，2018，40(05) : 872－879.

[5] 王弈，李传富．人工智能方法在医学图像处理中的研究新进展［J］．中国医学物理学杂志，2013，30(3) : 4138－4 143.

[6] Yen J C，Chang F J，Chang S. A new criterion form automatic multilevel thresholding［J］. IEEE Transactions on Image Processing，1995, 4(3) : 370-378

[7] Khan J F，Bhuiyan S M A，Adhami R R. Image segmentation and shape analysis for road-sign detection [J]. IEEE Transactions on Intelligent Transportation Systems，2011，12(1)：83-96. DOI：10.1109/ TITS. 2010.2073466.

[8] PHAM D L，XU C Y，PRINCE J L. A survey of current methods in medical image segmentation [J]. Annual Review of Biomedical Engineering，2000，2(1): 315-337.

[9] 朱卫. 基于随机森林算法的街道场景语义分割[D]. 哈尔滨理工大学, 2019.

[10] 亓祥惠. 基于 MRF 与模糊聚类的图像分割算法研究[D]. 兰州理工大学, 2019.

[11] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).

[12] Karen S, Andrew Z. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[13] Szegedy C , Liu W , Jia Y , et al. Going Deeper with Convolutions[J]. IEEE Computer Society, 2014.

[14] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[15] Long J , Shelhamer E , Darrell T . Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(4):640-651.

[16] Badrinarayanan V, Handa A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling[J]. 2015.

[17] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation[J]. arXiv preprint arXiv: 1505.04597, 2015.

[18] 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.

[19] Liangpei, Zhang, Lefei, et al. Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art[J]. IEEE Geoscience & Remote Sensing Magazine, 2016.

[20] Fu J, Zheng H, Mei T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[21] Li R, Liu W, Yang L, et al. DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, 11(11):3954-3962.

[22] Yang X, Li S S, Chen Z C, et al. An Attention-Fused Network for Semantic Segmentation of Very-High-Resolution Remote Sensing Imagery[J]. arXiv preprint arXiv:2105.04132, 2021.

[23] Ashish V, Noam S, Niki P, et al. Attention is all you need [J]. arXiv preprint arXiv:1706.03762, 2017

[24] Dosovitskiy A , Beyer L , Kolesnikov A , et al. An Image is Worth 16x16 Words: Transformers

for Image Recognition at Scale[J]. 2020.
[25] Liu Z, Lin Y T, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows[J]. arXiv preprint arXiv: 2103.14030, 2021
[26] Li R, Zheng S Y, Duan C X, et al. Multi-stage Attention ResU-Net for Semantic Segmentation of Fine-Resolution Remote Sensing Images[J]. arXiv preprint arXiv: 2011.14302, 2020
[27] Hu J,Li S，Samuel A. Squeeze-and-Excitation Networks.[J]. IEEE transactions on pattern analysis and machine intelligence, 2019.
[28] 苏健民, 杨岚心, 景维鹏. 基于 U-Net 的高分辨率遥感图像语义分割方法[J]. 计算机工程与应用, 2019, 55(07):207-213.
[29] 黄鹏,郑淇,梁超.图像分割方法综述[J].武汉大学学报(理学版), 2020, 66(06):519-531.
[30] 卫佳杰. 基于 U-Net 和注意力机制的双源遥感图像语义分割方法研究[D].西安电子科技大学,2020.

装 订 线

致 谢

在一个温和的午后，我终于修改完了我的论文，当我按下左上角的保存时，我也终于长吁了一口气。完成这样几十页的一篇论文，着实是个浩大的工程，我需要感谢许多人，当然也要感谢自己。

首先我最要感谢的，就是我的指导老师席江波老师了。他非常负责、耐心和认真，在毕业设计的各个阶段，毕业设计的开题，实验过程中困难问题的解决，毕业论文的写作，都给予了我极大的帮助。但这些都不是他对我最大的帮助，他教会我最重要的东西是严谨的治学态度。他告诉我们不能只顾着埋头编程实验，你的成果需要用足够的耐心细心和严谨的逻辑来表达，否则再好的成果也不过是表格中冷冰冰的数字。我从他的教诲中还体会到了我们要用对待毕业论文的方式去对待生活学习中的重要事情，这是一种生活上的态度。我还要感谢在毕业设计上给过我帮助的学姐们，也要感谢给我极大帮助的同一组的同学们。回过头来看，我本应把毕设做得更好，但也只能怪自己，只愿自己在人生的下一章里能吸取教训，砥砺前行。

另外我要感谢在我大学四年求学历程中每一位给予我帮助的老师、同学和朋友，感谢我的室友们，他们是我快乐的源泉，四年的朝夕相处笑声从没停过；感谢所有与我一起热爱篮球的兄弟，希望我们还能再聚首球场；感谢陪我走过前两年的 405，也感谢最后两年的 405。

感谢在申请的路上每一个帮助我的人，感谢自己的勇气和信心，最终能有机会去看看更广阔的世界。最后我还要感谢我的父母，是他们的言传身教塑造了一个具备完整独立人格的我，也让我充满永远向前的动力。

致谢是我的论文的最后一部分，也预示着四年青春的结尾，回忆总会在不经意间被勾起。四年来，我下过秦始皇陵，在秦岭的脚下埋过木桩，在陕西历史博物馆里见过各代珍宝，也曾在厚重的城墙下辨别过历史的痕迹。坐在图书馆和教室的桌子前，彷佛又见到了曾经的自己。每个人的青春都会有遗憾，但无悔与否是自己决定的。

装 订 线