

摘要

正确且高效的识别岩石的种类一直都是地质工作者的首要任务。如果采集到的岩石特征明显，那么可以很容易将其分类，但是如果岩石之间的特征相似且研究人员的经验不足的话就会对识别任务带来一定的麻烦。电气石的地质显微图像是电气石在电子显微镜下拍摄的。在地质领域，电气石是一种找矿指示物，而且其具有压电性等工业用途。电气石的地质图像鉴定在矿物存储层勘探和评估、环境保护等方面具有重要作用。电气石地质显微图像分割是指将电气石地质显微图像中的每个电气石颗粒划分成一个区域，它是电气石地质显微图像鉴定的一个重要步骤，具有很重要的意义。深度学习近些年的发展迅速，在图像，语音方面都取得巨大的突破。生活中到处可见深度学习的影子。打开手机要利用人脸识别解锁，支付账单需要用指纹支付，平时闲暇与智能音箱聊天，用智能音箱控制家电，购物等。虽然深度学习技术在我们的生活中随处可见，但是仍然有很多其他领域还没有利用深度学习技术。例如深度学习在图像方面的发展有目共睹，但是在岩石方面的进步却寥寥无几。我校有专业的岩石领域专家与大量的数据，将深度学习运用到岩石的识别与分类上不仅充分利用我校的优势，也能实现跨专业的合作，实现专业之间的互补，也为培养应用型，全能型人才奠定了基础。

本文借助利用迁移学习的方法，搭建 BiSeNet 网络模型，其中的 Context Path 采用 ResNet 网络提取电气石的特征。利用我校相关实验室的岩石薄片标本和电子显微镜，建立一个电气石的数据集；然后搭建的 BiSeNet 模型进行训练。最后根据模型在测试集上的表现来分析，然后进行超参数和网络模型的优化。

经过研究，本文最终实现了预期功能。在本文提供的数据集上训练出了好的模型，预测的准确率接近人类水平，可以进一步改进后用于实际场景中。

关键词：深度学习；语义分割；tensorflow；电气石显微图像分割

Abstract

Correct and efficient identification of rock types has always been the primary task of geologists. If the characteristics of the collected rocks are obvious, they can be easily classified, but if the characteristics of the rocks are similar and the experience of the researchers is insufficient, it will bring some trouble to the identification task. The geological microscopic image of tourmaline was taken by tourmaline under electron microscope. In the geological field, tourmaline is a prospecting indicator, and it has piezoelectric and other industrial uses. Geological image identification of tourmaline plays an important role in mineral reservoir exploration and evaluation, environmental protection and so on. Tourmaline geological micro-image segmentation refers to the division of each tourmaline particle into a region in tourmaline geological micro-image. It is an important step in the identification of tourmaline geological micro-image and has great significance. Deep learning has developed rapidly in recent years, and has made great breakthroughs in image and voice. The shadow of deep learning can be seen everywhere in life. Open the mobile phone to use face recognition to unlock, pay bills with fingerprints, chat with smart speakers in spare time, use smart speakers to control household appliances, shopping and so on. Although in-depth learning technology is ubiquitous in our life, there are still many other areas that have not yet utilized in-depth learning technology. For example, the development of in-depth learning in image is obvious to all, but the progress in rock is very few. Our school has professional experts in the field of rock and a large number of data. The application of in-depth learning to rock identification and classification not only makes full use of the advantages of our school, but also achieves cross-professional cooperation and complementarity among specialties. It also lays a foundation for training applied and versatile talents.

In this paper, we use the method of migration learning to build a BiSeNet network model, in which Context Path uses the ResNet network to extract the characteristics of tourmaline. Using a rock slice specimen and an electron microscope from the relevant laboratory of our school, a data set of tourmaline was built; then the BiSeNet model was built for training. Finally, the model is analyzed according to the performance of the model on the test set, and then the optimization of the hyperparameter and network model is performed.

After research, this paper finally realized the expected function. A good model is trained on the dataset provided in this paper. The accuracy of the prediction is close to the human level and can be further improved for use in actual scenarios.

Key words: deep learning; semantic segmentation; tensorflow; tourmaline microscopic image segmentation

目录

1. 绪论.....	1
1.1 电气石地质显微图像分割的背景和意义.....	1
1.1.1 电气石地质背景	1
1.1.2 选题意义	2
1.2 神经网络介绍.....	3
1.2.1 人工智能的发展	3
1.2.2 深度学习的崛起	4
1.3 论文研究的主要内容.....	6
2. 电气石数据集的制作.....	9
2.1 数据集介绍	9
2.2 数据标注	11
2.3 数据处理	13
2.4 本章小结	15
3. 深度学习框架.....	17
3.1 本文实验环境介绍.....	17
3.2 搭建实验环境.....	18
3.2.1 安装 CUDA 和 cuDNN	18
3.2.2 安装 tensorflow-gpu.....	18
3.3 本章小结	19
4. 电气石分割的 BiSeNet 神经网络模型.....	21
4.1 图像分割的神经网络发展.....	21
4.1.1 LeNet5	21
4.1.2 AlexNet.....	22
4.1.3 VGG.....	22
4.1.4 GoogleNet.....	23
4.1.5 ResNet	23
4.2 卷积神经网络基本概念.....	25
4.2.1 前向传播	25
4.2.2 后向传播	25
4.2.3 梯度下降	26
4.2.4 卷积	26

4.2.5 激活函数.....	27
4.2.6 池化.....	29
4.2.7 Softmax 分类.....	30
4.3 基于 tensorflow 的 BiSeNet 模型实现	31
4.3.1 Context Path	34
4.3.2 Spatial Path	34
4.3.3 ARM	34
4.3.4 FFM	35
4.4 实验结果分析	35
4.4.1 模型训练	35
4.4.2 模型预测	37
4.5 本章小结	38
5. 总结与展望.....	39
5.1 本文总结	39
5.2 以后研究的工作	39
致谢.....	41
参考文献.....	43

1. 绪论

1.1 电气石地质显微图像分割的背景和意义

1.1.1 电气石地质背景

在地质学的研究过程中，找矿一直是一个重要且艰巨的任务。矿物的生成的特征一般不是很明显，这给研究人员带来了不少困难。但是通过研究发现有些矿物的周围常常会有一些其它种类岩石的出现，所以可以通过这些岩石间接的找到矿物的所在地。即如果在一个地方发现了大量的此类岩石，那么这个地方就有很大的概率有矿物存在。所以对于岩石种类的识别在地质学中意义重大。

电气石^[1]是一种常见的热液流体（岩浆热液）对应的矿物，它是在变质岩里面的，或者说广义的石变岩里面的。它是一种热液矿物，所以电气石里面有很多的卤族元素，尤其是氟，氢氧根，即指示了热液里含有很多的氟，而氟是一种非常强的阴离子，它能与很多的阳离子发生反应，有氟的存在，很多原本不动的金属阳离子都可以迁移了，典型的有金、锡、铀。往往有氟的存在都可以形成这些矿物，或者说不只是只有一种矿，而是有很多种矿，所以电气石是一种非常常见的找矿指示。电气石也非常稳定，形成以后在地表风化十万年依然完好无损，所以说电气石非常好识别，往往和石英长在一起。电气石有很多分类，颜色通常是黑的，常呈放射状，还有绿的、红的，比较好看的还有碧玺等。电气石在工业上还有一个作用：压电效应。电气石整体上是一个柱状的矿物，两头用机械一压就出现电压了。比如在电气石的两端用电表测量电压，可以发现电势差接近 0，但是用机械力压过之后再测量就会发现带电气石两端产生了电势差。这可以运用到电气工业中。

电气石具有以下几个特点：

(1)成分:钠、镁、铁、锂、锰的复杂硼铝硅酸盐。镁元素和钠元素是镁电气石组成中的最重要元素。在黑电气石中，钠和铁是主要的。在锂电气石中，钠和锂是主要的。

(2)形态:具有三角形或六方断面的长柱状晶体。横断裂开常常把柱面分为几个部分。

(3)鉴定:大多数电气石具有颜色和多色性。当晶体延长方向垂直于下尼科尔棱镜（下偏光镜）平面时具有最强吸收性。形态、平行消光、延长方向快光等特征也能帮助鉴定。大多数电气石的横截面由于强吸收而在薄片巾常是暗色的，因

此很难获得干涉图，所以对于初学者去认识它是很困难的。富含黑云母的岩石，如变质泥质岩和片岩中，电气石难以鉴定，但是可以根据它的低重折和不同的最强吸收面而和黑云母区别。薄片中的电气石是淡黄色到无色。黑电气石是褐到黑色，暗绿或蓝到黄、褐色或淡紫色。而锂电气石无色，粉红或淡绿或蓝。在大晶体中颜色常呈环带。

(4)产状:在片岩、变质泥岩和角闪岩中电气石是少量的，但却又是无所不在的组分。在伟晶岩中它常是显著的，也产于某些热液脉中。它作为碎屑颗粒而是沉积物中常见的且少量的组分。

1.1.2 选题意义

电气石具有压电效应^[2]，它具有最外层破裂的电子辐射，并产生氧离子，抗菌和除臭功能。因此被广泛应用于工业与生活中的各种领域。正确高效的识别电气石不仅是专家们需攻克的难题，也需要我们相关专业研究者的助力。用深度学习的技术对电气石这种高价值的岩石进行识别分类，不仅仅提高自身对所学知识的应用能力，而且具有非常高实用价值。本文也参考了近期来的相关文章^[3]并对其做出了总结。

另外采矿专业是我校的特色专业，如图 1-1 所示，通过这次机会与我校特长专业进行交叉学科的研究，不仅仅拓展自己的知识面，而且将深度学习技术带入实际的应用场景，具有非常大的现实意义。



图 1-1 我校资环学院

1.2 神经网络介绍

1.2.1 人工智能的发展

我们正处于一个巨大变化的时代。人工智能^[4]已成为这个时代的主题。现在的人工智能就像 20 世纪的计算机一样，是下次工业革命的源动力。中国和主要发达国家正在积极分配人工智能作为新工业革命的触发点，增加科技投入，加快工业化进程。人工智能出现后，无论是身体还是心理上的重复性工作都可能慢慢消失。人工智能在逻辑推理理论和专家系统方面经历了两次繁荣，两次寒冷的冬天(如图 1-2 所示)。

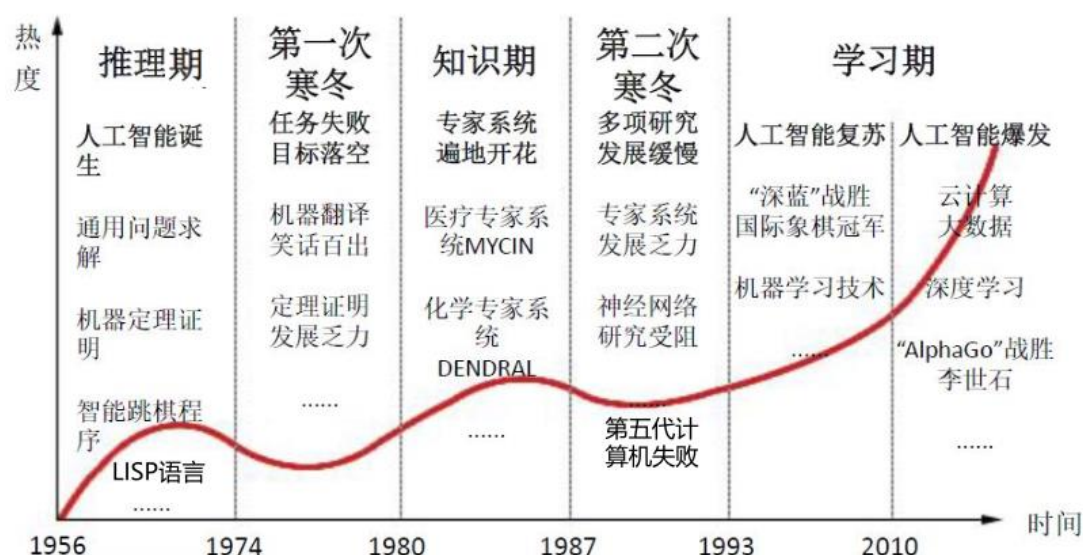


图 1-2 人工智能的三次浪潮

在过去十年中，人工智能技术再次经历了前所未有的爆炸性增长和繁荣。机器学习技术的兴起，特别是深度学习技术。以互联网为代表，国内外企业都设立了以人工智能为重点的研发部门。在人工智能浪潮的推动下，学术界和工业界密切合作，逐步向各行各业发展科技成果。

这一轮人工智能的快速发展离不开三个关键要素。

首先，高速硬件是深度学习的基础。目前，深度学习算法^[5]模型结构复杂，规模大，参数多，模型的开发门槛很高。面对这些挑战，高性能硬件，如 GPU，FPGA，ASIC^[6]等，特别是人工智能应用专用芯片的开发，已经多次提高了机器学习的培训速度。

其次，数据是推动人工智能发展的动力。机器学习^[7]技术需要大量的标签数据培训，在海量数据样本的基础上挖掘信息，并获得有用的知识。互联网和 LoT 物联网的发展不仅允许移动电话等移动设备访问互联网。云是人工智能的容器。如果我们将人工智能与电力进行比较，云就是一个非常强大的发电站，为各行各业提供持续的智能核电。

第三，不断开发新的人工智能算法，从卷积网络^[8]（CNN）到递归神经网络（RNIN），从生成性对抗网络（GAN）到迁移学习。每种新算法的引入或改进都带来了应用效果的显著改善。人工智能技术在语音，图像，自然语言等诸多领域的应用促进了人工智能潮流的不断发展。

当今人工智能的发展中最受欢迎的技术当属深度学习，此次人工智能的崛起最终要看深度学习的发展，可以说深度学习是本轮人工智能崛起的核心。

1.2.2 深度学习的崛起

在深度学习崛起之前,神经网络曾几经波折,经历了两个低谷,三个阶段。

(1)神经网络的第一次高潮

神经网络的第一个高潮是由感知器^[9]带来的。感知器是一种由算法构造的“分类器”。它是一种线性分类模型。原则是通过不断寻找合适的表面来分离证据。

1958 年，罗森布拉特在“纽约时报”“电子‘脑’教导自己”中发表了一篇文章。正式将该算法命名为“perceptor”。在介绍感知器之后，Roseblat 对此非常有信心。他乐观地认为这种感觉最终可以“学习，做出决定，翻译语言”。主要投资机构也为他注入资金。美国海军训练营为他提供资金，并期望感知机可以“去，说，读，读，自我复制，甚至有自我意识”。这可以被认为是神经网络研究的起源和第一次高潮。

(2)神经网络的第一个冬天

虽然单层感知器简单而优雅，但它显然功能有限，只能对线性问题进行分类，无法要求 XOR^[10]。什么是线性问题？简单地说，一条线用于将图形分为两类。之后 Marvin Minsky 分析并直接指出大多数研究感知器是没有价值的。“此时，距离传感器^[9]已经炙热了十年，人们的高期望与感知器的能力不一致。单层感知器完全失去了人们对这次攻击的追求。很明显，单层感知器的问题是无法解决非线性问题，人们试图通过添加隐藏层来创建多层感知器，这是最简单的前馈神经网络。

对多层感知器的研究表明，随着隐藏层中层数的增加，该区域可以形成任意形状，从而解决任何复杂的分类问题，但是如何控制隐藏层的重量？对于每个层

的节点，它们没有预期的输出，因此它们无法传递感知。机器的学习环境训练多层感知器，人们还没有找到可靠的学习算法来解决这个问题。Marvin Minsky 对感知机器的批评无法突破瓶颈，使得人工神经网络的发展成为人们的第一个低谷。

(3)神经网络的第二个高潮

在提出反向传播算法（BP 算法）^[11]之前，它确实解决了感知器的局限性，并再次将神经网络培养到了高潮。当神经网络进入第一个低谷时，Geoffrey Hinton 刚获得了心理学学士学位，并准备毕业。随着案例科学的发展，1982 年，总部位于加利福尼亚的生物物理学家 John Hopfield 开发了一种反馈神经网络（Hopfield Network），这是一个解决了识别和约束优化问题的网络，并激发了神经网络领域的研究人员。之后随着 XOR 问题的解决，计算机速度的提高，神经网络又得到了发展。

(4)神经网络的第二个冬天

神经网络的第二次高潮持续了很长一段时间，在此期间，研究人员继续探索 BP 网络的应用，在此期间，Yann Lecun 开发出实用的 LeNet 运用于邮票手写数字的识别，让神经网络首次出现在实用的生活场景中。后来，他提出了第一个真正的基于 BP 算法的深度学习。它也是深度学习卷积神经网络（CNN）中应用最广泛的神经网络结构，并开发了商业银行支票阅读软件。在手写体数字方面，该校验识别系统在 20 世纪 90 年代末占据了美国市场近 20% 的份额，虽然 BP 算法将神经网络引入了实际应用阶段，但在当时的神经网络中仍然存在许多缺陷。第一个是浅极限问题。发现神经网络中的参数越远离输出层，训练难度越大，层数越多，问题就越明显，称为“梯度爆炸”问题。此外，在当时计算资源不足的情况下，数据集非常小，无法满足深网训练的要求。随着神经网络发展速度的减慢，传统的机器学习算法也取得了突破。除了基本的线性分类外，该分类算法还可以利用“核机制”非线性映射算法将线性不可分割样本转换为高级特征空间，使数据样本线性不可分割。样品可以分割。1998 年，该算法在手写邮政编码问题上的误码率降至 0.8% 以下，大大超过了同期神经网络算法的性能，并迅速成为研究的主流。与支持向量机算法相比，神经网络的理论基础不明确，缺点更加突出。这样，神经网络进入第二个冬季。

(5)深度学习的到来

即使神经网络再次进入冬季，社会似乎已经失去了对这一领域的耐心。投资公司已将注意力转移到其他领域，甚至与神经网络相关的文章也一再被拒绝，但 Hinton 和其他人仍然不放弃。直到 2006 年，Hinton 发表了一篇突破性文章，一篇关于 Deep Belief Nets 的 A Fast Learning Algorithm 算法，其中 Hinton 介绍了一种成功的

方法来训练多层神经网络，他称之为深层信念网络。一旦深层信念网络启动，它就会立即击败 SVM^[12]，这使得许多研究人员将注意力转回神经网络。深度学习的诞生伴随着更优化的算法，更高性能的计算能力（GPU）和更大数据集的背景，一旦出现就引起了极大的轰动。首先，我们应该提到算法的优化。从 Hinton 2006 年提出的深层信念网络成功培养多层神经网络开始，后来的研究人员不断在这一领域进行创新，提出了越来越多的优秀模型，并将其应用于各种场景。深度学习兴起的另一个条件是强大的计算能力的出现。

CPU 集群^[13]是人们之前在高性能计算中可以想到的。现在 GPU 用于深入研究。使用 GPU 集群可以加速只能在一个月到几个小时内完成培训的网络。时间大大缩短，这使得研究人员培养了大量的网络。除了硬件的快速发展，深度学习也得到了充分的推动：大数据。与传统的神经网络相比，虽然我们确实在算法中对深层架构^[14]的训练进行了改进，但最重要的进展是我们拥有成功训练这些算法所需的资源。可以说只有数据驱动才能实现 AI 的深度学习，迭代模型，变得越来越智能化。

因此，算法，硬件和大数据是必不可少的。我们不能忽视这一点。如今，深度学习在许多领域都表现出色。 本论文主要研究深度学习在图像方面的应用。

1.3 论文研究的主要内容

在论文的第一章中，我们介绍了电气石的地质背景知识以及选题意义，还列举了人工智能与深度学习的曲折发展历程，分析了为何如今深度学习会如此火爆以及介绍了深度学习可以应用的领域，从交叉学科的角度阐述了用深度学习的方法分割电气石地质显微图像的意义。

在论文的第二章中，我们介绍了数据集的准备、标注和处理，重点介绍了标注方法以及数据处理过程中的数据增强，从数据集处理初期阶段和与其他专业数据集对比的角度对实验结果不理想的原因做出了分析。

在论文的第三章中，我们对实验所需的环境进行了搭建，由于不同的系统，不同的软件版本都会导致安装的不成功，所以本章的内容是在作者电脑上安装成功的，我对自己安装成功的版本做了较为详细的阐述，对于自己安装过程中的失败原因和常见安装错误做了总结。

在论文的第四章中，我们从 imagenet 比赛^[15]开始介绍，简介了近几年深度学习在图像处理方面的发展，之后对图像处理中最常用的网络结构卷积神经网络的基本概念作了介绍，之后对本文所用的 BiSeNet 模型的原理及基于 tensorflow

的搭建做了重点说明，较为清晰的介绍了本文所用的技术，最后对实验结果进行了分析。

在论文的第五章中，我从本文的研究方向出发，基于电气石的实用场景，对下一步研究的方向作了分析和展望。下一步工作更加注重电气石的实际用途，综合了各个领域，各个专业，也更加能够体现深度学习对图像处理方面的作用。

2. 电气石数据集的制作

2.1 数据集介绍

原数据是在 Leica 电子显微镜^[16]下（如图 2-1 所示），观察电气石薄片样本,如图 2-2 所示，进行图像的采集。采集到的原图片总共 108 张，图片格式为 jpg，分辨率为 2560*1920，位深度为 24，如图 2-3 所示。



图 2-1 Leica DM2700 P 偏光显微镜



图 2-2 岩石薄片样本

属性	值
版权	
图像	
图像 ID	
分辨率	2560 x 1920
宽度	2560 像素
高度	1920 像素
水平分辨率	300 dpi
垂直分辨率	300 dpi
位深度	24
压缩	
分辨率单位	

图 2-3 原图片详细信息

由于原图片较大而且目标物体在原图片中所占比例较小(如图 2-4 所示),所以第一次时实验效果不是很好,在之后的原图片处理中我对原图片进行了截图处理将 108 张原图片截成 200 张 jpg 格式的目标物占比较大的图片,如 2-5 所示。



图 2-4 显微镜采集到的原图片

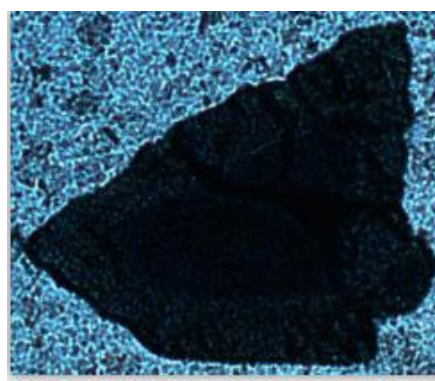


图 2-5 截图后的图片

本文训练神经网络需要提供三中图片:

1.jpg 原图片:即将截图后的图片统一处理成分辨率为 600*600,水平、垂直分辨率均为 96dpi, 位深度为 24 的图片,命名为 000001.jpg-000200.jpg。如图 2-6 所示。

2.png 原图片:将 jpg 原图片转换为 png 格式,命名为 000001.png-000200.png。如图 2-7 所示。

3. png 填充图:用 photoshop 对 png 原图片进行填充后的图片, 命名为 0000 01.png-000200.png。因为图片中只含两类像素一类是电气石, 一类是背景, 所以 png 填充图中只含有两种颜色, 背景颜色为红色, 前景颜色为绿色。如图 2-8 所示。



图 2-6 显微图像截图后的详细信息

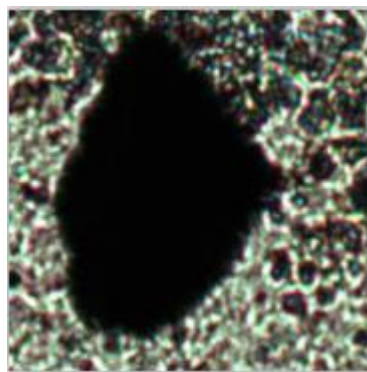


图 2-7 转换格式后的图片

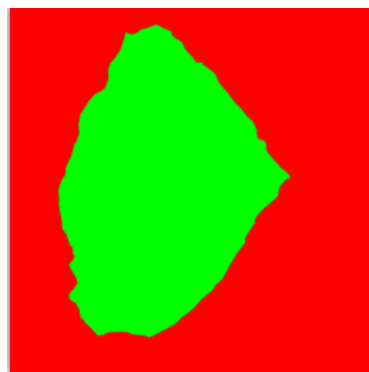


图 2-8 填充图

2.2 数据标注

用 PS 对 jpg 原图进行标记, 标记工具为钢笔, 标记好之后默认保存路径, 如图 2-9 所示。对 200 张 jpg 原图片进行标注之后用 photoshop 的动作命令建立一个把背景填充为红色 (R:255,G:0,B:0), 电气石填充为绿色(R:0,G:255,B:0), 把填充完之后的保存为 png 图像的动作, 如图 2-10。之后对标注后的 jpg 原图进行批处理操作, 如图 2-11。生成的 png 填充图片如图 2-12 所示。

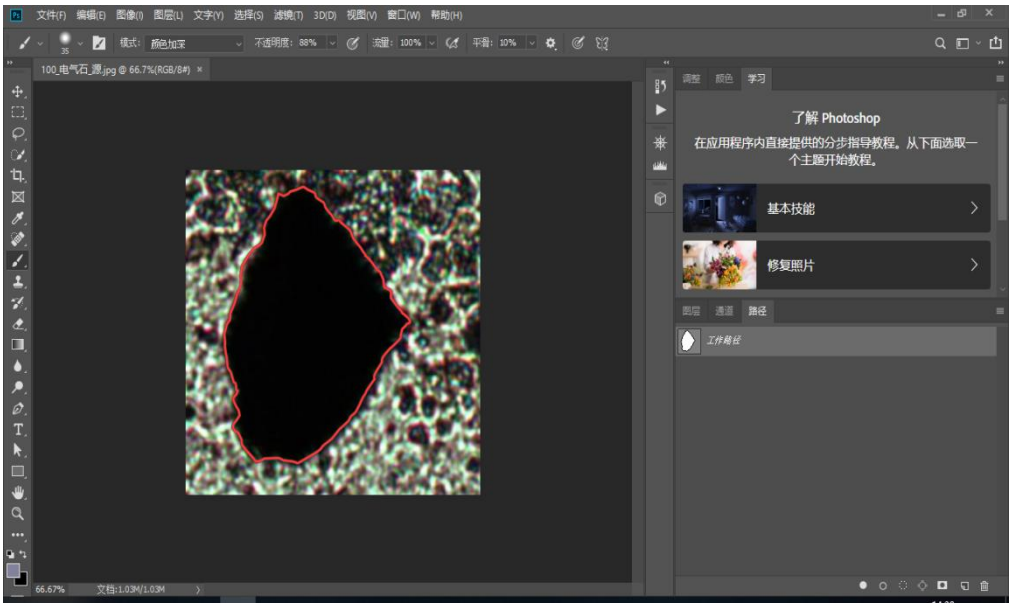


图 2-9 用钢笔工具对原图进行标注



图 2-10 建立填充颜色的并导出图片的动作

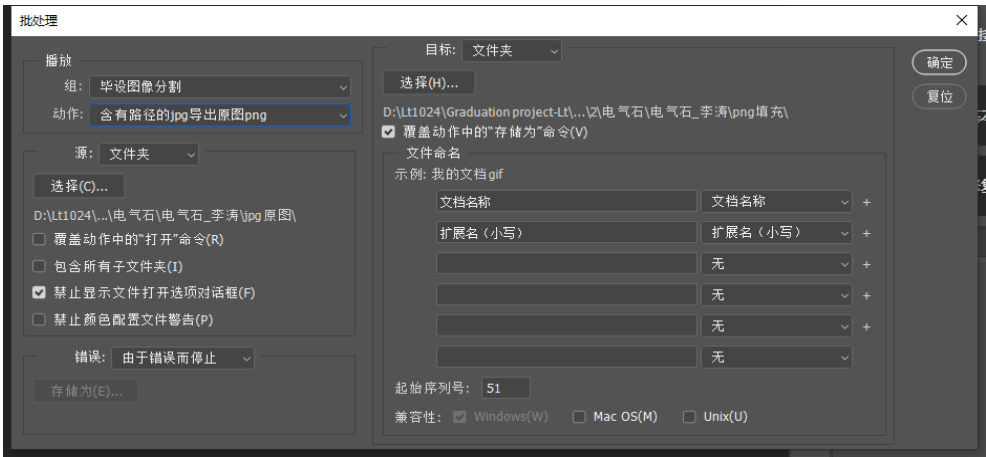


图 2-11 对标注图片进行批处理操作生成相应的填充图

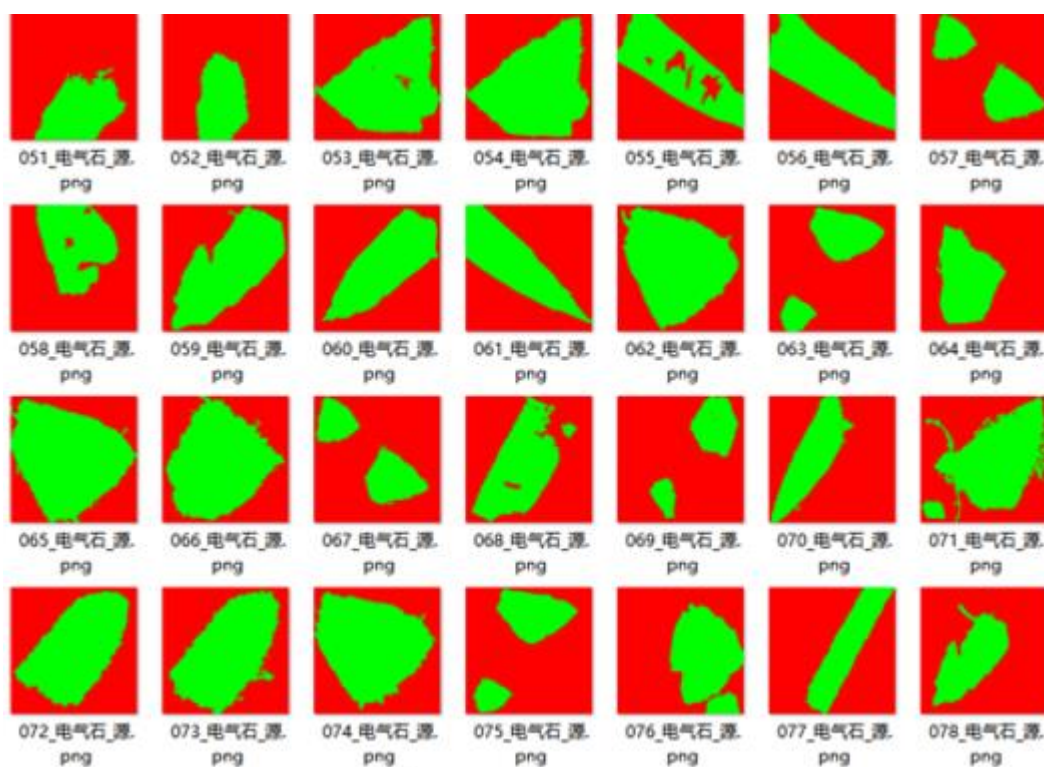


图 2-12 批处理之后生成的填充图片

2.3 数据处理

为了解决显微镜采集到的原图片目标物占比太小，我采取截图的方法来提高目标物在图片中的占比，大大提高了实验结果的精确度。

在前几次实验之中我发现原图片数量并不多，训练轮数也不大，学习率也么问题，但是训练时间太长了，经过和其他成功的实验案例相比较之后我发现是原图片太大的问题(2560*1920)，截图的方法减小了图片大小，但是截图之后的图片大小各异，为了兼顾训练速度和模型输入图片大小(本文采用的神经网络模型输入图片大小必须大于512*512),所以统一将截图处理为600*600的jpg格式图片。

经过上述处理之后图片总共有200张，这样的数据集大小远远不够，所以我采用常用的数据增强方法对这200张图片进行了处理，形成了最终的数据集，大小为800张。数据增强^[16]主要用来防止过拟合，用于dataset较小的时候。数据增强方法有以下几种方法：

(1) 随机旋转

随机旋转^[17]一般情况下是对输入图像随机旋转[0,360)

(2) 随机裁剪

随机裁剪是对输入图像随机切割掉一部分

(3) 色彩抖动

色彩抖动^[18]指的是在颜色空间如 RGB 中，每个通道随机抖动一定的程度。在实际的使用中，该方法不常用，在很多场景下反而会使实验结果变差

(4) 高斯噪声

是指在图像中随机加入少量的噪声。该方法对防止过拟合比较有效，这会让神经网络不能拟合输入图像的所有特征

(5) 水平翻转

(6) 竖直翻转

因为在神经网络模型之中已经含有随机旋转和裁剪的数据增强功能，所以对数据集的初期处理中采用以下三种方法进行处理：对 jpg 原图进行水平翻转，垂直翻转，水平翻转+垂直翻转，每一个处理都可以增加 200 张图片，值得注意的是对相应的 png 原图和 png 填充图也进行相应的翻转处理，因为之前的命名都是一一对应的，所以翻转处理之后的图片也是一一对应的，处理之后再按神经网络模型的命名规则统一命名即可。算法的核心思想是遍历文件夹的每一个图片，将每张图片保存为 PIL 模块中的 Image 对象，然后通过 Image 对象的内置方法 transpose 方法对图像进行翻转处理，然后对翻转处理后的图片进行重命名，之后将其保存到特定文件夹，其中的操作涉及 os 模块，水平和垂直翻转的具体实现代码如下，其它操作相似：

```
for filename in filenames:
```

```

    im = Image.open(current_path)
    current_path = os.path.join(parent, filename)
    tmp = im.transpose(Image.FLIP_LEFT_RIGHT)
    out = tmp.transpose(Image.FLIP_TOP_BOTTOM)
    zero_fill = str(rename_num).zfill(fill_rename_num)
    rename = op_all + f'\{zero_fill}' + suffix + file_type
    out.save(rename)
    print(rename_num, end=" ")
    rename_num = rename_num + 1

```

```
print("水平和垂直翻转完成！")
```

2.4 本章小结

本章重点介绍了数据集是如何从最原始的状态一步步变成实验所需要的格式的，数据集的构建不仅仅要求实验者对 photoshop 等图片处理软件有较为熟练的掌握，而且要求实验者对其他成功案例有较为清晰的了解，以及时分析实验中所出现的错误原因。总之，数据集的构建极为关键，它对最终的实验结果有着最直接的影响。

3. 深度学习框架

3.1 本文实验环境介绍

如今支持深度学习的框架有很多，其中最流行的当属 tensorflow^[19](如图 3-1 所示)，它是深度学习最流行的库之一。Tensorflow 是所有深度学习框架中最适合用来进行工程运用的。所以选择 Tensorflow 是必要的。

对于 tensorflow 框架 CPU 版与 GPU 版，首先要看自己的机器是否支持 GPU 加速，如果不支持的话，那么只能安装 CPU 版本的了，如果支持 GPU 版本的话，一定要安装 GPU 版本的。对于小型的网络结构还感受不到太大的差别，但是如果网络结构稍微大一点的话就会明显感受到他们之间的速度差异，特别对于一些大型的网络，使用 GPU 加速是必需的。

以下是我的环境配置选择：

训练机器的配置是内存为 12G,显卡为 1060Ti,系统版本为 win10,python 版本为 3.6, tensorflow-gpu 版本为 1.10.0,cuda 版本为 9.0, cudnn 版本为 7.5。

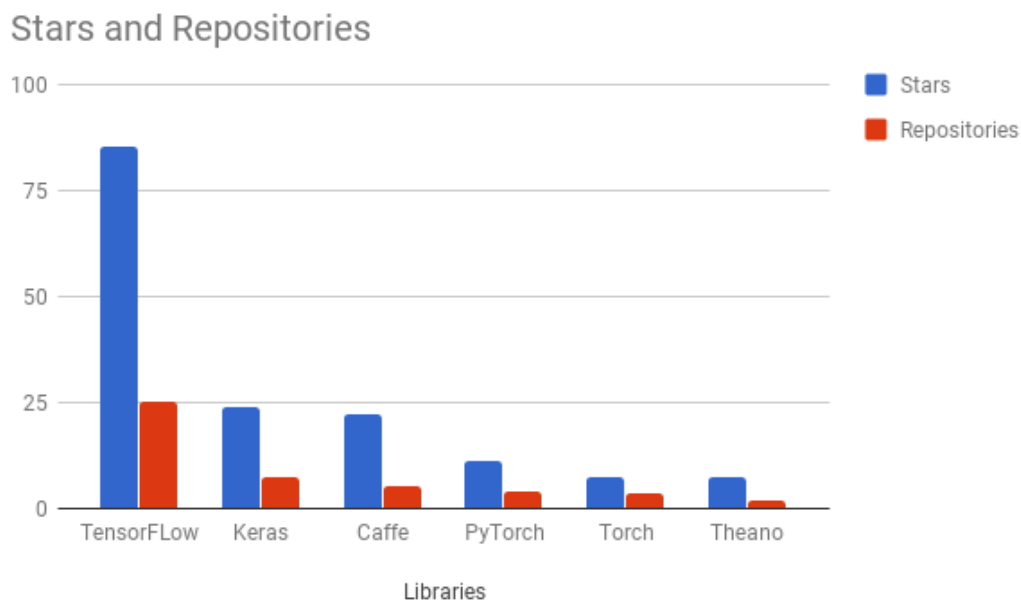


图 3-1 Github 上 tensorflow 与其他框架 Stars 和 Repositories 的对比

3.2 搭建实验环境

3.2.1 安装 CUDA 和 cuDNN

由于 CUDA Toolkit^[20]需要在指定版本显卡驱动环境下才能正常使用的，所以如果我们已经安装了 nvidia 显卡驱动（很显然，大部分人都安装了），再安装 CUDA Toolkit 时，会因二者版本不兼容而导致 CUDA 无法正常使用，这也就是很多人安装失败的原因。而 CUDA Toolkit 安装包中自带与之匹配的显卡驱动，所以务必要删除电脑先前的显卡驱动。

到官网下载好 CUDA Toolkit 9.0 安装后，我们开始下载 cuDnn 7.0^[21]，需要注意的是，下载 cuDNN 需要在 nvidia^[22]上注册账号，使用邮箱注册就可以，免费的。登陆账号后才能下载。接下来，解压，将解压后的三个文件夹，拷贝到 CUDA 安装的根目录下这样 CUDA Toolkit 9.0 和 cuDnn 7.0 就已经安装了，下面要进行环境变量的配置。

将下面四个路径加入到环境变量中，注意要换成自己的安装路径。

C:\Software__Study__Other\NVIDIA GPU Computing Toolkit\CUDA\v9.0

C:\Software__Study__Other\NVIDIA GPU Computing Toolkit\CUDA\v9.0\
bin

C:\Software__Study__Other\NVIDIA GPU Computing Toolkit\CUDA\v9.0\
lib\x64

C:\Software__Study__Other\NVIDIA GPU Computing Toolkit\CUDA\v9.0\
libnvvp

下面开始进行 tensorflow-gpu 的安装。

3.2.2 安装 tensorflow-gpu

百度搜索 Anaconda 下载最新版本安装。如图 3-2 所示，以管理员的方式打开 Anaconda Prompt，输入 `conda create -n tensorflow-gpu pip python=3.6` 创建一个名为“tensorflow-gpu”，python 版本为 3.6 的 conda 环境，通过以下命令 `activate tensorflow-gpu` 激活 conda 环境，然后输入 `pip install tensorflow-gpu==1.10.0` 安装 tensorflow-gpu，等待安装成功后输入 `python` 回车后输入 `import tensorflow` 等待数秒后不报错即安装成功如图 3-3 所示。

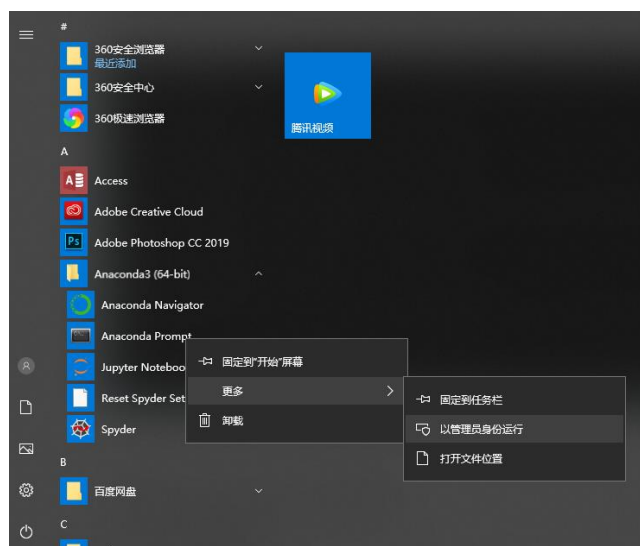


图 3-2 打开 Anaconda Prompt

```
(base) C:\Windows\system32>activate tensorflow-gpu
(tensorflow-gpu) C:\Windows\system32>python
Python 3.6.2 |Continuum Analytics, Inc.| (default, Jul 20 2017, 12:30:02) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>> import tensorflow
>>>
```

图 3-3 检测 tensorflow-gpu 是否安装成功

3.3 本章小结

本章在自己电脑上搭建了 tensorflow-gpu 版本的环境，对 tensorflow 进行了简介，说明了为什么选择 tensorflow 而不选择其他的框架，也说明了 tensorflow 的 cpu 版和 gpu 版本的差别，最后通过一步步安装说明步骤较为清晰地展现了作者搭建环境时的基本流程。本章基本上都是实践操作，而且因为深度学习框架发展还不是很成熟，所以安装过程中可能出现各式各样的问题，作者也在本章中对安装容易出错的地方做出了总结。通过本章实验环境的成功搭建，可以为后续实验提供简单的操作，更加方便实验人员进行实验，对所出现的错误也能更加快速的找到出错原因，节省了研究人员的宝贵时间。

4. 电气石分割的 BiSeNet 神经网络模型

4.1 图像分割的神经网络发展

LeNet5 是最早出现的卷积神经网络，虽然在手写数字识别上取得一定的成果。但是受限与当时的计算能力，所以网络的深度只有五层，而且输入图片比较小，无法像更深、更广的网络结构发展，直到 2012 年上 ImageNet 大赛的冠军 AlexNet 出现之后，才又点燃了卷积神经网络在图像方面的应用，之后 ImageNet 的参赛者都开始使用卷积神经网络，得益于计算机硬件的发展,网络结构可以设计的更加复杂多元化，因此进行图像分类的效果也越来越好，每一年都在刷新着记录，直到 2017 年比赛效果太好（如图 4-1 所示），图像分类的错误率已经非常低，所以 ImageNet 比赛正式取消，回首再看运用卷积神经网络参加 ImageNet 比赛取得了巨大的成功，其中不乏出现十分经典的网络结构，下面对介绍几种常见的结构。

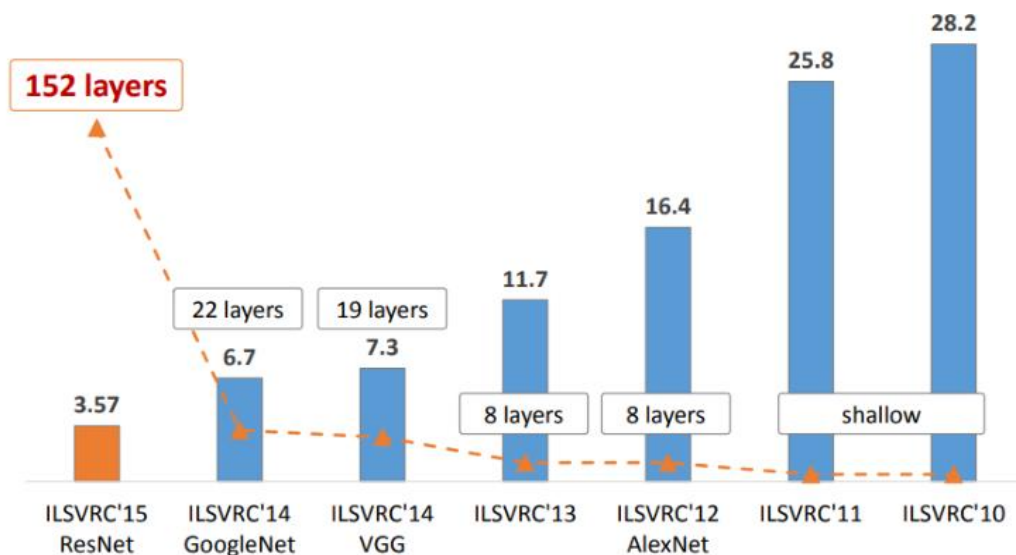


图 4-1 ImageNet 图像分类错误率（top-5）

4.1.1 LeNet5

LeNet5 是最早出现的 CNN。LeNet5 可以说是近年来大量网络架构的起源，为现代深度学习领域的发展做了重要铺垫。其中 5 代表五层模型，其网络结构如图 4-2 所示

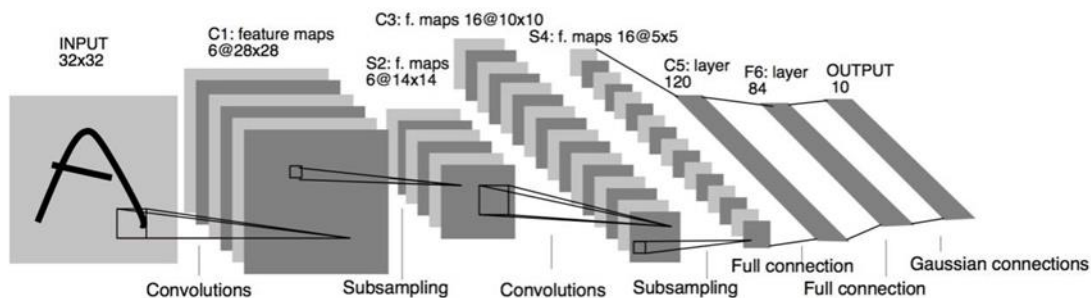


图 4-2 Lenet5 结构

4.1.2 AlexNet

AlexNet 是以其作者 Alex Krizhevsky 命名的网络架构。AlexNet 网络结构设计如图 4-3 所示。

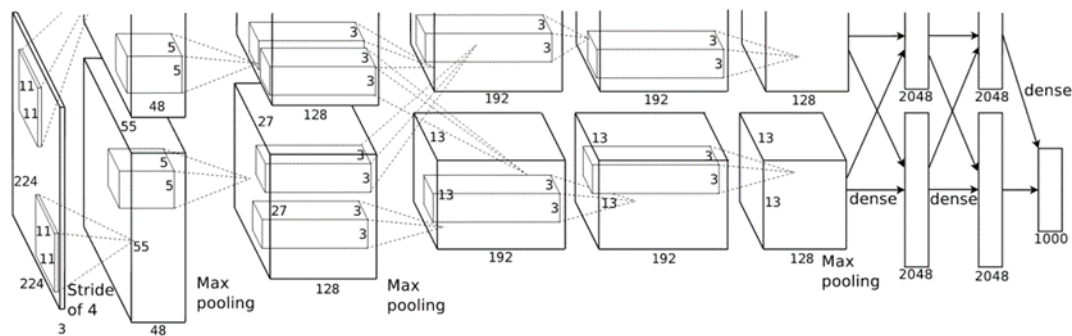


图 4-3 AlexNet 结构图

4.1.3 VGG

作为 2014 年 ILSVRC 挑战的亚军，来自牛津大学的 VGG(Visual Geometry Group,牛津大学计算机视觉组)网络很好地继承了 AlexNet 的衣钵，意在使用更深的网络来获取更好的训练效果，其结构如图 4-4 所示。

不同于 LeNet5 和 AlexNet，VGG 使用的 3x3 滤波器变得更小。这看似离脱离了 LeNet5 的设计初衷，反而接近 LeNet5 竭力避免的 1x1 卷积。实际上，VGG 通过依次采用多个 3x3 卷积，能够达到与更大的感受野(如 5x5 与 7x7)类似的效果，以提取更多复杂特征以及这些特征的组合。这样的思想后来被许多新生网络采纳，如 ResNet。

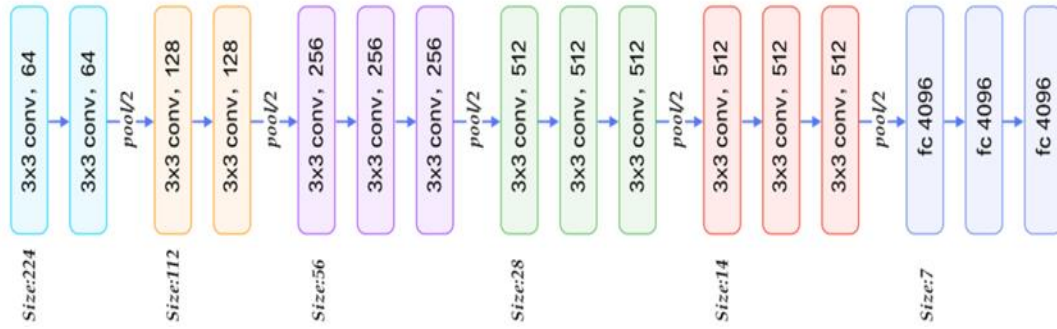


图 4-4 VGG 结构图

4.1.4 GoogleNet

以机构命名的 GoogLeNet 网络是 2014 年 ImageNet 挑战的冠军。GoogLeNet 充分利用了硬件的优势，扩大了网络结构的深度和广度，借鉴了 NIN(Network in Network)的设计思想即设计将几个网络并联起来，或者利用其他的方法复杂化网络结构，提高网络的拟合能力。

Inception 模块如图 4-5 所示所示，(a)图为 Inception 降维模块，（b）为 Inception 含降维模块。

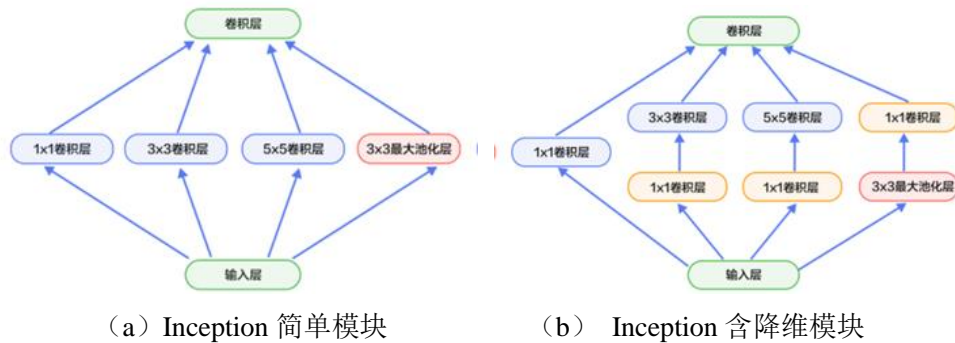


图 4-5 Inception 模块

4.1.5 ResNet

ResNet 是深度卷积网络的又一次飞跃，它主要通过残差模块的引入解决了卷积网络不能过深的问题。如图 4-6 所示，每个残差模块包含两条路径，其中一条路径的设计借鉴了 Highway Network 思想，相当于在旁侧专门开辟一个通道使得输入 x 可以直达输出；另一条路径对输入特征做 2 到 3 次卷积操作得到该特征对应的残差 $F(x)$ ；最后再将两条路径上的输出相加，即优化的目标由原来的拟合输出 $H(x)$ 变成输出和输入的差 $F(x)=H(x)-x$ 。残差模块这一设计将要解决的

问题由学习一个恒等变换转化为学习如何使 $F(x)=0$ 并使输出仍为 x ，使问题得到了简化。

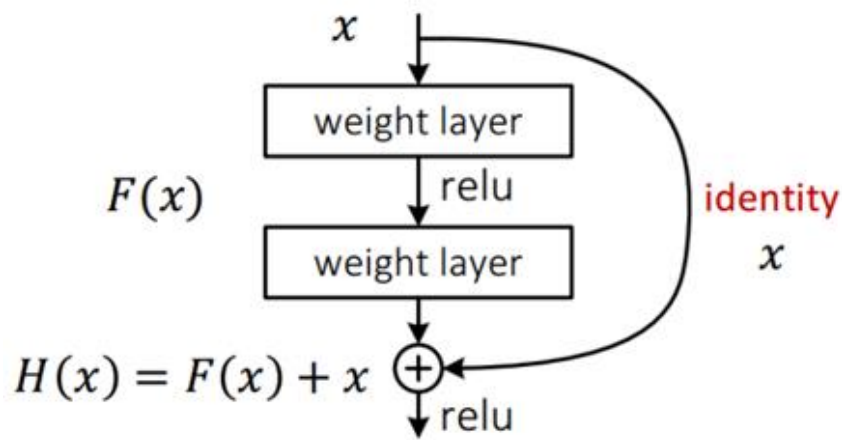


图 4-6 残差模块

残差模块的不同形式如图 4-7 所示，左侧是基本模块连接方式，由两个输出通道数相同的 3×3 卷积层组成。右侧是瓶颈模块(Bottleneck)连接方式。因为先使用了 1×1 的卷积层来对输入进行降维(对应图中示例由 256 维下降至 64 维)，然后又使用 1×1 卷积层来对输入进行升维(对应图中示例由 64 维上升至 256 维)；如此一来，相比原始的输入和最终的输出，中间 3×3 卷积层的输入和输出通道数都较小(对应图中示例由 64 维至 64 维)，整体形似瓶颈，因此得名“瓶颈模块”。

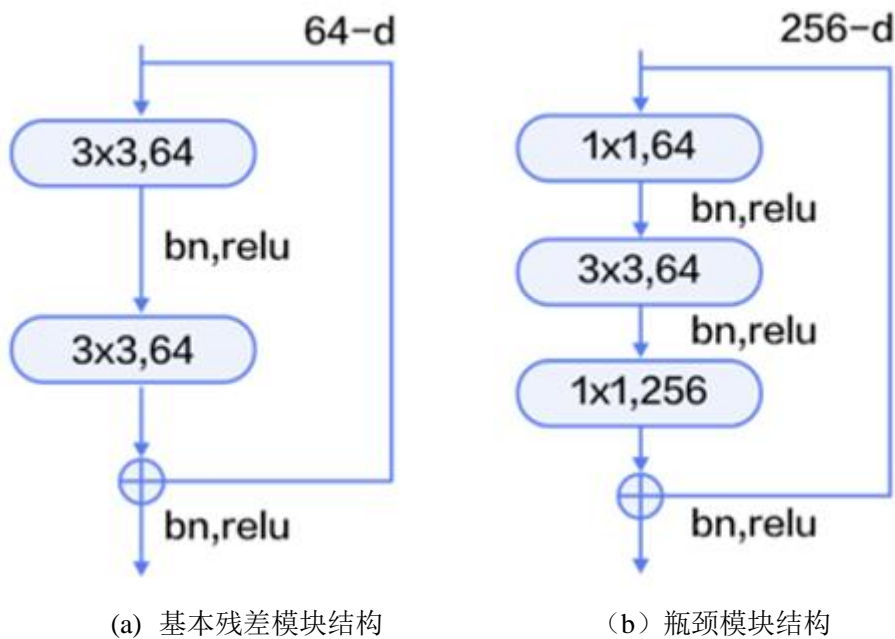


图 4-7 残差模块的不同实现形式

4.2 卷积神经网络基本概念

4.2.1 前向传播

前向传播其实就是神经网络的后一层根据前一层的输入和权重、偏置计算出后一层的值，然后通过激活函数激活，在将此层的输出作为下一层的输入继续计算下一层的输出值，直至计算出最出最后一层的输出，前向传播算法才结束。

从数学角度讲，深层网络的前向传播过程就是由输入向量 x 得到输出的函数计算过程。每一个节点实际上表示一个线性变换与一个非线性变换的复合。比如一个 3 层全连接神经网络的正向传播过程如图 4-8 所示：

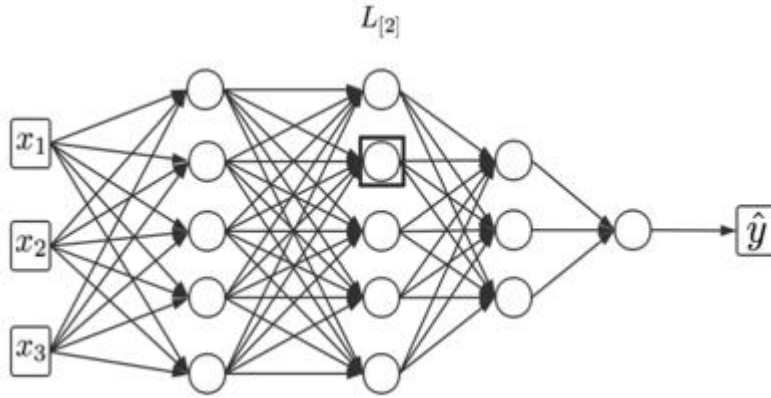


图 4-8 3 层全连接神经网络正向传播过程

4.2.2 后向传播

后向传播其实就是将前向传播中计算出来的误差反向逐层传播回去，即反向逐层计算出对于每个参数的“代价”，反向传播把代价函数关于每一个参数的偏导数计算出来之后就可以用梯度下降算法来更新参数，完成自动学习的效果了。在反向传播算法计算各个参数的偏导数的时候要用到高等数学中求导的链式法则。神经网络反向逐层求偏导的计算顺序与链式法则(如公式 4-1 所示)十分契合，只需要将后一层的偏导数作为输入传入前一层，前一层根据激活函数等求出本层各个参数的偏导数用作参数的更新，同时将有用的偏导数传给更前一层，直到计算出第一层参数的偏导数算法结束。反向方向传播的传播过程如图 4-9 所示。

$$\begin{aligned}
 \frac{\partial L}{\partial a_2^{[2]}} &= \frac{\partial L}{\partial a_1^{[4]}} \frac{\partial a_1^{[4]}}{\partial a_1^{[3]}} \frac{\partial a_1^{[3]}}{\partial a_2^{[2]}} + \frac{\partial L}{\partial a_1^{[4]}} \frac{\partial a_1^{[4]}}{\partial a_2^{[3]}} \frac{\partial a_2^{[3]}}{\partial a_2^{[2]}} + \frac{\partial L}{\partial a_1^{[4]}} \frac{\partial a_1^{[4]}}{\partial a_3^{[3]}} \frac{\partial a_3^{[3]}}{\partial a_2^{[2]}} \\
 &= \frac{\partial L}{\partial a_1^{[4]}} \left(\frac{\partial a_1^{[4]}}{\partial a_1^{[3]}} \frac{\partial a_1^{[3]}}{\partial a_2^{[2]}} + \frac{\partial a_1^{[4]}}{\partial a_2^{[3]}} \frac{\partial a_2^{[3]}}{\partial a_2^{[2]}} + \frac{\partial a_1^{[4]}}{\partial a_3^{[3]}} \frac{\partial a_3^{[3]}}{\partial a_2^{[2]}} \right)
 \end{aligned}
 \tag{4-1}$$

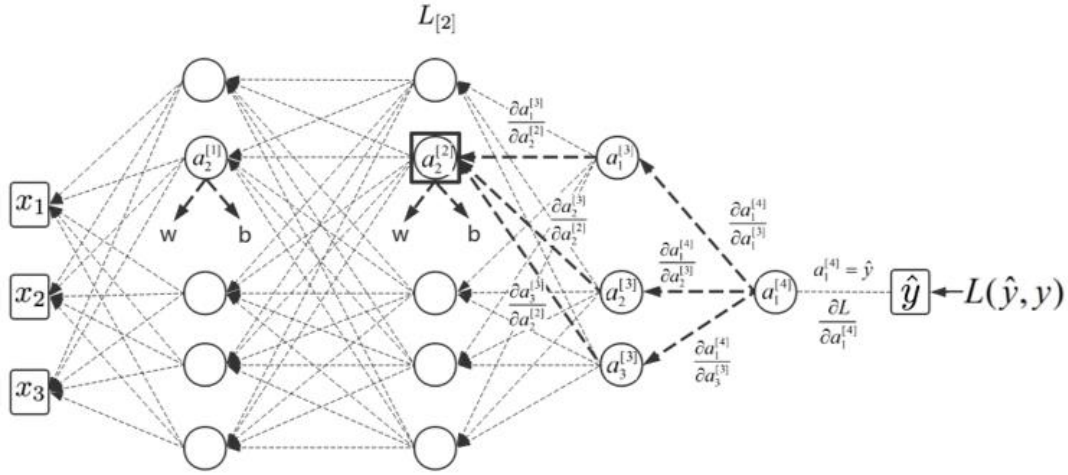


图 4-9 反向传播算法传播过程

4.2.3 梯度下降

梯度下降算法是深度学习之中最基础且最常用的参数更新算法，它的核心思想就是高等数学之中的函数的某一点沿梯度的反方向下降最快。我们将其运用到损失函数的优化之中，让损失函数沿梯度的反向更新参数，那么损失函数的优化速度就会相当迅速。我们可以借助深度学习框架自动完成反向传播和梯度下降以更新参数，利用深度学习框架大大减少了我们的工作量，可以有更多的时间来考虑更高层的问题，而不用考虑底层梯度更新的具体步骤。梯度下降算法的公式为：

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (4-2)$$

4.2.4 卷积

卷积操作是卷积神经网络中最重要的组成部分。在全连接神经网络中前一层中每个神经元与后一层中每个神经元都有一条边即一个参数，如果全连接网络的深度加大的话就会导致参数量过大而计算时间太长，而且通过研究发现在全连接神经网络中很多的参数都是用不到的。而卷积操作就可以避免这两个问题。首先，卷积操作是基于生物视觉神经网络中“感受野”概念在深度学习领域的推广，这能够很好地解释现实问题，生物在观察一个物体时对局部的特征比较感兴趣。另外在一个区域观察到的某种特征方法也同样适用于其他的区域，所以可以利用一个卷积核（滤波器）基于区域进行特征提取，不同的卷积核会提取到不同的特征。利用卷积操作不仅仅使神经网络的参数量大大减少而且训练的效果更好。在操作

上卷积就是对图像的局部区域与卷积核进行对应位置的相乘然后将乘得的结果相加，具体操作如图 4-10 所示。

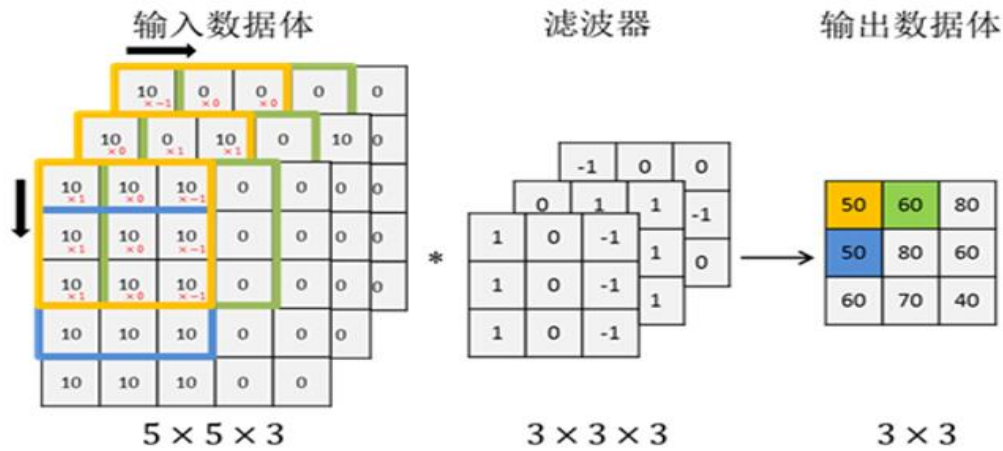


图 4-10 三维卷积操作示例

4.2.5 激活函数

激活函数的实际意义：生物神经网络中不是每个神经元都对和他相连的神经元传递信号，某些信号可能会被阻塞掉，即神经元的信号可能会传递给下个神经元，也可能不会被阻塞掉，控制这个信号是否被阻塞掉的开关在深度学习神经网络中就称为“激活函数”，即现实生活中的函数大多都不会是线性函数，如果只用线性函数来模拟现实生活中的各个变量之间的关系，那么这样明显就不符合现实意义，所以每一层神经元都通过一个激活函数来将线性函数转换为非线性函数。如图 4-11 所示。常见的激活函数有以下几种：

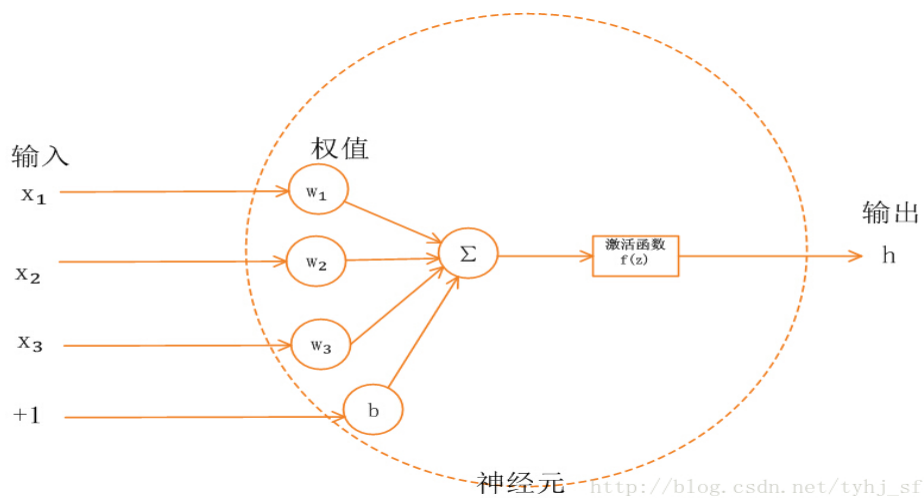


图 4-11 神经网络中的激活函数

(1) Sigmoid 函数

Sigmoid 函数数学公式为:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4-3)$$

Sigmoid 函数及其导函数图像如图 4-12 所示

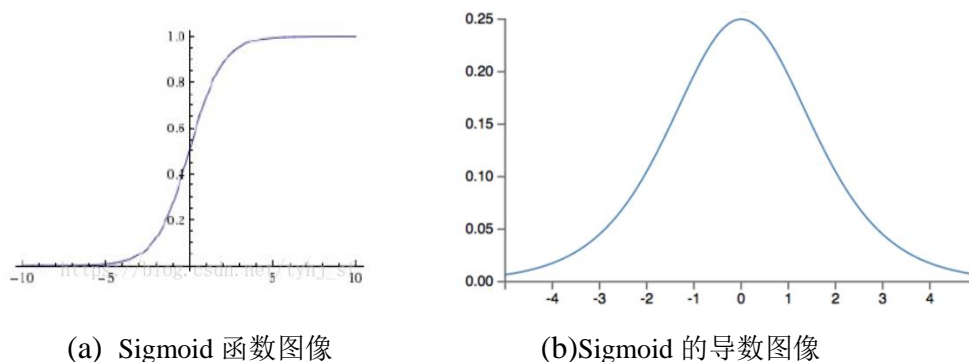


图 4-12 Sigmoid 及其导函数图像

(2) Tanh 函数

Tanh 函数数学公式:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4-4)$$

Tanh 函数及其导函数图像如图 4-13 所示, 由图可知当自变量在 0 附近时导数值较大, 但当自变量偏离零点较远时导数值接近于 0, 这一现象被称为“梯度消失”即当神经网络的层数比较多时, 根据链式法则多个比较小的导数值相乘的值越来越小, 导致传到浅层的梯度值过小从而学习速度太慢。

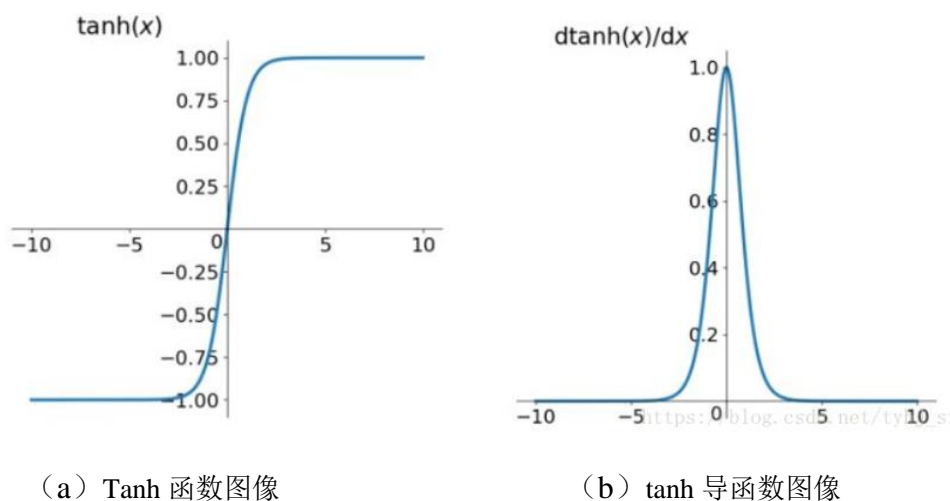


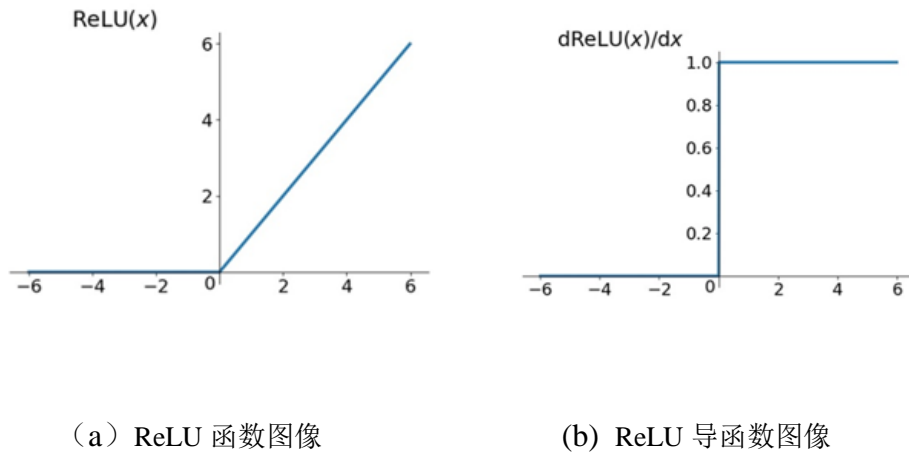
图 4-13 tanh 函数及其导函数图像

(3) ReLU 函数

ReLU 函数的数学公式：

$$Relu = \max(0, x) \quad (4-5)$$

ReLU 函数及其导函数的图像如图 4-14 所示。根据图像可知 ReLU 函数在自变量为负数的时候导数值为 0，当自变量为正时导数值恒为 1。这样就会使神经网络的正向传播非常容易计算，在正向传播时如果当前神经元的输出大于 0 那么就将其传递给下一层的神经元，如果小于 0 那么就不向后面的神经元传递值，而且在反向传播时不会出现梯度消失的问题（在正区间上），收敛速度是非迅速。



(a) ReLU 函数图像

(b) ReLU 导函数图像

图 4-14 ReLU 函数及其导函数图像

4.2.6 池化

池化的作用就是保留显著特征，增大卷积核的感受野。池化层可以大大减少神经网络的参数数量，但是也会一定程度上导致训练效果降低。常见的池化有最大池化和平均池化，最大池化就是将特征图的某一个区域的所有值的最大值作为输出（如图 4-15 所示），而平均池化就是把特征图特定区域的所有值的均值作为此区域的输出。在神经网络发展的早期阶段通常使用的是平均池化，而现在常用的池化为最大池化，因为研究者发现其在工作时表现较好。在本实验中的 BiSeNet 模型的 ARM 中和 FFM 模块中使用的另外一种池化方式为全局平均池化(Global average pooling, GAP) (如图 4-16 所示),每个特征图对应经过 GAP 输出一个值，它可以代替分类任务中的最后一个全连接层，使分类任务更加清晰，而且它不含参数，这样也对过拟合的问题有所缓解。

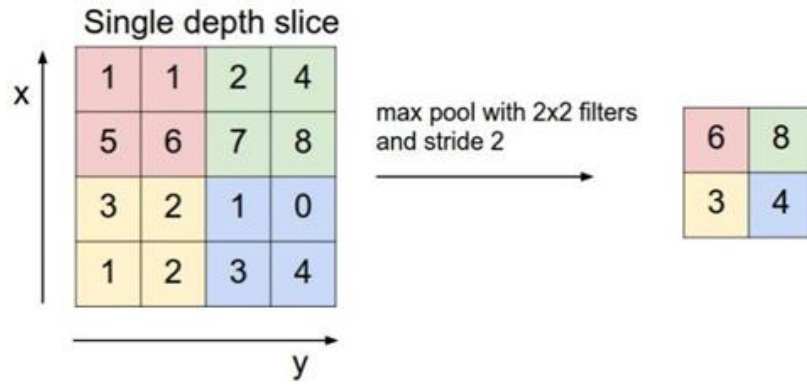


图 4-15 最大池化

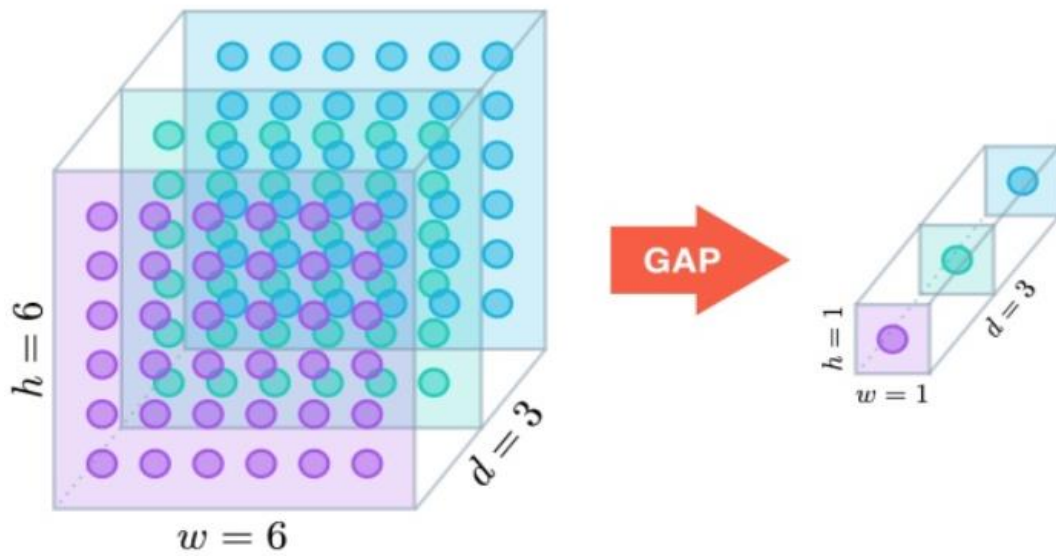


图 4-16 GAP全局平均池化

4.2.7 Softmax 分类

Softmax 用于分类，基本原理为数值越大概率越大，它从多个选项中选择出现概率最大的选项作为答案。它将各个选项的值映射为（0，1）的概率值，且所有选项之和为 1，其数学公式为：

$$\text{Softmax}(S_i) = \frac{e^{s_i}}{\sum_j e^{s_j}} \quad (4-6)$$

Softmax 层的具体计算如图 4-17 所示：

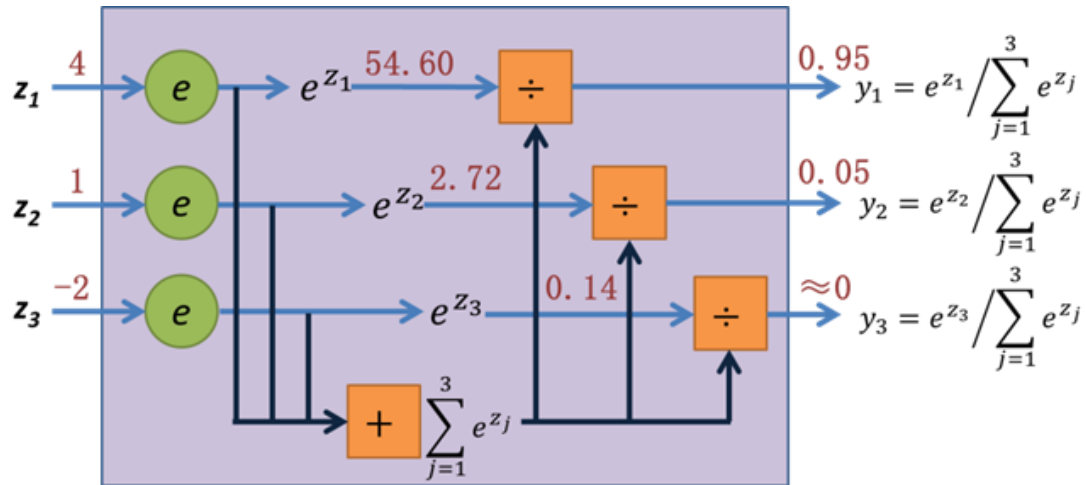


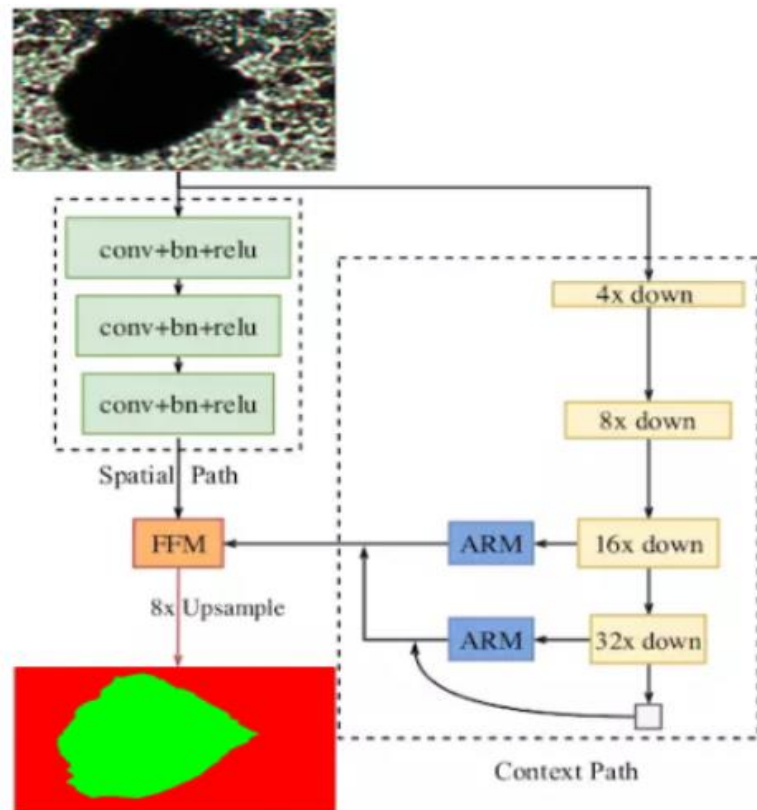
图 4-17 Softmax 层的计算过程

4.3 基于 tensorflow 的 BiSeNet 模型实现

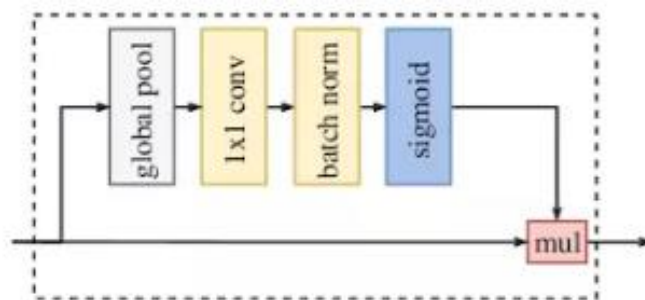
BiSeNet^[23]是旷世科技（如图 4-18 所示）在 2018 年提出的用于实时语义分割的神经网络结构。它包括两个部分：Spatial Path(SP)和 Context Path(CP)，它们解决了空间信息缺失与感受野缩小的问题，最后通过特征融合模块(Fusion Module/FFM)这两个部分提取的特征融合在一起。此外，还有另外一个注意力优化模块(Attention Refinement Module/ARM)，借助全局平均池化捕获全局并计算注意力以指导特征学习。本文对原网络结构做了一点改动，使用 ResNet152 作为 Context Path 来提取更深层的上下文信息，以获得更好的实验效果。剩下的模块 Spatial Path、ARM、FFM 模块与原网络结构相同。Spatial Path 通过三层卷积层来实现提取原图的空间信息。BiSeNet 的网络结构如图 4-19 所示：



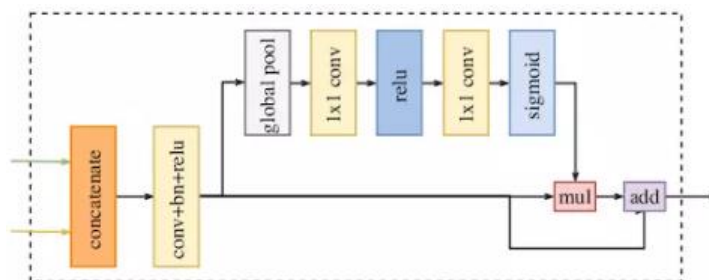
图 4-18 旷视科技



(a) BiSetNet 网络结构



(b) ARM 模块组成



(c) FFM 模块结构

图 4-19 BiSenet 网络结构

BiSeNet 的实现:

```
def build_bisenet(inputs,num_classes,preset_model='BiSeNet',frontend="Res
Net101", weight_decay=1e-5, is_training=True, pretrained_dir="models"):
    spatial_net = ConvBlock(inputs, n_filters=64, kernel_size=[3, 3], s
trides=2)
    spatial_net = ConvBlock(spatial_net, n_filters=128, kernel_size=[3,
3], strides=2)
    spatial_net = ConvBlock(spatial_net, n_filters=256, kernel_size=[3,
3], strides=2)
    ### Context path
    logits, end_points, frontend_scope, init_fn = frontend_builder.buil
d_frontend(inputs, frontend, pretrained_dir=pretrained_dir, is_training = is
_training)
    net_4 = AttentionRefinementModule(end_points['pool4'], n_filters=
512)
    net_5 = AttentionRefinementModule(end_points['pool5'], n_filters=
2048)
    global_channels = tf.reduce_mean(net_5, [1, 2], keep_dims=True)
    net_5_scaled = tf.multiply(global_channels, net_5)
    ### Combining the paths
    net_4 = Upsampling(net_4, scale=2)
    net_5_scaled = Upsampling(net_5_scaled, scale=4)
    context_net = tf.concat([net_4, net_5_scaled], axis=-1)
    # 在通道上面直接 concat, 然后卷积到 32 个通道
    net = FeatureFusionModule(input_1=spatial_net, input_2=context_n
et, n_filters=num_classes)
    ### Final upscaling and finish [1,64,64,32]-->[1,512,512,32] resi
ze
    net = Upsampling(net, scale=8)
    net = slim.conv2d(net, num_classes, [1, 1], activation_fn=None, s
cope='logits')
    return net, init_fn
```


4.3.1 Context Path

本论文中的 Context Path 采用 ResNet152 为前缀网络提取特征，ResNet152 网络结构如 4-20 所示。

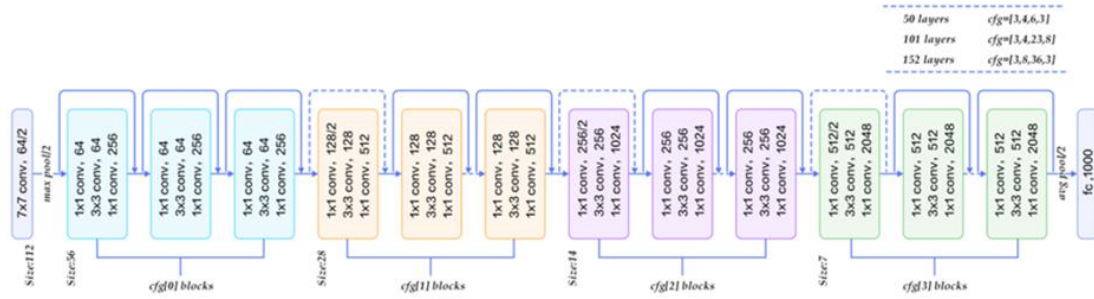


图 4-20 ResNet152 结构图

ResNet152 的网络层数为 152 层，除去开始的卷积核大小为 7*7 的卷积层和最后一层的全连接层，总共还有 150 层，分成 4 大块，每个块中含有的瓶颈模块个数分别为 3，8，36，3，每个瓶颈模块有 3 个卷积层，第一个卷积层和第三个卷积层的卷积核大小为[1,1],第二个卷积层的卷积核大小为[3,3],瓶颈模块的网络结构如图 4-7（b）所示。

4.3.2 Spatial Path

本文的 Spatial Path 与原结构相同为的是尽量保存原图像的空间信息，只用三个 3*3 的卷积，通道升为 256，利用底层的实现的 ConvBlock 函数（输入图像和卷积核的个数，尺寸，每次移动的步长等参数，输出卷积运算之后的图像）来实现。

4.3.3 ARM

ARM 又称为注意力优化模块，如图 4-19（b）所示，ARM 模块先通过 tensorflow 中的内置函数 reduce_mean 函数对输入图像做全局的平均池化，然后通过 slim 模块中的二维卷积函数 conv2d 对图像做 kernel_size 为[1,1]的卷积操作，之后再用 batch_norm 函数对图像做局部归一化操作，通过设置 fused 参数为 True 尽可能使用更快的融合方式，这步操作不仅加快了运算速度，而且提高了模型的泛化能力。然后再经过 Sigmoid 函数的激活，最后将激活后的输出和原始输入做元素级别的对应乘，使用的是内置函数 multiply，将结果返回。

4.3.4 FFM

特征融合模块，将 Spatial Path 与 Context Path 提取的特征图进行融合，组成如图 4-19 (c)，FFM 函数的参数有三个，一个是 Spatial Path 的输出，记作 input_1, 另一个是 Context Path 的输出，记作 input_2, 还有一个参数 n_filters 表示卷积核个数。FFM 函数首先对 input_1 和 input_2 沿最后一维进行连接成一个 Tensor, 然后做一个卷积核大小为 3*3 的卷积，卷积后的输出记作 input, 然后对 input 做全局平均池化，方法与 ARM 中的相同，都是通过 `tf.reduce_mean(inputs, [1, 2], keep_dims=True)` 实现，然后再通过两次卷积，卷积核大小都为 1*1, 前一次卷积激活函数为 relu, 后一次激活函数为 Sigmoid, 将输出结果记为 net, 然后对 net 和 input 做对应元素乘，将结果记为 net1, 最后对 net1 和 input 相加，将结果作为返回值。

4.4 实验结果分析

4.4.1 模型训练

经过前几次实验分析，将模型训练的轮数设为 800 轮，batch_size 大小为 7, learning_rate 为 0.0001。把数据集分成 train、test、validation, 按照 8:1:1 的比例来划分数据集。它们分别对应相应 train_label、test_label、validation_label 的标签的，如图 4-21 所示。

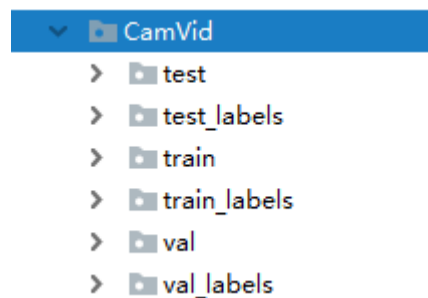


图 4-21 数据集

训练的模型的 Average validation accuracy 随轮数的变化趋势如图 4-22 所示所示，由图可知当在 1-100 轮时 Average validation accuracy 迅速上升，说明模型学习电气石图像特征的能力较强，在。

短时间内可提取到图片之间共同的特征，在 100 轮之后 Average validation accuracy 呈现小幅度的振荡，说明模型在前 100 轮之内提取的特征大多正确，但

要在学习到更为深层的特征时比较难，这不仅仅与模型和数据集有关，而是涉及全面的影响因素。

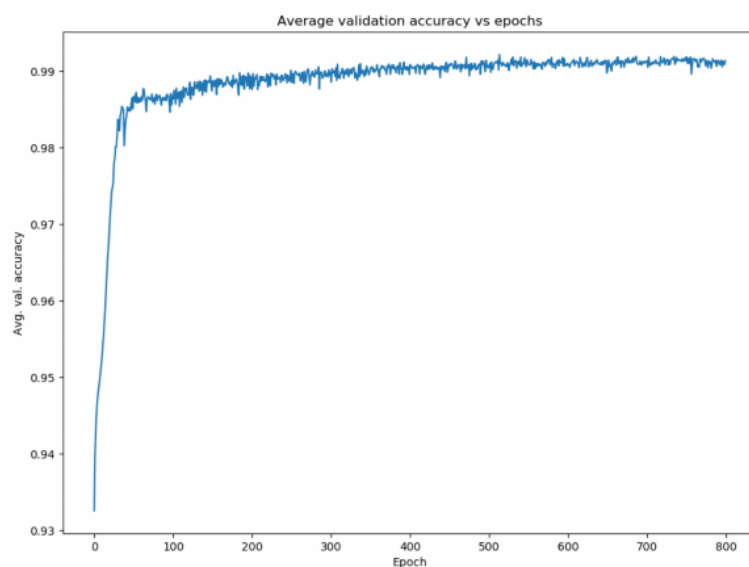


图 4-22 Average validation accuracy 随轮数的变化趋势图

模型的 Average loss 随轮数的变化趋势如图 4-23 所示，由图可知在训练的早期阶段，模型的平均损失较大，但是随着轮数的递增，平均损失迅速下降到较低水平，在 100 轮之后展示出非常平稳的状态。

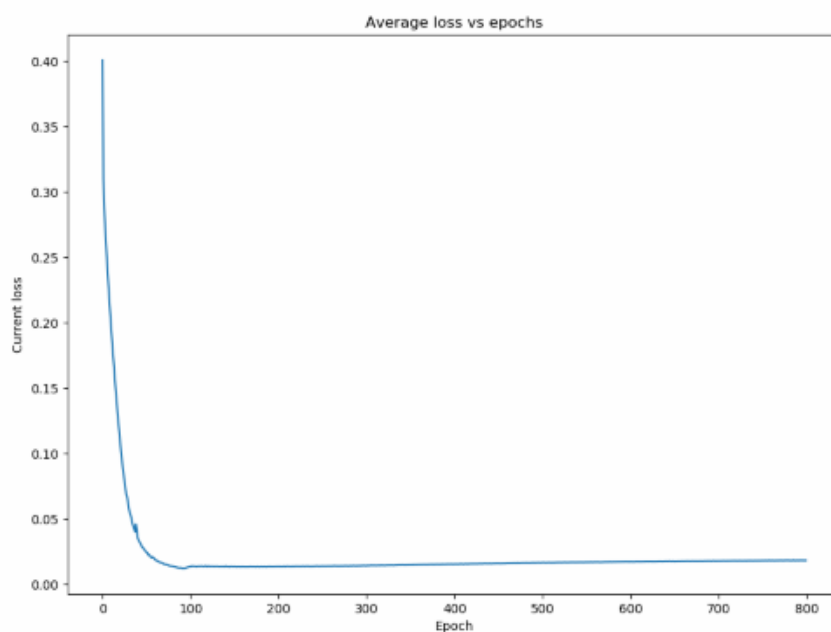


图 4-23 Average loss 随轮数的变化趋势

Average IoU 随轮数的变化趋势如图 4-24 所示，与图 4-22 相似，其在 100 轮之内也飞速增长，但是在 100 轮之后呈现小幅振荡且小幅上升趋势，说明模型的学习能力已经最大化，无法再获得非常有用的特征，模型已趋于稳定。

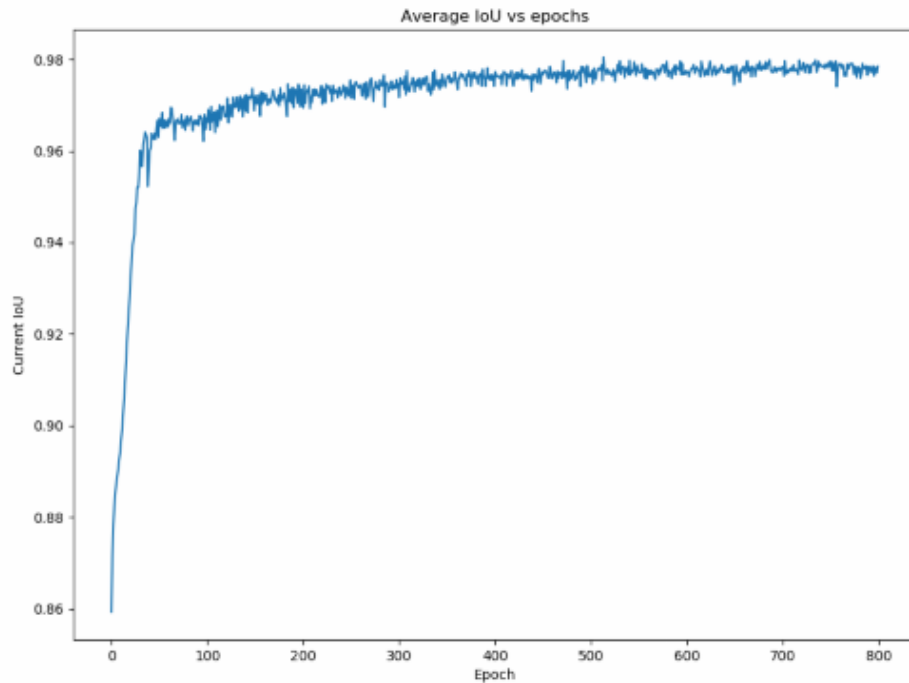
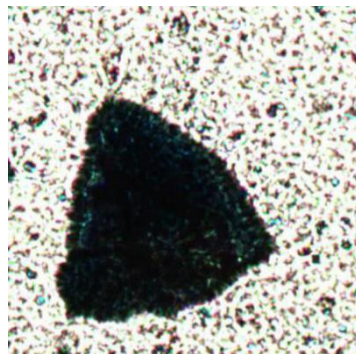


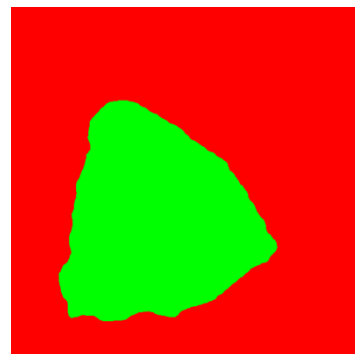
图 4-24 Average IoU 随轮数的变化趋势图

4.4.2 模型预测

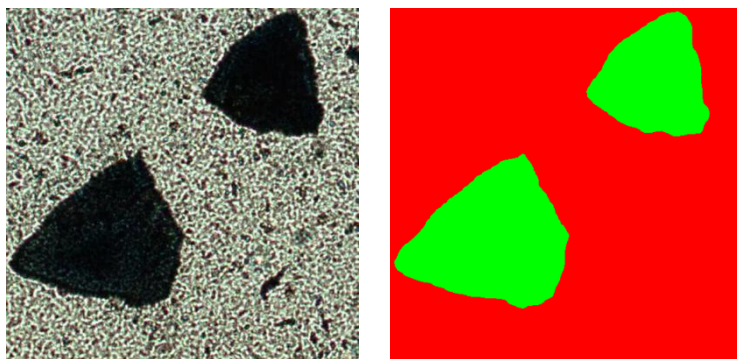
模型训练完成后就可以运行测试程序进行验证，限于篇幅，挑出两组原图与模型预测图如图 4-25 所示。由图对比可知模型的预测较为精确。



(a)原图 1



(b)预测 1



(c) 原图 2

(d) 预测 2

图 4-25 原图与模型预测图片

4.5 本章小结

本章首先介绍了 ImageNet 大赛中的近代网络结构，因为后面网络都是基于这几个经典的结构深入研究的,而且这几个经典的网络中采用的优化思想值得我们学习，之后介绍了图像中应用最多的卷积神经网络中的常用概念，进一步为之后的 BiSeNet 搭建作了铺垫。重点介绍了 BiSeNet 模型的具体实现。之后对实验的结果进行分析，展现了模型预测结果。

5. 总结与展望

5.1 本文总结

在之前的几次实验中跑出来的结果都不太好，我根据根据实验的结果通过对比其他成功实验得出失败原因，并对数据集、网络结构和超参数进行了调整，最终得到了比较理想的结果。

由于深度学习对数据的需求量很大，所以很多专业领域还没有运用到太多的深度学习技术来发展，所以我们深度学习的研究人员一定要多了解其他相关领域的知识，以将深度学习技术推广到更多的领域，推动深度学习在各个领域的发展。例如本文所研究的电气石显微图像分割就是其中一个，我查阅资料发现研究这个方向的并不是太多，所以利用本校的优势，进行交叉学科的研究。从结果来看，效果符合预期结果，实现深度学习的一个实际应用。

本文主要从向项目的时间顺序进行讲解，对电气石的工业背景等开始入手，到处理数据集，搭建实验所需要的环境，建立网络，训练模型，进行预测。对本实验所用到的技术进行了比较详细的介绍，通过这一个简单的实验也体现出了深度学习用途之广。

5.2 以后研究的工作

根据这段时间的对电气石的了解，我将之后的研究方向分成两个。

第一：电气石种类很多，颜色形状各异，每种用途也不同，如果之后有更多的数据，那么可以对所有种类的电气石运用深度学习进行图形分割，将所有的电气石分类。

第二：因为电气石具有压电效应，在工业上的用途很大，所以可以在本文的研究基础上对所采集到的电气石图像中电气石的面积和周长进行计算，然后作为电气石实际用途的指标。

致谢

首先要感谢我的指导老师芦碧波老师，老师首先了解我的兴趣，然后根据我的兴趣给我推荐了深度学习的项目，给我点明了方向，通过之后老师的深度学习课程我进一步加深了对深度学习的了解，也激发了我要完成整个项目的热情。老师还根据以往的经验对我们进行了 PhotoShop 的培训，让我们在最短的时间内掌握了标注数据的方法。在之后的工作之中，老师每一星期都会召集我们一起讨论本周的完成情况，然后对我们的工作做出指导，消除了我们前进路上的拦路虎。老师为了我们请来资环学院的老师为我们讲解岩石的背景知识和用途，然我们知道了自身工作的实际意义。老师还对我们进行了论文格式的统一，减少了我们书写论文的错误。再一次感谢老师不辞辛苦的指导。

其次要感谢杨文艺学长和同组同学的帮助，一开始的时候并不了解深度学习，杨文艺学长给我推荐了很多非常实用的资料，并且给我们讲了他学习深度学习中遇到的问题，这对我的帮助很大，让我可以迅速的进入主题。每当我遇到新的问题，我们一个小组的都会在一起讨论解决，这不仅能很快解决问题，而且拓展了我的思路。

最后感谢家人和学校给我的支持，给我提供了良好的环境来完成这个毕设，让我可以全心地思考问题，谢谢你们。

参考文献

- [1] W.W.Morrhouse. 岩石薄片研究入门[M]. 马志先、吴国忠、马绍周, 译. 蒋荫昌, 校. 北京:地质出版社, 1986-06
- [2] 李苍柏, 范建福, 宋相龙. 深度学习在地质上的应用[J]. 地质学刊, 2018, 42 (01) , 1 15-121
- [3] 张野, 李明超, 韩帅. 基于岩石图像深度学习的岩性自动识别与分类方法[J]. 岩石学报, 2017, 34(2): 333-342
- [4] 白林, 姚钰, 李双涛, 徐东晶, 魏昕. 基于深度学习特征提取的岩石图像矿物成分分析[J]. 中国矿业大学学报, 2018, 27: 178-182
- [5] 刘祥龙, 杨晴虹, 谭中意, 蒋晓琳等. PaddlePaddle 深度学习实战[M]. 北京: 机械工业出版社, 2018
- [6] 高扬、卫峥. 白话深度学习与 TensorFlow[M]. 万娟, 插画设计. 北京: 机械工业出版社, 2017-08
- [7] 孙志军, 薛磊, 许阳明, et al. 深度学习研究综述[J]. 计算机应用研究, 2012, 29(8):280 6-2810.
- [8] 刘建伟, 刘媛, 罗雄麟. 深度学习研究进展[J]. 计算机应用研究, 2014, 31(7).
- [9] 余凯, 贾磊, 陈雨强, et al. 深度学习的昨天、今天和明天[J]. 计算机研究与发展, 2013, 50(9):1799-1804.
- [10] 谢承泮. 神经网络发展综述[J]. 图书情报导刊, 2006, 16(12):148-150.
- [11] 喻宗泉. 人工神经网络发展五十五年[J]. 自动化与仪表, 1998(5):1-4.
- [12] 马艳霞. 人工神经网络发展之浅见[J]. 吉林广播电视大学学报, 2007(5):111-113
- [13] 龙伟, 张金. 人工神经网络发展前景[J]. 机械, 1998(1):47-50.
- [14] 郑宝玉, 王雨, 吴锦雯, et al. 基于深度卷积神经网络的弱监督图像语义分割[J]. 南京邮电大学学报(自然科学版), 2018, 38(05):5-16.
- [15] 王德培, 谢云. 模糊图像语义分割[J]. 电子世界, 2018, 555(21):34-35.
- [16] 彭琮, 曹建军, 吴凡, et al. 基于数据源分析的数据增强方法[J]. 兵工自动化, 2017.
- [17] 陈文兵, 管正雄, 陈允杰. 基于条件生成式对抗网络的数据增强方法[J]. 计算机应用, 2018, 38(11):259-265.
- [18] 张一珂, 张鹏远, 颜永红. 基于对抗训练策略的语言模型数据增强技术[J]. 自动化学报, 2018, v.44(05):126-135.
- [19] Abadi, Martín, Barham P, Chen J , et al.TensorFlow: A system for large-scale machine learning[J]. 2016.

- [20] Baylor D , Koc L , Koo C Y , et al. TFX: A TensorFlow-Based Production-Scalable Machine Learning Platform[C]// the 23rd ACM SIGKDD International Conference. ACM, 2017.
- [21] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems[J]. 2016.
- [22] Matthews A G D G , Mark V D W , Nickson T , et al. GPflow: A Gaussian process library using TensorFlow[J]. 2016.
- [23] Changqian Yu; Jingbo Wang; Chao Peng; Changxin Gao; GangYu; Nong Sang. BiSENet: Bilateral Segmentation Network for Real-time Semantic Segmentation.In: The European Conference on Computer Vision (ECCV), 2018, pp. 325-341