|  |
| --- |
|  |
|  |
| **毕 业 设 计** |
|  |
| 2020年6月 |

|  |
| --- |
| **基于深度学习的几何体语义分割系统设计** |

**摘要**

随着深度学习的快速发展以及全卷积神经网络的出现，图像语义分割技术得到了迅速发展。语义分割目前已成为计算机视觉中一个十分热门的发展领域。通过图像语义分割算法，可以对图像进行像素级(pixel)分类，不仅可以对图像的内容进行识别，还可以获取图像的整体信息。在科技迅速发展、文明不断进步的今天，人机交互、地质检测、无人驾驶等领域均需要高效、精确的语义分割作为前提，因此对图像语义分割技术的研究具有重要的现实意义。

一个通用的语义分割架构是编码器-解码器网络架构，语义分割不仅需要在像素级有判别能力，还需要有能将编码器在不同阶段学到的可判别特征投影到像素空间的机制。不同的架构采用不同的机制作为解码机制的一部分。基于全卷积网络的语义分割，存在一个问题，输入经过几个卷积和池化后，输出特征图的分辨率下降，因此，FCN 的直接预测分辨率低，导致了相对模糊的物体边界。为了解决这个问题，设计新的网络，先下采样，经过不同程度的卷积，学习了深层次的特征，再经过上采样恢复为原图大小，上采样用反卷积实现。上采样部分会融合特征提取部分的输出，这样做实际上是将多尺度特征融合在了一起，以最后一个上采样为例，它的特征既来自第一个卷积block的输出(同尺度特征)，也来自上采样的输出(大尺度特征)，这样的连接是贯穿整个网络的，这样网络中有多次融合过程，相对应的FCN网络只在最后一层进行融合。生成密集的特征图后，最后的输出特征映射被送入soft-max分类器进行像素级分类。

最后本文在新网络模型之下，在 Windows 系统中进行了图像语义分割系统的设计与实现。

**关键词**　语义分割；全卷积神经网络；U-Net；tensorflow

|  |
| --- |
| **Design of Geometry Semantic Segmentation System Based on Depth Learning** |

**Abstract**

With the rapid development of deep learning and the emergence of full convolution neural network, image semantic segmentation technology has developed rapidly. Semantic segmentation has become a very popular development field in computer vision. Through the image semantic segmentation algorithm, the image can be classified at the pixel level, which can not only recognize the content of the image, but also obtain the overall information of the image. With the rapid development of science and technology and the continuous progress of civilization, human-computer interaction, geological detection, driverless and other fields need efficient and accurate semantic segmentation as the premise, so the research on image semantic segmentation technology has important practical significance.

A general semantic segmentation architecture is the encoder decoder network architecture. Semantic segmentation requires not only the ability to distinguish at the pixel level, but also the mechanism to project the discriminative features learned by the encoder at different stages into the pixel space. Different architectures adopt different mechanisms as part of the decoding mechanism. There is a problem in the semantic segmentation based on the full convolution network. After several convolutions and pooling, the resolution of the output characteristic image decreases. Therefore, the direct prediction resolution of FCN is low, resulting in relatively fuzzy object boundary. In order to solve this problem, a new network is designed. First, the lower sampling is used, and then the deep level features are learned after different degrees of convolution, and then the upper sampling is restored to the original image size, and the upper sampling is realized by deconvolution. The upper sampling part will fuse the output of the feature extraction part. In fact, it will fuse the multi-scale features together. Taking the last upper sampling as an example, its features come from both the output of the first convolution block (the same scale features) and the output of the upper sampling (the large-scale features). Such a connection runs through the whole network, so there are multiple fusion processes in the network The corresponding FCN network is only fused at the last layer. After generating dense feature map, the final output feature map is sent to soft Max classifier for pixel level classification.

Finally, under the new network model, this paper designs and implements the image semantic segmentation system in Windows system.

**Keywords** semantic segmentation；FCN；U-Net；tensorflow

目录

摘要

Abstract

[1 绪论 1](#_Toc41317388)

[1.1 语义分割概述 1](#_Toc41317389)

[1.1.1 语义分割介绍 1](#_Toc41317390)

[1.1.2 国内外研究现状 1](#_Toc41317391)

[1.2 现有的语义分割方法 2](#_Toc41317392)

[1.2.1 基于区域的语义分割 2](#_Toc41317393)

[1.2.2 弱监督语义分割 2](#_Toc41317394)

[1.2.3 基于全卷积神经网络的语义分割 3](#_Toc41317395)

[1.3 本文主要内容及章节安排 3](#_Toc41317396)

[2 深度学习理论与相关技术 4](#_Toc41317397)

[2.1 深度学习的概述 4](#_Toc41317398)

[2.1.1 深度学习的发展 4](#_Toc41317399)

[2.1.2 深度学习的特点 4](#_Toc41317400)

[2.2 卷积神经网络 5](#_Toc41317401)

[2.2.1 全连接网络 5](#_Toc41317402)

[2.2.2 局部相关性和权值共享 5](#_Toc41317403)

[2.2.3 卷积层 6](#_Toc41317404)

[2.2.4 池化层 7](#_Toc41317405)

[2.2.5 经典的卷积神经网络 8](#_Toc41317406)

[2.3 卷积神经网络训练过程 8](#_Toc41317407)

[2.3.1 前向传播 8](#_Toc41317408)

[2.3.2 启动函数 8](#_Toc41317409)

[2.3.3 误差计算 11](#_Toc41317410)

[2.3.4 单神经元梯度 11](#_Toc41317411)

[2.3.5 反向传播算法与优化方法 12](#_Toc41317412)

[2.3.6 过拟合和欠拟合 12](#_Toc41317413)

[2.4 本章总结 13](#_Toc41317414)

[3 网络模型的搭建与实现 14](#_Toc41317415)

[3.1 全卷积神经网络 14](#_Toc41317416)

[3.1.1 全卷积神经网络的架构 14](#_Toc41317417)

[3.1.2 全卷积神经网络优缺点 15](#_Toc41317418)

[3.2 网络改进方法 16](#_Toc41317419)

[3.2.1 BN层 16](#_Toc41317420)

[3.2.2 拼接结构 16](#_Toc41317421)

[3.2.3 数据增强 17](#_Toc41317422)

[3.3 语义分割网络实现 18](#_Toc41317423)

[3.3.1 数据准备 18](#_Toc41317424)

[3.3.2 网络模型 18](#_Toc41317425)

[3.4 实验结果与分析 19](#_Toc41317426)

[3.4.1 评价指标 19](#_Toc41317427)

[3.4.2 新网络与FCN直观对比 20](#_Toc41317428)

[3.4.3 新网络精确度作对比 21](#_Toc41317429)

[3.4.4 训练自己数据集 21](#_Toc41317430)

[3.5 本章总结 22](#_Toc41317431)

[4 系统的设计与实现 23](#_Toc41317432)

[4.1 需求分析 23](#_Toc41317433)

[4.2 系统功能设计 23](#_Toc41317434)

[4.2.1 图像语义分割模块 24](#_Toc41317435)

[4.2.2 用户交互模块 24](#_Toc41317436)

[4.3 系统实现与演示 24](#_Toc41317437)

[4.3.1 系统开发环境 24](#_Toc41317438)

[4.3.2 系统图形用户界面 24](#_Toc41317439)

[4.4 本章小结 26](#_Toc41317440)

[结论 27](#_Toc41317441)

[参考文献 28](#_Toc41317442)

[附录 29](#_Toc41317443)

[致谢 32](#_Toc41317444)

# 绪论

## 语义分割概述

### 语义分割介绍

语义分割[1]是当今计算机视觉领域的重要问题之一。从宏观上看，语义分割是一项高层次的任务，为实现场景的完整理解铺平了道路。场景理解作为一个核心的计算机视觉问题，其重要性在于越来越多的应用程序通过从图像中推断知识来提供营养。其中一些应用包括自动驾驶汽车、人机交互、虚拟现实等[2]。

语义分割是通过算法自动分割并识别出图片中的内容，可以将语义分割理解为每个像素点的分类问题，分析每个像素点的物体的类别信息，实现对更加复杂场景的理解。如图1-1所示，语义分割所要完成的效果即是为每个像素点实现分类，但不区分不同个体。



图1-1 语义分割效果图

### 国内外研究现状

图像语义分割是AI领域中，有关计算机视觉一个重要的研究方向，为图像语义理解提供了必要支持，同时语义分割也是深度学习算法的一个应用。随着深度学习算法的提出与发展,传统图像语义分割方法在分割的准确率和精度上，都已经显得很是落后。最初，全卷积神经网络将可用于分类的卷积神经网络应用到图像语义分割,卷积层替换全连接层，使得图像语义分割发展迅速,表现在语义分割的精度，准确率以及速度的很大提升。随着全卷积神经网络的提出，出现了很多改变卷积方式的网络，用以提高语义分割的准确率，精度等一系列评价指标。其中常用的有：深度可分离卷积，此卷积是在牺牲少量性能的基础上，很大程度上降低了计算量；DCN，通过改变卷积核的长宽比例，用以应用多种形状特征图的卷积，更关注特征图想要关注的位置；致密上采样卷积，可以得到更多的特征图信息。以及出现的利用全局信息的多种语义分割方法，使得语义分割的发展十分迅速，在准确率，平均并交比，平均像素精度，频率权重交兵比都有了很大的提升。

基于深度学习的方法，以及卷积层的应用，常见的语义分割模型有 FCN、U-net、SegNet、DeepLab 系列等[3]。各种方法的不断提出，使得在图像语义分割方面取得了不小成功,但为了使计算机可以像人类一样去理解周围环境，还需要很多工作去做。

## 现有的语义分割方法

一个通用的语义分割架构是编码器-译码器网络架构：编码器网络通常是一个预训练分类网络，例如 VGG/ResNet。译码器网络的任务是将编码器学到的可判别特征（低分辨率）在语义上将其投影到像素空间（更高分辨率），来得到一个密集分类。

不同于分类任务中深度网络的最终结果（即类存在的概率）是唯一重要的事，语义分割不仅需要在像素级有判别能力，还需要有能将编码器在不同阶段学到的可判别特征投影到像素空间的机制。不同的架构采用不同的机制作为译码机制的一部分。现在主要讨论三个主要方法：基于区域的语义分割，基于全卷积神经网络的语义分割，弱监督语义分割。

### 基于区域的语义分割

基于区域的方法通常遵循“使用识别进行分割”流程[4]，首先从一张图像中提取任意形状的区域并且标识出来，然后进行基于区域的分类。在测试时，基于区域的预测被转化成像素级的预测，通常根据包含像素的最高分数区域来标注像素。

R-CNN 是一个具有代表性的基于区域的方法，它基于目标检测结果执行语义分割。具体地，R-CNN 首先利用选择性搜索提取大量目标建议，然后对每个建议计算 CNN 特征，最后使用特定类线性支持向量机对每个区域进行分类。与传统用于图像分类的 CNN 结构相比，R-CNN 可以解决更复杂的问题，例如目标检测和图像分割，并且已经成为了这两个领域一个重要的基础网络。除此之外，R-CNN可以建立在任意 CNN 基准架构之上。

对于图像分割任务，R-CNN 对每个区域提取两类特征：全区域特征和前景特征。将这两类特征合并在一起作为区域特征可以使网络获得更好的性能。由于使用了高度可判别 CNN 特征，R-CNN 获得了显著的性能提升。然而，R-CNN 在处理分割任务时还有一些缺点：

特征不兼容分割任务。

特征没有包含足够的空间信息来生成精确的边界。

生成基于分割的建议需要花费大量时间并且会大大影响最终的性能。

### 弱监督语义分割

语义分割中大多数方法都依赖于大量带有像素级标注的图像，然而，手工标注相当费时费力。弱监督问题是指为实现某个计算机视觉任务[5]，采用了比该任务更弱的一种人工标注作为监督信息。弱监督的标注比原始的标注更容易获取。例如，对于目标检测任务，图像层面的卷标相比物体的边界框是一种弱监督的标注；对于语义分割任务，图像层面的卷标和物体的边界框相比像素层面的卷标则是一种弱监督的标注。弱监督的视觉学习，得到像素级的标记通过分类卷标是最简单的，是指仅利用图像层面的标注信息，通过弱监督方式学出一个模型，该模型可以预测出图像的语义分割结果。在比较专业的领域，运用比较广，比如说，在医学图像方面弱监督运用很广，标注工作一般人做不了，因为标注需要医生。这种弱监督学习的关键问题是如何去构建图像层面的卷标语义和像素点的关联，推断出图像所对应的分割屏蔽，从而利用全卷积神经网络去学习分割模型。

因此，一些弱监督方法被提出。例如，Boxsup 利用边界框标注作为监督信号来训练网络，迭代地提升预测屏蔽。Simple Does It 将弱监督限制作为一个输入卷标噪声，并将递归训练作为去噪策略。Pixel-level Labeling 解释了多实例学习框架内的分割任务，通过添加一个额外的层来限制模型将更多的权重分配给图像级分类中重要的像素。

### 基于全卷积神经网络的语义分割

全卷积网络（FCN） 学习像素到像素的映像，没有提取区域建议。FCN 网络是传统 CNN 的扩展，主要思想是使传统 CNN 可以输入任意大小的图像。传统 CNN 只能接受特定大小的输入的原因在于全连接层是固定的。相反，FCN 仅使用卷积和池化层，使得网络可以对任意大小的输入进行预测。FCN 存在一个问题，输入经过几个卷积和池化后，输出特征图的分辨率下降，因此，FCN 的直接预测分辨率低，导致了相对模糊的物体边界。

## 本文主要内容及章节安排

本文介绍从机器学习到深度学习的方法的改变，主要介绍卷积神经网络的架构，进而利用卷积神经网络搭建全卷积神经网络以实现图像的语义分割，由于全卷积神经网络直接预测分辨率比较低，因此本文提出改进方法，最终可以在要求不高的配置上提高精度，具体工作如下：

首先详细讲解CNN网络模型的结构，在CNN基础上，将最后一层的全连接层转换为全卷积层，使用 1x1 卷积。分析FCN的工作原理，得出分析率较低的原因：输入经过几个卷积和池化后，输出特征图的分辨率下降。为了解决这个问题，本文的设计取消全连接层，译码器使用从编码器传输的较大池化索引（位置）对其输入进行非线性上采样，从而使得上采样不需要学习，生成稀疏特征映像。然后，使用可训练的卷积核进行卷积操作，生成密集的特征图。最后的译码器输出特征映像被送入soft-max分类器进行像素级分类。

在章节方面，本文将研究过程分为5个章节，具体内容如下：

第一章，绪论。介绍了本文的研究背景，选题意义，以及现在主要的语义分割方法，并对现有基础的语义分割方法的实现原理和缺点。

第二章，深度学习的发展与相关技术的介绍。介绍从机器学习到深度学习的发展，主要介绍卷积神经网络结构，搭建，训练过程。

第三章，介绍FCN的优缺点，介绍unet的拼接，构造网络。并对其进行测试与实现。

第四章，系统的实现与测试。对上一章节提出的网络进行实现。

第五章，结论。对全文进行总结，分析新网络的优缺点，并对未来研究进行性展望。

# 深度学习理论与相关技术

人工智能是让机器获得像人类一样具有思考和推理机制的智能技术，人工智能的发展主要经历过三个阶段。1970年，科学家们尝试用知识库加推理的方式解决人工智能，通过构建庞大复杂的专家系统来仿真人类专家的智慧水平。这种方式存在一个最大的难题，就是很多复杂抽象的概念无法用具体的代码实现。比如人类对图片的识别，对语言的理解，这些问题没办法通过既定规则模拟，为了解决这些问题，一门通过让机器自动从数据中学习规则的研究学科诞生，称为机器学习。

## 深度学习的概述

### 深度学习的发展

机器学习中，有一门通过神经网络来学习复杂，抽象逻辑的方向，称为神经网络。神经网络算法是一类基于神经网络从数据学习的算法，它仍然属于机器学习的范畴，受限于计算力和数据量，早期的神经网络层数较浅，一般在1-4层，网络表达能力有限。随着计算能力和大数据时代的到来，高度并行化的GPU和海量数据，让大规模神经网络的训练成为可能。

2006年，Geoffrey Hinton等人发现通过逐层预训练的方式可以较好地训练多层神经网络，并在MNIST桑取得了优于SVM的错误率。在论文中，Geoffrey Hinton首次提出Deep Learning的概念，这也是（深层）神经网络被叫做深度学习的由来。。2011 年，Xavier Glorot 提出了线性整流单元(Rectified Linear Unit，简称 ReLU)启动函数，这是现在使用最为广泛的启动函数之一。2012 年，Alex Krizhevsky 提出了 8 层的深层神经网络 AlexNet，它采用了 ReLU 启动函数，并使用 Dropout 技术来防止过拟合，同时抛弃了逐层预训练的方式，直接在两块 NVIDIA GTX580 GPU 上训练网络。AlexNet 在 ILSVRC-2012 图片识别比赛中获得了第一名的成绩，比第二名在 Top-5 错误率上降低了惊人的 10.9%。 自AlexNet 模型提出后，各种各样的算法模型相继被发表，其中有 VGG 系列、 GoogLeNet 系列、ResNet 系列、DenseNet 系列等。ResNet 系列模型将网络的层数提升至数百层、甚至上千层，同时保持性能不变甚至更优。它算法思想简单，具有普适性，并且效果显著，是深度学习最具代表性的模型。

除了有监督学习领域取得了惊人的成果，在无监督学习和强化学习领域也取得了巨大的成绩。2014 年，Ian Goodfellow 提出了生成对抗网络，通过对抗训练的方式学习样本的真实分布，从而生成逼近度较高的样本。此后，大量的生成对抗网络模型相继被提出，最新的图片生成效果已经达到了肉眼难辨真伪的逼真度。2016 年，DeepMind 公司应用深度神经网络到强化学习领域，提出了 DQN 算法，在 Atari 游戏平台中的 49 个游戏上取得了与人类相当甚至超越人类的水平；在围棋领域，DeepMind 提出的 AlphaGo 和 AlphaGo Zero 智能程序相继打败人类顶级围棋专家李世石、柯洁等；在多智慧体协作的 Dota2 游戏平台，OpenAI 开发的 OpenAI Five 智能程序在受限游戏环境中打败了 TI8 冠军队伍 OG 队，展现出了大量专业级的高层智慧操作[6]。

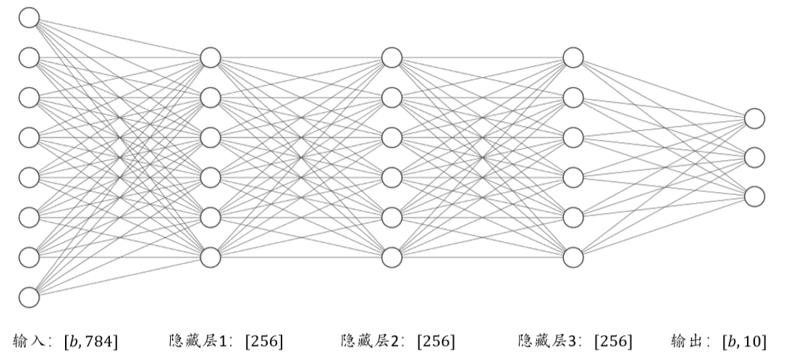
### 深度学习的特点

对数据量，计算能力的要求大。随着计算机技术的发展，设计的算法越来越复杂，对数据量的需求也越来越大。随着深度学习的兴起和计算能力的提高，AlexNet(8 层)、VGG16(16 层)、 GoogLeNet(22 层)、ResNet50(50 层)、DenseNet121(121 层)等模型相继被提出，同时输入图片的大小也从28 × 28逐渐增大，变成224 × 224、299 × 299等，这些变化使得网络的总参 数量可达到千万级别，网络规模的增大，使得神经网络的容量也相应增大，从而能够学习到复杂的数据模态，模型的性能也会随之提升；另一方面，网络规模的增大，意味着更容易出现过拟合现象，训练需要的数据集和计算代价也会变大。

## 卷积神经网络

### 全连接网络

我们来分析一下全连接网络存在的问题。考虑一个简单的4层全连接网络，输入是28\*28打平后为784节点的手写数字图片向量，中间三个隐藏层节点数都是256，输出层的节点是10，如图2-1所示。



输入：[b,784] 隐藏层1：[256] 隐藏层2：[256] 隐藏层3：[256] 输出：[b,10]

图2-1 4层全连接网络结构示意图

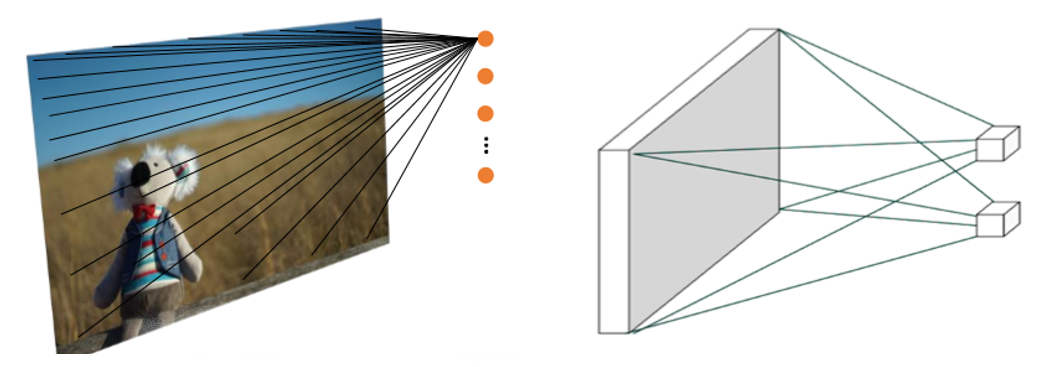
通过Tesnsorflow，我们可以可以这个网络模型，添加4个Dense层，使用Sequential容器封装成一个网络对象，利用summary()函数打印出每一层参数量的统计结果，结果如表2-1所示。我们可以看出，一张简单的28\*28灰度图片，就已经有如此之多的网络参数量。

表2-1 网络参数量统计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 层数 | 隐藏层1 | 隐藏层2 | 隐藏层3 | 输出层 |
| 参数量 | 200960 | 65792 | 65792 | 2570 |

### 局部相关性和权值共享

以图片类型数据为例，我们来讨论避免全连接网络的参数量过大的缺陷。对于灰度图片，即2D图片，在进入全连接层之前，需要将二维矩阵合并成一维向量，然后每个向量与每个输出节点相连，我们可以把连接关系形象对应到图片的像素位置如图2-2。



（a）像素全连接示意图 （b）2D特征全连接示意图

图2-2 特征全连接示意图

我们可以看出网络层的每个输出节点都与所有的输入节点相连接，用于提取所有输入节点的特征信息，这种稠密的连接方式是全连接层参数量大、计算代价高的根本原因。

在现实生活中，存在着大量以位置或距离作为重要性分布衡量标准的数据，比如和自己居住更近的人更有可能对自己影响更大(位置相关)，股票的走势预测应该更加关注近段 时间的数据趋势(时间相关)，图片每个像素点和周边像素点的关联度更大(位置相关)。种基于距离的重要性分布假设特性称为局部相关性，它只关注和自己距离较近的部分节点，而忽略距离较远的节点。在这种重要性分布假设下，全连接层的连接模式变成了如图1.3.2-1所示，输出节点𝑗只与以𝑗为中心的局部区域(感受野)相连接，与其它像素无连接。

每个输出节点仅与感受野区域内𝑘×𝑘个输入节点相连接，输出层节点数为 ‖𝐽‖，则当前层的参数量为𝑘×𝑘×‖𝐽‖，相对于全连接层的‖𝐼‖×‖𝐽‖，考虑到𝑘一般取值较小，如1、3、5等，𝑘×𝑘≪‖𝐼‖，因此成功地将参数量减少了很多。通过权值共享的思想，对于每个输出节点𝑜𝑗，均使用相同的权值矩阵𝑾[7]，那么无论输出节点的数量‖𝐽‖是多少，网络层的参数量总是𝑘×𝑘。使用权值矩阵与对应感受野内部的像素相乘累加，此时网络层的参数量很小，且与输入输出节点数无关。

通过运用局部相关性[8]和权值共享的思想，我们成功把网络的参数量从 ‖𝐼‖×‖𝐽‖减少到𝑘×𝑘(准确地说，是在单输入通道、单卷积核的条件下)。这种共享权值的“局部连接层”网络就是卷积神经网络。

### 卷积层

在局部相关性的先验下，我们提出了简化的“局部连接层”，对于窗口𝑘 × 𝑘内的所有像素，采用权值相乘累加的方式提取特征信息，每个输出节点提取对应感受野区域的特征信息。这种运算是信号处理领域的一种标准运算：离散卷积运算。离散卷积将积分运算换成累加运算。

在计算机视觉中，卷积运算基于2D 图片函数𝑓(𝑚 𝑛)和 2D 卷积核𝑔(𝑚 𝑛)，其中𝑓(𝑖 𝑗) 和𝑔(𝑖 𝑗)仅在各自窗口有效区域存在值，其它区域视为 0，每次通过移动卷积核，并与图片对应位置处的感受野像素相乘累加，得到此位置的输出值。卷积核即是行、列为k大小的权值矩阵w，对应到特征图上大小为𝑘的窗口即为感受野，感受野与权值矩阵w相乘累加，得到此位置的输出值。通过权值共享，我们从左上方逐步向右、向下移动卷积核，提取每个位置上的像素特征，直至最右下方，完成卷积运算如图2-3所示。

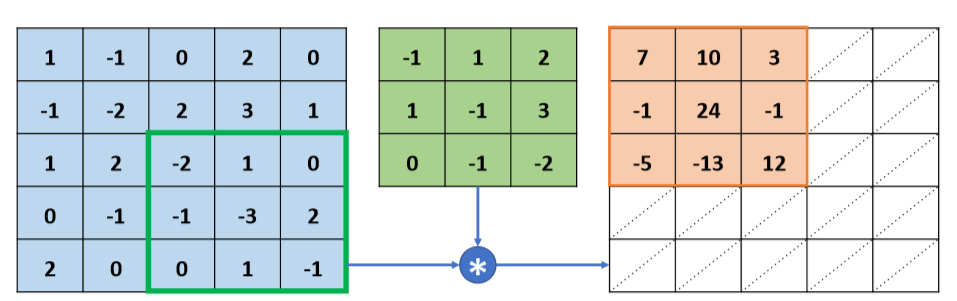
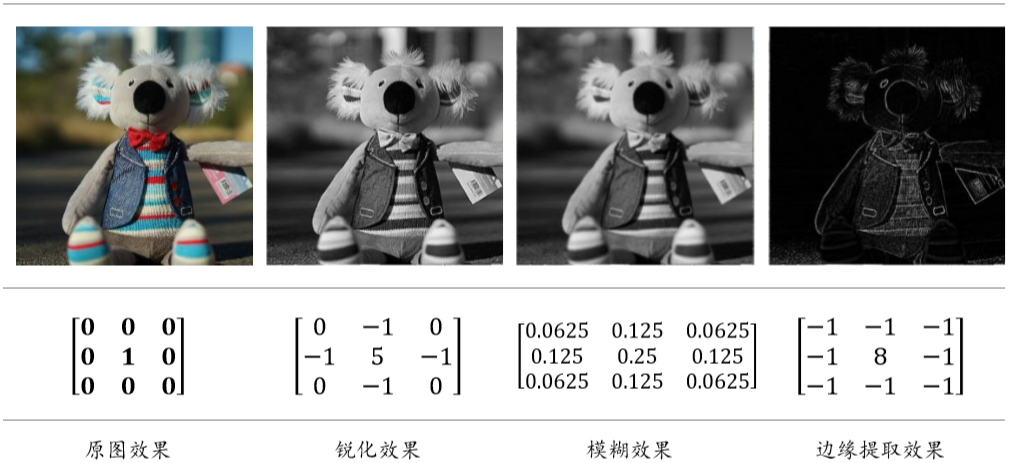


图2-3 3\*3卷积核运算

从数学角度理解，卷积神经网络即是完成了 2D 函数的离散卷积运算；从局部相关与权值共享角度理解，也能得到一样的效果。通过这两种角度，我们既能直观理解卷积神经网络的计算流程，又能严谨地从数学角度进行推导。正是基于卷积运算，卷积神经网络才能如此命名。

在计算机视觉领域，2D卷积运算能够提取数据的有用特征，通过特定的卷积核与输入图片进行卷积运算，获得不同特征的输出图片，卷积层[9]即是为了获得不同特征的图片。一些经典卷积核效果，如图2-4。



原图效果 锐化效果 模糊效果 边缘提取效果

图2-4 常见卷积核及其效果

### 池化层

在卷积层[10]中，可以通过调节步长参数𝑠实现特征图的高宽成倍缩小，从而降低了网络的参数量。除了通过在卷积层设置步长实现特征图的变化，还有一种专门的网络层可以实现尺寸缩减功能，即池化层(Pooling Layer)。

池化层同样基于局部相关性的思想，通过从局部相关的一组元素中进行采样或信息聚合，从而得到新的元素值。常见的池化层，最大池化层，即从局部相关元素中选取一个最大元素值，如图2-5；平均池化层，即从局部相关元素中计算平均值，如图2-6。

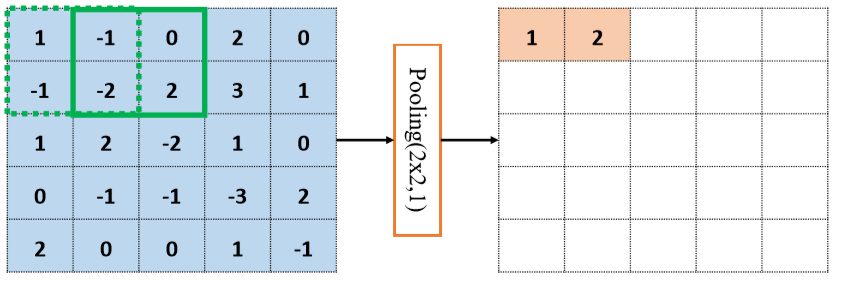


图2-5 最大池化层

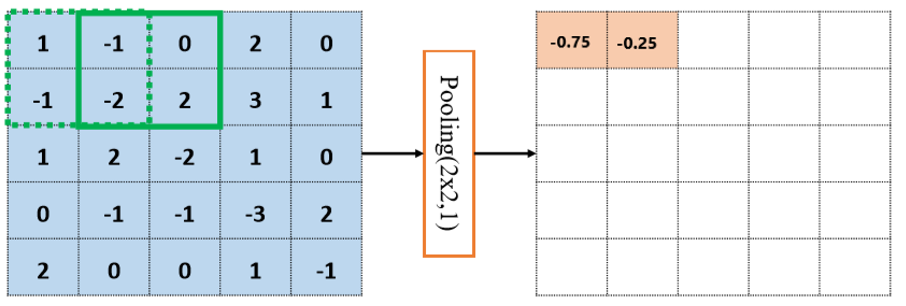


图2-6 平均池化层

### 经典的卷积神经网络

在 AlexNet 出现之前的网络模型都是浅层的神经网络。自2012年 AlexNet的提出以来，各种各样的深度卷积神经网络模型相继被提出，其中比较有代表性的有 VGG系列，GoogLeNet系列 ，ResNet 系列 ，DenseNet 系列等，他们的网络层数整体趋势逐渐增多。以网络模型在 ILSVRC 挑战赛 ImageNet 数据集上面的分类性能表现为例，网络层数的增加，错误率也在降低。

## 卷积神经网络训练过程

### 前向传播

我们把神经网络从输入到输出的计算过程叫做前向传播[11]。神经网络的前向传播过程，也是数据张量(Tensor)从第一层流动(Flow)至输出层的过程，即从输入数据开始，途径每个隐藏层，直至得到输出并计算误差，这也是 TensorFlow 框架名字由来。 前向传播的最后一步就是完成误差的计算。我们希望通过在训练集上面学习到一组参数，使 得训练的误差最小。上述的最小化优化问题一般采用误差反向传播算法来求解 网络参数的梯度信息，并利用梯度下降算法迭代更新参数。

### 启动函数

神经网络中的常见启动函数，与阶跃函数和符号函数不同，这些函数都是平滑可导的，适合于梯度下降算法。

1. Sigmoid 函数也叫 Logistic 函数，定义为：

|  |  |
| --- | --- |
|  | （2-1） |

它的一个优良特性就是能够把 𝑥∈𝑅 的输入“压缩”到 𝑥∈(0,1) 区间，这个区间的数值可以用来表示信号强度，一般可以将0-1理解为某种信号的强度，如像素的颜色强度，1 代表当前通道颜色最强，0 代表当前通道无颜色；抑或代表门控值(Gate)的强度，1 代表当前门控全部开放，0 代表门控关闭。

Sigmoid 函数连续可导，如图2-7, 可以直接利用梯度下降算法优化网络参数。

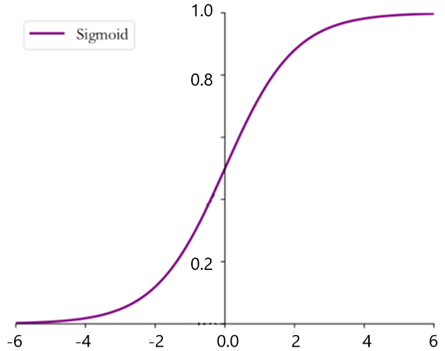


图2-7 Sigmoid函数图像

2.ReLU函数[12]，在 ReLU启动函数提出之前，Sigmoid 函数通常 是神经网络的启动函数首选。但是 Sigmoid 函数在输入值较大或较小时容易出现梯度值接 近于 0 的现象，称为梯度弥散现象。出现梯度弥散现象时，网络参数长时间得不到更新，导致训练不收敛或停滞不动的现象发生，较深层次的网络模型中更容易出现梯度弥散现象。ReLU 函数定义为：

。 （2-2）

函数图像，如图2-8可以看到，ReLU 对小于 0 的值全部抑制为 0；对于正数则直接输 出，这种单边抑制特性来源于生物学。类似于单侧抑制，相对宽松的特性。

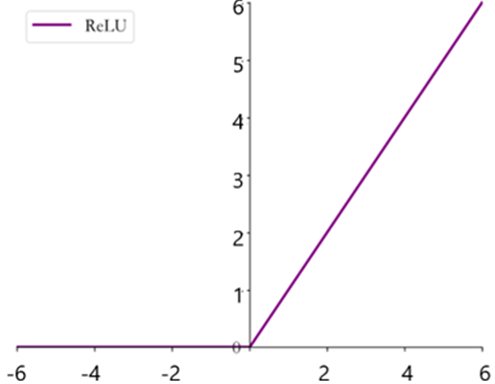


图2-8 Rule函数图像

3.LeakyReLU[12], ReLU 函数在𝑥 < 0时导数值恒为 0，也可能会造成梯度弥散现象，为了克服这个问题，LeakyReLU 函数被提出，LeakyReLU 的表达式为:

|  |  |
| --- | --- |
|  | (2-3) |

其中𝑝为用户自行设置的某较小数值的超参数，当p = 0时，LeayReLU 函数退化为 ReLU 函数；当p ≠ 0时，x < 0处能够获得较小的导数值p，从而避免出现梯度弥散现象。函数图像如图2-9。

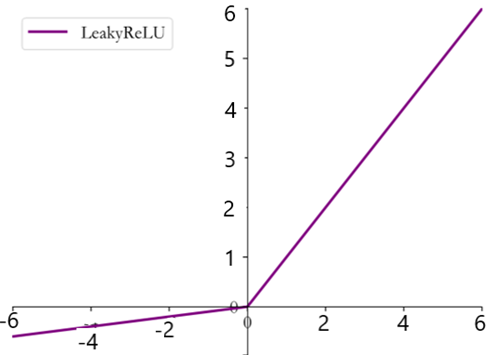


图2-9 LeakyRule图像

4. Tanh函数[13]。Tanh 能够将 x∈𝑅 的输入“压缩”到(−1,1)区间，定义为：

|  |  |
| --- | --- |
|  | (2-4) |

可以看到 tanh 启动函数可通过 Sigmoid 函数缩放平移后实现，图像如图2-10

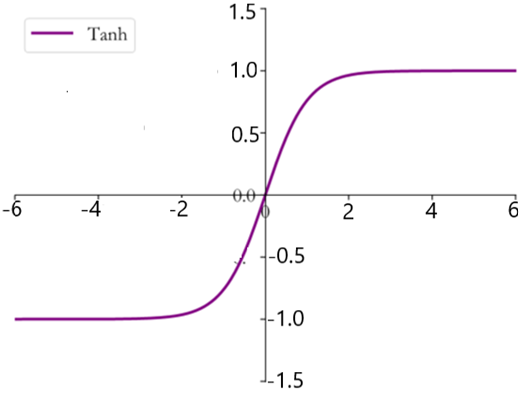


图2-10 tanh函数图像

### 误差计算

在搭建完模型结构后，下一步就是选择合适的误差函数来计算误差。常见的误差函数 有均方差、交叉熵、KL 散度、Hinge Loss 函数等，其中均方差函数和交叉熵函数在深度学习中比较常见。

1. 均方差误差函数，均方差(Mean Squared Error，简称 MSE)误差函数把输出向量和真实向量映射到笛卡尔 坐标系的两个点上，通过计算这两个点之间的欧式距离(准确地说是欧式距离的平方)来衡 量两个向量之间的差距。MSE 误差函数的值总是大于等于 0，当 MSE 函数达到最小值 0 时，输出等于真实标签，此时神经网络的参数达到最优状态。

|  |  |
| --- | --- |
|  | (2-5) |

2. 交叉熵误差函数[14]，将热力学中熵的概念引入到信息论中，用来衡量信息的不确定度。熵在信 息学科中也叫信息熵，或者香农熵。熵越大，代表不确定性越大，信息量也就越大。们基于熵引出交叉熵(Cross Entropy)的定义：

|  |  |
| --- | --- |
|  | (2-6) |

交叉熵有一个好处是使用激活函数sigmoid在梯度下降时能避免均方误差损失函数学习速率降低的问题，学习速率可以被输出的误差所控制。在描述特征中，可以衡量两个随机变量之间的相似度。

### 单神经元梯度

对于采用 Sigmoid 激活函数的神经元模型，它的数学模型可以写为：

|  |  |
| --- | --- |
|  | (2-7) |

其中变量的上标表示层数，我们将神经元模型绘制[15]，如图2-11。

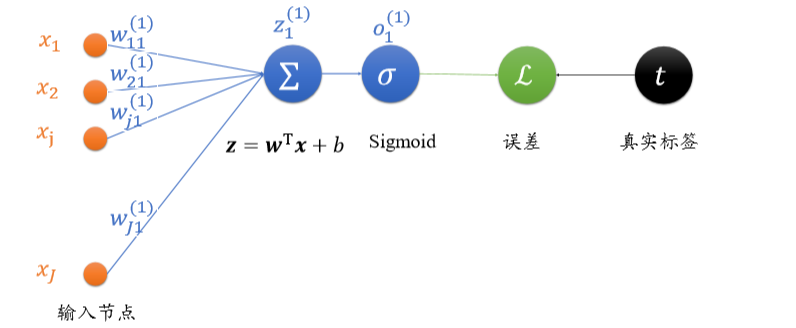


图2-11 单个神经元模型

经过梯度计算，误差对权值𝑤 的偏导数只与输出值𝑜1、真实值𝑡以及当前权值连接的输入𝑥有关。

### 反向传播算法与优化方法

反向传播是“误差反向传播”的简称[16]，是一种与最优化方法（如梯度下降法[17]）结合使用的，用来训练人工神经网络的常见方法。整个网络，即正向传播求误差，反向传播回传误差，该方法对网络中所有权重计算损失函数的梯度。这个梯度会反馈给最优化方法，用来更新权值以最小化损失函数。

常用的优化函数有BGD 批量梯度下降，SGD随机梯度下降，MBGD小批量梯度下降。

梯度下降法，是无约束条件中最常用的方法，假设是具有一阶连续偏导的函数，现在的目标是要求取最小的。核心思想：负梯度方向是使函数值下降最快的方向，在迭代的每一步根据负梯度的方向更新x的值，从而求得最小的。因此我们的目标就转变为求取的梯度。当是凸函数的时候，用梯度下降的方法取得的最小值是全局最优解。

在批量梯度下降法中，因为每次都遍历了完整的训练集，其能保证结果为全局最优，但是也因为我们需要对于每个参数求偏导，且在对每个参数求偏导的过程中还需要对训练集遍历一次，当训练集很大时，这个计算量是惊人的。所以，为了提高速度，减少计算量，提出了SGD随机梯度下降的方法，该方法每次随机选取一个样本进行梯度计算，大大降低了计算成本。随机梯度下降算法和批量梯度下降的不同点在于其梯度是根据随机选取的训练集样本来决定的，其每次对参数的更新，都是针对单个样本数据，并没有遍历完整的参数。当样本数据很大时，可能到迭代完成，也只不过遍历了样本中的一小部分。因此，其速度较快，但是其每次的优化方向不一定是全局最优的，但最终的结果是在全局最优解的附近。

为了综合上述两种方法，提出了小批量梯度下降。降低在SGD中高方差的问题，能使得收敛更加稳定；可以利用深度学习中最先进的库进行矩阵优化的操作，加速操作；一般的小批量介于50～256，但是当适用很小的批量时，有时也统称为SGD。核心思想是在每次迭代时考虑一小部分样本，比如考虑10个样本，同时计算在这10个样本点上的每个参数的偏导数，对于每个优化参数，将该参数在这10个样本点的偏导数求和。

### 过拟合和欠拟合

机器学习的主要目的是从训练集上学习到数据的真实模型，从而能够在未见过的测试 集上也能够表现良好，我们把这种能力叫做泛化能力[18]。训练集和测试集都采样 自某个相同的数据分布。我们将模型拟合复杂函数的能力，记为模型的容量或表达能力。当模型的容量过大，网络模型除了学习到训练数据的模态之外，也把额外的噪音误差也学到了，这样会导致学习的模型在训练集上表现良好，但在一个新的测试集上的表现却欠佳。我们把这种现象记为过拟合。当模型的容量过小时，在训练集上的表现与在测试集上的表现都欠佳，我们把这种现象记为欠拟合。

解决过拟合的方法很多，现在一般有正则化，Dropout，数据增强。

Dropout 通过随机断开神经网 络的连接，减少每次训练时实际参与计算的模型的参数量；但是在测试时，Dropout 会恢复所有的连接，保证模型测试时获得最好的性能。

常用的正则化方式有 L0、L1、L2 。实际训练时，一般优先尝试较小的正则化系数𝜆，观测网络是否出现过拟合现象。然后 尝试逐渐增大𝜆参数来增加网络参数稀疏性，提高泛化能力。但是，过大的𝜆参数有可能导致网络不收敛，需要根据实际任务调节。

数据增强，增加数据集规模是解决过 拟合最重要的途径。但是收集样本数据和标签往往是代价昂贵的，在有限的数据集上，通过数据增强技术可以增加训练的样本数量，获得一定程度上的性能提升。数据增强(Data Augmentation)是指在维持样本标签不变的条件下，根据先验知识改变样本的特征，使得新 产生的样本也符合或者近似符合数据的真实分布。 一般可以通过旋转，翻转，裁剪，生成数据等方式。

## 本章总结

本章主要对卷积神经网络的理论进行了介绍。第一部分介绍了卷积神经网络常用的几个网络层，卷积层，池化层，全连接层。第二部分介绍了卷积神经网络在训练过程中遇到的问题以及比较常见的解决方法。包括损失函数，误差计算，反向传播，梯度下降以及模型构建完成可能造成的过拟合欠拟合的理解。

在下一章我们将介绍基于卷积神经网络改建全卷积网络，并分析其所存在的缺陷和不足，结合U-Net网络，设计出一种比较合适的网络。

# 网络模型的搭建与实现

在第二章，我们对于卷积神经网络相关理论以及网络训练过程进行了简短的介绍。计算机视觉的任务主要有图像分类，目标检测，语义分割等。语义分割更注重的是像素级的分类，相关模型需要具有像素级的密集预测能力。2006年，Geoffrey Hinton 在提出了一种逐层预训练的算法，可以有效地初始化 Deep Belief Networks(DBN)网络，从而使得训练大规模、深层数(上百万的参数量)的神经网络成为可能。深度学习这项技术被用在很多领域，如自动驾驶，医学成像等。全卷积神经网络是2005年提出的用于语义分割的一种框架，这一部分，将介绍全卷积神经网络的架构，并对分割的缺点进行分析，再联系U-net提出方法，并提出适合硬件环境较差的机器的网络模型。

## 全卷积神经网络

### 全卷积神经网络的架构

语义分割的架构，一般是编码器-解码器的架构，编码器网络通常是一个预训练网络，解码器任务是将编码器学到的可判别特征（低分辨率）在语义上将其投影到像素空间（更高分辨率），来得到一个密集分类。

由上一部分，卷积网络理论知识，我们认识一个常见的卷积神经网络，1990年，Yann LeCun等人提出了用于手写数字和机器打印字符图片识别的神经网络，被命名为 LeNet-5。可以看出，网络层数较少，在现在GPU的加持下，数分钟就可以训练好此网络，我们可以看出，该网络由卷积层，池化层，全连接层，最后经过Gaussian connections输出概率值，如图3-1所示。

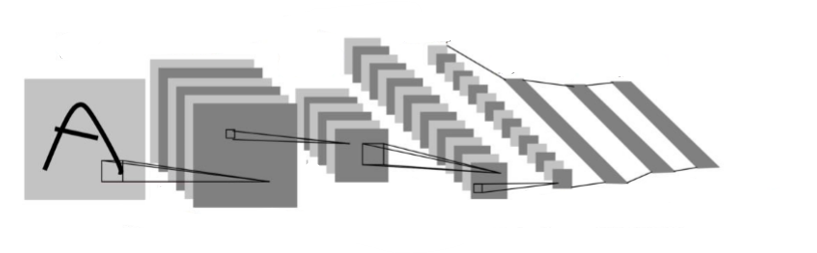


图3-1 LeNet-5网络模型

卷积神经网络的强大之处在于它的多层结构能自动学习特征，并且可以学习到多个层次的特征：浅层卷积层的感知域比较小，能学习到局部区域的特征。深层卷积层有相对比较大的感知域，能学习到抽象一些的特征。这些抽象的特征对物体的大小、位置和方向等敏感性更低，更有助于识别性能的提高。但是因为获得这些抽象特征，所以丢失了一些物体的细节，不能很好地给出物体具体的轮廓，从而不能对物体做精确的分割。传统的卷积神经网络有这样的缺点：存储开销与滑动窗口大，每个窗口都需要存储空间来存储信息，特别是到最后的全连接层，将会以指数级递增；存在大量重复计算，计算效率低下；滑动窗口是相对独立的，末端的全连接层只能约束局部特征。

为了解决卷积神经网络存在的问题，全卷积神经网络[19]将全连接层全部换成卷积层，如图3-2所示。现在的全卷积神经网络主要使用三种技术：卷积化，上采样，跳跃结构。

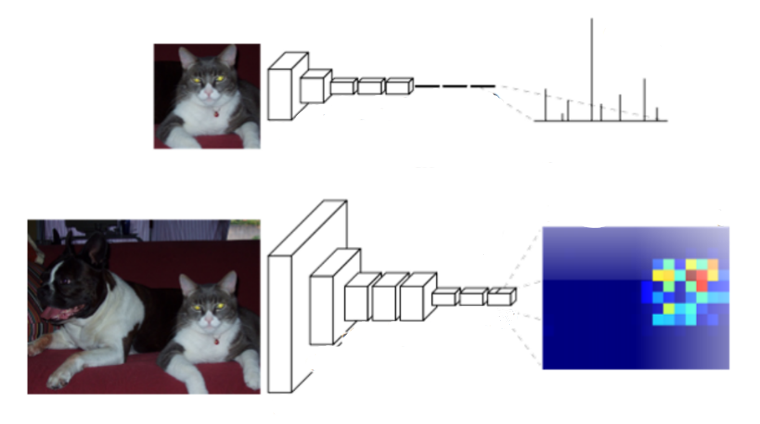


图3-2 全卷积神经网络模型演变

卷积化：全连接层换成卷积层，适应任何尺寸的输入，可以输出低分辨率的分割图片。全连接的结构是固定的，会给每一个连接分权重，不会改变连接关系，卷积核则是会学习有用的关系，没用的关系会直接弱化，这样的设计，共用了一套权重，减少了重复性计算，可以降低模型的复杂度。

上采样：由于在卷积的过程中，形成的热点图变得很小，为了得到原图像大小的稠密像素预测，进行上采样。常见的上采样方法有双线性插值、转置卷积。

跳跃结构：通过卷积，反卷积基本能定位到目标区域，但会发现模型前期是通过卷积、池化、非线性激活函数等作用输出了特征权重图像，我们经过反卷积等操作输出的图像实际是很粗糙的，丢了很多细节。因此我们需要找到一种方式填补丢失的细节数据，所以就有了跳跃结构，如图3-3所示。

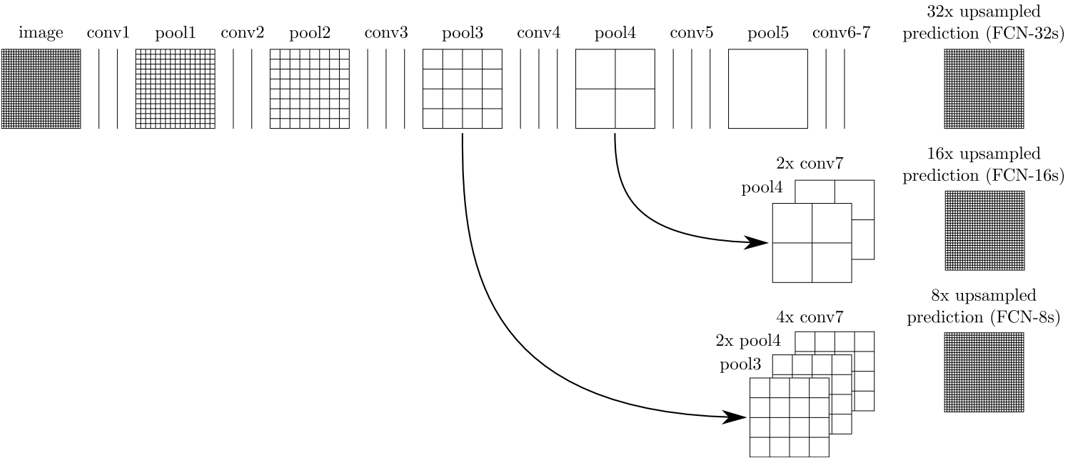


图3-3 加入跳跃结构的全卷积神经网络模型

### 全卷积神经网络优缺点

与传统卷积神经网络进行图像分割的方法相比，全卷积神经网络有两大明显的优点：一是可以接受任意大小的输入图像，而不用要求所有的训练图像和测试图像具有同样的尺寸。二是更加高效，因为避免了由于使用像素块而带来的重复存储和计算卷积的问题。

全卷积神经网络，整体的网络结构可以分为全卷积部分和反卷积部分。

全卷积神经网络存在的缺点：因为模型是基于卷积神经网络改进而来，即便是用卷积层替换了全连接层，但是依然是独立像素进行分类，没有充分考虑像素与像素之间的关系；分割的结果不够精细，图像过于模糊，虽然8倍上采样比32倍上采样效果好了很多，图像还是过于模糊或平滑，没有分割出目标图像的细节。

基于全卷积神经网络存在的缺点，提出可能改进的方法。

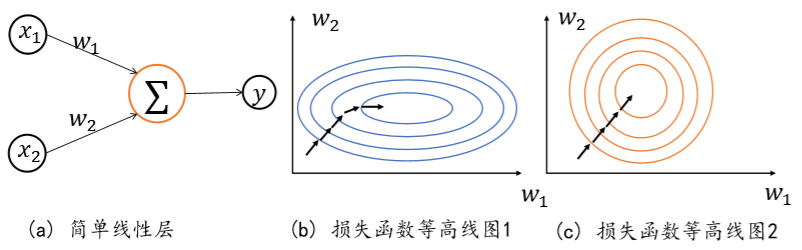
## 网络改进方法

### BN层

讨论在两种输入情况下的优化问题：I 输入

II 输入

分析简单的线性层，绘制出两种不同输入下的函数损失等高线，如图3-4所示。



(a)简单线性层 (b)损失函数等高线1 (c)损失函数等高线2

图3-4 损失函数等高线

图中的圆心是即为全局最优值。分别分析两种不同输入情况的优化轨迹图，我们可以得到，当输入x相差较大，损失函数在w2轴上过于陡峭，这样相对难以收敛。当输入x分布相近时，向中心收敛更快，可以更快的训练。由经验性归纳出，网络层输入分布相近时，并且分布在较小范围内时，更有利于函数的优化，为了实现这个目的，于是引入数据标准化。

通过数据标准化[20]操作可以将数据x映射到:

|  |  |
| --- | --- |
|  | （3-1） |

其中，来自统计的所有数据的均值和方差，是为防止除零错误而设置的较小的数字。

### 拼接结构

拼接：即是采用将特征在channel维度拼接在一起，形成更厚的特征。对应于tensorflow的tf.concat()函数，但存在一个问题，比较占用显存。使用拼接结构，用以实现对图像不同尺度信息的采集，这样做也是为了能尽可能利用图片中的信息。在语义分割发展的过程中，为了对医学影响进行有效的分割，提出了著名的u-net[21]网络，u-net网络便使用了拼接结构，很大程度提高了准确率。

拼接要实现的即是融合同纬度和下层共同的特征，如图3-5所示。

下采样学习到的特征

拼接

图3-5 拼接示意图

### 数据增强

在训练网络模型时候，有时候会出现数据集较小的情况，或者用户为了完成指定对象的语义分割任务，但由于用户在收集数据以及标记数据上，存在收集数据量不够的困难，因此数据增强[21]显得尤为可贵。

数据增强，既可以有效的训练分割对象的特征，也可以在一定程度上减轻过拟合现象。数据增强有多种方式，流行的数据增强技术主要有翻转，旋转，缩放比例，裁剪，移位，高斯噪声，以及一些其他的比较高级增强技术。

翻转和旋转：翻转和旋转都是将原始的图像像素在位置空间上做变换，图像的翻转是将原始的图像进行镜像操作，镜像操作在数据增强中会经常被使用，并且起了非常重要的作用，它主要包括水平镜像翻转，垂直镜像翻转和原点镜像翻转。

裁剪：做裁剪操作主要是考虑原始图像的宽高扰动，在大多数图像分类网络中，样本在输入网络前必须要统一大小，所以通过调整图像的尺寸可以大量的扩展数据。

缩放比例：图像可以向外或向内缩放。向外缩放时，最终图像尺寸将大于原始图像尺寸，为了保持原始图像的大小，通常需要结合裁剪，从缩放后的图像中裁剪出和原始图像大小一样的图像。另一种方法是向内缩放，它会缩小图像大小，缩小到预设的大小。缩放也会带了一些问题，如缩放后的图像尺寸和原始图像尺寸的长宽比差异较大，会出现图像失帧的现象，如果在实验中对最终的结果有一定的影响，需要做等比例缩放，对不足的地方进行边缘填充。

移位：移位只涉及沿X或Y方向（或两者）移动图像，如果图像的背景是单色被背景或者是纯的黑色背景，使用该方法可以很有效的增强数据数量。

高斯噪声：当神经网络试图学习可能无用的高频特征（即图像中大量出现的模式）时，通常会发生过度拟合。具有零均值的高斯噪声基本上在所有频率中具有数据点，从而有效地扭曲高频特征。这也意味着较低频率的组件也会失真，但你的神经网络可以学会超越它，添加适量的噪音可以增强学习能力。基于噪声的数据增强就是在原来的图片的基础上，随机叠加一些噪声，最常见的做法就是高斯噪声。更复杂一点的就是在面积大小可选定、位置随机的矩形区域上丢弃像素产生黑色矩形块，从而产生一些彩色噪声

色彩抖动：色彩抖动主要是在图像的颜色方面做增强，主要调整的是图像的亮度，饱和度和对比度。工程中不是任何数据集都适用，通常如果不同背景的图像较多，加入色彩抖动操作会有很好的提升。

## 语义分割网络实现

### 数据准备

深度学习发展到现在，已经发展了很多很多模型，本文模型的搭建在下面讲解，除了模型的搭建，深度学习所做的工作，花费时间最多的应该是数据的预处理和后处理，如果尝试去做一个自己的语义分割项目，那就也包括自己数据集的制作。

图像语义分割是一种像素级的图像分类操作，目的是在图像中同一类别上打上相同的标签，以表示这个类别是同一类。python中可以用Labelme进行对自己的数据进行标注。

在学习模型的搭建之初，需要指定图像的大小，因此，在训练图片之前，也需要对图片大小，数据格式进行预处理，使其满足输入格式。

文章网络是用tensorflow实现的，在tensorflow中有数据集的创建的函数，同时我们可以把创建的数据集分成训练集和测试集。数据集此时保存的格式是string格式，而string格式是无法输入到神经网络中来训练网络的权重参数。同样，tensorflow中有对png格式的图片进行解码的函数，对转换后的数据集进行解码，tensorflow中map函数可以对整个数据集解码，此时数据集满足深度学习网络模型的输入。

图片数据满足网络模型的输入，便可以对网络进行训练。当网络模型训练成功后，可保存模型，保存模型时候，神经网络的模型以及权重参数都一同保存，可供预测时候再次调用。

在图片预测中，也是需要对图片进行处理，因为预测也是需要将图片输入网络模型，模型对其进行计算，最后得出分割结果。因此，对预测图像也需要做和图片预处理相同的操作。在图片经过处理后，经过网络模型，输出的是做语义分割所标记的标签。为了将图片分割后的图片可视化，也是需要做处理。在tensorflow中，可以实现这种处理。可将处理后的图片保存在本地，供对比观察。

### 网络模型

通常，语义分割网络架构可以概括为编码器-解码器架构。

在编码器部分，可以理解为特征提取，收缩路径，文章网络中，由简单的卷积，池化下采样组成。 在特征提取上，每两个3\*3的卷积层后，跟一个BN层处理，再跟一个2\*2的最大池化层，并且，每次激活函数都采用LeakyReLU函数做处理，除此之外，每次降采样都会增加一倍通道数。

在解码器部分，可以理解为特征融合，扩展路径，文章网络中，上采样由转置卷积进行，然后进行特征融合，但此时的特征融合和全卷积神经网络不一样，即采取拼接。在上采样过程中，每一步会有一个2\*2的卷积层，和两个3\*3的卷积层，于此同时，每一步上采样的过程，都会加入相对应收缩路径的特征图，经过裁剪以保持相同的大小。

在网络的最后一层是一个1\*1的卷积层，通过操作可以将多通道的特征向量转换为所需要的分类结果的数量，最终，经过处理，便可以得到图像语义分割的结果。

文章网络有一个特点，可以输入任意大小的图片，对其进行语义分割。网络模型如下图所示，如图3-6所示。

2Conv3\*3+bn 

Max pool 

Up-conv 

Conv1\*1 

Copy and crop

图3-6 网络模型示意图

## 实验结果与分析

本文的网络搭建，是以tensorflow为框架，以常见的语义分割算法FCN为基础，同时借鉴unet的优点，设计一个新的网络，此网络在kaggle中进行测试，采用的数据集是CamVid。

### 评价指标

为了衡量语义分割网络的作用，判断各个网络的能力，需要使用公用的标准，方法去评估每一个新网络。通常上，一个神经网络系统要从多方面去评价，主要是执行时间，内存占用和精确度。

执行时间：速度是一个非常有价值的度量，很多系统需要保证推理时间可以满足硬实时的需求。在某些情况下，知晓系统的训练时间是非常有用的，执行时间又通常不是非常明显，除非某个系统特别慢。从某种意义上说，提供方法的确切时间可能不是非常有意义，因为执行时间非常依赖硬件设备及后台实现，致使一些比较是无用的。

内存占用：内存是语义分割方法的另一个重要的因素。和执行时间相比，对内存占用的限制比较小，内存可以比较灵活地获得，但其仍然是一个约束因素。执行时间和内存占用，基本上是计算机研究领域中算法都需要考虑的因素。

精确度：语义分割算法中，通常使用多种标准来衡量算法的精度，这些大部分的标准都是像素精度或者IoU的引申。我们采用将逐个介绍。

1. 像素精度(PA)：为标记正确的像素占总像素的比例。

|  |  |
| --- | --- |
|  | （3-2） |

1. 平均像素精度(MPA)：是像素精度的引申，即求出每个类的被正确分类的像素比例，再求其所有类的平均。

|  |  |
| --- | --- |
|  | (3-3) |

1. 平均交并比(MIoU)：是交并比的延申，即求出每一类的交并比，再求出所有类的平均值。

|  |  |
| --- | --- |
|  | (3-4) |

1. 频权交并比：可以理解为根据每一类出现的频率对各个类的IoU进行加权求和

|  |  |
| --- | --- |
|  | （3-5） |

### 新网络与FCN直观对比

CamVid数据集[21]：是来自剑桥的道路与驾驶场景图像分割数据集，图像数据来自视频帧提取，原始分辨率大小为960x720，包括32个类别。分为367张训练图像，100张验证图像，233张测试图像。在测试中，本文用其中的12类，分别是sky，building，road,road\_marking,pavement,tree,signsymbol,fence,car,pedestrian,bicyclist,pole加上背景，一共有13个标签。在kaggle上，利用免费的GPU进行计算，加快神经网络的运算速度，以改变自己本机的低配置带来的不便。在kaggle上测试，结果如图3-7所示。

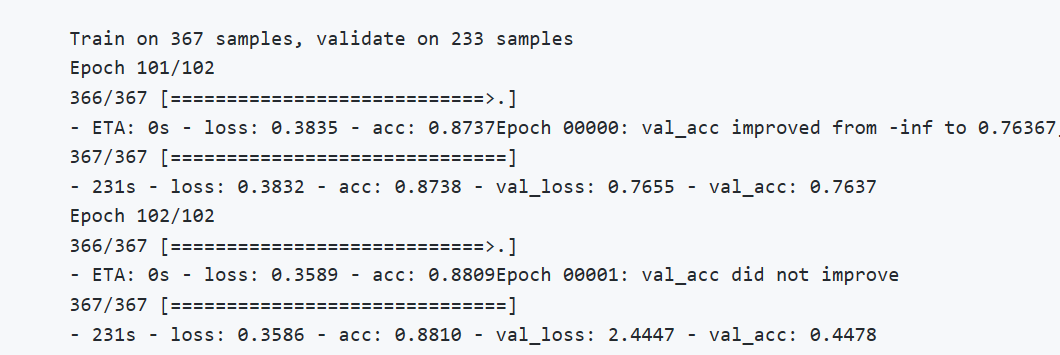
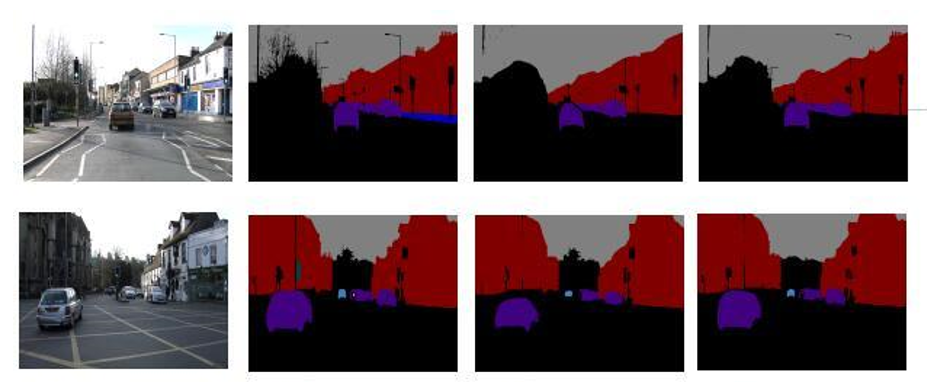


图3-7 准确率训练结果

训练完数据后，准确率达到88%，在图像分割结果做对比，直观显示如图3-8所示。在下图中我们可以看出，在第一幅图中，FCN在电线杆上的分割不如新网络精确，在第二幅图中，FCN在树的分割上，比新网络精确。由此，可以得到新网络在一些问题上还存在一些缺陷。



|  |  |  |  |
| --- | --- | --- | --- |
| 原图 | 原标签 | FCN | 新网络 |

图3-8 分割图片对比

### 新网络精确度作对比

为了更方便的看出各种网络的精确度，在利用实验数据时，将CamVid数据集处理，将其中的各种建筑合并为一类，只保留汽车，天空，这样算上背景，一共有4类。通过在同一个数据集上训练不同的网络模型，用以验证新网络的可行性。对比结果如表3-1。

表3-1 各种网络模型的精确度

|  |  |  |
| --- | --- | --- |
| 网络 | PA（%） | MPA（%） |
| FCN-16 | 88.5 | 73.9 |
| U-Net | 90.1 | 76.4 |
| 新网络 | 90.7 | 77.1 |

在用python处理完精确度的比较后，发现每种网络，其MPA要比PA低上不少。同时，新网络比上FCN有优势，但相对于U-Net来说，比率不大。从对网络模型的分析来看，影响因素可能是激活函数。说明LeakyReLU函数虽然更接近于人类神经网络的机制，但有时候，效果可能不如Rule函数好。

### 训练自己数据集

在kaggle训练Camvid数据集，达到了88.10%的准确率，说明此网络可供日常较为简单的数据集的训练与使用。同时训练自己的数据集，供系统使用。训练结果如图3-9所示。可以看到，准确率已经很高了，这是在数据集稍微简单的情况下达到的结果，证明此网络的可用性是可以的。

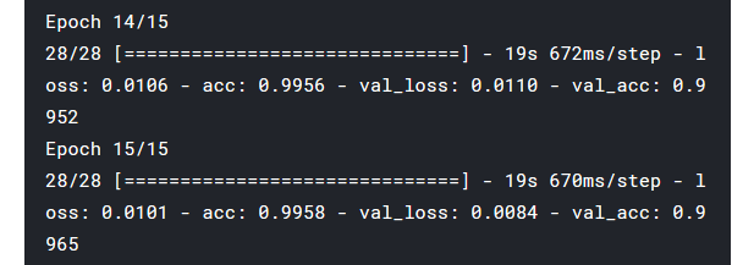


图3-9 自己数据集训练结果

## 本章总结

本章从介绍全卷积神经网络，并针对缺点提出改进方法，搭建一个新的网络。将此网络在kaggle上测试，采用CamVid作为数据集。得出网络在CamVid的准确率以验证此网络的可用性。在下一章将描述文章系统的搭建和可视化界面的设计。

# 系统的设计与实现

近些年来，随着计算机硬件以及深度学习算法的迅速发展，计算机视觉领域有很大的发展潜力。一般的研究都提供了优秀的算法，在语义分割方面，CNN，FCN，SegNet，Ｕ-net等等都可以比较好的进行语义分割，但在查询过程中，很难找到一些提供图形用户界面的软件，本章设计一个可以供普通用户体验图像语义分割的系统，该系统具有选择图像，图像分割，图像对比功能。下文将从各功能的需求分析，进行系统设计，实现各个功能，并进行测试。该系统主要是提供试用功能，即让用户了解什么是语义分割和语义分割可以完成的事情。

## 需求分析

本设计要完成的系统，是根据上部分所提出的新网络，可以实现从本地选择图片，对图片进行预处理，使其符合深度学习模型的输入，从而可以对图像进行语义分割，最后，在分割后的基础上，将原图片与分割后的图片进行对比，展示给用户。在设计好的系统中，将上述功能可以在图形用户界面实现，供方便使用。

系统所涉及到的功能：

1. 图像选择功能：用户可以从本地选择图片，并对图像的大小做预处理，

符合模型的输入格式。

1. 图像分割功能：语义分割功能是系统的核心，将用户选择的图片输入

到模型参数接口，对图片进行分割。

1. 图像对比功能：将用户选择的图片与分割后形成的图片的进行直接直

接对比，让用户可以直接看出分割效果。

由以上简单分析，文章系统的流程如图如图4-1所示。

开始

选择图像

语义分割

图像对比

结束

图4-1 系统流程图

## 系统功能设计

图像分割模块是系统的核心模块，主要功能的实现是由此模块实现。图像分割的网络模型，由tensorflow进行描述实现。与用户进行交互的实现，是在图像分割实现基础上，对函数实现进行组合，以及利用python自带库tkinter实现。

### 图像语义分割模块

图像语义分割模块，是在tensorflow框架上，搭建网络模型，对数据集进行训练，完成对参数的训练。功能完成模块图如下，如图4-2所示。

语义分割

数据集制作

网络搭建

参数设置

网络训练

图4-2 功能模块

### 用户交互模块

要实现与用户的交互，文章系统利用python自带的tkinter，设置图形用户界面，将各个函数可以响应在按钮上，功能如下描述：

用户在本地选择图片，可以返回图片的绝对地址，并在同一界面显示用户选择的图片；用户点击进行语义分割按钮，系统可以接受选择图片返回的绝对地址为参数，将参数传给语义分割模型，模型对传入的绝对地址所表示的图片进行分割，并且可以显示；最后，可以实现图片和语义分割后图像的对比。

## 系统实现与演示

### 系统开发环境

系统采用个人计算机，Window10家庭中文版，具体的开发环境如表4-1所示。

表4-1 系统开发环境

开发环境

详细配置

操作系统

CPU

RAM

开发语言

开发工具

Windows 10家庭中文版

Intel(R)Core(TM)i5-8250U CPU@1.80GHz

8.00GB

Python3.6 tensorflow2.00

Anaconda

### 系统图形用户界面

如图所示，是系统的主模块，可以进行系统的功能，如图4-3所示。

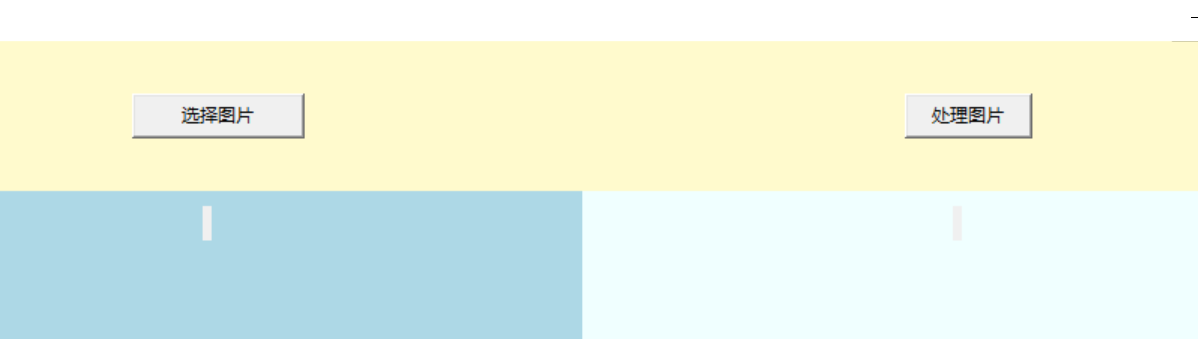


图4-3 系统主界面

从主界面中，有两个按钮，一个是选择图片，一个是处理图片。选择图片的功能即是从本地选择要处理的图片，此处要实现图形用户界面，采用tkinter实现，如图4-4

4-5所示。

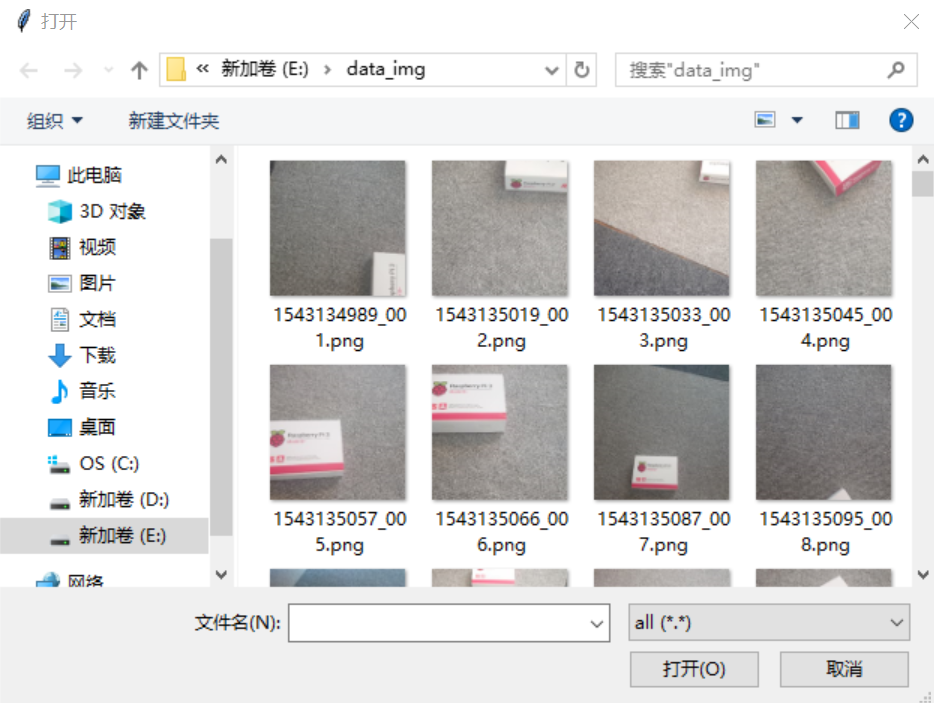


图4-4 选择本地图片



图4-5 本地图片展示

另一个按钮事处理图片的按钮，按钮响应函数即为用tensorflow实现的深度神经网络，此按钮将图片传入到神经网络，经过输出显示在图形用户界面。最后，整个图像用户界面，将会显示两幅图片的对比，直接明显的对比，给用户直接的观感，体会语义分割所能得到地结果，如图4-6所示。

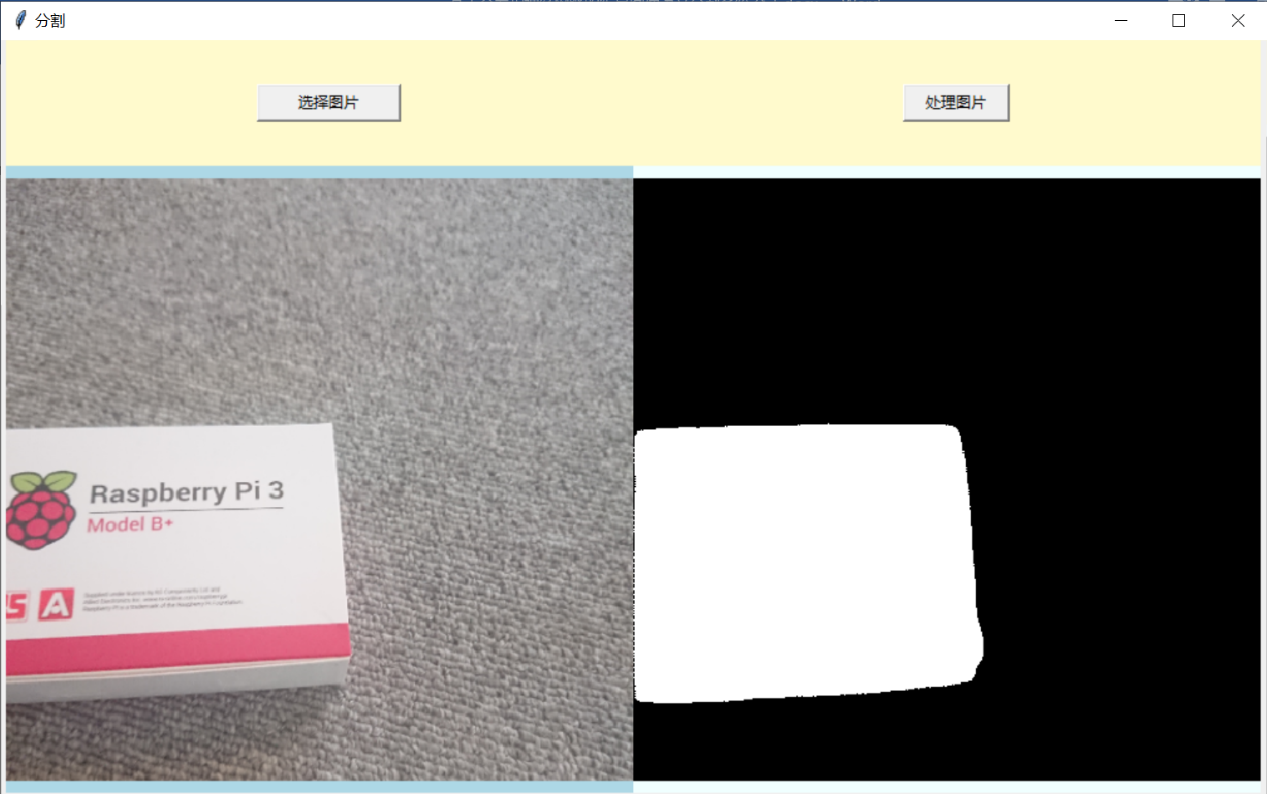


图4-6 结果展示

## 本章小结

本章主要介绍系统的简单实现，展示系统的界面设计以及操作流程。系统主要功能是让用户了解语义分割的所要做事情，系统主要由选择，处理，展示三功能模块组成。本章系统重在体验语义分割功能，功能较少，还需以后打磨各种功能。

千万不要删除行尾的分节符，此行不会被打印。“结论”以前的所有正文内容都要编写在此行之前。

结论

信息技术是人类历史上的第三次工业革命，计算机，互联网等的普及极大方便了人们的日常生活。通过编程的方式，人类可以将提前设计好的交互逻辑交给机器重复快速的执行，从而将人类从简单枯燥的重复劳动工作中解脱出来。但是对于需要较高智能水平的任务，比如人脸识别，自动驾驶等，很难设计明确的逻辑规则，而人工智能是有望解决此问题的关键技术。随着深度学习算法的崛起，人工智能在某些任务上取得了类人甚至超人的智力水平。深度学习在视觉方面主要有图片识别，目标检测，语义分割，视频理解，图片生成。

本文主要讲述的是语义分割方面的内容。介绍深度学习的发展过程，主要介绍实现神经网络的各个部分，各个函数。从而借鉴比较流行的神经网络模型，设计一个可能低配，可能高配的模型。基于全卷积神经网络，模型的具体改进工作如下：

1. 采用U-net网络中的拼接，来增加特征的厚度。
2. 添加BN层，使得数据归一化；同时采用LeakerRule激活函数，相比Rule函

数，LeakerRule函数更接近人类大脑神经元的特征。

（3）在kaggle上测试新的网络在CamVid的准确率，比较这个网络的优劣性，最后将此网络用于训练自己的数据，来满足系统的要求。

本系统所需要的硬件设施要求较低，因此对于普通用户，都可以将自己准备的数据集在此网络上进行训练，不需要消耗很多CPU时间用于数据计算。

本文所设计的系统主要功能是为了让用户体验语义分割的工作过程，体验语义分割所要完成的事务。系统存在一些不足，实现的功能相对较少，用户无法直接观察每层网络所提取出来的特征。这些都是需要改进的地方。

现在我们的生活处处被人工智能所环绕，尽管目前能达到的智能水平离通用人工智能还有一段距离，我仍坚定地相信人工智能时代已经来临。

参考文献

1. 肖朝霞, 陈胜. 图像语义分割问题研究综述.软件导刊, 2018, 17(08):6-8+12.
2. Nekrasov V , Ju J , Choi J . Global Deconvolutional Networks for Semantic Segmentation[J]. 2016.
3. 周莉莉, 姜枫.图像分割方法综述研究.计算机应用研究, 2017, 34(07):1921-1928.
4. 卢飞. 基于区域与深度残差网络的图像语义分割[D]. 2019.
5. 顾攀, 张烽栋. 基于神经网络的图像弱监督语义分割算法[J]. 计算机应用与软件, 2018, 035(002):284-288.
6. 郭丽丽, 丁世飞. 深度学习研究进展[J]. 计算机科学, 2015, 042(005):28-33.
7. Engineering; Researchers from Xiangtan University Report New Studies and Findings in the Area of Engineering (Multi-modal Weights Sharing and Hierarchical Feature Fusion for Rgbd Salient Object Detection)[J]. Journal of Engineering,2020.
8. 陈书贞, 李光耀, 练秋生. 结合图像的局部相关性及非局部相似性的多尺度分块压缩感知[J]. 燕山大学学报, 2013, 000(006):547-553.
9. 谢宝剑. 基于卷积神经网络的图像分类方法研究[D]. 2015.
10. 常祥, 杨明. 基于改进的卷积神经网络的图像分类性能[J]. 重庆理工大学学报, 2017, 031(003):P.110-115.
11. 刘进锋,郭雷. 神经网络前向传播在GPU上的实现[J]. 微型机与应用(18):69-71,75.
12. 刘小文,郭大波,李聪.卷积神经网络中激活函数的一种改进[J].测试技术学报,2019,33(02):121-125.
13. 赖策.卷积神经网络中的激活函数分析[J].科学技术创新,2019(33):35-36.
14. 薛景浩, 章毓晋, 林行刚. 图像分割中的交叉熵和模糊散度算法[J]. 电子学报, 1999(10):132-135.
15. 刘荣.人工神经网络基本原理概述[J].计算机产品与流通,2020(06):35+81.
16. Jinchuan Qian,Li Jiang,Zhihuan Song.Locally Linear Back-propagation Based Contribution for Nonlinear Process Fault Diagnosis[J].IEEE/CAA Journal of Automatica Sinica,2020,7(03):764-775.
17. 常永虎,李虎阳.基于梯度的优化算法研究[J].现代计算机,2019(17):3-8+15.
18. 任义丽,罗路.卷积神经网络过拟合问题研究[J].信息系统工程,2019(05):140+142.
19. Junhao Wu,Ziyu Gan,Wei Guo,Xuan Yang,Adan Lin. A fully convolutional network feature descriptor: Application to left ventricle motion estimation based on graph matching in short-axis MRI[J]. Neurocomputing,2020,392.
20. Biswapriya B Misra. Data normalization strategies in metabolomics: Current challenges, approaches, and tools. 2020, 26(3):165-174.
21. Yunyun Yang,Chong Feng,Ruofan Wang. Automatic segmentation model combining U-Net and level set method for medical images[J]. Expert Systems With Applications,2020,153.

附录

图形用户界面

from PIL import Image,ImageTk

import tkinter as tk

from tkinter import filedialog

file\_name=''

def biaoshi(path):

new\_model = tf.keras.models.load\_model('E:/unetv11.h5')

image = tf.io.read\_file(path)

image = tf.image.decode\_png(image,channels=3)

image = tf.cast(image,tf.float32)

image = image/127.5-1

image = tf.expand\_dims(image,axis=0)

pred = new\_model.predict(image)

pred = tf.argmax(pred,axis=-1)

pred = pred

pred = tf.squeeze(pred)

pre = pred.numpy()\*255

cv.imwrite("E:/one.png",pre)

path = "E:/one.png"

return path

def take\_picture(): # 打开图片函数

global file\_name

fileName =filedialog.askopenfilename(filetypes=)

file\_name = fileName # 保存到全局变量里其他函数可以调用

#####加载图片显示到label\_1里

pilImage1 = (Image.open(fileName))

size = pilImage1.size

if size/size<1: #这几行if判断作用是resize选择的图片，使其比例不畸变地放入500\*500的框中

y = size\*500//size

pilImage1 = pilImage1.resize((500,y))

else:

x = size\*500//size

pilImage1 = pilImage1.resize((x,500))

tkImage1 = ImageTk.PhotoImage(image=pilImage1) #转化成tkinter的图片

label\_1.config(image=tkImage1) #在label\_1里更新图片

label\_1.image=tkImage1

def update\_image():

global file\_name

print(file\_name)

path1 = biaoshi(file\_name) #得到分割后的图片路径

pilImage1 = (Image.open(path1))

size = pilImage1.size

if size/size<1: #这几行if判断作用是resize选择的图片，使其比例不畸变地放入500\*500的框中

y = size\*500//size

pilImage1 = pilImage1.resize((500,y))

else:

x = size\*500//size

pilImage1 = pilImage1.resize((x,500))

tkImage1 = ImageTk.PhotoImage(image=pilImage1)

label\_2.config(image=tkImage1)

label\_2.image=tkImage1

if \_\_name\_\_=='\_\_main\_\_':

window = tk.Tk()

window.title('分割') #程序名字

window.geometry('1000x600') # 这里的乘是小x

choice = tk.Frame(window, width=1000, height=100, bg='LemonChiffon') # 放按钮的frame bg='blue'是背景颜色，删掉就是默认颜色

choice.pack(side='top')

choice.pack\_propagate(0) # 固定大小

show\_frame = tk.Frame(window, width=1000, height=500, bg='black') # 放图片的frame

show\_frame.pack(side='top')

show\_frame.pack\_propagate(0)

show1 = tk.Frame(show\_frame, width=500, height=500, bg='LightBlue') # 放第一个图片的frame

show1.pack(side='left')

show1.pack\_propagate(0)

show2 = tk.Frame(show\_frame, width=500, height=500, bg='Azure') # 放第一个图片的frame

show2.pack(side='left')

show2.pack\_propagate(0)

label\_1 = tk.Label(show1)

label\_1.pack(side='top',pady=10)

label\_2 = tk.Label(show2)

label\_2.pack(side='top',pady=10)

fontsize=15 #设置字体大小

b\_1 = tk.Button(choice, text='选择图片', width=fontsize, height=1, command=take\_picture) # 定义按钮1

b\_1.pack(side='left', padx=200)

b\_2 = tk.Button(choice, text='处理图片', width=fontsize, height=1, command=update\_image) # 定义按钮2

b\_2.pack(side='right', padx=200)

window.mainloop()

**毕业设计评审意见表A**

|  |  |  |  |
| --- | --- | --- | --- |
| **毕业设计题目** | 基于深度学习的几何体语义分割系统设计 | | |
| **学生姓名** | 韩林凯 | **专业班级** | 计算机科学与技术2016级4班 |
| **指导教师评语:**  该同学设计论文从深度学习对于更高需求的图像处理这一研究点出发，针对已有的全卷积网络的语义分割算法中存在的问题设计网络，下采样经过不同程度的卷积实现对目标的深度理解，再由上采样恢复目标原状，将多尺度特征融合，生成密集的特征图后，将输出特征映射送入soft-max分类器，完成对图形的特性处理和分类。实现了基础的图片语义分割处理，在人脸识别、自动驾驶等涉及图片处理的人工智能研究中有一定应用价值。该同学有较强的分析和解决问题的能力，高质量的完成了论文任务书内规定的内容。  毕业设计什么书结构组织合理、条理清晰、内容完整、语言通顺、实验结果清晰、图表规范，达到了本科毕业论文的要求，同意参加毕业论文答辩。  不足：  （1）系统实现的功能相对较少，用户无法直接观察每层网络所提取出来的特征。  （2）二级、三级标题的字体格式需要统一。图的大小需要微调，保证足够清晰。图下的注释字体不可用截图。 | | | |
| 建议成绩：95  指导教师（签字）： 任洪娥 教授    2020 年 6 月 1日 | | | |

**毕业设计评审意见表B**

|  |  |  |  |
| --- | --- | --- | --- |
| **毕业设计题目** | 基于深度学习的几何体语义分割系统设计 | | |
| **学生姓名** | 韩林凯 | **专业班级** | 计算机科学与技术2016级4班 |
| **评阅人评语:**  该同学针对当前深度学习以及全卷积神经网络中的研究热点，设计符合当下人工智能对图像处理的要求，将复杂的场景信息转换为按特性分类的像素信息，选题准确，有一定的参考价值。  毕业设计中使用的全卷积神经网络能够识别任意大小的图像，而且相对于卷积神经网络更加高效，避免了重复存储和计算卷积的问题，同时针对设计中全卷积网络处理的缺点进行了改进，利用数据标准化、拼接结构、数据增强来详细的描述图片细节，得到更加清晰地输出结果。设计的系统用户界面简单清晰，方便操作，由选择图片和处理图片两个选择框构成，最终的显示结果可以直观的放在一起进行对比。  毕业设计内容逻辑清晰完整，能够发现并有效处理遇到的问题，查阅大量中外文献自主学习，实验过程及实验结果清楚完整，对于图片的语义分割处理有较为系统的认知。设计说明书结构组织合理，内容完整，语言规范，达到了本科毕业设计的要求，推荐该同学参加毕业设计答辩。  不足：  （1）摘要及正文中的英文格式须为罗马字体。框图中字体须为宋体五号  （2）参考文献中的有些文献格式需要调整，以及文献引用的并不完整（如9）  （3）正文中的表格须为三线表，不可截图（如表2-1） | | | |
| 建议成绩：94  评阅人（姓名、职称）: 高心丹 副教授    2020 年 6 月2 日 | | | |

**毕业设计评审意见表B**

|  |  |  |  |
| --- | --- | --- | --- |
| **毕业设计题目** | 基于深度学习的几何体语义分割系统设计 | | |
| **学生姓名** | 韩林凯 | **专业班级** | 计算机科学与技术2016级4班 |
| **评阅人评语:**  该同学的毕业设计内容详实合理，选题符合本专业当前的研究热点及方向，在对三种语义分割方法分析及基础上，设计全卷积神经网络几何体语义分割系统。  设计首先在CNN基础上，将其最后一层的全连接层转换为全卷积层，然后分析FCN的工作原理，得出分析率较低是因为输入经过几个卷积和池化后，输出特征图的分辨率下降。为了解决分析率低的问题，设计中取消了全连接层，从而使上采样不需要学习，生成稀疏特征映像，然后使用可训练的卷积核进行卷积操作，生成密集的特征图。最后进行像素级分类。通过以上步骤该设计改进了全卷积神经网络直接预测分辨率较低的问题，实现了无需提高配置即可增加精度的目的。  该同学的设计思路清晰、结构合理，能够发现并处理设计中遇到的问题，改进设计中的不足，提高了设计可行性。该设计行文规范，语言通顺流畅，实验结果详细，达到了本科毕业设计的要求，推荐该同学参加毕业设计答辩。  不足：   1. 正文中的格式个别存在瑕疵，有的公式、表格和图片格式需要调整。 2. 设计结论中对自身工作阐述较少，对于前提背景以及基础过于冗余。 | | | |
| 建议成绩：95  评阅人（姓名、职称）: 李艳娟 副教授    2020 年 6 月2 日 | | | |

答辩委员会(教师姓名、职称)：

答辩委员会意见:

（答辩委员会成员的签字要手写）

毕业设计成绩：