

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘要

斜视是一种常见且严重的眼科疾病，及时检测斜视有助于其预后和治疗。远程医疗具有极大的潜力，可以缓解眼科疾病诊断不断增长的需求，是实现及时斜视检测的有效方法。随着计算机视觉在医学领域的不断发展，特别是深度学习对医学图像的影响，产生了积极的作用。但是深度学习在斜视检测领域的研究为数较少，主要的局限在于数据集的缺乏，以及专家医生标注的困难等。

本论文提出了可以进行斜视检测的框架 **RF-CNN**，并开发了实现远程医疗的界面软件。同时我们建立了经过专家医生标注的斜视数据集，实现自动斜视检测。针对软件使用过程中的缺点，在框架的基础上对网络进行了轻量化，提出了 **RF-CNN Lite**。本文首先建立基于 **R-FCN** 和搭建的卷积神经网络组成的斜视检测模型，然后针对该模型进行了神经网络轻量化，包括使用计算量更小的深度分离卷积替代原有的普通卷积操作，把模型的参数量和运算量大为减小，最后通过 **PyQt** 开发界面软件实现远程斜视检测。本文的创新点包括：针对斜视识别提出了斜视识别框架；在该框架上对神经网络进一步轻量化；实现了远程医疗的斜视检测系统。

关键词：斜视识别；远程医疗；轻量化网络；反残差块；深度分离卷积

Abstract

Strabismus is a commonly encountered and serious ophthalmologic diseases, timely detection of strabismus contributes to its prognosis and treatment. Telemedicine has great potential and can alleviate the growing demand for the diagnosis of ophthalmic diseases, and it is an effective method to achieve timely strabismus detection. With the continuous development of computer vision in the medical field, especially the impact of deep learning on medical images, it has produced a positive effect. However, there are few studies on deep learning in the field of strabismus detection. The main limitations are the lack of datasets and the difficulty of labeling by experts.

This paper proposes an RF-CNN framework for strabismus detection, and develops interface software for telemedicine. In the meanwhile, we establish a strabismus dataset marked by an expert doctor to achieve automatic strabismus detection. In view of the disadvantages in the use of software, the network was lightened on the basis of the framework, and RF-CNN Lite was proposed. This paper first establishes a strabismus detection model based on R-FCN and a built-in convolutional neural network, and then performs lightweighting of the neural network for this model, including the use of a smaller computational separable convolution to replace the original ordinary convolution operation , Greatly reduce the parameter and calculation of the model, and finally realize the remote strabismus detection by the development of PyQt interface software. The innovations of this paper include: the strabismus recognition framework is proposed for strabismus recognition; the neural network is further lightened on this framework; and the strabismus detection system for telemedicine is realized.

Keywords: Strabismus Detection; Telemedicine; Lightweight Network; Inverted Residual Block; Separable Convolution

目 录

摘要	I
Abstract.....	II
目 录	I
第 1 章 绪论	1
1.1 课题研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.3 主要研究内容.....	3
1.4 章节安排.....	3
第 2 章 深度学习模型理论	5
2.1 深度学习的概念和发展.....	5
2.2 卷积神经网络基本结构.....	6
2.3 经典卷积神经网络.....	8
2.4 常用的深度学习框架.....	11
2.5 本章小结.....	12
第 3 章 斜视检测模型 RF-CNN 的创建	14
3.1 斜视图像的检测.....	14
3.2 R-FCN	15
3.2.1 目标检测网络简介.....	15
3.2.2 模型结构	16
3.2.2 训练细节	18
3.3 用 R-FCN 对斜视图像进行眼部区域定位.....	18
3.3.1 训练细节	18
3.3.2 网络及训练的实现.....	19
3.4 用 CNN 对裁剪眼部区域进行斜视识别	20
3.4.1 模型结构	20
3.4.2 数据预处理和网络实现.....	21
3.4.3 斜视检测结果.....	21

3.5 本章小结	22
第 4 章 轻量化斜视检测模型 RF-CNN Lite 的构建	23
4.1 神经网络轻量化的架构	23
4.2 神经网络轻量化的架构	24
4.2.1 权重裁剪	24
4.2.2 权重量化	24
4.2.3 低秩分解	24
4.2.4 知识蒸馏	25
4.3 基于神经网络架构搜索的自动化神经网络架构设计	25
4.4 人工设计轻量级网络	25
4.4.1 使用小卷积核代替大卷积	25
4.4.2 限制中间特征通道数	26
4.4.3 分解卷积计算	26
4.5 用反残差结构对模型进行轻量级设计	29
4.5.1 眼部区域检测网络 R-FCN 轻量化设计	29
4.5.2 斜视识别网络轻量化设计	32
4.6 本章小结	33
第 5 章 总结与展望	34
5.1 本文总结	34
5.2 未来工作和展望	34
参考文献	35
攻读学位期间主要研究成果	39
致 谢	40

第 1 章 绪论

1.1 课题研究背景与意义

斜视是一种常见且严重的眼科疾病，它的临床症状为双眼未准确对准同一位置^[1]^[2]，这种情况通常发生在儿童时期。斜视是由视神经、大脑或者眼外肌相关问题引起的^[3]。其存在的风险因素包括早产和家族遗传^[4]。斜视使得人们遭受痛苦，严重影响了正常的生活。斜视能够阻止大脑将从双眼接收到的图像进行融合^[5]，从而导致弱视。没有得到及时治疗的弱视眼睛可能会退化，最终导致失明^[6]。同样地，在斜视人群中复视和深度视觉困难更为常见。从而，斜视的诊断和治疗变得越来越重要，其中斜视检测是第一步也是最重要的一个步骤。

传统的斜视检测是在医院进行的。医生使用 Hirschberg 检测法^[7]来判断患者是否斜视，具体做法为：将一束光束发送到患者的眼睛中，以验证每只眼睛的反射是否位于两个角膜上反射的同一位置。它们由专业的眼科医生来进行诊断，这将使诊断变得昂贵，从而使人们无法进行专业检查。此外，眼科医生根据他们的经验做出决定，因此诊断结果是主观的。并且该方法耗时耗力，对专业医生的资源需求较高，不能很好地满足庞大的斜视患者人群。

实现远程医疗的斜视检测的研究意义总体来说可以归结为以下方面。相比较于传统的斜视检测的方法，可以很大程度地节省时间和成本，借助于远程医疗的方法，斜视患者不再需要实地到医院给专业医生进行诊断，只需要对眼部进行图片拍摄，使得医生资源需求和诊断耗时大大节省。另外，使用斜视检测系统进行诊断，可以减少专业医生决策的主观性，从而减少人为错误。

自从 2012 年 AlexNet^[8]在 ImageNet 竞赛中识别准确率以大幅度领先其他方法后，深度学习被广泛应用于各种图像处理任务中，并取得了突出的效果，其中包括图像识别、目标检测、图像分割等领域。因此，在实际应用中，考虑用深度学习算法实现斜视检测的目标，即斜视检测的任务。

另一方面，当深度学习算法取得的准确率越高时往往需要耗费越大量的计算资源，包括存储空间和运算时间，这限制了深度学习在移动端和嵌入式端的应用，在配置较低的电脑中也极大消耗电脑的计算资源。因此，如何对深度学习模型进行轻量化，并加速其运算时间，也是一个具备挑战性的问题。深度学习在提升准确率的学术研究上已经硕果累累，而算法模型的轻量化的研究还处于不断探索的阶段，目前尚未在领域内有权威的算法，因而具有研究意义。

1.2 国内外研究现状

由于斜视检测具有重要的意义，在学术中有所研究，但总体来说数量并不多。其中，因为斜视数据的特殊性，涉及到患者的医疗数据隐私，这种限制一直未打破，加上斜视数据需要专业医生的标注，需要耗费大量的精力，因此在学术界并没有公开的斜视数据集。

在基于传统的图像识别方法中，主要利用了准确的瞳孔边缘定位以及反射的光亮特征。瞳孔边缘定位是指利用瞳孔的圆形特征得到中心点，并且得到亮点位置，其中主要方法采用边缘检测算法如 Canny^[9]，以及形状特征检测算法如霍夫变换^[10]等对其位置进行检测。其他提出的图像方法也是围绕此两种特征，并结合机器学习算法，通过学习提取更加抽象的特征，或者提出更加高级的特征，通过学习特征来完成斜视检测的模式识别任务。Almeida^[11]利用数码相机整合了多种图像处理方法，通过有监督学习 SVM 学习瞳孔中心点以及光亮点之间的位置距离特征进行斜视识别。Valente^[12]利用数字视频结合 cover 测试对斜视进行识别。Helveston^[13]提出了一种方法，在没有专业医生的情况下使用医学方法来治疗斜视。为此，数码相机被用于捕获患者的图像，并且使用计算机将图像通过电子邮件发送给医生，以便可以对图像进行分析。

随着深度学习在不同的图像任务得到广泛应用，今年来也有提出用卷积神经网络进行斜视识别的文章。Chen^[14]使用眼动追踪仪器和深度学习算法，通过卷积神经网络提取眼动追踪数据的抽象特征，最后对斜视与否进行识别。但是此方法只能对满足拥有专业设备条件的医生有用，并且获取眼动追踪数据耗时耗力，不够便利，推广性较差。目前并没有结合深度学习算法针对拍摄图片进行斜视识别的研究。此外，以上方法均是针对算法解决斜视检测的研究，目前针对应用的轻便性的研究较少，对于斜视检测的研究暂时没有。

另一方面，由于神经网络中参数量及计算量大，其占用的计算资源较大，包括储存空间和运算时间，这往往也是在实际应用中制约其发展的瓶颈^[15]。因此，针对神经网络模型的轻量化，显得尤为重要，特别是对端上的应用领域中。其中轻量化的操作也有很多相关领域的不同方法，包括有人工设计轻量化神经网络模型、神经网络模型的压缩以及基于神经网络架构搜索(Neural Architecture Search, NAS)的自动化神经网络架构设计三种方法。人工设计轻量化神经网络模型是指设计更高效的网络计算方式，其针对的是卷积的计算方法。其中在轻量级神经网络架构的设计领域已经取得了一定的成果，如 Google 在 2016 年提出的通过深度可分离卷积代替标准卷积的思想，设计了轻量级网络架构 MobileNet V1^[16]，使用深度卷积(Depthwise Convolution)和逐点卷积(Pointwise Convolution)两个操作，极大地降低了卷积的计算量。旷视科技提出

了 ShuffleNet V1^[17], 其结合了逐点卷积核通道混洗操作, 同样地取得了很好的效果。另外, Google 提出的 Deep compression^[18]通过阈值来评估权重的重要程度, 删去低于阈值的权值。并通过剪枝、权值的共享和量化以及哈弗曼编码的操作来达到卷积神经网络模型的压缩目的。为了使神经网络模型具备更高的压缩比, 以保持更好的神经网络的性能, AMC(AutoML for Model Compression, AMC)^[19]使用了强化学习方法来自动学习模型压缩策略。在神经网络架构搜索领域, MnasNet^[20]和 NasNet^[21]同样地也是使用强化学习方法学习神经网络架构搜索策略, 以达到在移动端设备上的轻量化神经网络的自动化构建。与 MnasNet 不同的是, NasNet 提出了基于块的搜索空间, 能够使得搜索的速度大大增加。

1.3 主要研究内容

本论文聚焦于斜视检测这一实际应用问题, 首先尝试把深度学习的算法应用于斜视检测, 并封装成软件提供给专业医生进行实际场景实验, 然后尝试应用卷积神经网络的轻量化技术, 对斜视检测的神经网络模型进行轻量化。主要研究内容如下:

- (1) 研究深度卷积神经网络在斜视检测上的应用。
- (2) 研究轻量化神经网络的方法, 介绍人工设计轻量化神经网络模型的理论和应用。
- (3) 主要介绍深度分离卷积的操作及意义, 从空间、时间和准确率层面对比验证轻量化算法的有效性。

1.4 章节安排

根据上面的主要的研究内容, 本文的组织结构安排如下:

第 1 章: 绪论。简述斜视检测的背景意义, 简述斜视检测算法以及神经网络模型轻量化算法的研究现状, 并要概述文章研究内容。

第 2 章: 深度学习模型理论。介绍深度学习的基本理论与发展, 卷积神经网络的基本组成结构, 经典的卷积神经网络以及常用的深度学习框架。

第 3 章: 斜视检测模型的构建。首先介绍了用于图像检测与识别的经典模型 R-FCN 以及手工设计的 CNN 组成的 RF-CNN 框架, 然后介绍基于 PyTorch 实现 RF-CNN 并用于斜视检测的实验过程, 得到基于 RF-CNN 的斜视检测实验结果。

第 4 章: 斜视检测的模型轻量化。针对斜视检测这一具体问题, 提出了可应用的模型轻量化算法框架。介绍了深度分离卷积算法及其思想, 针对 RF-CNN 进行神经网络模型的轻量化得到 RF-CNN Lite, 将其应用于斜视检测。并介绍本实验所采用的评价指标, 对实验结果进行分析。

第 5 章：总结与展望。该部分主要是对本文研究工作和实验的总结，同时也是对以后的研究工作的展望。

第 2 章 深度学习模型理论

在 2012 年 AlexNet 在 ImageNet 竞赛中以非常大的优势超出其他方法，由此开始深度学习便获得了快速的发展，在图像领域的其他任务中，如目标检测、图像分割等方向大放异彩，广泛应用于安防、教育、汽车自动驾驶、医疗及新零售等领域，共同组成了当下的人工智能行业。然而早在上个世纪 80 年代，深度学习的基础算法即是神经网络就已经被提了出来，从那以后 30 年来不断曲折发展，直到现在成为了热门的研究领域。相较于传统的机器学习算法，深度学习由于其可解释性较差一直被人们所诟病，但这并没有妨碍其在各个领域中的应用，以及其相较于其他传统算法显示出的巨大的性能优势。在计算机视觉领域中的表现尤为突出。下文我们将介绍深度学习的基本理论及其发展。

2.1 深度学习的概念和发展

深度学习的概念是在人工神经网络的基础上得到发展而来，而最早的人工神经网络则可以追溯到 1943 年提出的人工神经元模型。如图 2-1 所示，该模型将神经元共分成三个过程：输入信号的加权、求和以及非线性激活^[22]。在 1958 年，在人工神经元基础上发展的感知机^[23]被提了出来，该模型在训练数据集中利用了梯度下降的方法能够不断更新权值。然而，接下来在 1969 年学者^[24]提出了感知机存在无法解决非线性问题，对于最简单的异或（XOR）问题也束手无策，只能够处理线性问题。一直到上个世纪 80 年代，Hinton 提出了多层感知机以及反向传播算法^[25]。1989 年，LeCun 利用卷积神经网络应用于手写数字识别，并得到了较好的识别准确率^[26]。但由于神经网络存在可解释性较差的缺点，以及在网络层数较高的情况下存在梯度消失等问题，学术界对于神经网络的研究热度慢慢减退。

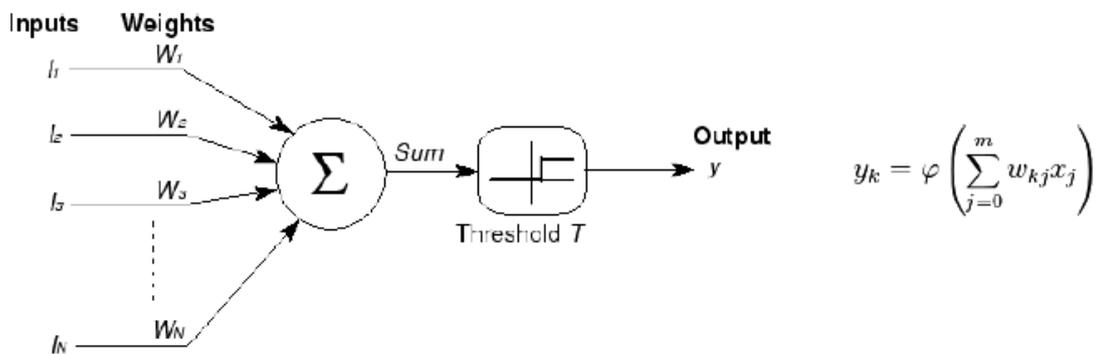


图 2-1 人工神经元模型

一直到 2006 年，依然是 Hinton，他作为第一个提出了深度学习概念的学者，同

样地，这一年也被称为是深度学习元年。在研究中 Hinton 使用了无监督预训练对权值进行初始化，并使用有监督训练对神经网络进行微调，提出了该套训练方法。并在此后一直到 2012 年间，学术界对神经网络的研究热情开始恢复，并得到了快速的发展。2012 年，ReLU 非线性激活函数、以及 Dropout 等方法在 AlexNet 中得到了应用，并以极大优势的识别准确率超过了其他传统方法赢得了当年的 ImageNet 竞赛。至此之后深度学习的研究及其应用便呈井喷之势爆发，成为了人工智能的基本算法。一系列的著名的经典神经网络结构被陆续提出，其中包括 VGG-Net^[27]、Inception-Net^[28]、ResNet^[29]，到后来提出的轻量级网络 MobileNet、SqueezeNet^[30]、ShuffleNet^[31]等，特别是在计算机视觉领域，深度学习的研究被学术界所追捧，得到了极大的发展。

2.2 卷积神经网络基本结构

卷积神经网络是构成深度学习的主要算法，简称 CNN(Convolutional Neural Network)，其主要由三种结构组成：卷积层、池化层和非线性函数。

卷积层的操作如图 2-2 所示，输入为原图像或者特征图，通过滑动窗口的操作，卷积核与输入矩阵中的每一个等大的区域的对应元素一一相乘并进行相加，由此得到了输出矩阵对应的每一个元素。假设输入为 I ，卷积核为 K ，输出为 S ，卷积操作的数学公式表示为：

$$S(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2-1)$$

卷积层的核心思想在于局部连接以及权值共享。局部连接的概念是与传统神经网络的全连接概念相对的，全连接层的每一个权值都会与每一个输入节点进行相乘，而局部连接是指卷积核每次只和输入节点的局部进行卷积运算。权值共享是指当卷积核与不断进行滑窗的输入数据进行卷积时，卷积核的权值一直保持不变。

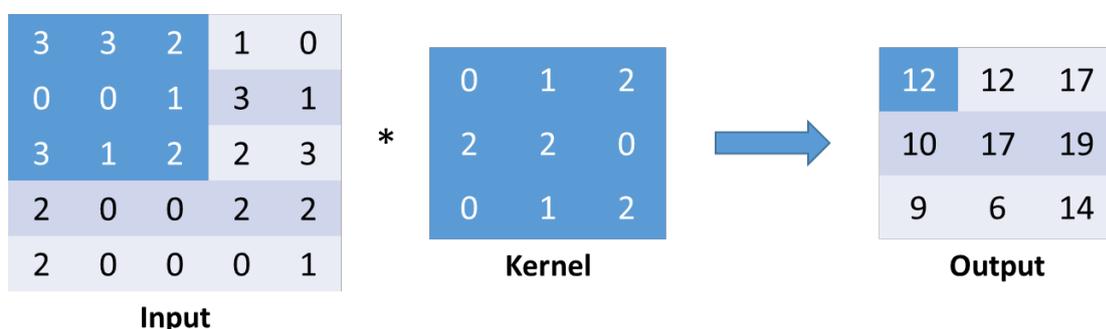


图 2-2 卷积操作

池化层与卷积层同理，同样是以一个滑动窗口对输入的局部区域数据进行计算，不同的是池化层不存在权值参数。在池化操作中最常用的是最大池化，其就是选取窗

口内最大值，最大池化的设计初衷在于选择最大的特征，从而去掉其他没那么重要的特征，可以达到减少计算量并加速收敛的目的。池化操作另一种常见池化方式为平均池化，即计算窗口内的平均值，该池化操作在最后一层卷积层之后比较常见，用以替换原先的全连接展开。

上述的卷积层和池化层表示的是线性运算，要对神经网络赋予非线性能力，从而得到更好的表征能力，需要引入可以实现非线性表达的函数。当前最常见的非线性函数为 ReLU (rectified linear unit)，在 ReLU 之前最为常用的非线性函数为 sigmoid，sigmoid 的缺点在于其过渡区域小，很容易就会饱和，并且在深层的神经网络中会存在梯度消失的现象。但是在二分类或者多标签分类问题中，在最后一层神经网络还是会使用 sigmoid，将数值映射至[0, 1]。Sigmoid 与 ReLU 的数学表达式以及函数图像如下图所示：

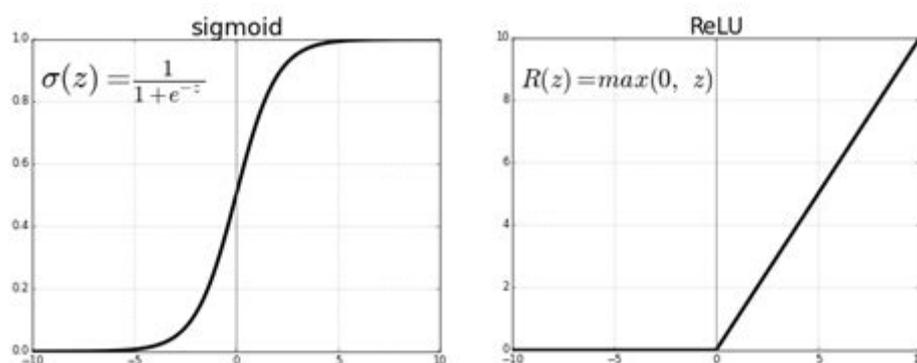


图 2-3 sigmoid 及 ReLU 激活函数

此外，与卷积层、池化层以及非线性函数这三种基本结构外，在早期的神经网络中全连接层同样地被广泛使用。全连接层指的是在一层的神经网络的神节点都与下一层的所有节点相连接，如下图所示。全连接层网络结构由早期的多层感知机发展而来，但是全连接层存在参数量过大、训练困难、容易陷入过拟合等缺点，在现在新提出的神经网络中多数使用平均池化层用以代替全连接层，由此可以带来减少参数数量的效果。

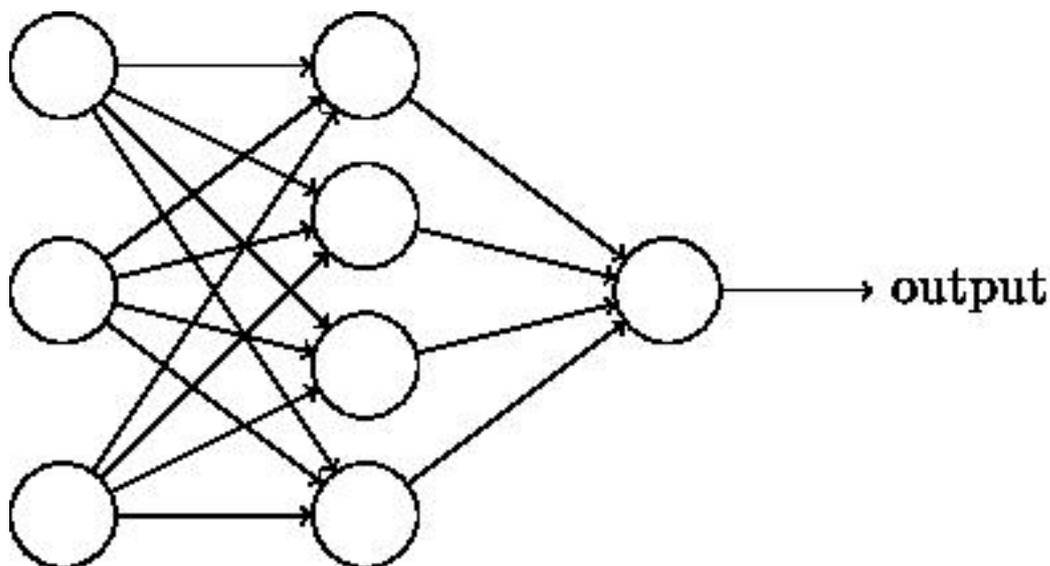


图 2-4 全连接层

2.3 经典卷积神经网络

深度学习在 2012 年后呈现爆发式发展，特别地在计算机视觉领域中更是提出了很多的著名经典卷积神经网络，在不同的图像任务中得到广泛的应用，在图像识别方向中尤为突出，并且成为深度学习领域研究的基准。

LeNet-5 作为第一个经典的卷积神经网络，总共包含 7 层，其中包括 2 个卷积层，2 个下采样层，2 个全连接层以及 1 个为径向基函数的全连接层。如下图所示，输入为大小为 32×32 的图片，卷积层采用的卷积核都为 5×5 ，下采样层则是采用平均池化操作。LeCun 在 1998 年提出的 LeNet-5 网络结构满足卷积神经网络所需要的基本要素，并在手写字母识别应用中取得不错的效果，当时并没有引起足够的重视是因为存在其他方法也可以与卷积神经网络的结果相当。

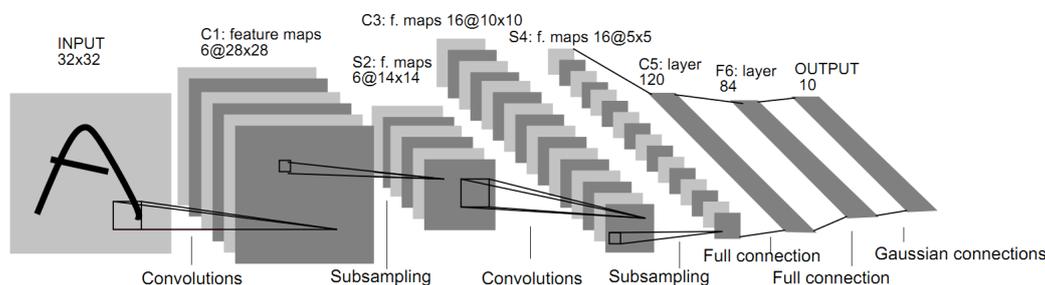


图 2-5 LeNet-5 结构

2012 年的 AlexNet 的提出标志着深度学习的爆发。该网络的结构一共有 8 层，其中包含 5 个卷积层，3 个全连接层，和最大池化层。AlexNet 在 ImageNet 竞赛中大放异彩，该网络对卷积神经网络进行了以下改进：1) 在激活函数方面使用 ReLU 代替

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 VGG-Net 组成

GoogLeNet，一共包括 4 个版本的网络[28] [32] [33] [34]，是由 Google 提出来的系列网络。与之前提到的网络相比较，GoogLeNet 的网络层数增加至上百层，在之前的准确率基础上进一步提高，创新点包括有：如图所示，提出了 Inception 模组，核心点在于将不同大小的卷积核进行组合使用，这也是体现了多尺度的思想；2) 将 3*3 卷积核分解成 3*1 卷积核和 1*3 卷积核以降低参数量；3) 提出了 batch normalization 的方法，该方法可以对网络训练的收敛可以起到加速的作用，并带有正则化作用，起到减少过拟合的效果，在往后的发展中 batch normalization 得到了大量的应用，收到了很好的效果。

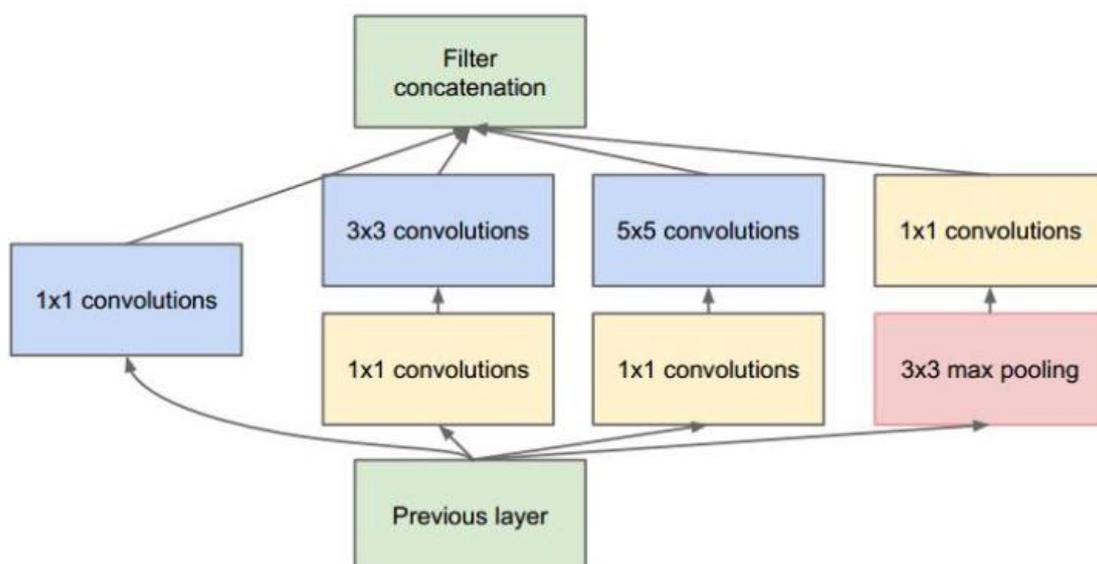


图 2-8 Inception 模块结构

ResNet，由残差结构组成的网络，由何恺明提出。如图 2-9 所示，残差结构是指将输入层和输入层直接进行相加，因为在深层网络中存在梯度爆炸或者消失从而难以训练的问题，而残差结构很好地解决了这个问题，由于该结构具有简单且实用的特性，在后来的研究中也广泛引用。

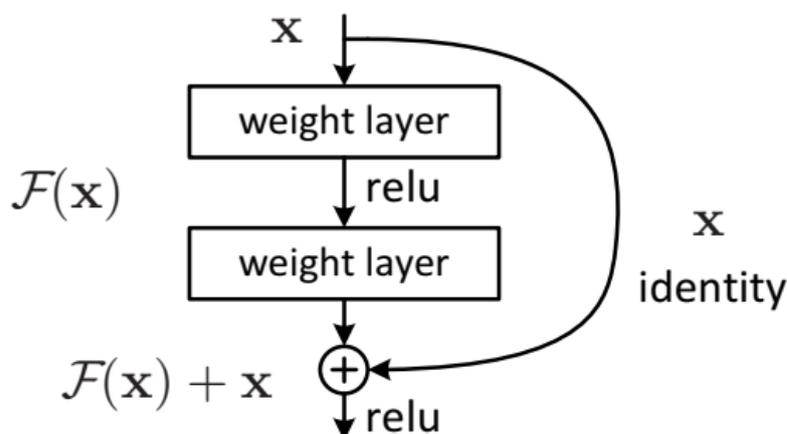


图 2-9 残差结构

2.4 常用的深度学习框架

深度学习的快速发展出了算法理论之外还有一个很重要的原因是 GPU 加速技术的支持和代码框架的不断发展，GPU 的加速技术可以极大地减少神经网络的训练时间，深度学习代码框架的不断更新迭代使得研究人员可以更加专注于在算法本身的设计，而不需要再代码开发和维护的过程中花费更多的时间。下面对几个主流深度学习

框架的特点进行简单的介绍，包括 TensorFlow、Caffe、Caffe2、PyTorch、MXNet 等。

TensorFlow 是在 2015 年 Google 发布的深度学习开源框架。它的特点是通过数据流图（Data Flow Graph）的方式进行计算，计算图中的节点代表数学运算，数据形式则是以张量（tensor）的形式存储并进行交互。TensorFlow 的底层由 C++ 进行开发，提供了 Python、C++、Java 等编程接口。该框架能够支持分布式，可以方便部署在一台或多台 CPU、GPU 机器中，并且能够实现自动求导，从而不需要使用者单独实现反向传播并求梯度的功能。但是该框架也存在缺点，由于其实现过于底层化，导致上层代码在调用方式上不够抽象，使得使用性复杂繁琐。但是 TensorFlow 目前仍然是使用人数最多且社区最庞大的深度学习框架。

Caffe，全称为 Convolutional Architecture for Fast Feature Embedding，是由当时在加州大学伯克利分校就读的贾扬清主导开发并开源，是最早的几个深度学习框架之一。Caffe 同 TensorFlow 一样底层开发语言是 C++，该框架的优点是运行效率高，使用简单，使用时的网络结构定义为文本格式，因此使用起来较为方便。缺点是不够灵活，使用人员当定义新网络结构时需要在 C++ 源码中定义好 CPU 以及 CUDA 的运算规则，从而扩展性较差。

Caffe2 是继 Caffe 后升级的版本，其继承了 Caffe 的大部分的特点，并针对在应用和部署上的问题进行了解决。Caffe2 的优点包括轻量级、模块化以及高性能，可以支持不同的平台，因此对工业场景的适应性更强。缺点在于 Caffe2 的开发者社区还不成熟，使用文档不够规范化。

PyTorch，由 Facebook 开发的一套深度学习框架，一经发布就引起大量的关注。PyTorch 的前身为 Torch，后者是基于 Lua 语言进行开发，在现在 Python 主导深度学习的开发大环境下，Torch 较为小众。PyTorch 继承了 Torch 简洁高效易用的特点，在 Torch 的基础上对重构了新的模块，增加了自动求导的功能，最终实现了最为普遍的动态图框架。动态图机制和 TensorFlow 的静态图相比有很多优点，例如可以非常方便修改动态图中的节点，而在静态图中实现起来则复杂许多。PyTorch 凭借这种简洁易用的特点，使用人数不断增加，在学术界使用的频率甚至超过了 TensorFlow。

MXNet 是由 DMLC（Distributed Machine Learning Community）开发并开源的深度学习库，现在由亚马逊官方推荐。MXNet 的特点有如下，一是支持多个编程语言封装，包括 Python、R、C++、Go、JavaScript 等，二是支持分布式运算，并且分布式性能良好，对内存和显存优化相比其他深度学习框架做的更好。缺点在于文档并不规范，生态圈还不够完善。

2.5 本章小结

本章主要介绍了与深度学习相关的知识，包括深度学习的发展，卷积神经网络的

基本组成结构，经典的卷积神经网络，以及常用的深度学习框架。深度学习的基础为神经网络，由神经网络发展而来，在学术界的发展曲折，并在 2012 年开始迅速发展。深度学习的核心为卷积神经网络，其主要由卷积层、池化层和非线性激活函数组成。经典的卷积神经网络包括 LeNet-5、AlexNet、VGG-Net、Inception Net 和 ResNet 等。常见的深度学习框架有 TensorFlow、Caffe/Caffe2、PyTorch 和 MXNet 等。

第3章 斜视检测模型 RF-CNN 的创建

3.1 斜视图像的检测

斜视是常见而且严重的眼科疾病，它严重影响着患者的日常生活，常发生于儿童时期。能够及时地发生斜视并且实施对应的治疗可以很好地改善患者的生活状况。传统的方法是根据医学领域的检测法对患者进行斜视检测，其所体现的缺点也很明显，诊断效率低，耗费时间长，以及对专业医生的需求大使得传统的斜视检测弊端更加凸显。随着技术的进步，现在可以实现斜视检测的自动化，其中一种普遍的方法是通过相机获得斜视的患者照片后，对斜视进行检测。本论文解决的问题是对相机拍摄的患者图片进行斜视检测，如下图所示，根据输入的患者斜视图片，经过斜视检测系统，最后得到斜视检测的结果。

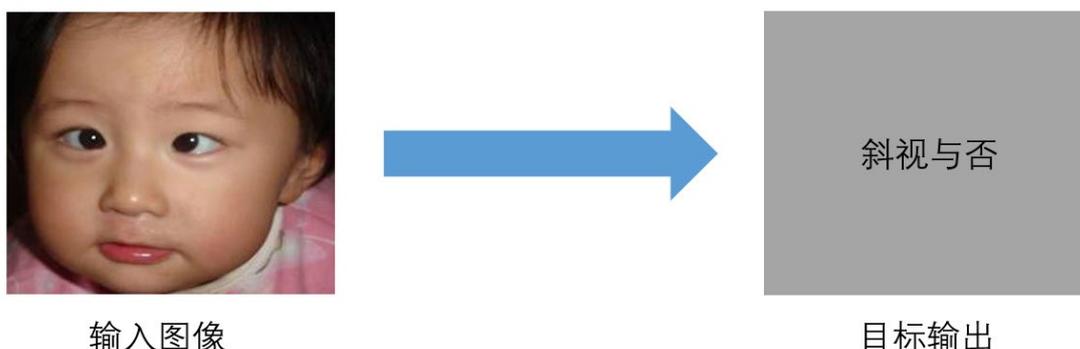


图 3-1 斜视图像检测的目标

斜视检测是一个特定场景下的图像识别问题，而识别模型的建立从数学角度而言，即是建立从输入图像到目标输出的映射关系，该问题能够看作是斜视与否分类问题。具体地，设 x 为输入的一张图像，存在 f 使得：

$$f(x) = \begin{cases} 1 & x \text{ 为斜视图像} \\ 0 & x \text{ 为正常图像} \end{cases} \quad (3-1)$$

建立模型的目标就是需要找到合适的关系 f 。在传统的斜视检测法中， f 为利用瞳孔的中心点与反射光点间的距离与斜视与否之间关系的特征，将输入的每一张图像分类为斜视与否。在卷积神经网络中，逐层的卷积会遍历输入图像的每个像素，用以提取像素及其周围领域特征，卷积的操作可理解为在提取特征时融合了传统意义上的形状及颜色等特征。

该论文提出了斜视检测模型 RF-CNN，该模型由两个子网络组成，第一个子网络

用于眼部区域的检测,第二个子网络用于对在第一个子网络裁剪的眼部区域进行斜视识别。在第一个子网络中, R-FCN^[35]用于对眼部区域的检测,该子网络目的是消除背景的影响,在第二个子网络中,卷积神经网络 CNN 用于对眼部区域的斜视识别。该斜视检测模型 RF-CNN 基于 PyQt 将斜视检测算法封装至界面软件供医院进行实验。

3.2 R-FCN

3.2.1 目标检测网络简介

在 2.3 节介绍的经典卷积神经网络全都应用于图像识别,这些经典网络使得深度学习在计算机视觉领域奠定了坚实的基础。同样在计算机视觉的其他领域中,深度学习也在其中大放异彩,提出了众多的流行的网络结构。其中如在图像分割领域的 FCN^[36]、Unet^[37]、Deeplab^[38]、Mask R-CNN^[39],人体姿态检测的 OpenPose^[40]、cnn^[41],人脸识别的 DeepFace^[42]。本论文所讨论的其中之一目标检测领域也有流行的网络结构及方法框架,分为两阶段(two-stage)方法,如 Faster-RCNN^[34]、R-FCN 等,以及一阶段方法(single-stage)如 YOLO^[43]、SSD^[44]等。

Faster-RCNN 网络是 Ross B. Girshick 在 2016 年提出 RCNN 目标检测系列网络的最新研究成果,RCNN 网络是深度学习在计算机视觉领域的首次成功应用,后来的深度学习方法基本都是基于 RCNN 系列网络所设计的网络架构。Faster-RCNN 提出了 RPN (Region Proposal Networks) 网络用于生成 region proposals 即区域候选框,通过一定规则设置不同尺度的锚点(Anchor)在 RPN 的卷积特征层提取候选框来代替 Selective Search 等传统的候选框生成方法,实现了网络的端到端训练,经过 RoI Pooling 层,该层收集输入的特征图及候选框,最后根据所得到的特征综合输出候选框的类别和检测框最终的精确位置。这也是第一个真正意义上的深度学习目标检测算法。

R-FCN 由当时在微软亚洲研究院工作的代季峰提出,该网络结构依然是基于 Faster-RCNN 检测框架的研究,但是对该框架的缺点进行了改进,缺点有计算量大,体现在基础网络的复杂度、候选框数量的多少以及分类和位置回归子网络的复杂度。R-FCN 提出了通过 position-positive score maps (位置敏感得分图)来解决这个矛盾,位置敏感得分图通过预测 RoI 中不同部位的类别投票表决产生该 RoI 的类别预测。R-FCN 沿用了 Faster RCNN 的框架结构,不同的是在 Faster R-CNN 的基础上通过引入位置敏感得分图,去掉了 RoI-wise subnetwork,直接在位置敏感得分图上利用 ROI Pooling 进行信息采样融合分类和位置信息。本工作选取 R-FCN 作为眼部区域检测的基础网络,对于 R-FCN 的结构将在 3.2.2 小节进一步介绍。

2015 年，随着 YOLO 算法的出现，深度学习目标检测算法开始有了两阶段方法和单阶段方法之分。区别于 R-CNN 系列为代表的两步检测算法，YOLO 舍去了候选框提取分支（Proposal 阶段），直接将特征提取、候选框回归和分类在同一个无分支的卷积网络中完成，使得网络结构变得简单，检测速度相较于 Faster R-CNN 也有了非常明显的提升。

SSD 于 2016 年提出，对 YOLO 进行了改进，达到了和两阶段方法相当的精度，同时又保持了较快的运行速度。SSD 也采用了网格划分的思想，和 Faster RCNN 不同的是它将所有的操作整合

在一个卷积网络中完成。为了检测不同尺度的目标，SSD 对不同卷积层的特征图像进行滑窗扫描，在前面的卷积层输出的特征图像中检测小的目标，在后面的卷积层输出的特征图像中检测大的目标。它的主要特点是：基于多尺度特征图像的检测：在多个尺度的卷积特征图上进行预测，以检测不同大小的目标，一定程度上提升了小目标物体的检测精度；借鉴了 Faster R-CNN 中的 Anchor boxes 思想，在不同尺度的特征图上采样候选区域，一定程度上提升了检测的召回率以及小目标的检测效果。

3.2.2 模型结构

首先介绍 R-FCN 的基础网络 Faster-RCNN 目标检测框架，如下图所示，Faster-RCNN 将目标检测分为两个阶段，第一阶段为 RPN 网络通过一定规则设置不同尺度的锚点（Anchor）在 RPN 的卷积特征层提取候选框，第二阶段为讲上一阶段得到的候选框输入进 RoI 池化层进行候选区域特征提取，最终进行框回归和分类。

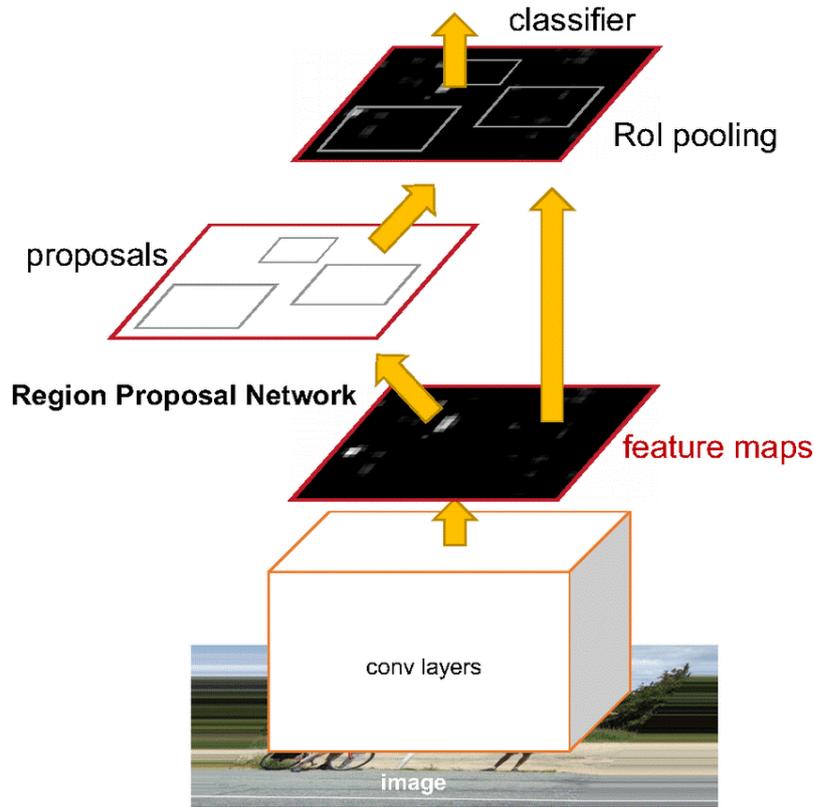


图 3-2 Faster-RCNN 结构

而 R-FCN 是 Faster-RCNN 的基础上提出改进，其网络结构如图所示，它在第一阶段提取候选框的操作与 Faster-RCNN 无异，区别在于第二阶段，通常我们的分类是要增加物体的平移不变性，而目标检测是减少物体的平移变化，通常我们所用的网络都是 ImageNet 的分类任务训练得到的，在目标检测的时候进行微调。由于得到的初始模型基于分类任务，那么会偏向于平移不变性，这和目标检测就出现了矛盾。因此，提出了 position-positive score maps（位置敏感得分图），如果一个 RoI 含有一个类别 c 的物体，那么作者将该 RoI 划分为 $k * k$ 的区域，分别表示该物体的各个部位，比如假设该 RoI 中含有人这个物体且 $k = 3$ ，那么就将“人”划分为了 9 个子区域，top-center 区域毫无疑问应该是人的头部，而 bottom-center 应该是人的脚部，而将 RoI 划分为 $k * k$ 个区域是希望这个 RoI 在其中的每一个区域都应该含有该类别 c 的物体的各个部位，即如果是人，那么 RoI 的 top-center 区域就必须含有人的头部。而当这所有子区域都含有各自对应的该物体的相应部位后，那么分类器才会将该 RoI 判断为该类别。物体的各个部位和 RoI 的这些子区域是“一一映射”的对应关系。

以上为 position-sensitive score map 设计的核心思想，R-FCN 会在共享卷积层的最后再接上一层卷积层，而该卷积层就是“位置敏感得分图 position-sensitive score map”，首先它就是一层卷积层，它的 height 和 width 和共享卷积层的一样，但是它的 $channels = k * k(C + 1)$ ， C 表示物体类别种数再加上 1 个背景类别，每个类别都有

$k * k$ 个 score maps。RoI 的 $k * k$ 个子区域在各个类别的 score maps 上找到其每个子区域的响应值。后续就是通过位置敏感 RoI 池化操作 (Position-sensitive RoI pooling) 找到上述的每个子区域的响应值。最终根据这些响应值就可以得到属于各个类别的概率以及检测框的长宽及坐标了。

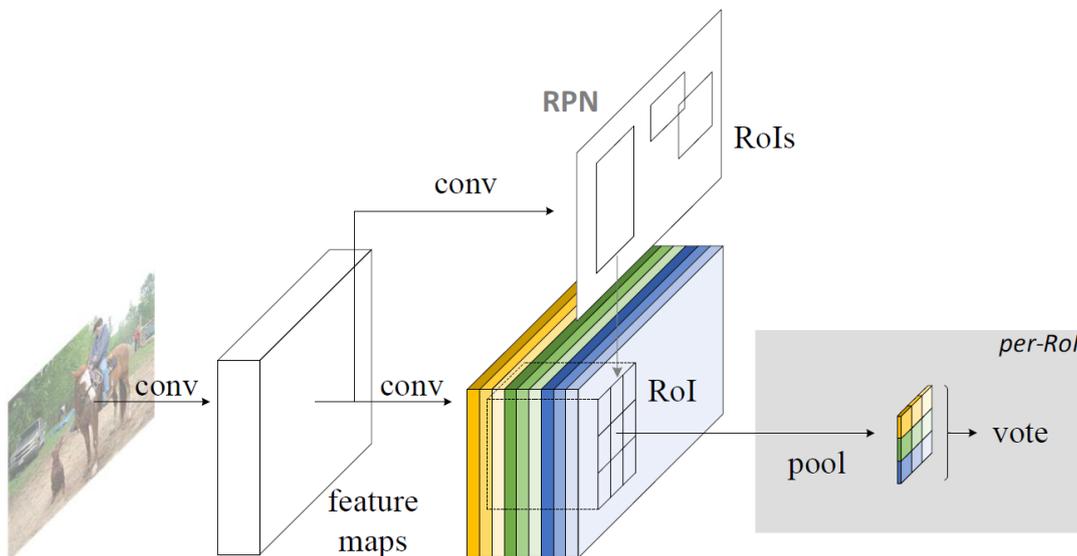


图 3-3 R-FCN 结构

3.2.2 训练细节

针对网络权值的初始化的问题，在初始化网络参数时，R-FCN 参考了经典的卷积神经网络的权值初始化方法，使用均值等于 0，标准差等于 $\sqrt{2/N}$ 的高斯分布对网络进行初始化，其中 N 代表着对应卷积核权值的个数。

3.3 用 R-FCN 对斜视图像进行眼部区域定位

3.3.1 训练细节

该论文首先用 R-FCN 对斜视图像进行眼部区域进行定位。在数据集方面，我们建立的斜眼数据集包含图像一共 5943 张，其中包含斜视图像 3000 张，正常图像 2943 张。其中用于训练集与测试集各占总图像张数百分之六十与百分之四十，用于训练使用的图像为 3566 张，测试使用的图像为 2377 张。

一般而言，输入至神经网络的图像数据需要进行预处理，将其归一化，该做法能加速损失收敛并减缓过拟合的现象。在这里我们采用了对于图像数据最简单的归一化方式，具体操作为：

$$y = \frac{x - \mu}{\sigma} \quad (3-2)$$

式中的 x 和 y 分别是归一化操作的输入和输出， μ 和 σ 分别是均值和标准差，通常

根据经验或者取统计值，在这里因为图像数据的范围是[0, 255]， μ 和 σ 均取为 127.5 能够保证数据可以做到归一化。

3.3.2 网络及训练的实现

将 R-FCN 应用于斜视图像的眼部区域的定位，其整体架构如下图所示，在提取候选框阶段，候选眼睛区域是通过候选框提取网络（RPN）获得的[35]。在生成候选的眼睛区域后，R-FCN 会将 RoI 进一步分类为眼睛区域类别或背景类别。R-FCN 的最后一层为眼睛区域和背景生成 k^2 个位置敏感得分图，这将得到 k^2 个 2 通道层。本文中， k 设置为 3[36]，并且 9 个得分图对眼睛区域或背景进行编码（左上，顶部中心，右上，...，底部中心，右下）。然后通过平均 k^2 得分对每个 RoI 进行投票，公式如下所示：

$$s_c(P) = \sum_{x,y} s_c(x,y|P) \quad (3-3)$$

其中 s_c 是第 c 个类别的平均响应值； P 是在 R-FCN 网络中学习到的参数； s_c 表示在第 c 个类别中第 (x, y) 个 bin 中的响应值，其定义如下：

$$s_c(x,y|P) = \sum_{(i,j) \in \text{bin}(x,y)} o_{x,y,c}(i + i_0, j + j_0|P)/n \quad (3-4)$$

其中 $o_{x,y,c}$ 是 $k^2 * 2$ 个得分图中的一个得分图， $(i_0; j_0)$ 是 RoI 的左上角情况， n 表示 bin 中的像素数。

最后，R-FCN 网络通过优化下面损失函数进行训练：

$$L = L_{cls} + L_{reg}^{eye} \quad (3-5)$$

其中 L_{reg}^{eye} 表示在[46]中定义的眼部区域边界框回归值， L_{cls} 是分类的交叉熵损失，定义如下：

$$L_{cls} = -\log(e^{r_{c^*}(P)} / \sum_{c'=0}^1 e^{r_{c'}(P)}) \quad (3-6)$$

其中 c^* 表示区域候选框的真实标签， $c' = 0$ 和 $c' = 1$ 分别表示背景区域和眼部区域。在该网络使用了在线难例挖掘（OHEM）策略用来训练，超参数学习率为 0.0003，损失函数优化策略使用 stochastic gradient descent（SGD），动量 momentum 为 0.9，网络输入尺寸均为重调整后的 $224 * 224 * 3$ 。

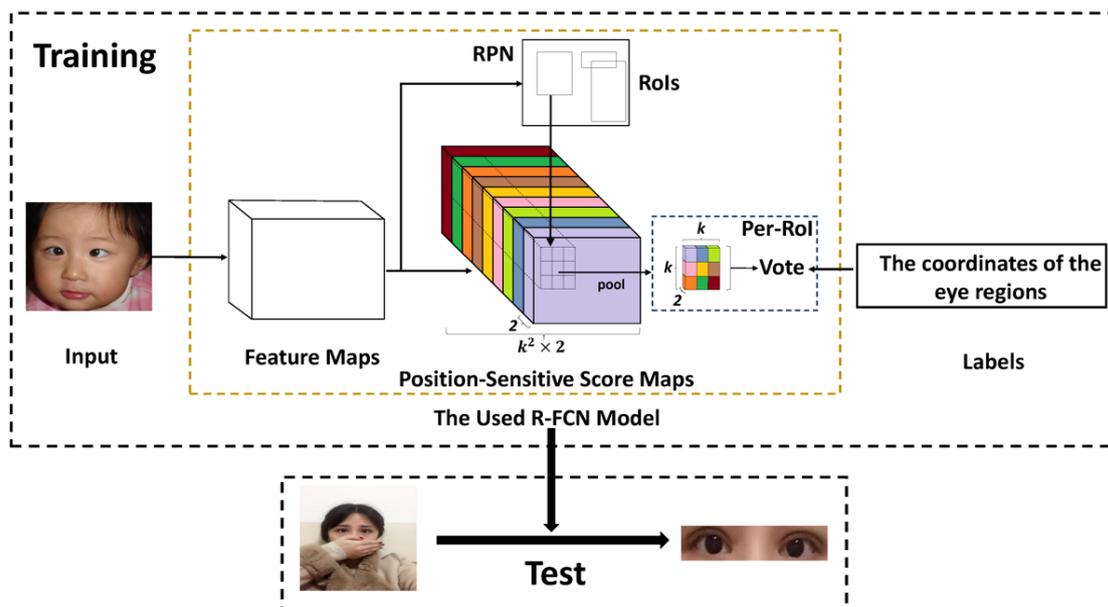


图 3-4 用于斜视图像眼部区域定位的 R-FCN 整体结构

3.4 用 CNN 对裁剪眼部区域进行斜视识别

3.4.1 模型结构

斜视检测模型 RF-CNN 中第一阶段由 3.3 节中 R-FCN 对眼部区域进行检测，并对目标框内眼部区域进行裁剪，第二阶段为得到眼部区域后输入至卷积神经网络进行斜视识别，结构网络如图所示，卷积层名称后紧跟特征图的通道数，方括号内为卷积核大小以及步长，“conv”，“maxpool”和“fc”分别代表卷积层，最大池化层和完全连接层。卷积神经网络是深度学习最常用的基本结构，在 2.3 节中介绍了多种经典卷积神经网络，在本文中该斜视识别网络由 5 个卷积层、3 个最大池化层、以及 3 个全连接层组成，如图所示。在卷积核大小的设置上前 3 个卷积层采用了最常见的 $5 * 5$ 卷积核大小，最后两个卷积层的卷积核大小设置为 $4 * 4$ ，而最大池化层的卷积核大小设置为 $2 * 2$ 。

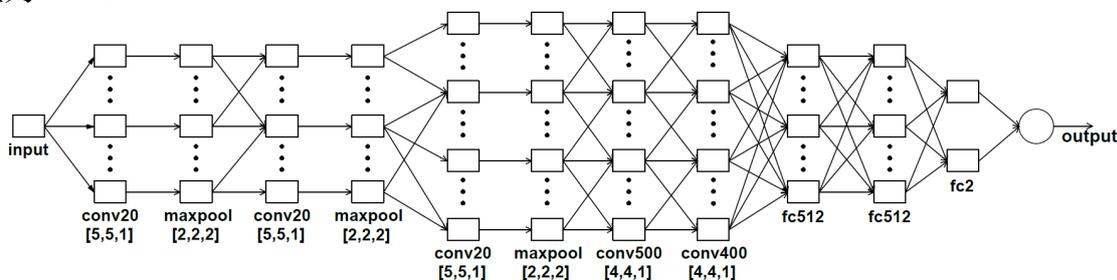


图 3-5 用于对眼部区域进行斜视检测的 CNN 结构

3.4.2 数据预处理和网络实现

将上述的卷积神经网络用于对 3.3 节中裁剪得到的眼部区域进行斜视识别，因此同样地，数据集包含裁剪眼部区域图像一共 5943 张，其中包含斜视眼部区域图像 3000 张，正常眼部区域图像 2943 张。其中用于训练集与测试集比例为 6 : 4，训练使用的图像为 3566 张，测试使用的图像为 2377 张。

对图像数据的归一化上同样采用均值和标准差均为 127.5 的处理方式。针对网络权值的初始化的问题，使用均值等于 0，标准差等于 $\sqrt{2/N}$ 的高斯分布初始化网络，其中 N 代表对应卷积核权值的个数。

在训练中针对前两层全连接层使用了 dropout 策略，dropout 率设为 0.5。损失函数策略使用了 stochastic gradient descent，并使用了 L2 正则化方法，其中权重衰减率为 $5 * 10^{-4}$ 。其他的超参数 batch size 为 32，学习率为 0.01，并训练迭代数为 5000。

3.4.3 斜视检测结果

为了量化评估斜视检测模型及算法的有效性，我们使用精准率 Se ，召回率 Sp ， Acc 以及 AUC 作为评估指标，其数学定义如下：

$$Se = \frac{TP}{TP+FP} \quad (3-7)$$

$$Sp = \frac{TN}{TN+FN} \quad (3-8)$$

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \quad (3-9)$$

其中 TP 、 FP 、 FN 分别表示测试集中的 True Positive (预测为真真值为真)、False Positive (预测为真真值为假) 和 False Negative (预测为假真值为真)。AUC 表示 ROC 曲线下的面积，指的是随机给定一个斜视样本和一个正常样本，神经网络输出该斜视样本为斜视的那个概率值比神经网络输出该正常样本为正常的那个概率值要大的可能性。对于斜视和正常两个类别均可用上式计算，但由于我们只关心斜视类别，评估时以上均只是针对斜视类别计算。实验得到以下的评估结果：

表 3-1 RF-CNN 斜视检测图像结果

评价指标	TP	TN	FP	FN	Se	Sp	Acc	AUC
RF-CNN	1056	1146	78	68	0.9329	0.9440	0.9326	0.9854

从实验结果看，算法本身具有不稳定因素，选取不同的训练集和验证集数据得到

的结果差异很大，而且召回率与精准率之间有相对矛盾之处，在同一模型下选定不同置信度时，召回率高则精准率低，反之亦然。但**F1**的定义中权衡了召回率和精准率的关系，且通过反复实验可得到不同的验证集的平均**F1**相对稳定，因而我们主要评估的标准也是所有验证集的平均**F1**。

允许像素误差的评估方式有工程指导意义，但为了论证像素级分割算法的效果差异，论文以下的实验仍主要以原始的精准率**Pr**，召回率**Re**，以及**F1**的原始定义进行评估。

3.5 本章小结

本章我们主要介绍了斜视检测系统 RF-CNN 的建立过程。首先我们介绍了斜视图像检测的目标，然后我们介绍了在实现这个目标上，使用了深度学习中目标检测与图像识别领域的方法。接着我们详述了 R-FCN、手工搭建的卷积神经网络的网络结构以及训练细节。最后我们给出使用上述神经网络进行斜视检测的实验结果。

第 4 章 轻量化斜视检测模型 RF-CNN Lite 的构建

4.1 神经网络轻量化的架构

随着模型参数量的增加，模型的准确率和复杂度也逐步增加，但同时，神经网络中存在较大冗余。设计更加高效的网络结构对于压缩模型大小、加快运行速度以及减轻训练难度都能够起到重要的作用。神经网络轻量化架构工分为三种方法，其中包括神经网络模型压缩算法、基于神经网络架构搜索的自动化神经网络架构设计以及人工设计轻量级神经网络，下节中将对这三种方法进行介绍。

由于原始网络中的参数量以及计算量巨大，在医院使用上述软件的过程中存在多种缺点：1) 软件容量大；2) 网络模型庞大，前向推导时计算量大，占据电脑大量内存空间。因此对该斜视检测模型进行轻量化显得尤为重要，本文使用了人工设计轻量级神经网络方法从参数量和运算量方面优化算法，取得了不错的效果。

本论文经过实验验证，提出了以下模型轻量化的框架：

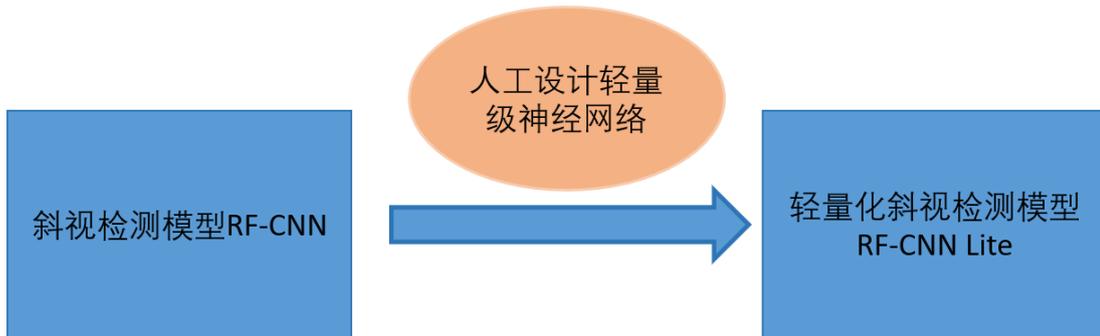


图 4-1 模型轻量化框架

为了评估轻量化的效果，引入神经网络的参数量和计算量的计算方法，参数量主要为卷积以及反卷积层的权值个数，单个卷积层的计算公式为：

$$Params = k_w \cdot k_h \cdot C_{in} \cdot C_{out} \quad (4-1)$$

k_w, k_h 表示卷积核的大小， C_{in} 表示卷积层输入的通道数， C_{out} 表示卷积层输出的通道数。参数量反映的是神经网络占用的存储空间。

理论计算量以 FLOPs 表示，把一次乘加运算定义成一个 FLOPs，只考虑乘法，不考虑加法和加偏置，当卷积核的遍历步长为 1 时，单个卷积层的 FLOPs 计算公式如下：

$$FLOPs = k_w \cdot k_h \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out} \quad (4-2)$$

$k_w, k_h, C_{in}, C_{out}$ 的含义与上面描述一致, H_{out}, W_{out} 表示输出卷积层的分辨率大小。计算量反映的是神经网络前向推理的时间。

4.2 神经网络轻量化的架构

神经网络模型压缩算法根据在不同方面的冗余可分为权重裁剪、权重量化、低秩分解以及知识蒸馏四种方法对已有的神经网络模型进行压缩,从而降低对空间和计算能力的需求,最终能够实现在便携设备上的实时运行。以下对该四种方法进行介绍。

4.2.1 权重裁剪

权重裁剪指的是去除神经网络结构的权重,从而可以达到压缩模型的目的。在网络中存在很多的参数是不重要或者不需要的,可以通过权重裁剪的方法去除这些参数。该方法有两个优点:1)通过减少参数量可以使得储存空间相应地较少;2)在网络前向计算时由于参数的减少可以使得网络运行时计算量也相应减少,对设备的计算能力要求也可以降低,从而增加计算速度。

于 2015 年 Deep Compression 被谷歌提出,该研究通过对权值的重要性进行排序,使用阈值对小于该阈值的权重进行删减,在实际操作中使用稀疏矩阵对权重进行保存才能够实现压缩的目的,这种直接对权重进行操作的方法称之为非结构性剪枝。另外其他的方法如 channel pruning^[45]和 Thinet^[46]在实现上使用了更简便实用的方法,都是通过特征图的新构造来对权值的重要性进行评估,关于误差最小化的问题,其中 channel pruning 和 Thinet 分别使用了 Lasso 回归(L1 正则化)和贪婪方法来解决。

4.2.2 权重量化

权值量化指的是使用更少的计算机位数(bit)对权重或者激活值进行储存。在 Deep Compression 中使用的是权值共享的方法,通过非监督方法聚类的方式来确定权值的中心,并使用该中心替代原有的权值,因此可以只需要保存这些聚类中心就能够达到压缩的目的。另一种方法为将神经网络的权值只用+1 和-1 来表示,这样可以很大程度地减少储存空间并加快网络前向计算的速度。一般做法上为对 32 位浮点数表示的权值使用更低位数的整型^[47]、浮点数^[48]或者定点数来表示。

4.2.3 低秩分解

低秩分解的方法主要是通过对卷积核进行分解以减少冗余,其关键点在于如何对卷积核的参数进行排列,并确定在哪个维度上对秩进行约束。低秩分解通常对矩阵或张量以及其运算以更少的秩来表示^[49],该方法一般是使用奇异值分解的方法对矩阵进行分解。在张量计算时使用更少的秩的张量计算来替代,因此低秩分解的研究关键在于使用多少秩来表示原矩阵或者张量。

4.2.4 知识蒸馏

和其他的压缩或者加速的方法不同的是，知识蒸馏是建立一个“学生”网络来去拟合“教师”网络分布的方法，将“教师”网络学到的知识通过迁移学习的方式指导“学生”网络学习，一般来说，“教师”网络是一个较大的网络，而“学生”网络是一个轻量级的网络。其为利用不同大小网络之间的知识传递，而不是像传统方法一样直接去训练一个轻量级网络。

Hinton 首先提出了知识蒸馏的概念，让小网络在学习分类目标的同时，也尽量拟合大网络对不同类别的分类结果^[50]，其思路是根据训练好的模型包含的类别信息，通过迁移学习来提高小网络的准确率。

4.3 基于神经网络架构搜索的自动化神经网络架构设计

神经网络架构搜索(Neural Architecture Search, NAS)是指根据某种搜索策略，在特定的搜索空间内，最终自动设计出解决特定任务的高性能神经网络架构。神经网络架构搜索共分为三部分，分别是搜索空间，搜索策略和性能评估策略。搜索空间定义了用以构建神经网络的基本架构单元，不过根据不同的特定任务可以利用其先验知识进而缩小搜索空间的大小。搜索策略指的是在指数级甚至无限大的搜索空间中如何能够搜索到用于解决特定任务的神经网络架构，并解决在神经网络中不同层或者不同模块之间的连接方式以及参数等问题。性能评估策略指的是高效、准确地对神经网络的性能进行度量。

神经网络架构搜索的优点在于可以不用通过人工设计轻量化的神经网络架构，可以自动化地实现新的神经网络架构并在性能上优于前者，但是缺点也是非常地明显，巨大的资源和时间消耗并不适合资源匮乏的研究者，像谷歌这种科技巨头占有巨大的优势。

4.4 人工设计轻量级网络

不同于神经网络模型压缩算法以及基于神经网络架构搜索的自动化神经网络架构设计，人工设计轻量级神经网络主要是针对卷积运算重新设计更加高效的网络单元，从而损失网络性能相对较小的条件下减少网络参数量，从而提升网络的运行速度。在本文提出对斜视检测模型的轻量化中，采用的是人工设计轻量级神经网络的方法。人工设计轻量级神经网络主要从减少卷积核的冗余、减少输入特征的通道数，设计更加高效的模块三个方面来进行研究，对三个方面介绍如下：

4.4.1 使用小卷积核代替大卷积

通过多层小卷积核替代一层大卷积核可以有效减少网络的参数，在 2.3 节中介绍

的 VGG 网络中, 使用 3×3 卷积核代替 5×5 和 7×7 大小的大卷积核, 对于一个大小 5×5 的感受野, 可以通过两层 3×3 大小的卷积实现。对于一个 7×7 的卷积核可通过三层 3×3 卷积实现, 如下图所示:

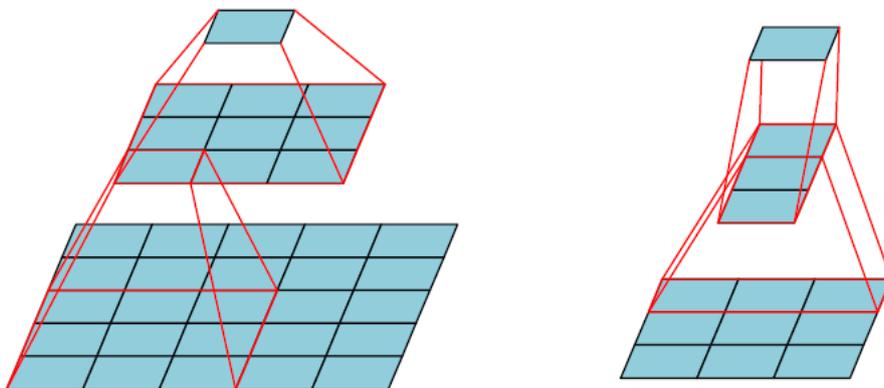


图 4-2 模型轻量化框架

在参数量上, 5×5 大小的卷积核参数量为 25, 而两层 3×3 的卷积核参数量为 18, 参数量相比减少了 28%。从浮点运算数角度, 针对输入大小为 $H \times W \times c_{in}$ 的特征, 输出为 $H \times W \times c_{out}$ 大小的特征图时, $Flops = 2$, 而两层卷积的 $FLOPs(3 \times 3)$ = 计算量也可以相应的减少。并且两层 3×3 卷积可以合并为两层非线性层, 对比一层大卷积核的非线性能力更强。

在^[51]中提出将一个 3×3 的卷积操作分解为连续的 1×3 和 3×1 的卷积, 分解之后参数量和 FLOPs 相比之前都减少了 33%。^[52]中提出使用大的 $k \times k$ 卷积核能够增加感受野, 并且为了减少计算量, 使用两层 $1 \times k$ 和 $k \times 1$ 的卷积进行代替, 最后参数量以及 FLOPs 都减少为原来一半。

4.4.2 限制中间特征通道数

在 4.1 节中介绍到对于不带 bias 的标准卷积操作, $FLOPs = k_w \cdot k_h \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out}$, 运算量受到输入通道数 C_{in} 和卷积核数量 C_{out} 的影响。通常来说卷积核数量表示提取的特征通道数量, 降低会影响网络的准确率, 因此可以选择降低输入通道数 C_{in} 来减少运算量。在 SqueezeNet 中提出了 Fire module, 如图所示, 该模块可以在保证准确率的同时降低计算量。该模块共分为两部分, 压缩层(squeeze)层和扩张层(expand)层, 通过减少 squeeze 层的通道数量来降低整个模型需要的计算量。与 AlexNet 相比在获得相同性能的情况下模型大小压缩近 50 倍。

4.4.3 分解卷积计算

对于标准的卷积操作为使用一个卷积核作用在输入的特征图所有通道上, 这样做

不但导致模型的参数量较大，而且当合并所有通道的卷积运算会存在冗余很大。通过分组卷积的操作，ResNeXt^[53]将多个分支合并为一个分支，和标准的 ResNext 在消耗相等计算量情况下，准确率更高。

在 MobileNet V1 中提出了深度可分离卷积(Depthwise Separable Convolution)，该操作对标准的卷积操作进行了分解，如图所示。深度可分离卷积(Depthwise Separable Convolution)可分为深度卷积 (Depthwise Convolution) 和逐点卷积 (Pointwise Convolution)两个操作。深度卷积(Depthwise Convolution)对于每个输入通道采用不同的卷积核，即一个通道对应一个卷积核，卷积操作是按照通道进行分解的，然后分别做卷积操作，这样相当于收集了每个通道的空间特征；逐点卷积(Pointwise Convolution)是卷积核大小为 1*1 的标准卷积，作用在输入的所有通道上，将来自不同通道的特征进行融合，相当于收集了每一个点的特征。

对于标准卷积可以知道计算量为 $k_w * k_h * C_{in} * C_{out} * H_{out} * W_{out}$ ，而两步的深度可分离卷积的计算量为 $k_w * k_h * C_{in} * H_{out} * W_{out} + C_{in} * C_{out} * H_{out} * W_{out}$ ，与前面的标准卷积的计算量相比得到计算量大致只有原来的 1/9。

我们常用的卷积核大小一般为 $3 * 3$ ，可以得到一个普通的深度可分离卷积的计算量为 $3 * 3 * C_{in} * H_{out} * W_{out} + C_{in} * C_{out} * H_{out} * W_{out}$ ， C_{in} 为输入特征通道数， H_{out} 和 W_{out} 为输入特征高和宽， C_{out} 为当前层卷积核数量。一般来说 C_{in} 远大于 9，因此，深度卷积的主要计算量集中到 1*1 的逐点卷积运算中了。针对这个问题，ShuffleNet 提出了 Group 逐点卷积，对逐点卷积进行分组卷积操作，用以减少逐点卷积操作方面的计算量消耗，如下图所示。

ShuffleNet 首先对输入通道进行分组，并在每个组中进行单独的卷积运算操作，如果单纯对分组卷积操作进行堆叠，会导致各个组之间的信息无法流通，从而影响网络的表征能力。为了让分组卷积能够和其他分组产生的特征进行交流，ShuffleNet 提出了混洗(Shuffle) 操作，对来自不同组的特征进行重新排列，使得在新的分组中包含来自之前各个组的特征，从而保证了各个组之间的信息流通，提高了网络的特征表达能力。

ShuffleNet 使用混洗(Shuffle)操作和分组逐点卷积(Group Pointwise Convolution)组合成 ShuffleNet Unit。当输入特征的大小为 $H * W * C_1$ 时，而 Bottleneck 的通道数为 C_2 时，经典的残差网络 ResNet 单元的计算量为 $2 * H * W * C_1 * C_2 + 9 * H * W * C_2^2$ FLOPs，而经过轻量化后的 ResNeXt 的计算量 FLOPs 为 $2 * H * W * C_1 * C_2 + 9 * H * W * C_2$ ，而与此同时 ShuffleNet 单元相比 ResNeXt 进一步将计算量 FLOPs 减小为 $2 * H * W * C_1 * C_2/g + 9 * H * W * C_2$ 。最后 ShuffleNet 在与 AlexNet 获得相同准确率的情况下比后者快 13 倍。

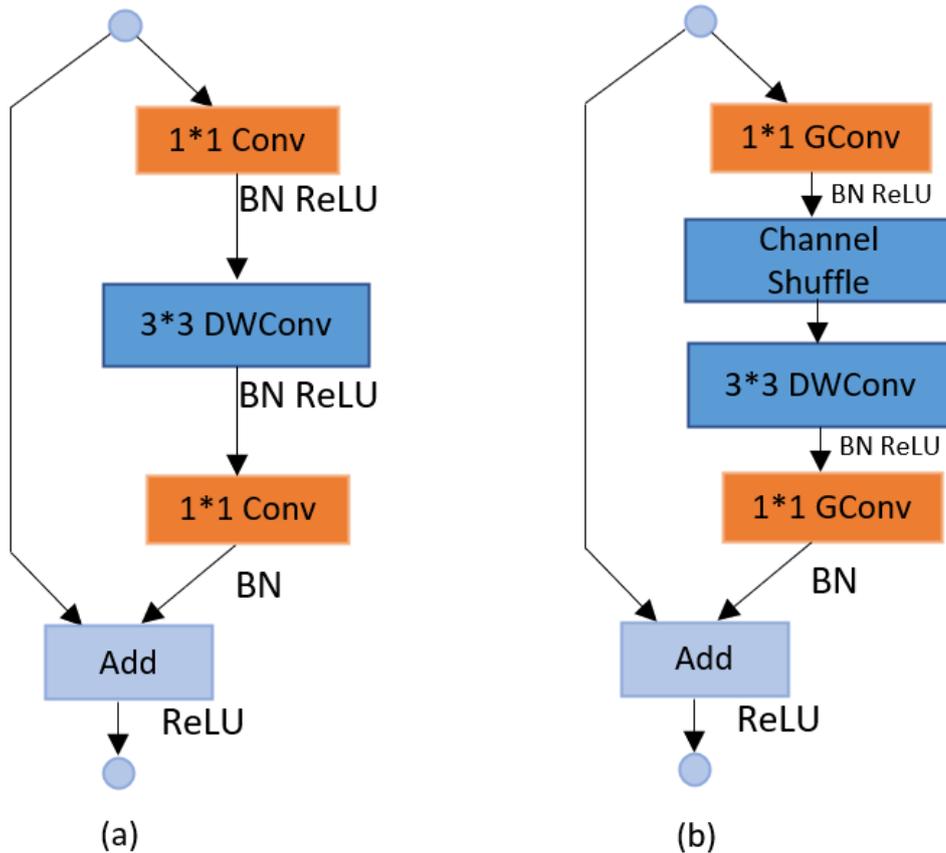


图 4-3 ShuffleNet Unit 结构

为了能够在保持网络规模的前提下进一步提高网络的性能，后来提出了 MobileNet V2^[54]该网络引入了残差结构，并对残差结构进行了设计，得到反残差结构。首先，因为深度卷积本身并不能改变特征的通道数，而且在输入通道较少的情况下提取特征能力较差，因此在残差分支中使用逐点卷积来先增加后减少特征通道数，使得深度卷积层作用在高维的特征之中，与之前经典的残差分支刚好相反。其次，在高维的空间中激活函数可以有效的增加非线性能力，但是在低维空间中会破坏特征的表达能力，因为第二个逐点卷积起到了降维的作用，所以去掉第二个逐点卷积之后的激活层，也就是相当于使用了线性的 Bottleneck，如下图所示。由于增强了网络的表征能力，MobileNet V2 达到与 MobileNet V1 相同的性能时计算量只有后者的 2/5。

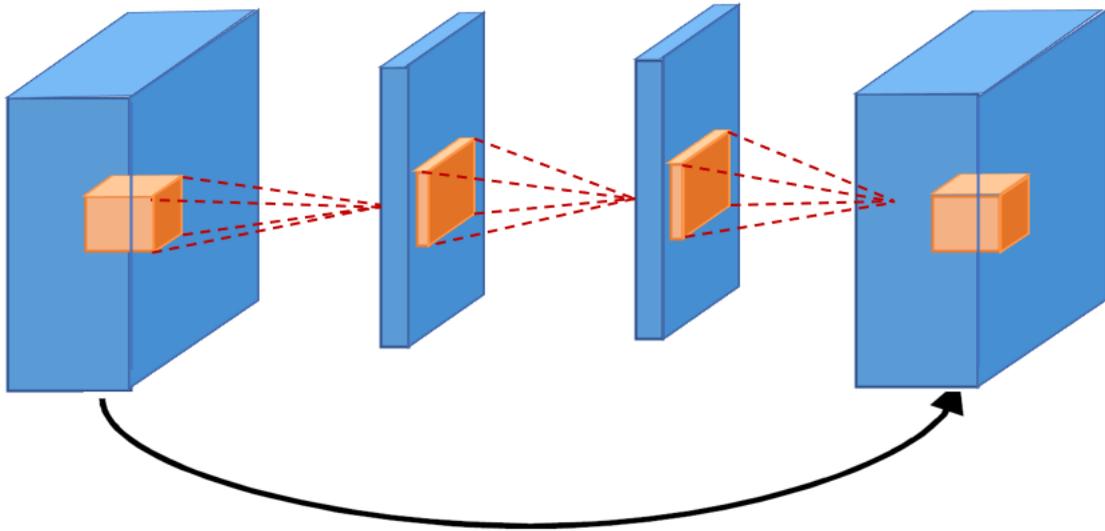


图 4-4 普通残差网络结构

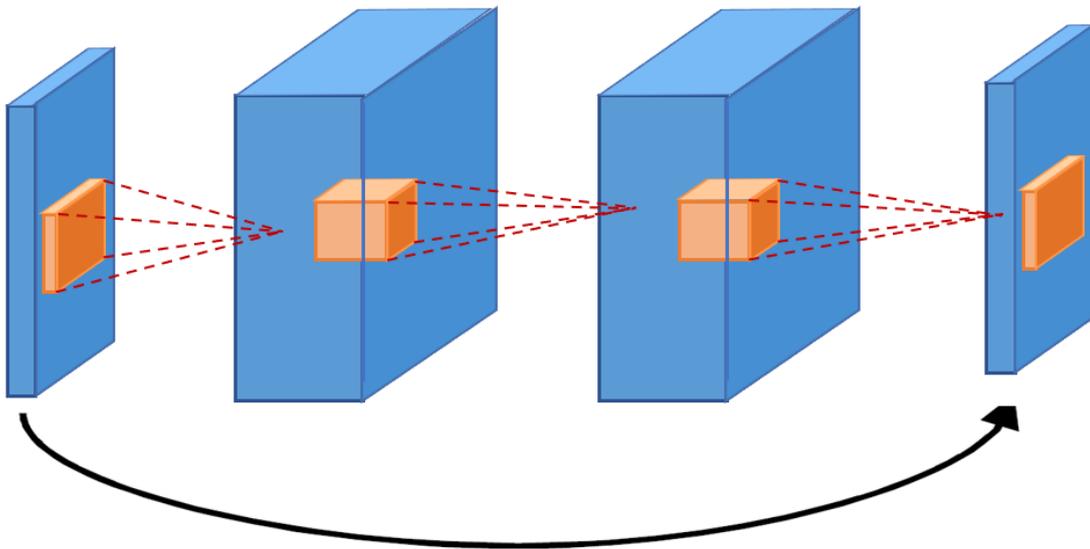


图 4-5 Bottleneck 结构

4.5 用反残差结构对模型进行轻量化设计

在本论文的斜视检测模型 RF-CNN 中，由负责眼部区域检测的 R-FCN 和负责对裁剪眼部区域进行斜视识别的 CNN 级联组成。本文中我们使用反残差结构分别对 RF-CNN 模型中两个阶段进行改造，下面章节分别介绍如何使用反残差结构对模型 RF-CNN 进行轻量化设计，最终得到轻量化斜视检测模型 RF-CNN Lite。

4.5.1 眼部区域检测网络 R-FCN 轻量化设计

在设计轻量化神经网络时，考虑使用反残差结构对斜视检测模型进行改造，在前

半部分的 R-FCN 中，位于候选框提取网络前面的基础分类神经网络负责对输入图像进行特征提取，后面再将共享特征输入至后面的 RPN 网络和 RoI-wise subnetwork 中，最后得到目标分类及检测框的位置坐标。在第一部分中，基础网络用来在整张图像上做比较耗时的卷积操作，这些操作与候选框无关，是计算共享的，第二个用来产生候选框的网络以及第三个用来分类或进一步对检测框进行回归的网络则是和每一个候选框有关系的，每一个候选框必须单独进行后面的卷积计算，衔接在候选框提取网络和最后一部分网络之间的是 ROI Pooling。我们希望的是耗时的卷积操作都尽量移到第一部分的网络中去。因此，最直接的想法是对第一部分的基础网络进行改造，使用 4.2 节中的反残差结构替代该部分的基础网络，如下图所示。

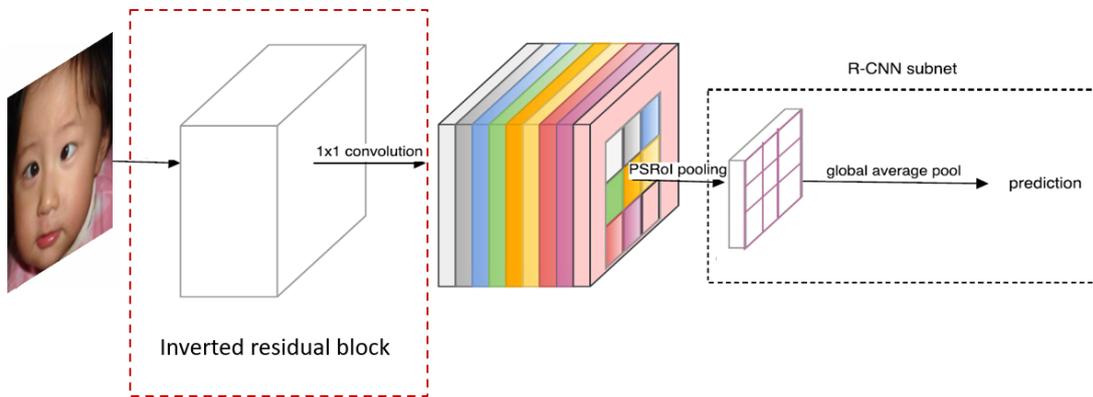


图 4-6 R-FCN 轻量化网络结构

在 R-FCN 中该部分的基础网络为 ResNet-101，参数量以及计算量都十分巨大。在本工作中，在设计由反残差结构组成的卷积神经网络时，网络的层数、中间特征通道数如下表所示，其中 t 为反残差结构中升维的膨胀系数， c 为输出通道数， n 为 bottleneck 重复次数， s 为步长。该轻量化网络输入的图片分辨率重调整为 $256 * 256$ ，并由 1 个卷积层、5 个 bottleneck 模块以及 1 个 $1 * 1$ 卷积层组成，在网络中共包含下采样 4 次，每次下采样后接下来除了第一个卷积层外均把特征图通道数翻倍。

表 4-1 R-FCN 基础网络的替代网络结构

Input	Operator	t	c	n	s
256*256*3	Conv2d	-	32	1	2
128*128*32	bottleneck	1	16	1	2
64*64*16	bottleneck	6	32	2	2
32*32*32	bottleneck	6	64	3	2
16*16*64	bottleneck	6	128	4	2
8*8*128	bottleneck	6	256	3	1
8*8*256	Conv2d1*1	-	1024	1	1

在进行对比实验时，基础网络使用了业界内提取特征能力较强的 ResNet50 以及 ResNet101 两个版本的经典残差网络，与使用反残差结构组成的卷积神经网络进行比较，实验结果对比如下图所示，其中 Average Precision 表示平均精度，Param memory 表示参数所占的储存空间，flops 表示模型进行前向计算时的浮点运算数，time/img 表示每张图片的检测时间：

表 4-2 R-FCN 与轻量化后的 R-FCN 实验结果对比

Network	Average Precision	Param memory	flops	time/img(Titan Xp)
R-FCN + ResNet50	86.52%	122MB	79 GFLOPS	0.12sec
R-FCN + ResNet101	88.27%	194MB	117 GFLOPS	0.14sec
R-FCN + Inverted residual block	84.64%	22MB	2 GFLOPS	0.09sec

从上表中显示的实验结果来看，使用反残差结构的卷积神经网络作为 R-FCN 的基础网络相较于使用普通卷积层平均精度损失 4%左右的情况下，在参数储存空间上从 ResNet50 的 122MB，ResNet101 的 194MB 下降至 22MB，在参数量上从 ResNet50 的 79GFLOPs 以及 ResNet101 的 117GFLOPs 下降至 2GFLOPs，轻量化后的效果明显。并且对眼部区域的检测速度进行了对比，从使用 ResNet50 的 0.12 秒以及 ResNet101 的 0.14 秒每张的检测速度下降至 0.09 秒，检测速度提升明显。

4.5.2 斜视识别网络轻量化设计

针对第二部分中对裁剪眼部区域进行斜视识别的卷积神经网络，对其进行轻量化与改造后的 R-FCN 级联组成新的轻量化斜视检测网络 RF-CNN Lite。在 RF-CNN 的该部分网络中，由 5 个卷积层、3 个最大池化层、3 个全连接层组成，一共包含下采样 3 次。全连接层的层数过高导致该网络所占的参数数量以及计算量的负担过大，因此同样地使用反残差结构对该网络进行替代，网络结构如下表所示，其中参数代表的含义与前面眼部区域检测网络一致。该网络由 1 个卷积层、4 个 bottleneck 模块、2 个 1*1 卷积层以及 1 个平均池化层组成，在网络中共包含下采样 4 次，同样地除了第一个卷积层外均把特征通道数翻倍，并针对原来的 3 个全连接层使用平均池化层以及 1*1 卷积层来替代，极大地减少了参数数量以及计算量。

表 4-3 轻量化后的 CNN 网络结构

Input	Operator	t	c	n	s
224*224*3	Conv2d	-	32	1	2
112*112*32	bottleneck	1	16	1	2
56*56*16	bottleneck	6	32	2	2
28*28*32	bottleneck	6	64	3	2
14*14*64	bottleneck	6	128	4	2
7*7*128	Conv2d1*1	-	1024	1	1
7*7*1024	Avgpool7*7	-	-	1	-
1*1*1024	Conv2d1*1	-	2	-	-

在进行对比实验时，使用了反残差结构组成的卷积神经网络进行比较，实验结果对比如下表所示：

表 4-4 轻量化后的 CNN 实验结果对比

Network	Param mem	flops	TP	TN	FP
CNN(普通卷积层)	51MB	2GFLOPs	1085	1146	78
CNN(反残差结构)	16MB	579 MFLOPs	1012	1161	115
	FN	Se	Sp	Acc	AUC
CNN(普通卷积层)	68	0.9329	0.9440	0.9326	0.9854
CNN(反残差结构)	89	0.8980	0.9288	0.9163	0.9742

从实验结果来看，使用反残差结构的卷积神经网络在参数储存空间上显示出明显的优势，使用普通卷积层的神经网络参数占用的空间为 51MB，而轻量化后的神经网络参数储存空间仅为 16MB，仅为前者的三分之一。针对卷积神经网络在前向计算时的浮点运算量指标，使用了普通卷积层的神经网络所占运算量为 2GFLOPs，相反地使用了反残差结构的神经网络运算量仅为 579MFLOPs，为前者的四分之一，取得了不错的轻量化结果。在斜视检测的结果指标上看，主要从敏感度、精确率、准确率以及 AUC 四个方面进行对比，使用反残差结构的神经网络相比于普通卷积层的神经网络，敏感度下降了 2.4%左右，精确率下降了 1.5%左右，在准确率上下降了 1.6%左右，AUC 则是下降 0.11。上述轻量化的操作可以使得斜视检测在准确率损失 2%以内可以达到参数量减少三分之二，计算量减少三分之二的效果。

4.6 本章小结

本章我们首先介绍了神经网络轻量化架构的三种方法包括神经网络模型压缩算法、基于神经网络架构搜索的自动化神经网络架构设计以及人工设计轻量级神经网络，并总体介绍了应用斜视检测算法的轻量化框架 RF-CNN Lite，接着我们介绍了反残差结构的设计思想和原理，针对轻量化框架第一阶段眼部区域检测的轻量化操作，主要讲解了使用反残差结构对 R-FCN 检测模型的基础网络进行设计，最后对第二阶段进行斜视检测的卷积神经网络同样使用反残差结构进行设计，给出了轻量化前后的实验结果对比。从实验结果来看，本论文提出的斜视检测轻量化框架在准确率损失 2%以内可以使得参数量和计算量减少为原来的三分之一。

第 5 章 总结与展望

5.1 本文总结

本文工作主要是针对斜视检测能够在医院进行远程医疗的实现而展开。斜视是最常见的眼科疾病之一,能够及时对患者进行斜视检测非常重要,随着技术的不断发展,实现斜视检测的自动化是目前发展趋势。斜视检测的目标是通过拍摄的照片从而判断患者是否斜视。本文的工作包括搭建斜视检测的模型,并针对该模型进行轻量化的工作,具体如下:

(1) 对斜视检测问题进行了分析,与当前流行的深度学习算法联系起来,探究了应用于图像分类的主流深度学习方法,选择了 R-FCN 以及卷积神经网络进行复现和修改,将其应用于斜视检测问题的模型 RF-CNN 构建,最终得到不错的实验结果。

(2) 对斜视检测的模型 RF-CNN 提出了轻量化的框架 RF-CNN Lite。对之前构建的模型 RF-CNN 第一阶段眼部区域检测的神经网络先进行轻量化设计,再对第二阶段的斜视识别的卷积神经网络进行轻量化设计。在对第一阶段进行轻量化步骤中,使用了反残差结构进行设计代替了 R-FCN 中的基础网络,并紧接着对第二阶段中斜视识别的卷积神经网络同样使用反残差结构进行轻量化设计,得到比 RF-CNN 更加轻量化的斜视检测模型 RF-CNN Lite。

5.2 未来工作和展望

在本研究工作中,因为研究的资源和时间限制,以及侧重点等原因,当前的研究工作存在一定的局限性,在以后的研究中有以下改进空间:

1. 训练数据可以采用更加丰富多样的图像以拓宽模型的应用场景。本文采用的数据集只针对去医院就诊的患者,对其他应用场景的数据的泛化性能并不是很好。此外本文未采用太多的数据增广方式,对于一些特殊场景可以预见数据的增加可以进一步提高模型的准确率。

2. 采取轻量化设计后的斜视检测系统 RF-CNN Lite 在测试集上的准确率下降至 91%,仍有较大的提升空间,可以采用其他提取特征能力更强的网络进行实验结果验证,以进一步提高斜视检测准确率,满足医生的诊断准确率的标准。

3. 可以使用其他的神经网络轻量化方式验证模型效果,如基于神经网络架构搜索的自动化神经网络架构设计、神经网络模型压缩算法。当前只使用了人工设计轻量级神经网络方式对模型进行轻量化设计,并进行轻量化效果验证。

参考文献

- [1] Burian, Hermann M., and Gunter K. Von Noorden. Burian-von noorden's binocular vision and ocular motility: Theory and management of strabismus[M]. CV Mosby, 1985.
- [2] Van d G E S , Van d S G W , Polling J R , et al. Amblyopia & Strabismus Questionnaire: design and initial validation[J]. Strabismus, 2004, 12(3):181-193.
- [3] Rutstein, R. P., Cogen, M. S., Cotter, S. A., Daum, K. M., Mozlin, R. L., & Ryan, J. M.. Optometric clinical practice guideline care of the patient with strabismus: Esotropia and Exotropia[M]. Lindbergh Blvd. St. Louis: American Optometric Association, 2011.
- [4] Lorenz, Birgit. Genetics of isolated and syndromic strabismus: Facts and perspectives[J]. Strabismus, 2002, 10(2):147-156.
- [5] Kiorpes, L., Kiper, D. C., O' keefe, L. P., Cavanaugh, J. R., & Movshon, J. A.. Neuronal correlates of amblyopia in the visual cortex of macaque monkeys with experimental strabismus and anisometropia[J]. Journal of Neuroscience, 1998, 18(16), 6411-6424.
- [6] Tommila V , Tarkkanen A . Incidence of loss of vision in the healthy eye in amblyopia.[J]. British Journal of Ophthalmology, 1981, 65(8):575-577.
- [7] Eskridge J B , Wick B , Perrigin D . The Hirschberg Test: A Double-Masked Clinical Evaluation[J]. American Journal of Optometry & Physiological Optics, 1988, 65(9):745-750.
- [8] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012.
- [9] Bao P , Zhang L , Wu X . Canny edge detection enhancement by scale multiplication[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(9):1485.
- [10] Ballard D H . Generalizing the Hough transform to detect arbitrary shapes[J]. Pattern Recognition, 1981, 13(2):111-122.
- [11] Almeida J D S D , Aristófanes Corrêa Silva, Paiva A C D , et al. Computational methodology for automatic detection of strabismus in digital images through Hirschberg test[J]. Computers in Biology and Medicine, 2011, 42(1):135-146.

- [12] Valente T L A , De Almeida J D S , Silva, Aristófanés Corrêa, et al. Automatic diagnosis of strabismus in digital videos through cover test[J]. Computer Methods and Programs in Biomedicine, 2017, 140(Complete):295-305.
- [13] Helveston E M , Orge F H , Naranjo R , et al. Telemedicine: Strabismus e-consultation[J]. journal of aapos the official publication of the american association for pediatric ophthalmology & strabismus, 2001, 5(5):0-296.
- [14] Zenghai C , Hong F , Wai-Lun L , et al. Strabismus Recognition Using Eye-Tracking Data and Convolutional Neural Networks[J]. Journal of Healthcare Engineering, 2018, 2018:1-9.
- [15] Cheng J , Wang P S , Gang L I , et al. Recent advances in efficient computation of deep convolutional neural networks[J]. Frontiers of Information Technology & Electronic Engineering, 2018, v.19(01):67-80.
- [16] Howard A G , Zhu M , Chen B , et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.
- [17] Zhang X , Zhou X , Lin M , et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[J]. 2017.
- [18] Han S , Mao H , Dally W J . Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding[C]// ICLR. 2016.
- [19] He Y , Lin J , Liu Z , et al. AMC: AutoML for Model Compression and Acceleration on Mobile Devices[J]. 2018.
- [20] Tan M , Chen B , Pang R , et al. MnasNet: Platform-Aware Neural Architecture Search for Mobile[J]. 2018.
- [21] Zoph, B., Vasudevan, V., et al. Learning transferable architectures for scalable image recognition[J]. 2018.
- [22] McCulloch, W. S., & Pitts, W.. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4), 115-133.
- [23] Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain[J]. Psychological Review, 1958, 65:386-408.
- [24] 马卫飞. 基于深度学习的桥梁裂缝检测算法研究[D].
- [25] Rumelhart D E , Hinton G E , Williams R J . Learning Representations by Back Propagating Errors[J]. Nature, 1986, 323(6088):533-536.
- [26] Lecun Y , Boser B , Denker J , et al. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 1989, 1(4):541-551.

- [27] Simonyan, K., & Zisserman, A.. Very deep convolutional networks for large-scale image recognition[J]. 2014.
- [28] Szegedy C , Liu W , Jia Y , et al. Going Deeper with Convolutions[J]. 2014.
- [29] He K , Zhang X , Ren S , et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2016.
- [30] Iandola F N , Han S , Moskewicz M W , et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[J]. 2016.
- [31] Ioffe S , Szegedy C . Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015.
- [32] Szegedy, Christian, et al. Rethinking the inception architecture for computer vision[J]. 2016.
- [33] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6):1137-1149.
- [34] Dai, Jifeng, et al. R-fcn: Object detection via region-based fully convolutional networks[J]. 2016.
- [35] 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.
- [36] Ronneberger O , Fischer P , Brox T . U-Net: Convolutional Networks for Biomedical Image Segmentation[J]. 2015.
- [37] 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, 2018, 40(4):834.
- [38] He, Kaiming, et al. Mask r-cnn[J]. 2017.
- [39] Cao Z , Hidalgo G , Simon T , et al. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields[J]. 2018.
- [40] Chen Y , Wang Z , Peng Y , et al. Cascaded Pyramid Network for Multi-Person Pose Estimation[J]. 2017.
- [41] Taigman, Yaniv, et al. Deepface: Closing the gap to human-level performance in face verification[J]. 2014.
- [42] Redmon J , Divvala S , Girshick R , et al. You Only Look Once: Unified, Real-Time Object Detection[J]. 2015.

- [43] Liu W , Anguelov D , Erhan D , et al. SSD: Single Shot MultiBox Detector[J]. 2016.
- [44] Girshick R . Fast R-CNN[J]. Computer Science, 2015.
- [45] Xiangyu Z , Yihui H , Jian S . Channel Pruning for Accelerating Very Deep Neural Networks[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2017.
- [46] Luo J H , Wu J , Lin W . ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression[C]// 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017.
- [47] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2704-2713.
- [48] Zhou S, Wu Y, Ni Z, et al. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients[J]. arXiv preprint arXiv:1606.06160, 2016.
- [49] Astrid M, Lee S I. Cp-decomposition with tensor power method for convolutional neural networks compression[C]//2017 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2017: 115-118.
- [50] Hinton G , Vinyals O , Dean J . Distilling the Knowledge in a Neural Network[J]. Computer Science, 2015, 14(7):38-39.
- [51] Jaderberg, M., Vedaldi, A., et al. Speeding up convolutional neural networks with low rank expansions[J]. 2014.
- [52] Peng C , Zhang X , Yu G , et al. Large Kernel Matters -- Improve Semantic Segmentation by Global Convolutional Network[J]. 2017.
- [53] Xie S , Girshick R , Dollár, Piotr, et al. Aggregated Residual Transformations for Deep Neural Networks[J]. 2016.
- [54] Sandler, M., Howard, A., Zhu, M., et al. Mobilenetv2: Inverted residuals and linear bottlenecks[J]. 2018.

攻读学位期间主要研究成果

一、发表的学术论文

[1] Jiewei Lu, Zhun Fan, Zheng Ce, Jingan Feng, Longtao Huang, Wenji Li, Erik D. Goodman. Automated Strabismus Detection for Telemedicine Applications[J]. arXiv preprint arXiv: 1809.02940, 2018.

二、获奖情况

[1] 黄龙涛；伍宇明；段一鸣；朱贵杰。基于无人机的高层建筑墙体裂缝检测。“兆易创新杯”第十三届中国研究生电子设计竞赛，获华南赛区三等奖。

三、申请专利

[1] 范衡；黄龙涛；朱贵杰；郑策。一种基于卷积神经网络的斜眼检测方法。[P]. 中国专利，申请号：201811127832.4

[2] 范衡；伍宇明；卢杰威；朱贵杰；莫嘉杰；黄龙涛。一种基于深度学习的蝴蝶自动分类方法。[P]. 中国专利，申请号：201811070920.5

致 谢

光阴荏苒，转眼间三年的硕士生活就要结束了。回首这三年的学习生活，对那些曾经给予我帮助、引导我的人，我心中充满着感激之情。

首先我要特别感谢我的导师范衡教授，是范衡老师在学术上严谨的治学作风、务实的工作态度深深地影响着我，引领我打开了这扇学术研究的大门。在这三年中是在他的悉心指导下，我不断地学习新的知识，提高了我的科研能力。是他在每周固定的学术研讨会上，让我学习到清晰表达自己的观点的能力，在逻辑思考能力上不断提高，他还特别强调思辨和表达的能力是我们在科研上必须具备的一项技能。范衡老师在学习以及生活上都给予了我很大的帮助，不管在工作还是生活上一直是我学习的榜样，他一直激励我在学习和生活中不断提高自己。

然后我要感谢实验室的同学们，感谢卢杰威师兄的细心教导，是他教会我如何做好科研，感谢朱贵杰师兄教会我如何撰写发明专利材料以及软件著作权材料，感谢实验室冯靖安、邱本章、姜涛、袁宇彤、卞新超、王诏君、安康的帮助，是他们在科研学习上的热情不断感染着我，在每一次的学术探讨中我从中学习到了很多东西。

最后我要感谢我的父母和家人，是你们一直在背后默默地支持我，让我可以心无旁骛地完成我的学业，如果没有你们的支持和付出，在学习和生活上给予我帮助和指导，我也不能顺利完成学业。同时，我要感谢所有支持和帮助过我的人，谢谢你们！

黄龙涛
2020年4月