

---

# 北京理工大学

## 本科生毕业设计(论文)

### 基于眼底图像的糖尿病视网膜病变智能分类 系统设计与实现

Research and implementation of intelligent classification  
method for diabetic retinopathy based on color fundus image

学    院：\_\_\_\_\_  
专    业：\_\_\_\_\_  
学生姓名：\_\_\_\_\_  
学    号：\_\_\_\_\_  
指导教师：\_\_\_\_\_

20  年  月  日

# 基于眼底图像的糖尿病视网膜病变智能分类系统的设计与实现

## 摘 要

糖尿病视网膜病变是糖尿病最严重的并发症之一，也是目前全世界适龄工作人口中最主要的致盲原因。眼底图像在临床上对糖尿病视网膜病变严重程度的诊断至关重要，眼科医生通过对眼底图像的分析，根据微动脉瘤、渗出物等病变来诊断糖尿病视网膜病变的严重程度。但是目前的人工诊断对医生的临床经验的依赖很大，十分耗时；而且部分地区糖尿病视网膜病变患者很多，而医疗资源十分缺乏。因此，通过图像处理、深度学习等领域的方法，实现对彩色眼底图像的智能分类，对糖尿病视网膜病变进行及时的诊断、预防和治疗具有十分积极的意义。

基于上述需求，本课题搭建了糖尿病视网膜病变智能分类系统。本文基于 Python 的科学计算软件包和 Keras 深度学习框架，使用增加了注意机制的深度学习网络 Inception V3 训练模型并实现对彩色眼底图像的分类，利用 Tkinter 搭建系统界面，构建一个能够对眼底图像进行病变分类的智能系统。本文主要完成了以下工作：对彩色眼底图像进行预处理与数据增强、搭建增加注意力机制的深度学习网络 Inception V3、输入数据利用网络训练能够对糖尿病视网膜病变进行分类的模型、实现彩色眼底图像的获取和可视化、利用训练所得模型对彩色眼底图像进行智能分类。

**关键词：糖尿病视网膜病变分类；深度学习；彩色眼底图像**

## **Design and Implementation of Intelligent Classification System for Diabetic Retinopathy Based on Fundus Image**

### **Abstract**

Diabetic retinopathy is one of the most serious complications of diabetes, and it is also the leading cause of blindness in the working-age population worldwide. Fundus images are clinically critical for the diagnosis of the severity of diabetic retinopathy. Ophthalmologists diagnose the severity of diabetic retinopathy by analyzing the fundus image and based on lesions such as microaneurysms and exudates. However, the current manual diagnosis relies heavily on the doctor's clinical experience and is very time consuming; and there are many patients with diabetic retinopathy in some areas, and the medical resources are very scarce. Therefore, the intelligent classification of color fundus images through image processing, deep learning and other fields has a very positive significance for timely diagnosis, prevention and treatment of diabetic retinopathy.

Based on the above requirements, this topic has established an intelligent classification system for diabetic retinopathy. Based on Python's scientific computing software package and Keras deep learning framework, this paper uses the deep learning network Inception V3 training model with attention mechanism to realize the classification of color fundus images, and uses Tkinter to build a system interface to construct a lesion classification intelligent system for fundus images. This paper mainly completed the following tasks: preprocessing and data enhancement of color fundus images, building deep learning network Inception V3 with increased attention mechanism, input data using deep learning network to train a model for classification of diabetic retinopathy, and achieving color fundus image acquisition and visualize the image, and use the trained model to intelligently classify color fundus images.

**Key words: classification of diabetic retinopathy; deep learning; color fundus image**



## 目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 绪论.....	1
1.1 课题研究背景与意义.....	1
1.2 国内外研究现状分析.....	2
1.2.1 彩色眼底图像二分类问题.....	2
1.2.2 彩色眼底图像三分类问题.....	3
1.2.3 彩色眼底图像五分类问题.....	4
1.3 主要研究内容.....	5
1.4 论文组织结构.....	6
第 2 章 基于眼底图像的糖尿病视网膜病变的分类算法实现.....	7
2.1 相关的开发平台、技术介绍.....	7
2.1.1 开发平台 Anaconda 与 Jupyter Notebook.....	7
2.1.2 深度学习框架 TensorFlow 与 Keras.....	8
2.2 开发环境搭建.....	10
2.2.1 算法实现环境介绍.....	10
2.2.2 算法实现环境搭建.....	10
2.3 数据集及预处理.....	13
2.3.1 数据集.....	13
2.3.2 数据的预处理.....	15
2.4 卷积神经网络 INCEPTION V3 和注意机制.....	16
2.5 模型训练.....	18
2.5.1 数据集准备.....	18
2.5.2 基于 Inception V3 的训练方法.....	20
2.6 实验结果.....	21
2.7 本章小结.....	22

第 3 章 基于眼底图像的糖尿病视网膜病变分类系统实现.....	23
3.1 系统需求分析与总体设计 .....	23
3.1.1 系统需求分析.....	23
3.1.2 系统总体设计.....	28
3.2 系统详细设计.....	32
3.2.1 开发环境搭建.....	32
3.2.2 系统详细设计.....	34
3.3 本章小结.....	38
第 4 章 系统运行与测试 .....	39
4.1 系统界面.....	39
4.2 系统测试运行.....	40
4.2.1 “选择图像”功能.....	40
4.2.2 “显示图像”功能.....	42
4.2.3 “图像分类”功能.....	43
4.3 本章小结.....	44
结 论.....	45
参考文献.....	46
致 谢.....	48

## 第1章 绪论

### 1.1 课题研究背景与意义

糖尿病视网膜病变(DR, Diabetic Retinopathy)是由于糖尿病患者的全身各个组织器官在长期处于高血糖的情况下,微血管发生病变,血管内液体渗出至组织中,导致视网膜发生病变,并且发生功能障碍。糖尿病视网膜病变是糖尿病患者中最严重和最常见的微血管并发症之一,也是糖尿病患者致盲的重要原因,并且也是四大主要致盲性眼病之一,是当今发展的世界中对人类视力健康十分有威胁性的一种疾病。根据相关的资料我们能了解到,全世界所有的糖尿病患者中有将近一半的患者会患有 DR,而且这种疾病会随着时间的变得越来越严重,大概有二十分之一到十分之一的患者会在 5 年内失明。此外同样有数据显示,到 2015 年为止,在很多发达国家中,糖尿病视网膜病变是这些国家里面处于工作阶段的人中失明的主要原因。据统计,截至 2018 年,中国患有糖尿病的人数在 4000 万以上,其中视力受到严重威胁的患者在 300 万人左右,也就是说有将近三分之一左右的患者会面临失明的困境。糖尿病视网膜病变跟长期患有糖尿病有关系,大概 40%至 45%的糖尿病患者会有不同程度的病变。如果及时检测到 DR,可以减缓或避免视力损害的进展,但是这可能是困难的,因为这种疾病通常几乎没有症状,直到提供有效治疗为时已晚。可见,糖尿病视网膜病变给人们的视力带来的威胁是十分巨大的。

在崭新的 21 世纪,人工智能技术如热浪袭来,人工智能技术与生活中各个领域的结合都产生了一些非常优秀出色的成果。目前,人工智能技术和医学领域融合成功,而且还会发展更好,包括医学影像方面、临床医疗智能诊断决策方面、医疗智能语音方面及“互联网+”医疗方面都具有很大的发展潜力<sup>[1]</sup>。不仅如此,本课题所涉及的深度学习技术在图像分类方面的发展飞速地进步,对图像识别及分类的网络逐渐发展和修正,克服各种难题,优化分类的准确率。随着卷积神经网络的出现,对图像分类的网络逐渐出现,从上个世纪的 LeNet-5 到开启深度学习元年的 AlexNet,以及此后的 VGG Net、GoogLeNet、ResNet 和 DenseNet。而在医学领域,许多疾病的诊断都是医生基于医学图像进行判断的。医学图像大量存在,为深度学习技术提供大量的数据,为很多疾病利用人工智能技术实现智能诊断和智能决策方面提供了很好的发展基础。

现在就全球来说，糖尿病患者非常之多，而其中相当大一部分的患者已经患有不同严重程度的糖尿病视网膜病变，并且糖尿病视网膜病变的致盲率极高。而相关专家建议，临床上的预防及诊断方法是糖尿病患者每年进行不少于两次的糖尿病视网膜病变检查。但是由于目前临床的疾病筛查诊断中，这种病变检查是一个完全依赖医生专家的过程，需要很有经验的眼科医生对采集到的患者的眼底图像进行完全人工的检查和评估，可见这个过程十分耗时和耗费资源。而且由于采集过程中导致图像的质量不好、医生的个人经验有限及眼底图像自身的特点<sup>[2]</sup>等问题，和设备资源的缺乏，加上糖尿病视网膜病变患者的增多，这种传统的临床诊断方法无法满足病患的需要。因此，实现糖尿病视网膜病变的智能诊断十分必要和迫切。

由于人工智能技术在图像领域和医学领域的突出表现，本课题借助人工智能对大数据的分析处理和自主学习能力，基于医生临床经验知识的学习，实现一个基于眼底图的糖尿病视网膜病变的智能诊断系统，自动完成对糖尿病视网膜病变程度的精确分类，旨在帮助眼科医生在临床诊断时提供智能决策，节省医生的诊断时间，也为病患的被正确诊断提供更大的可能性，并为严重病患争取更宝贵的治疗时间，实现糖尿病视网膜病变的早发现与早诊断。

## 1.2 国内外研究现状分析

从糖尿病视网膜病变眼底图像的采集技术、对眼底图像分类标准的主导及对眼底图像进行自动化智能化分级的角度来看，国外的研究现状都要比国内的情况明朗一些，而且占有一定的主导地位。

对于眼底图像的采集设备技术。糖尿病视网膜病变的检查方法是利用眼底检测设备采集患者的彩色眼底图像并让眼科医生专家对其进行人工的检查和评估，十分耗时且对工作经验要求很高。目前的眼底图像的采集主要是采用眼底相机进行的。

对于眼底图像的不同分类标准的技术发展。根据眼底图像糖尿病视网膜病变的严重程度，眼底图像的分类标准分为了三类：二分类问题、三分类问题和五分类问题<sup>[3]</sup>，其中目前使用最多的标准为五分类问题。

### 1.2.1 彩色眼底图像二分类问题

在二分类问题中，对彩色眼底图像的分类标准有两种：非 DR 和 DR 两类、增生性 DR 和非增生性 DR 两类。针对二分类问题，在 1996 年 Gardner 等人<sup>[4]</sup>利用 147



个糖尿病患者和 32 个正常图像使用反向传播神经网络进行训练，以识别视网膜图像中的特征。然后将 200 个糖尿病患者和 101 个正常图像随机化并用于评估网络与眼科医生检测糖尿病性视网膜病变的性能对比。结果表明训练出的网络对血管、渗出物和出血的成功识别的准确率分别为 91.7%，93.1% 和 73.8%。与眼科医生的结果相比，该网络对于检测糖尿病性视网膜病变的灵敏度为 88.4%，特异性为 83.5%。说明该可以检测血管、渗出物和出血，与眼科医生相比，该网络在检测糖尿病视网膜病变方面取得了良好的准确性。到 2014 年，Roychowdhury 等人<sup>[5]</sup>介绍了一种计算机辅助筛查系统（DREAM），该系统分析具有不同光照和视野的眼底图像，并使用机器学习生成糖尿病视网膜病变（DR）的严重程度等级。分析诸如高斯混合模型（GMM），k-最近邻（kNN），支持向量机（SVM）和 AdaBoost 的分类器，用于对来自非病变的视网膜病变进行分类。发现 GMM 和 kNN 分类器分别是用于明亮和红色病变分类的最佳分类器。与具有或不具有 DR 的图像分类相比，跟之前研究出来的最优秀结果相比较，DREAM 系统实现了 100% 灵敏度，53.16% 特异性和 0.904 准确率。在 2016 年，Varun Gulshan 等人<sup>[6]</sup>使用 128175 张视网膜图像的回顾性开发数据集训练了一种针对图像分类优化的特定类型的神经网络，称为深度卷积神经网络，其对于糖尿病性视网膜病变，糖尿病性黄斑水肿和图像可分级性分级为 3 至 7 次。结果表现为对测试数据集的准确率高达 99%。

### 1.2.2 彩色眼底图像三分类问题

在三分类问题中，大家通常将彩色眼底图像划分为非 DR、非增生性 DR 和增生性 DR 三类病变。在 2007 年 Jagadish Nayak 等人<sup>[7]</sup>提出了一种基于计算机的方法，将图像预处理，形态学处理技术和纹理分析方法应用于眼底图像，以检测硬性渗出物面积，血管面积和对比度等特征。这些特征被用作人工神经网络（ANN）的输入，用于自动分类。通过与专家眼科医师进行比较，验证检测结果，结果证明分类准确率为 93%，灵敏度为 90%，特异性为 100%。在 2018 年，ACarson Lam 等人<sup>[8]</sup>证明了卷积神经网络（CNNs）在彩色眼底图像上的应用，用于糖尿病视网膜病变分期的识别任务。其网络模型实现了与基线文献结果相当的测试指标性能，验证灵敏度为 95%。他们还探索了多项分类模型，并证明由于 CNN 无法检测到细微的疾病特征，误差主要发生在正常的轻度疾病的错误分类中。他们从 ImageNet 对预先训练的 GoogLeNet 和 AlexNet 模型进行迁移学习，分别在 2 分类，3 分类和 4 分类模型上将

峰值测试集精度提高到 74.5%，68.8%和 57.2%。

### 1.2.3 彩色眼底图像五分类问题

在五分类问题中，大家采用国际临床糖尿病视网膜病变严重程度分级标准，将彩色眼底图像划分为无明显糖尿病视网膜病变(No DR, No Diabetic Retinopathy)、轻度非增生性糖尿病视网膜病变(Mild NPDR, Mild Non-Proliferative Diabetic Retinopathy)、中度非增生性糖尿病视网膜病变(Moderate NPDR, Moderate Non-Proliferative Diabetic Retinopathy)、重度非增生性糖尿病视网膜病变(Severe NPDR, Severe Non-Proliferative Diabetic Retinopathy)和增生性糖尿病视网膜病变(PDR, Proliferative Diabetic Retinopathy)这五个等级的严重程度。在 2015 年，Kaggle 发布了糖尿病视网膜病变检测的比赛，提供高分辨率的眼底图像作为公开数据集，引起共 661 团队参加比赛，其中很多团队使用先进的深度学习网络并且表现优异。在 2016 年，Harry Pratt 等人<sup>[9]</sup>提出了一种 CNN 方法来诊断数字眼底图像中的 DR 并准确地对其严重程度进行分类。他们开发了具有 CNN 架构和数据增强的网络，可以识别分类任务中涉及的复杂特征，例如视网膜上的微动脉瘤，渗出物和出血，从而自动提供诊断而无需用户输入。他们使用公共可用的 Kaggle 数据集在高端图形处理器单元（GPU）上训练此网络，并展示令人印象深刻的结果，特别是对于高级分类任务。在使用的 80,000 张图像的数据集上，他们提出的 CNN 在 5,000 张验证图像上实现了 95%的灵敏度和 75%的准确度。在 2018 年，Google AI 医疗保健团队<sup>[10]</sup>利用深度学习技术来辅助眼科医生对糖尿病视网膜病变的分级，他们使用高质量标签使深度学习训练的模型更加真实可靠，模型透明，这种深度学习的辅助提高了诊断中的准确性和置信度。

在国内，对糖尿病视网膜病变的自动化检测方法与技术也在一直发展。在 2008 年，林蔚<sup>[11]</sup>提出通过 k 邻近图快速区域合并方法对视网膜图片进行分割从而实现硬性渗出区域的自动分割。在 2012 年，高玮玮等人<sup>[12]</sup>提出了基于高斯匹配滤波和局部熵确定阈值实现图像分割的方法，从而能够最大限度地获得完整的血管和微动脉瘤结构。2018 年，李琼等人<sup>[2]</sup>提出对眼底图像进行预处理并进行数据增强，然后在 AlexNet 网络的基础上，在网络的每一个卷积层和全连接层前引入一个批归一化层，得到一个网络层次更复杂的深度卷积神经网络 BNnet，并采用迁移学习的策略对视网膜图像进行训练，结果五分类的准确率高达 0.93，且具有较好的鲁棒性和泛化性。

### 1.3 主要研究内容

本课题的主要任务是实现基于眼底图像对糖尿病视网膜病变的严重程度进行智能化分期的系统。本课题的主要研究内容是：深度学习中常用的彩色图像的图像处理技术，为数据的预处理操作做准备；学习 CNN 网络发展的过程及技术知识，为实现核心的模型训练打基础，为实现系统中的细节问题打基础。糖尿病视网膜病变智能分期系统的主要研究过程如图 1-1 所示。

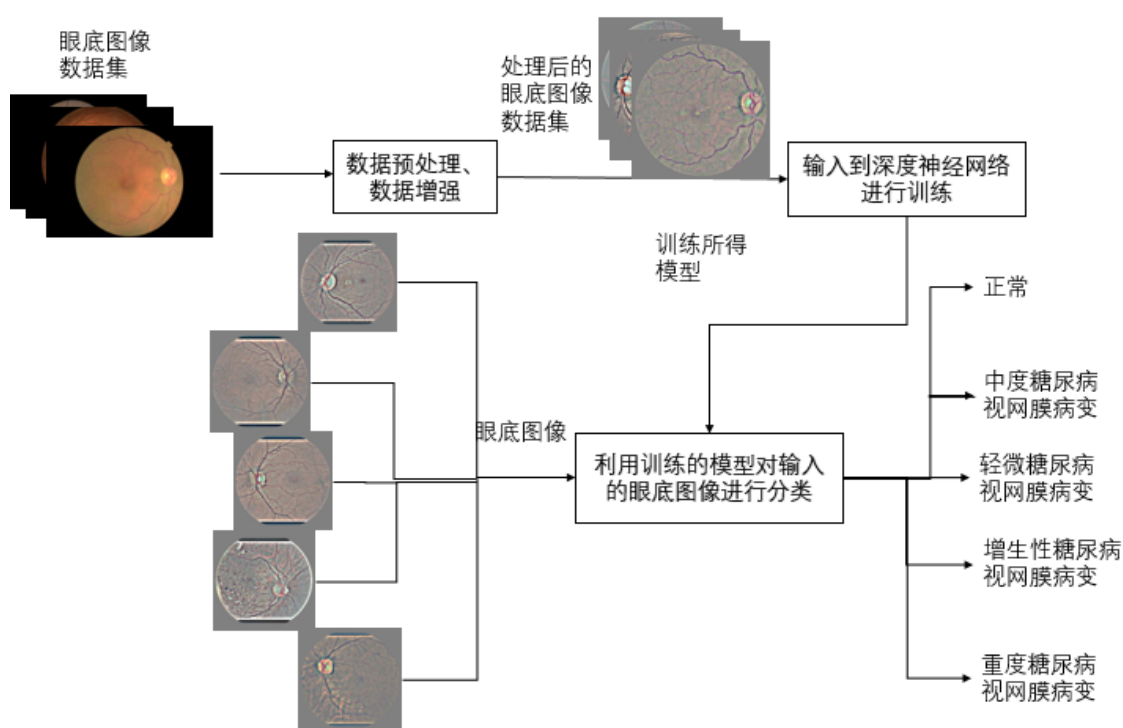


图 1-1 糖尿病视网膜病变智能分期系统的研究过程(该图是你自己画的吗?)

本文完成的工作如下：

1. 完成对彩色图像进行图像处理，实现数据预处理和数据增强。数据预处理部分主要是对彩色图像进行去除黑色边缘、色彩归一化等操作；数据增强主要是对彩色图像进行旋转、色彩增强等。
2. 研究卷积神经网络相关知识，选择预先训练的模型并搭建卷积神经网络，实现该系统的核心算法部分，在此过程中完成对网络所需的平台和环境的了解和搭建。
3. 根据一些基础的技术知识，结合之前的工作内容，对网络的性能进行评估，判断

对图像进行数据预处理这个操作对整个智能分类工作是否有积极的影响。

4. 实现对系统界面的设计与实现，将训练的模型集成到系统中，方便用户对系统的使用。

## 1.4 论文组织结构

本文的结构是严格按照此次课题的研究顺序进行安排的。本文的详细结构如下：

第 1 章：为绪论部分。详细介绍了糖尿病视网膜病变的发展及危害，人工智能技术的发展及人工智能技术与医学领域的结合，然后详细分析了人工智能技术与 DR 临床诊断可相结合的必要性和可实现性，最后简述了本课题的研究意义。

第 2 章：主要介绍了训练模型部分的算法实现。首先，对实现本论文中使用的算法所需的开发平台、相关技术进行介绍，并介绍开发环境的搭建；然后，介绍数据集和对其进行的预处理，简单介绍用于本课题中用到的 Inception V3 预训练的模型；最后，介绍实验部分，对智能分类模型的训练，并对模型的性能进行评估分析。

第 3 章：主要介绍了智能分类系统开发的流程，从需求分析开始，然后对系统进行概要设计说明，根据概要设计对系统进行详细设计说明，每一部分都会有详细的文字和图画说明。系统的开发流程是按照瀑布式开发进行，从确定需求到进行详细设计整个过程都是逻辑清晰明了的。

第 4 章：主要介绍了系统运行时的情景，从用户使用的角度来展示系统的界面和功能，并且对系统进行测试。

## 第2章 糖尿病视网膜病变的分类算法实现

本章主要对糖尿病视网膜病变智能分类系统的核心部分——利用深度学习网络训练出智能分类模型进行介绍。首先，对实现算法所需的开发平台、技术和深度学习相关知识进行介绍；其次，搭建实现算法所需的深度学习环境和框架；然后，对本算法实现过程中用到的数据集进行介绍，而且对数据进行预处理的方法进行介绍；最后，介绍实验部分，将处理过的数据输入到搭建的网络中进行训练来获取分类模型，并对模型的分类结果进行评估分析。

### 2.1 相关的开发平台、技术介绍

#### 2.1.1 开发平台 Anaconda 与 Jupyter Notebook

Anaconda 是免费开源的，主要是针对 Python 和 R 语言开发的一个科学集成的开发平台。它其中集成的功能能够让用户很方便地下载需要的软件包，并且可以便捷地管理软件包，与此同时用户可以自己创建开发所需的虚拟环境。Anaconda 主要用在当下火热的数据科学学科、机器学习和预测分析方面。Anaconda 拥有超过 1400 个软件包，附带了 Conda、虚拟环境管理和 Python 等在内 180 多个科学包和依赖项，它是在 Conda 基础上发展出来的。Anaconda 中包含了 Anaconda Navigator、Anaconda Prompt、Jupyter Notebook 和 Spider 等。Anaconda Navigator 是图形用户界面，可以在不使用命令行的情况下管理软件包、创建虚拟环境和管理路径。Conda 是一个开源的、跨平台和语言无关的软件包管理和系统管理系统。使用 Conda，可以安装、运行和升级科学软件包和相应的依赖项，而且能够在计算机中便捷地创建、保存、加载和切换环境。在开发人员需要多个不同的虚拟环境时，用 Conda 创建不同的虚拟环境，然后再进行切换加载环境对于开发人员来说是非常便捷的。

Jupyter Notebook 是一个开源的网页应用，用户能够在网页中新建和分享包含多种形式的文档，主要包括可随时编写运行的代码、方程式、说明性的文本文字和可视化的内容等。Jupyter Notebook 的主要用途有清洗数据、转换数据、数据可视化和机器学习等等。通俗地讲，用户在 Jupyter Notebook 应用中创建的文档要在浏览器中打开，即用网页的形式展示给用户，用户可以在打开这个文档后随时编写代码和运行代码，代码是一块一块展示在页面中，可以把实现不同功能的代码放在不同的代

代码块里，便于用户观察和修改代码，运行一个代码块之后它的结果会直接在它的下面显示。同时，对于编程人员来说，在同一个代码文档中也可以写下注释类的内容，可以在需要解释说明的代码块上方再添加一个块，在这里可以写下文字文本，对下面这段代码的原理功能进行解释，这种文本是以 Markdown 语法编写的。Jupyter Notebook 主要由两部分构成：展示部分和存储部分，即网页和文档。网页应用可以说是其展示形式，网页中有代码块，也有写说明文字的位置。在网页中可以实现各种功能，用户在网页中进行编写代码、说明文档、数学公式等。Jupyter Notebook 本身就是文档，以文件的形式存储，这些文件是 JSON 文件，文件的后缀名为.ipynb，存储为 JSON 形式也是便于对不同的版本的控制，并且这种文件方便用户跟别人共享。

Jupyter Notebook 的便捷之处在于：编程方面，可以语法高亮、还可以自动缩进、还有 tab 补全代码的功能；编程人员可以在打开的网页中可直接通过浏览器运行代码，结果直接显示在这个代码块下，十分方便；它还可以以 HTML、LaTeX、PNG 等形式对结果进行展示，格式十分正规；在对代码进行解释说明的文字描述中使用 Markdown 语法进行编写；用户可以使用 LaTeX 编写数学性说明。Jupyter Notebook 是在 Anaconda 中集成好的，安装 Anaconda 之后就可以直接使用它，适合进行深度学习等的编写、说明和实现。

### 2.1.2 深度学习框架 TensorFlow 与 Keras

TensorFlow 是一个端对端的免费的开源软件库，主要应用于机器学习等。它是一个符号性的数学库，它使用数据流图进行数值计算。在数据流图中，图的节点代表数学操作，图的边代表多维数据数组，即张量（tensor），这些张量被这些数学操作联系起来。这也正是 TensorFlow 名字的来源。

TensorFlow 的主要特征有：十分灵活，只要用户能够将进行的计算呈现为一个数据流图，那么就可以使用 TensorFlow，同时用户也可以自己构建出数据流图，也可以在 TensorFlow 基础上写自己的“上层库”；多平台运行，它可以在 CPU 上计算运行，也可以在 GPU 上计算运行，在不同的系统中、在不同的平台上如台式机、服务器等都可以运行。

Keras 是一个高级神经网络的应用程序接口，用 Python 编写而成，它必须以 TensorFlow、CNTK 或 Theano 作为后端，这样才可以运行。用 Keras 开发的特点是

十分快速，相对于其他的深度学习的框架来讲它能够让开发人员在尽量短的时间内得到实验结果，从而判断自己的想法是否有意义。Keras 秉持着以下十分优良的设计准则：Keras 对用户十分友好，它提供给用户的应用程序接口十分的一致，也很简单，对于用户常用的用例，它能够尽量简化操作，把操作难度尽量降到最低，避免操作让用户形成认知上的困难；模块化，高内聚，低耦合，在 Keras 中，单独的模块可以被组合形成序列或者图，即所说的模型；容易扩展，用户可以很轻松地将新的功能模块加入到模型中；Keras 的模型是由 Python 语言实现的。Keras 中的模型是组织整个网络层的方式。Keras 十分适合在以下场景使用：在进行简单快速的原型设计中；在使用卷积神经网络（CNN）和循环神经网络（RNN）或者两者结合使用时；可以在 CPU 也可以在 GPU 中运行。由于 Keras 本身的原因，所以只有在 Python2.7-3.6 的开发环境中可用，低版本的 Python 是不支持的。

新手开始使用 Keras 很快也很容易。Keras 中主要的模型是 Sequential（序列）模型，以构建 Sequential 模型为例，具体构建和学习模型的流程如下。（通过流程图展现以下内容）

```
from keras.models import Sequential  
model = Sequential()
```

可以使用.add()方法来对模型进行堆叠，

```
from keras.layers import Dense  
model.add(Dense(units=64, activation='relu', input_dim=100))  
model.add(Dense(units=10, activation='softmax'))
```

构建成功模型后，使用.compile()方法来配置学习过程，

```
model.compile(loss='categorical_crossentropy',  
              optimizer='sgd',  
              metrics=['accuracy'])
```

然后，可以批量地迭代训练数据，

```
# x_train 和 y_train 是 Numpy 数组  
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

最后，对模型性能进行评估，

```
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

## 2.2 开发环境搭建

### 2.2.1 算法实现环境介绍

- 1) 软件环境:
  - a) 操作系统: Windows 10
  - b) 开发平台: Anaconda3 4.2.0、Jupyter Notebook
  - c) 开发语言: Python3.6.1
  - d) 运算平台: CUDA9.0
  - e) 深度学习加速库: CuDNN7.0
  - f) 深度学习网络框架: TensorFlow-gpu1.12.0、Keras2.2.4
- 2) 硬件环境:
  - a) CPU: Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
  - b) GPU: NVIDIA GeForce GTX 1070
  - c) 内存: 16GB

### 2.2.2 算法实现环境搭建

#### 1) Anaconda 平台

Anaconda 是一个科学管理科学包和虚拟环境的开源平台。根据本系统的需要, 下载 Anaconda3 4.2.0, 它自带的 Python 版本为 3.5, 但是不影响在虚拟环境中使用其他的 Python 版本。下载安装包后, 进行安装, 安装过程中开发人员不要将其安装路径加入到系统环境变量中。安装完成后, 进行验证: 打开 Anaconda Navigator, 若能成功打开, 则说明安装成功。

因为我们需要用到虚拟环境, 所以需要在 Anaconda 中确认 Conda 是否被安装。打开 Anaconda Prompt, 输入 “conda --version”, 若输出 Conda 的版本, 则说明已安装。

在使用 Anaconda 时会有一些常用命令, 在此罗列出来, 便于了解, 如下表 2-1 Anaconda 常用命令表:

表 2-1 Anaconda 常用命令表



命令	对应的操作
<code>conda/pip install &lt;package&gt;</code>	安装依赖包
<code>conda/pip uninstall &lt;package&gt;</code>	卸载依赖包
<code>conda create -n &lt;env&gt; python==x.x</code>	创建 Python 版本为 x.x 的虚拟环境
<code>activate &lt;env&gt;</code>	激活这个环境
<code>deactivate &lt;env&gt;</code>	退出当前环境
<code>conda list</code>	查看已安装的包列表

考虑到实现本系统的兼容性问题，为了不影响操作系统平台，我们利用上述命令创建一个 Python 版本为 3.6.0 的虚拟环境，然后在这个虚拟环境搭建深度学习的框架。

## 2) TensorFlow-gpu1.12.0 和 Keras2.2.4 的安装

TensorFlow-gpu 是作为 Keras 的后端才进行安装的，这一部分的安装需要注意的问题很多，包括版本兼容的问题等，比较耗时费精力，需要有一定的耐心。

- 根据 Keras 安装文档，首先确定自己使用的主机硬件配置和计算机的操作系统版本达到要求；
- 安装编译环境 Microsoft Visual Studio 2015 Update 3；
- 安装 Python 环境：

已经安装 Anaconda3 4.2.0，其中集成了 Python 发行版，而且也集成了很多优秀的科学计算 Python 库；

- 安装 CUDA：

CUDA Toolkit 是 NVIDIA 公司针对 GPU 编程的基础工具包，也是驱动显卡计算的核心技术工具。安装 CUDA9.0，下载安装包时，具体要求根据自己的安装环境进行选择即可，如图 2-1 所示。

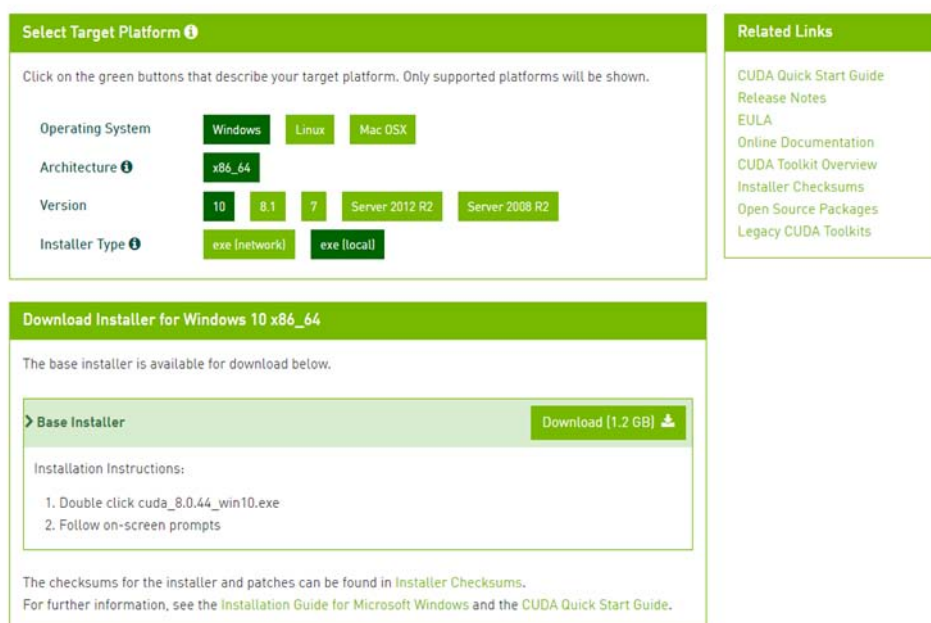


图 2-1 安装 CUDA 示意图（图不清楚）

e) 加速库 CuDNN:

开发人员首先注册 Nvidia 开发者账号，然后去官网下载。选择兼容的版本，这里选择 7.0 版本。下载解压，然后将解压后的文件夹复制到安装 CUDA 的地方覆盖对应文件夹。

f) Keras 框架的搭建:

首先，在 Anaconda 中使用前面介绍的命令，打开创建的虚拟环境，根据要求安装 TensorFlow 作为 Keras 的后端。使用以下代码安装 GPU 版本的 TensorFlow，考虑到版本的兼容问题，这里选择版本号为 1.12.0 的 TensorFlow:

```
pip install --upgrade tensorflow-gpu==1.12.0
```

然后安装 Keras:

```
pip install keras -U --pre
```

最后，查看 Keras 是否安装成功，在 Anaconda Prompt 中输入 Python 命令，进入 Python 环境中，输入以下代码:

```
import keras
```

没有报错就说明 Keras 已经成功安装。至此，实现该系统需要的基本环

境已经搭建完成。

## 2.3 数据集及预处理

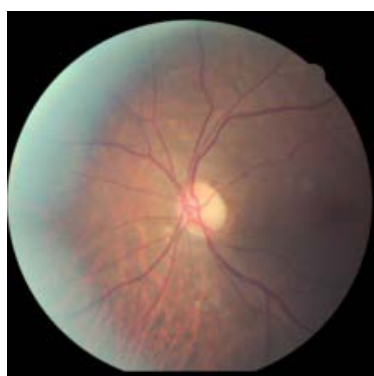
### 2.3.1 数据集

数据集是大量的在各种成像条件下拍摄的高分辨率的彩色眼底图像，每个病人各提供一张左眼底图像和一张右眼底图像，并且左右眼底图像的命名都是用病人的ID来标记的（例如，1\_left.jpeg是病人ID为1的左眼底图像）。这些彩色眼底图像来自不同型号和类型的相机，这可能会影响左右眼睛的视觉外观。从被标记的训练数据集可以看到，这些彩色眼底图像一共被分为5类，用0-4对每张图像进行分类来表示糖尿病视网膜病变的严重程度。数字与糖尿病视网膜病变的严重程度对应关系如下表2-1所示。

表 2-2 糖尿病视网膜病变严重程度分级表（表要规范，要有表头，全文用三线表）

0	无明显糖尿病视网膜病变
1	轻微非增生性糖尿病视网膜病变
2	中度非增生性糖尿病视网膜病变
3	重度非增生性糖尿病视网膜病变
4	增生性糖尿病视网膜病变

从数据集中随机选择这5种类别的各一张图像，可以看到如图2-2所示。



a)正常的



b)轻微的

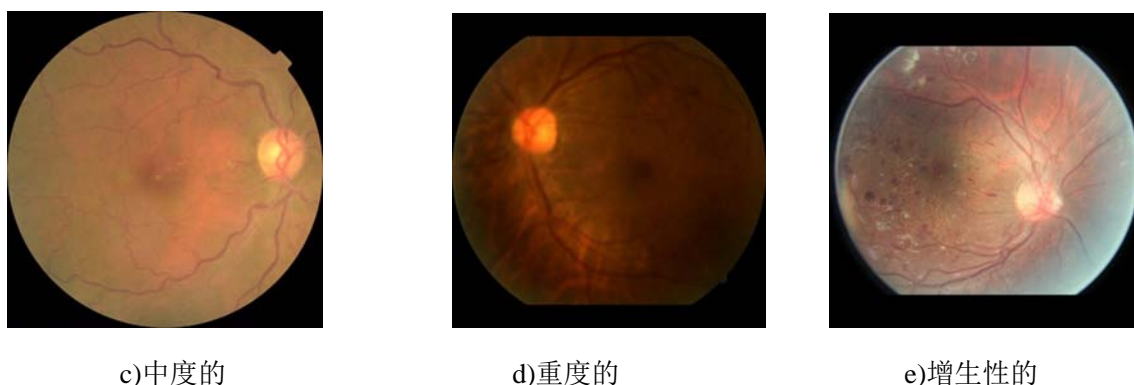


图 2-2 数据集抽样展示图

从图 2-2 中可以看出，图 2-2a 眼底图为 No DR 患者的眼底图，未发生视网膜病变；图 2-2b 眼底图为 Mild NPDR 患者的眼底图，且为非增生性的；图 2-2c 眼底图为 Moderate NPDR 患者的眼底图，未发生增生；图 2-2d 为 Severe NPDR 患者的眼底图，为非增生性的；图 2-2e 为 PDR 患者的眼底图。由于眼底图的特征与糖尿病视网膜病变的严重程度之间有非常紧密的联系，所以在本算法中通过对彩色眼底图特征的学习对糖尿病视网膜病变的严重程度进行分类。

本实验中共采用了 10,000 张原始的眼底图像数据集，并且对原始眼底图像进行归一化预处理，将原始眼底图像和经过归一化处理的眼底图像结合起来形成一个对比实验，后面对这两种数据集的数据增强的操作都是完全一样的。两组实验中训练集和测试集的比例分别为 75% 和 25%。这 10,000 张图像会被调整成为  $512 \times 512$  大小，训练集会经过一系列的数据预处理操作，而对测试集数据不会进行预处理。

利用代码查看数据集中每个类别的数据量，发现本课题的数据集是非常不均衡的，如图 2-3 所示。

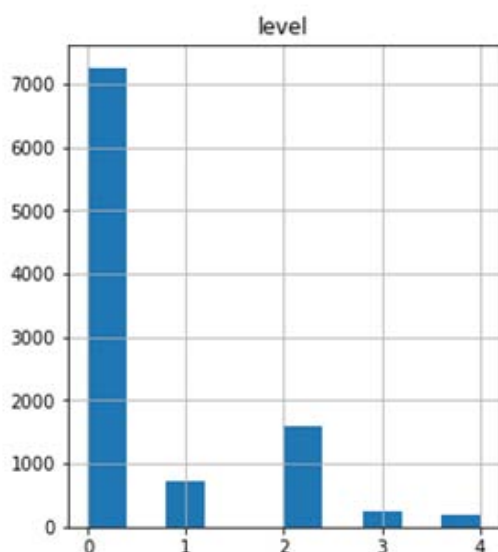


图 2-3 数据集中五种类别的分布

从图 2-3 的数据分布中可以看出，在这 10,000 张眼底图像中，属于正常糖尿病患者（对应上图中的数字 0）的眼底图的数据量是最多的，超过了 7000 张；而属于重度（对应数字 3）和增生性（对应数字 4）的糖尿病视网膜病变的眼底图的数据非常少，只有 100-200 张；属于轻微（对应数字 1）病变的眼底图有大概 800 张，属于中度（对应数字 2）病变的眼底图还是相对多一些，有 1600 张左右。由此可得知，数据集中这五个类别下的数据量差距很大，数据集是十分不均衡的，因此后续的操作中需要对数据进行数据增强。

### 2.3.2 数据的预处理

根据本课题的训练数据集图像的大小，考虑到网络输入的情况，将图像的大小调整为  $512 \times 512$ ，对其进行分类查看其每个类别的数量，发现数据存在类别的不均衡性，然后本课题对每个级别随机抽样相同的数量。由于有些类别的数据中含有相同的图像，所以对这些图像进行数据增强，即进行随机的水平旋转、色彩增强、饱和度和度增强等。

#### 1) 对数据集进行归一化

从原始的数据集中可以看到，每张图片的颜色和亮度都有十分大的区别，这是由相机的类型不同、拍摄的环境不同导致的。而这些巨大的差异与糖尿病视网膜病变的严重程度是无关的，所以，我们对感兴趣的部分进行截取，然后对图像进行重采样，最后对颜色、亮度进行归一化处理。

对感兴趣的部分进行截取：由于原始的眼底图像中有黑色背景，且这些黑色边缘部分中没有与 DR 相关的病理特征，所以我们首先去除边缘。

对截取后的图像进行重采样：截取后的图像尺寸偏小，对图像进行重采样。这里采用的实现方法是插值的方法。

对图像的颜色、亮度进行归一化：由于采集的图像之间出现颜色亮度较大的差异，所以利用下面的归一化公式进行， $r=R \div (R+G+B)$ ,  $g=G \div (R+G+B)$ ,  $b=1-r-b$ ，其中  $r$ 、 $g$ 、 $b$  是变换后的像素值， $R$ 、 $G$ 、 $B$  是变换前的像素值。

## 2) 眼底图像大小调整

该数据集中的原始图像的尺寸不一致，原始图像从  $2500 \times 2000$  到  $4000 \times 2500$  不等，而已经过归一化处理的图像的尺寸为  $1000 \times 1000$ 。对于神经网络来说，这种尺寸的图像输入太大，我们选择把图像的尺寸调整为  $512 \times 512$ ，原因主要有两点：第一，图像中诸如微动脉瘤等病变区域是很小的，图像的尺寸如果过小会容易丢失这些关键性的特征，是不利于对这些区域的特征学习的；第二，考虑到 GPU 的性能，图像的尺寸如果太大会使卷积神经网络的训练过于缓慢，所需耗费的时间太多<sup>[3]</sup>。

## 3) 数据增强

由于数据集中类别不均衡，所以对训练数据集进行了数据增强，其中包括旋转、对比度调整、颜色调整、饱和度调整等。旋转包括水平和垂直旋转，随机地选择水平或垂直旋转；对比度调整、色彩调整、饱和度调整都是在预先设置的阈值范围内进行随机地调整。

## 2.4 卷积神经网络 Inception V3 和注意机制

卷积神经网络的发展十分快速，从最开始出现的 LeNet，到确立卷积神经网络在深度学习中的地位 AlexNet，到将网络加深的 VGG 网络，再到网络更深更宽的 Google Inception Net，到里程碑性质的网络 ResNet，再到目前的 Dense Net。

在这一节中我主要介绍本实验中使用的深度学习网络 Inception V3，并且在这个网络的基础上新加了注意机制。

Inception V3 是 GoogLeNet 的第三个版本，由谷歌提出。GoogLeNet 在 2014 年的 ImageNet Large Scale Visual Recognition Competition (ILSVRC) 中取得第一名，该网络以结构上的创新取胜，通过采用全局平均池化层取代全连接层，极大的降低了参数量，是非常实用的模型，一般称该网络模型为 Inception V1。随后的 Inception V2

中，引入了 Batch Normalization 方法，加快了训练的收敛速度。在 Inception V3 模型中，通过将二维卷积层拆分成两个一维卷积层，不仅降低了参数数量，同时减轻了过拟合现象。

Inception V3 网络一共有 47 层，大致的网络拓扑结构如图 2-4 所示。

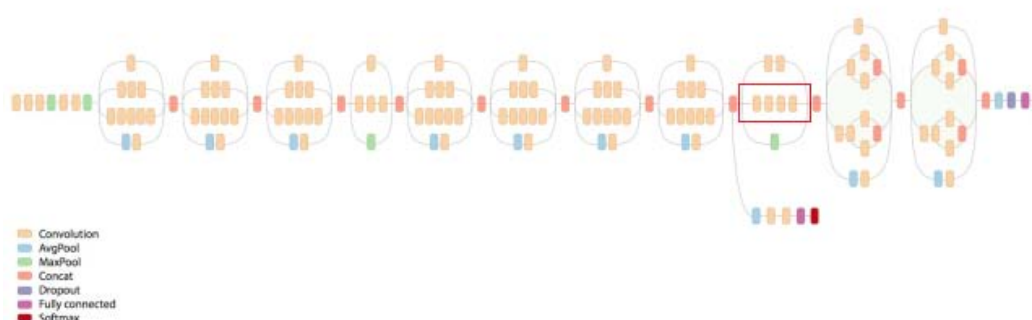


图 2-4 Inception V3 的网络拓扑结构

Inception V3 与之前的 GoogLeNet 比较，主要有三方面的改进：一、为了减少计算量，将大尺寸的滤波器替换为多层的小尺寸卷积滤波器，同时采用非对称卷积，将  $3 \times 3$  的卷积核分解为  $3 \times 1 + 1 \times 3$  的卷积；二、为了加快收敛，引入辅助分类器；三、同样是为了降低计算量，加入池化层，和卷积层响应来降低特征图大小。

Inception V3 网络与之前的卷积神经网络相比较的优点有：提供更“宽”的网络，使网络可以选择更好的特征；Max Pooling 可以去掉上一层卷积而产生的冗余信息；网络中的结构还会产生减少参数和模型融合的效果。

在这个网络的基础上引入了注意机制，Attention 机制的本质是从人类视觉注意力机制中获得灵感。大致含义是我们人类的视觉在感知东西的时候，一般不会是一个场景从到头看到尾每次都看，而往往是根据需求观察注意特定的一部分。而且当我们发现一个场景经常在某部分出现自己想观察的东西时，我们就会进行学习在将来再出现类似场景时把注意力放到该部分上。Attention 机制是一系列注意力分配参数，即权重参数。利用这种机制能够更好地选择特征，能够更好地把握全局和局部之间的联系。



## 2.5 模型训练

### 2.5.1 数据集准备

这一部分进行系统的核心部分，进行对模型的训练。首先，我们进行数据集的准备。将未经过处理的原始眼底图像和进行颜色、亮度等归一化操作的眼底图像各取出 10,000 张，然后接下来对训练集进行相同的数据预处理和数据增强的操作，对测试集的眼底图像只进行裁剪的处理操作。

#### 1) 数据预处理

对训练集的数据查看每个类别下的数据量，发现这个数据集是一个类别不均衡的数据集，所以我们考虑到在进行数据预处理后需要进行数据增强，使每个类别下的数据量保持一致。

之前已经提到，考虑到神经网络的训练速度和硬件条件的运行速度局限两个方面的因素，将输入到网络的图像尺寸调整到  $512 \times 512$  大小。具体实现如下：

```
if intermediate_trans == 'scale':  
    X = tf.image.resize_images(X, out_size)  
elif intermediate_trans == 'crop':  
    X = tf.image.resize_image_with_crop_or_pad(X, out_size[0], out_size[1])
```

调整尺寸后的图像如图 2-5 所示。



图 2-5 调整尺寸后的图像



## 2) 数据增强

对训练集进行随机的水平、垂直翻转、水平和垂直翻转或者不翻转；在设定的阈值内随机地调整亮度；在预设的色彩范围内随机地调整色彩；在设置的阈值范围内随机地调整饱和度和对比度。具体实现如下：

```
if horizontal_flip:
    X = tf.image.random_flip_left_right(X)
if vertical_flip:
    X = tf.image.random_flip_up_down(X)
if random_brightness:
    X = tf.image.random_brightness(X, max_delta=0.1)
if random_saturation:
    X = tf.image.random_saturation(X, lower=0.75, upper=1.5)
if random_hue:
    X = tf.image.random_hue(X, max_delta=0.15)
if random_contrast:
    X = tf.image.random_contrast(X, lower=0.75, upper=1.5)
```

数据增强后的图像如图 2-6 所示。

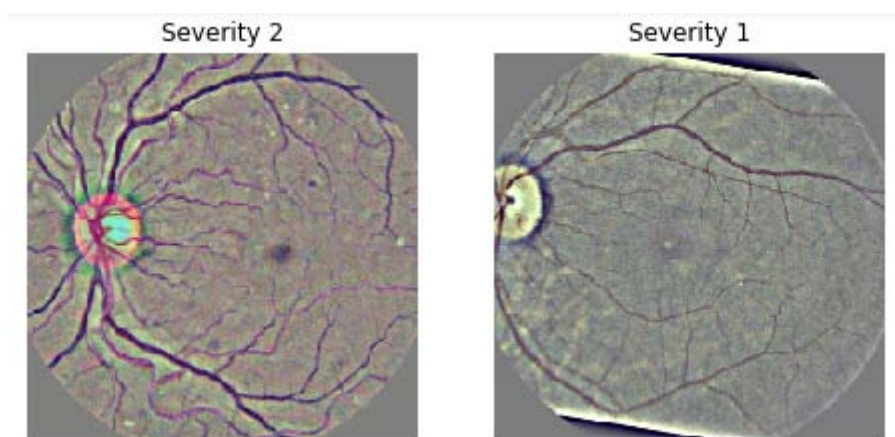


图 2-6 数据增强后的图像

### 2.5.2 基于 Inception V3 的训练方法

观察到注意机制的优点后，我们利用基于 Inception V3 的训练模型作为基础模型，将它和注意机制结合起来。构建这种注意机制，在池化之前将 GAP(Global Average Pooling，全局平均池化)中的像素打开和关闭，然后根据像素个数重新缩放结果。构建网络的结构如图 2-7 所示。

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 512, 512, 3)	0	
inception_v3 (Model)	(None, 14, 14, 2048)	21802784	input_1[0][0]
batch_normalization_95 (BatchNo	(None, 14, 14, 2048)	8192	inception_v3[1][0]
dropout_1 (Dropout)	(None, 14, 14, 2048)	0	batch_normalization_95[0][0]
conv2d_95 (Conv2D)	(None, 14, 14, 64)	131136	dropout_1[0][0]
conv2d_96 (Conv2D)	(None, 14, 14, 16)	1040	conv2d_95[0][0]
conv2d_97 (Conv2D)	(None, 14, 14, 8)	136	conv2d_96[0][0]
conv2d_98 (Conv2D)	(None, 14, 14, 1)	9	conv2d_97[0][0]
conv2d_99 (Conv2D)	(None, 14, 14, 2048)	2048	conv2d_98[0][0]
multiply_1 (Multiply)	(None, 14, 14, 2048)	0	conv2d_99[0][0] batch_normalization_95[0][0]
global_average_pooling2d_1 (Glo	(None, 2048)	0	multiply_1[0][0]
global_average_pooling2d_2 (Glo	(None, 2048)	0	conv2d_99[0][0]
RescaleGAP (Lambda)	(None, 2048)	0	global_average_pooling2d_1[0][0] global_average_pooling2d_2[0][0]
dropout_2 (Dropout)	(None, 2048)	0	RescaleGAP[0][0]
dense_1 (Dense)	(None, 128)	262272	dropout_2[0][0]
dropout_3 (Dropout)	(None, 128)	0	dense_1[0][0]
dense_2 (Dense)	(None, 5)	645	dropout_3[0][0]
Total params: 22,208,262			
Trainable params: 399,334			
Non-trainable params: 21,808,928			

图 2-7 增加注意机制后的网络结构

下面，对本实验中使用的网络结构进行介绍。从图 2-7 中可以看到，该网络中第 1 层为输入层，输入到下一层的为预处理的图像矩阵；第 2 层为在 inception V3 网络中预训练的模型，使用的是 Keras 中预先训练的模型；第 3 层为规范化 BatchNormalization 层，它可以将每个 batch 在前一层中的激活值进行规范化，使输出数据的均值接近 0，标准差接近 1，能够加速收敛；第 4 层为 Dropout 层，用于防

止过拟合；第 5 层开始为加入的注意机制，这一层为二维卷积层，有 64 个  $1 \times 1$  的卷积核，对二维输入进行卷积窗卷积运算，输出 64 维的输出张量；第 6 层为二维卷积层，有 16 个  $1 \times 1$  的卷积核，输出 16 维的输出张量；第 7 层为包括 8 个  $1 \times 1$  的卷积核的二维卷积层，能够输出 8 维的输出张量；第 8 层为有 1 个  $1 \times 1$  的卷积核的二维卷积层，激活函数为 S 型函数，输出 1 为的 4D 张量；第 9 层为卷积层，将这种注意机制传播到所有的 channel 中；第 10 层为 Multiply 层，将注意层和预先训练模型的张量进行逐元素积；第 11 层为 GlobalMaxPooling2D 层，为上一层中的输出张量施加全局最大值池化；第 12 层为 GlobalMaxPooling2D 层，为注意机制卷积层的输出张量施加全局最大值池化；第 13 层为 Lambda 层，主要是 11 层和 12 层的输出张量对应相除，解释注意模型中缺失的值；第 14、15 层为 Dropout 层，防止过拟合；第 16 层为 Dense 层，全连接层，实现分类，输出预测结果。

对原始数据和进行色彩归一化的图像分别进行训练。设置好训练的参数后，就可以进行训练，然后将训练所得模型保存下来，并且将效果最好的权重保存下来。

## 2.6 实验结果

利用未进行数据增强的测试集对训练所得模型的性能进行评估，结果如图 2-8 所示。（通过表给出实验结果统计，不要用图）

Accuracy on Test Data: 0.29				
	precision	recall	f1-score	support
0	0.75	0.24	0.36	1749
1	0.11	0.62	0.19	239
2	0.24	0.33	0.28	452
3	0.27	0.06	0.10	65
4	0.28	0.22	0.24	55
avg / total	0.58	0.29	0.32	2560

a) 未进行归一化的数据集

Accuracy on Test Data: 0.68				
	precision	recall	f1-score	support
0	0.70	0.98	0.81	1752
1	0.13	0.02	0.03	247
2	0.30	0.02	0.03	448
3	0.26	0.10	0.14	62
4	0.31	0.08	0.12	51
avg / total	0.55	0.68	0.57	2560

b)进行归一化的数据集

图 2-8 模型性能对比

图 2-8a 是利用未进行归一化的数据集训练的模型，对其利用测试集进行测试所得的准确率，结果为0.29；图 2-8b 是利用进行归一化处理的数据集训练所得的模型，对其利用相同的测试集进行测试所得准确率，结果为 0.68。从正确率可以看出，进行归一化的数据集训练出的模型比原始数据集训练出的模型的性能好很多，所以说明数据的归一化是非常必要的。

## 2.7 本章小结

本章主要介绍了实现系统的核心部分-训练模型。首先，对原始数据进行裁剪、调整尺寸和归一化处理，形成一个新的数据集，并与原始数据集形成对比实验组，从而评估这些预处理操作是否能够提高训练的模型的性能。然后，从数据集的介绍发现数据集的数据分布十分不均衡，对数据集进行尺寸的调整和数据增强，从网络角度和硬件支持角度为输入的图像选择合适的尺寸。从水平或垂直翻转、亮度调整、色彩调整、饱和度调整、对比度调整等方面对两组图像进行数据增强。最后，对两组数据集分别在同一个网络中进行训练，得到两个不同的模型，然后使用同一个测试集对两个模型进行性能评估，比较分析，发现使用预处理数据集训练出的模型效果明显优于原始数据集训练所得模型的分类效果。

## 第3章 糖尿病视网膜病变分类系统设计与实现

本章从整个系统的实现的角度出发，严格按照开发系统流程，完整地介绍开发糖尿病视网膜病变智能分类系统（简称智能分类系统）的整个过程，主要从系统需求分析、系统总体设计、系统详细设计、实现系统的开发环境搭建和实现过程这几个方面进行简单的介绍。

### 3.1 系统需求分析与总体设计

#### 3.1.1 系统需求分析

系统的需求分析是为了能够清晰地确定用户对糖尿病视网膜病变智能分级系统的需求，这样才能够使开发人员更加清楚自己的工作。这部分的系统需求分析工作主要包括以下几个方面：开发人员对糖尿病视网膜病变智能分级系统的了解、用户对糖尿病视网膜病变智能分级系统的需求、数据需求、功能流程需求、以及针对本系统的功能性需求和非功能性需求。系统的需求分析是完成该糖尿病视网膜病变智能分级系统的系统总体设计和各个功能子模块编写的重要基础。本次系统需求分析针对于糖尿病视网膜病变智能分级系统，采用用例分析的方法，采用统一建模语言（UML）完成对本智能分级系统的需求描述。

##### 1) 系统目标

本系统是在 Windows 平台下的糖尿病视网膜病变智能分级系统，可以读取单张有相机采集到的彩色眼底图，对彩色眼底图进行调整尺寸，主要的目标是从本地文件中选择彩色眼底图、对彩色眼底图进行尺寸调整、对彩色眼底图进行智能分级，从而向用户展示出选择的这张彩色眼底图的糖尿病视网膜病变的级别。

##### 2) 功能性需求分析

本系统是在 Windows 平台下的基于彩色眼底图的糖尿病视网膜病变智能分期系统，参与本系统的使用者只有一类用户，就是医生。通过业务流程的描述以及系统目标的明确，本系统的功能需求主要包括：彩色图像获取功能需求、彩色图像显示功能需求和对彩色图像进行糖尿病视网膜病变的分级。本系统的这三个功能需求是能够完整地实现系统的功能的。本系统的总体用例图如图 3-1 所示。

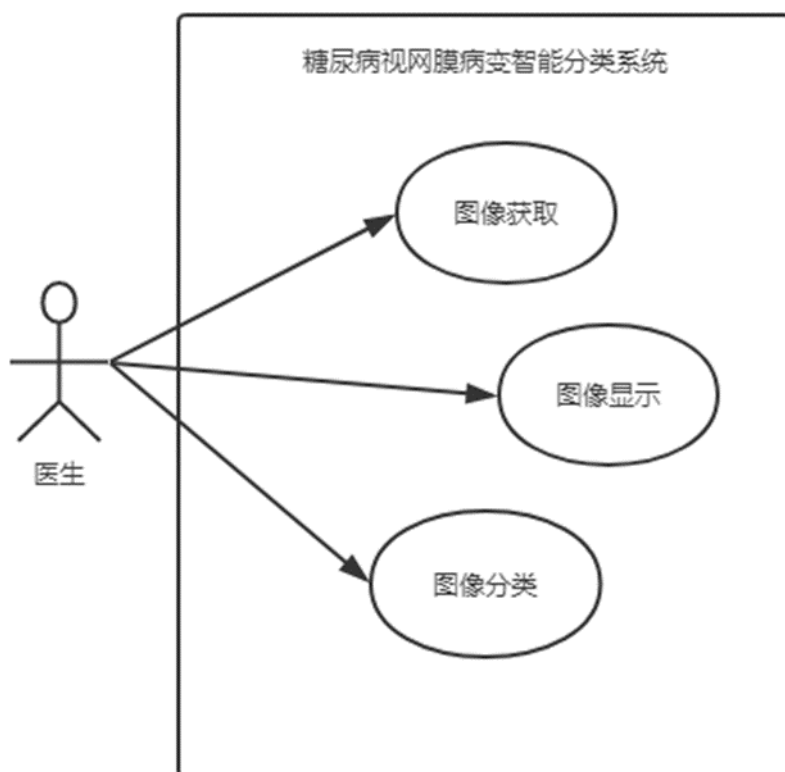


图 3-1 糖尿病视网膜病变智能分类系统的总体用例图

a) 对图像获取子功能的需求分析：

图像获取功能是指用户可从本地磁盘任意位置选择一张彩色图像。图像获取功能是本系统功能实现的第一步，也是十分重要的一步，图像获取功能的用例图如图 3-2 所示。

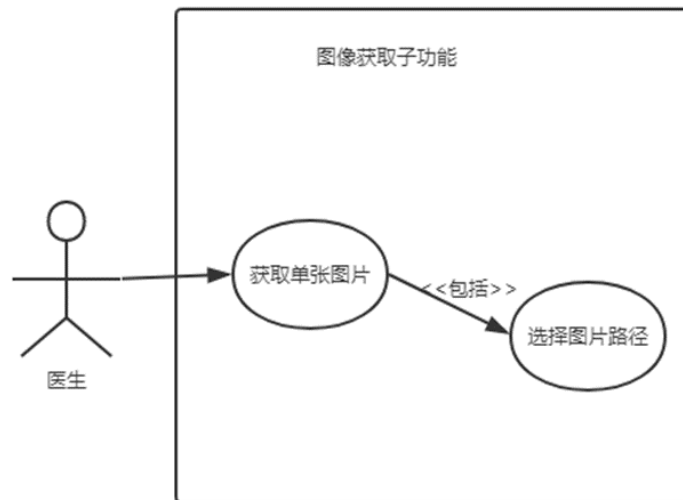


图 3-2 图像获取子功能的用例图

从上面的图像获取功能的用例图中可以看到，这个子功能只能选择单张图片，然后这个功能下包括选择图片在本地磁盘中所在的位置，根据路径才能获取图像。

b) 对图像显示子功能的需求分析：

图像显示是糖尿病视网膜病变智能分类系统中不可缺少的一个部分，它是图像获取的可视化部分，便于向用户进行展示，可以让用户（医生）再次确定自己选择的图像无误，使系统和用户之间的交互良好。图像显示子功能的用例图如图 3-3 所示。

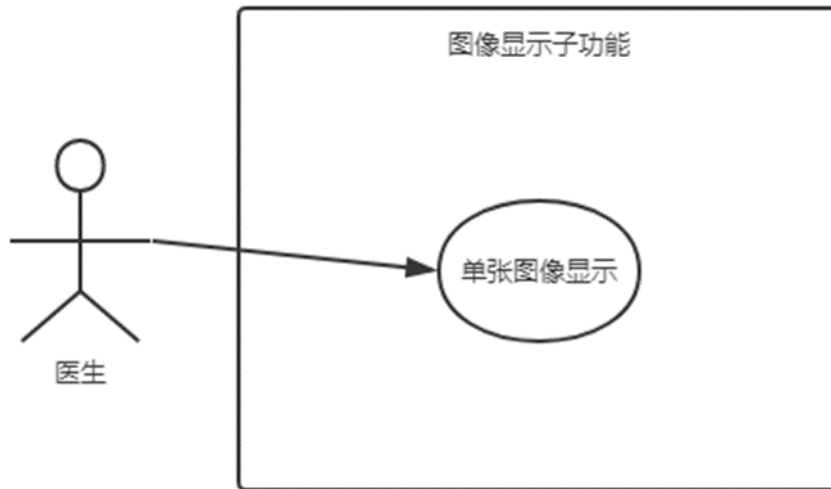


图 3-3 图像显示子功能的用例图

从上面的用例图中可以清晰地看到，图像显示子功能只支持显示单张图像，然后在单张图像显示功能下没有子结构的支持。

c) 图像分类子功能模块的需求分析：

图像分类子功能是指对图像进行智能分类并显示结果，这个子功能模块是系统的核心功能。这部分的主要功能就是将用户选择的图像在系统后台实现对这张彩色眼底图像的糖尿病视网膜病变的严重程度的判断，然后将结果在系统中以可视化的形式展现给用户。对图像进行智能分类并显示子功能的用例图如图 3-4 所示。



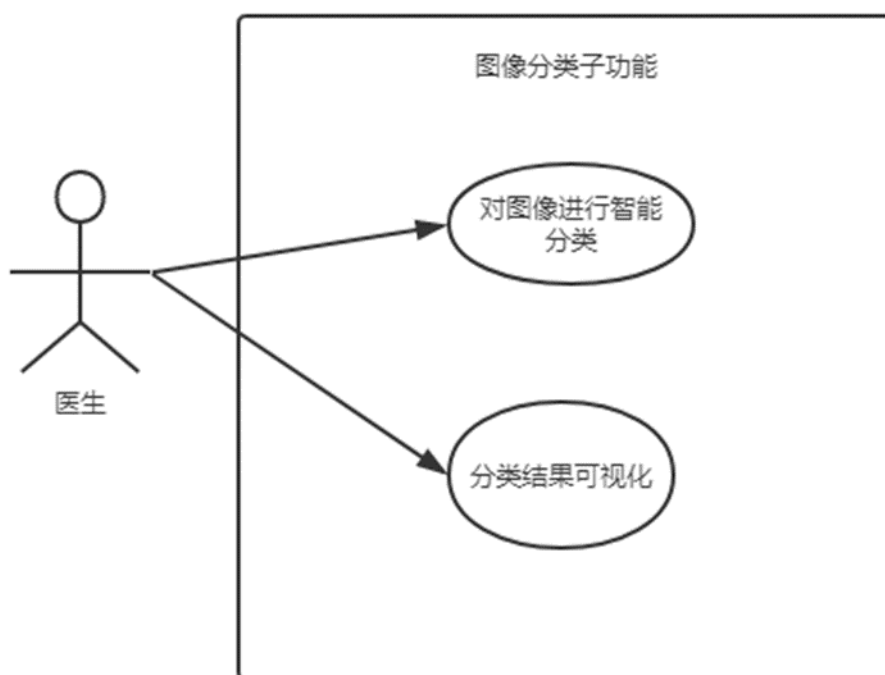


图 3-4 图像分类子功能的用例图

从上面的图像分类子功能的用例图中可以清晰地看到，该功能实现了系统的最重要的功能，它包括两个功能模块，即对用户选择的图像在系统中实现分类，并且将分类的结果可视化。

### 3) 非功能性需求分析

非功能性需求分析是指根据一些条件，判断系统运作的情形或其特性，而不是针对系统特定行为的需求。非功能性需求主要包括安全性、可靠性、互操作性、健壮性、易使用性、可维护性、可移植性、可重用性、可扩充性。这一节的非功能性需求分析主要针对糖尿病视网膜病变智能分类系统，主要从系统性能需求、易使用性、环境需求、系统的外部接口需求等这几个方面进行分析。

#### (1) 性能需求：

性能需求主要是指系统的响应时间、计算精度、资源占用率等。具体也就是说，在糖尿病视网膜病变智能分类系统中，用户使用该系统时对该系统的处理速度、资源占用等方面的要求和忍受程度。响应速度是指功能完成的时间，和客观环境、数

据量级、主观感受等都有关系。因此，本系统的响应时间用户必须能够忍受的范围之内，在 95% 的情况下，系统的响应时间不应该超过 3 秒，读取时间的耗时不能够超过 1 秒钟。从计算精度角度来讲，本系统在计算数据时不能够出现差错。从资源占用率来讲，在计算数据时占用计算机的系统资源要尽量少，CPU 占用率要小于等于 50%，内存占用率要小于等于 50%。

### （2） 易使用性：

易用性需求是指用户在使用糖尿病视网膜病变智能分类系统时对系统是否易用、操作的主观感受，系统中的功能操作是否简便易用，系统的界面是否美观，用户与系统的交互是否良好。本系统的易用性需求为界面简单，用户能够很轻松和系统之间进行交互，系统的操作简便明了。

### （3） 环境需求：

环境需求是指用户在使用糖尿病视网膜病变智能分类系统时的运行环境的要求。本系统的开发可以在 Windows 或者 Linux 操作系统下完成。要求本系统应当在 Windows 10 系统中部署环境并成功运行。

### （4） 外部接口需求：

外部接口需求是指用户对糖尿病视网膜病变智能分类系统对其他的第三方的外部软件、硬件等进行衔接的要求。要求本系统能够在硬件方面获取外接存储设备的数据。

## 3.1.2 系统总体设计

### 1) 系统总体结构设计

系统的总体设计是系统实现的重要一环，也是本糖尿病视网膜病变智能系统完成的重要的一部分。根据上一节中对本系统的功能性需求分析，确定了糖尿病视网膜病变智能分类系统的总体功能流程和各个子功能模块的用例图。根据上一节的非功能性需求，确定了实现本系统的性能需求、易用性、环境约束等非功能性的需求限制。结合两个方面的分析和考量，从本节开始，将从系统设计角度讨论并设计糖尿病视网膜病变智能分类系统的总体结构和功能结构。

本糖尿病视网膜病变智能分类系统主要实现的功能是用户选择需要进行分类的彩色图像，将彩色图像进行尺寸的调整，然后将其以可视化的形式展示给用户，最后系统后台对用户选择的彩色图像进行糖尿病视网膜病变智能分类，并将结果展示

给用户，至此，系统的全部功能完成与用户的交互。根据上一节的用例图，可知本系统的总体结构可以划分为图像获取、图像显示、图像分类这三个部分，具体系统的软件架构图如图 3-5 所示。

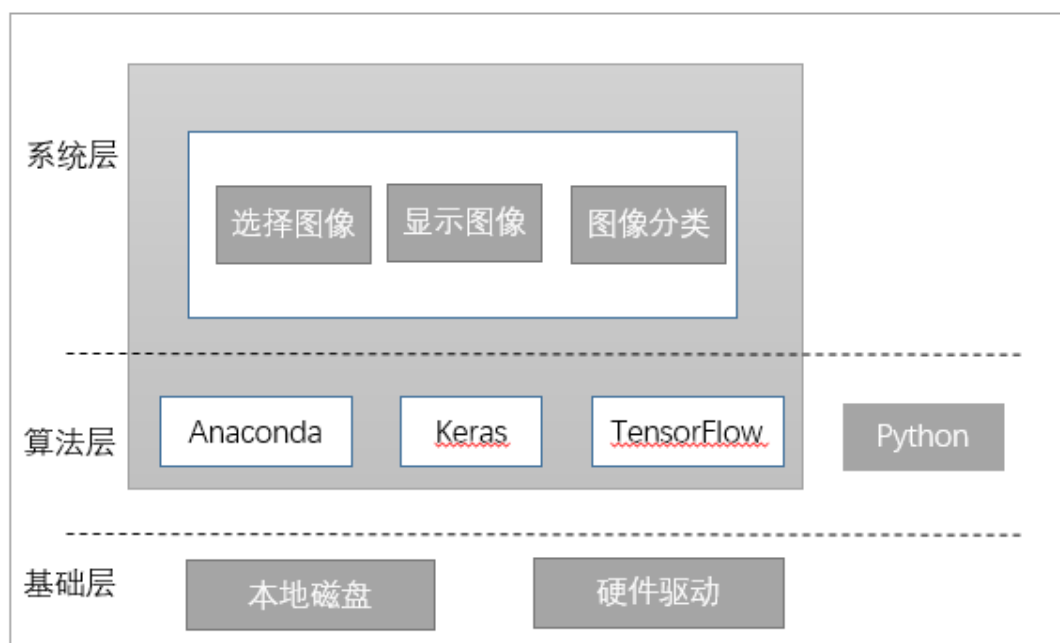


图 3-5 系统的软件架构图

从上面这个软件架构图中可以清晰地看出，系统基础层是由操作系统的硬件驱动程序组成。算法层包括包和环境管理工具 Anaconda、编码运行环境 Jupyter Notebook 和代码运行环境 TensorFlow、Keras，以及编程语言 Python。系统层主要由三个子功能模块，分别为图像选择模块、图像显示模块和图像分类模块，每个子功能模块都是在系统基础层和算法层的基础之上搭建起来的。

## 2) 系统功能结构设计

糖尿病视网膜病变智能分类系统的功能主要包括图像获取功能、图像显示功能和图像分类功能这三个部分。其中，图像获取功能为单张图像的读取；图像显示功能为将获取的图像进行在前端界面显示的功能；图像分类功能为对获取的图像根据训练所得的模型进行智能分类的功能。

糖尿病视网膜病变智能分类系统的软件结构图如图 3-6 所示。

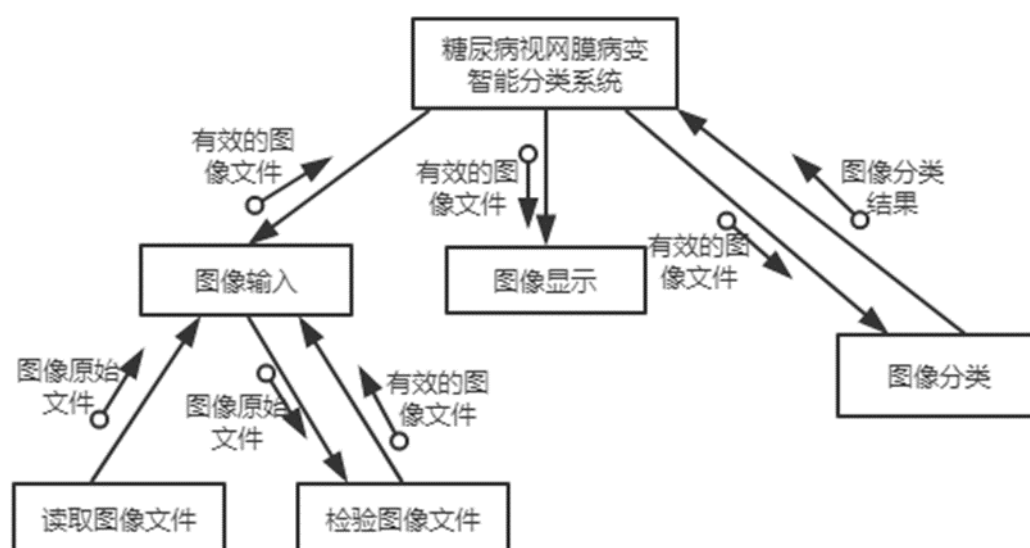


图 3-6 系统的软件结构图

从上面的软件结构图中，我们可以清晰地看出糖尿病视网膜病变智能分类系统的每个功能模块，以及每个子功能模块之间的数据流动关系。可以看到，系统由图像获取、图像显示和图像分类三个部分组成，能够清楚地了解每两个子模块之间的关联。下面，为了更直观地看到每个子功能模块的功能及子功能内部的处理，我们给出了这三个功能模块的 IPO 图。

系统名称：基于眼底图的糖尿病视网膜病变智能分类系统	模块名称：图像获取	模块编号：1
输入：无		
输出：有效的图像文件		
处理步骤： 1. 系统给用户可选择文件路径的入口 2. 用户通过入口选择彩色图像文件 3. 判断文件是否属于图片格式的有效图像文件：如果是，将读取用户选择的图像文件；如果不是，将向用户返回提示信息		

## 4. 给用户正确的响应

图 3-7 “图像获取”模块的 IPO 图

IPO 图能够非常直观的根据输入、加工、输出这三个部分来说明这个功能模块的功能。从图 3-7 “图像获取”模块的 IPO 图可以看出，图像获取功能无输入，经过上述的处理步骤，输出了有效的图像文件给系统。

系统名称：基于眼底图的 糖尿病视网膜病变智能 分类系统	模块名称：图像显示	模块编号：2
输入：有效的图像文件		
输出：将图像展示在前端界面		
处理步骤： 1. 获取用户选择图像文件的路径 2. 从路径下读取到图像再将其展示在前端界面		

图 3-8 “图像显示”模块的 IPO 图

从图 3-8 “图像显示”模块的 IPO 图可以看出，图像显示的输入为从图像获取中得到的有效的彩色图像文件；其实在系统中已经保存下了选择的图像文件的路径，经过具体的方法实现，将图像展示在前端界面；输出为展示在前端界面的这样一个效果。

系统名称：基于眼底图的 糖尿病视网膜病变智能 分类系统	模块名称：图像分类	模块编号：3
输入：有效的图像文件		
输出：图像的病变的严重程度		
处理步骤： 1. 读取图像		

- |  |
|--|
| <ol style="list-style-type: none"><li>2. 对图像进行尺寸的调整</li><li>3. 将图像输入到训练好的网络中进行分类</li><li>4. 将分类结果展示在前端界面</li></ol> |
|--|

图 3-9 “图像分类”模块的 IPO 图

从图 3-9 “图像分类”模块的 IPO 图可以看出，图像分类的输入为从“图像读取”中获得的有效的图像文件；图像分类的处理步骤上图中描述；图像分类的输出为图像的病变的严重程度，也就是 DR 的五个级别中的一个。

## 3.2 系统详细设计

### 3.2.1 开发环境搭建

#### 1. 系统开发环境介绍

##### 1) 软件环境：

- g) 操作系统：Windows 10
- h) 开发平台：Anaconda3 4.2.0、PyCharm Community Edition 2018.1.2 x64
- i) 开发语言：Python3.6.1
- j) 运算平台：CUDA9.0
- k) 深度学习加速库：CuDNN7.0
- l) 深度学习网络框架：TensorFlow-gpu1.12.0、Keras2.2.4
- m) 界面开发库：Tkinter

##### 2) 硬件环境：

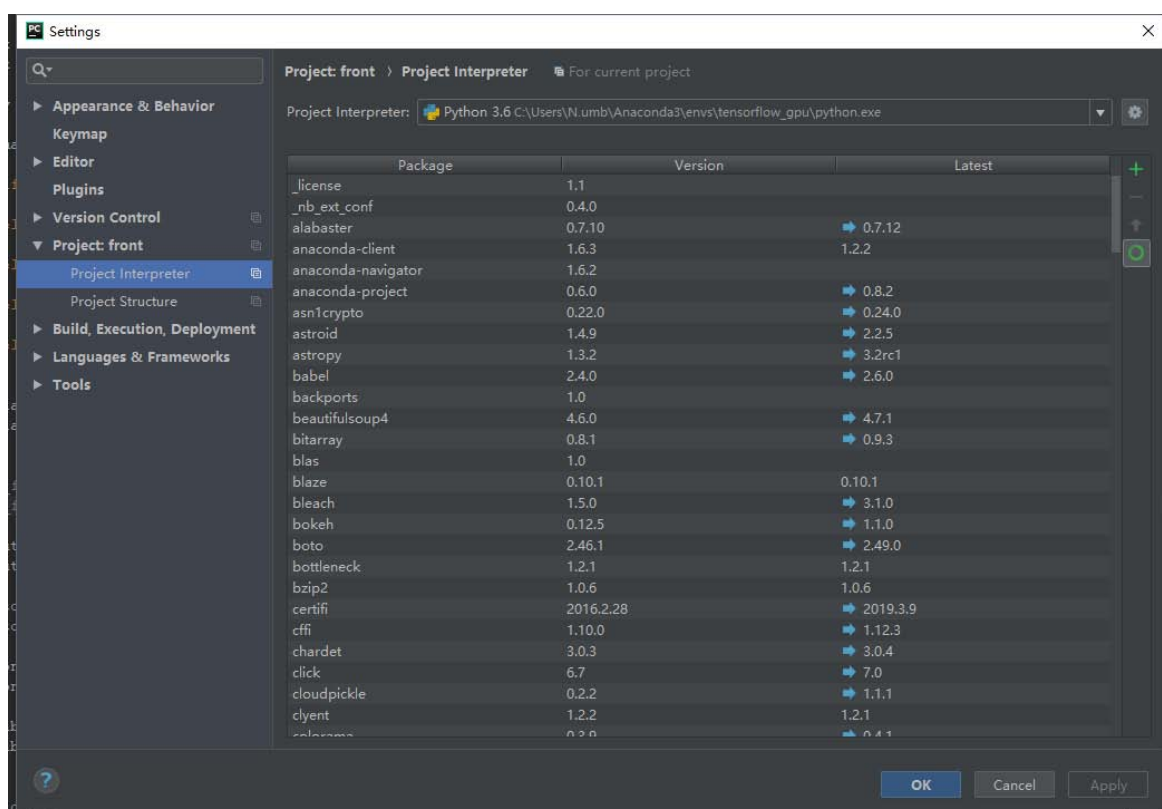
- d) CPU：Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
- e) GPU：NVIDIA GeForce GTX 1070
- f) 内存：16GB

#### 2. 系统开发环境搭建

这里的基本开发环境 TensorFlow 和 Keras 的运行环境在上一章中已经介绍，这里不再赘述。因为在开发系统时用到了 Python 中的 Tkinter 库，选择了 PyCharm 社区版作为开发环境。PyCharm 是一个集成开发环境(IDE, Integrated Development Environment)，主要用于 Python 语言开发，由捷克公司 JetBrains 开发，可以对代

码进行分析、对代码进行调试、可以与版本控制工具进行结合使用、可以在其中进行测试，集成功能十分强大，用于 Python 语言的开发也十分方便便捷。

这里主要介绍一下 PyCharm 和 Anaconda 中创建的虚拟环境的连接。在创建项目之后，只要在 PyCharm 工具栏中找到“File”，打开“Settings”中的“Project Interpreter”，进入界面后在系统解释器中选择在虚拟环境中的 Python 解释器，如图 3-10。



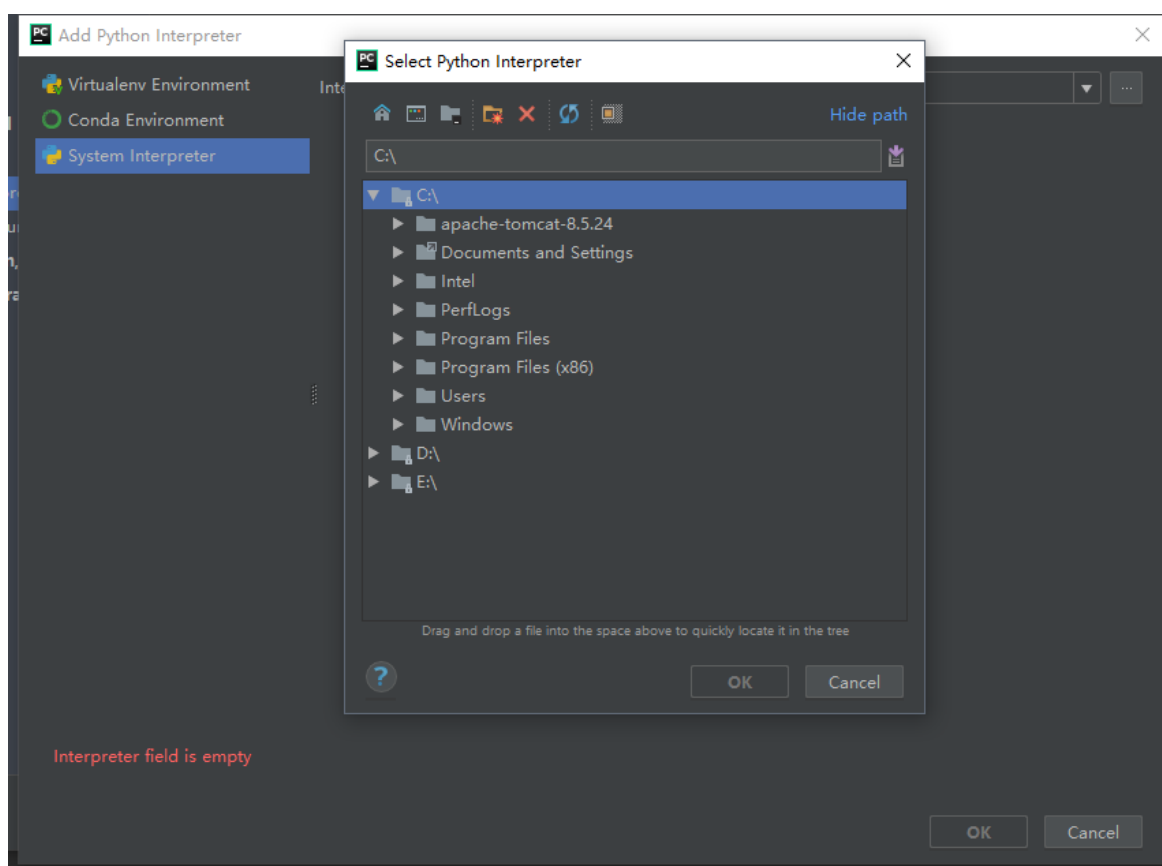


图 3-10 为 PyCharm 连接虚拟环境

### 3.2.2 系统详细设计

#### 1) 系统逻辑流程

本小节主要根据子功能模块的概要设计对系统进行详细的设计，以及所用到的各种算法的详细的介绍，以及使用的各种科学计算包的详细的说明。本系统的逻辑流程图如图 3-11 所示。



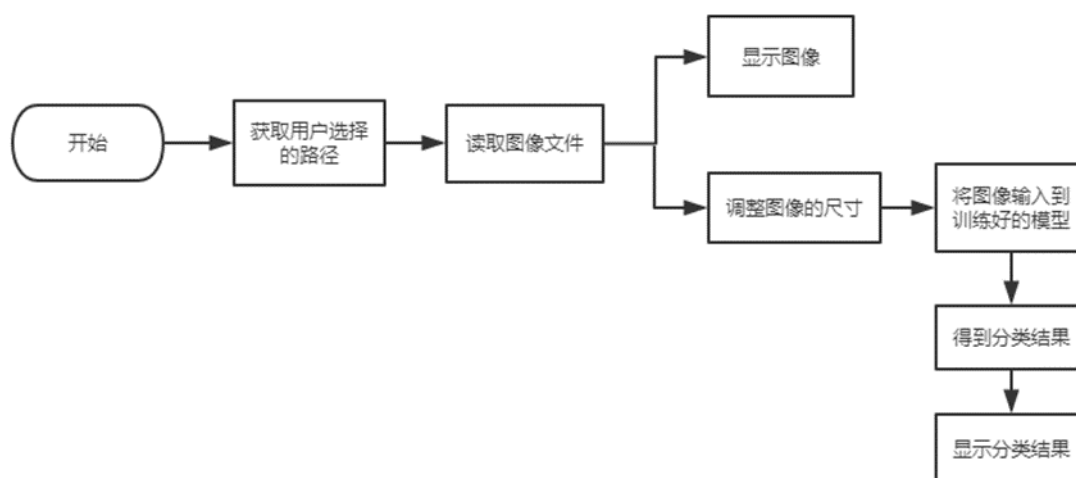


图 3-11 系统的总体逻辑流程图

本系统使用 Python 中自带的 Tkinter 库进行前端界面的设计，使用 Tkinter 对用户选择的图像进行显示，并且对图像的分类结果以文本的形式显示到前端界面。

图像文件的读取使用 Python 中的库实现；图像的分类使用第 3 章中训练好的模型进行分类。

本系统的实现较好地结合了 Python 中的深度学习预测方面的代码编写和实现前端的 Tkinter 的使用，使整个系统比较简易明了。

## 2) 界面设计

本系统使用 Python 自带的 Tkinter 模块来搭建实现主界面，整个系统的界面一共分为两个部分：一个是系统操作窗口，系统操作窗口中显示系统的功能操作，选择图像、显示图像和图像分类功能；另一个是显示图像子窗口，为用户使用显示图像功能时的弹出窗口。系统两个界面的构想图如图 3-12、图 3-13 所示。



图 3-12 系统主界面构想图

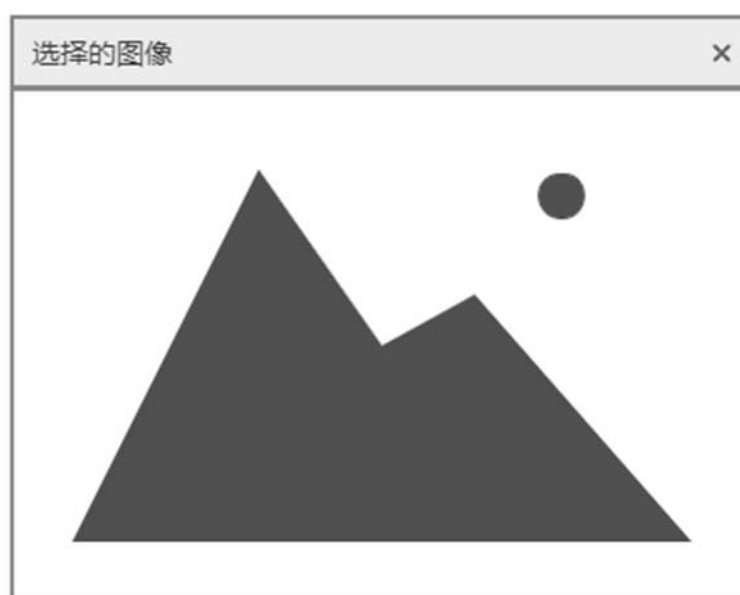


图 3-13 图像显示子窗口构想图

系统操作窗口的实现需要借助 Tkinter 模块，从图 3-12 可以看到，这个界面中需要一个主窗口、三个 Button 控件、一个文本框和一个 Label。主窗口的创建十分简单，使用 `tkinter.Tk()` 方法就可创建一个简易的窗口；对创建的窗口进行命名，使用 `title()`

方法即可，例如，`title('糖尿病视网膜病变智能分类系统')`；对创建的窗口的大小进行调整，使用 `minsize()` 方法，例如，`minsize(640, 480)`，第一个参数是指宽度为 640 像素，第二个参数是指高度为 480 像素。`Button` 控件的创建需要 `tkinter.Button()` 方法，例如

```
tkinter.Button(master, text="选择文件", command=choose_file)
```

界面设计中其他的按钮控件的实现类似，这里就不再赘述。

然后确定创建的 `Button` 控件的布局，这里使用 `grid()` 方法，使整个框架按照网格形式划分。文本框的创建需要使用 `tkinter.Entry()` 方法，例如

```
tkinter.Entry(master, width=60, textvariable=e)
```

即在 `master` 这个窗口上创建一个宽度为 60 像素的文本框，文本框中的内容变量存到 `e` 中。`Label` 的创建使用 `tkinter.Label()` 方法，例如

```
Tkinter.Label(master, text="您选择的文件路径为：")
```

文本框和 `Label` 的布局同样使用 `grid()` 方法。

图像显示子窗口的实现需要使用 `Toplevel()` 方法，然后对这个子窗口的标题和大小的设置和上面介绍的对系统操作窗口的操作相同。

### 3) 功能设计

本系统的功能为对彩色眼底图像进行获取、显示和智能分类。由于系统的功能单薄，所以对彩色眼底图像进行选择 and 显示利用 `Tkinter` 模块实现，而对彩色眼底图像的智能分类这一核心功能利用第 3 章中训练好的模型进行分类，并将分类结果显示。下面将详细介绍对彩色图像进行选择、显示、智能分类的设计和显示结果功能的设计。

1) 选择彩色图像：这里需要创建一个函数，命名为 `choose_file`，没有传入参数。在这个函数中，利用 `filedialog` 模块中的 `askopenfilename()` 方法，实现用户对图像文件的选择，然后将这个函数返回的文件绝对路径赋给一个字符串变量，便于寻找系统寻找图像并对它进行进一步的处理。

2) 显示彩色图像：这个功能是指将用户选择的彩色图像显示在一个新创建的子窗口中。此功能的实现需要创建一个函数，命名为 `show_image`，没有传入参数。在这个函数中，首先利用 `Toplevel()` 方法创建一个子窗口；然后利用 `PIL` 库中的 `Image` 模块下的 `open()` 方法，传入文件的绝对路径，将图片加载出来，利用 `ImageTk` 模块下的 `PhotoImage()` 方法将图片转化为一张 `Tkinter` 兼容的照片图像，可在 `Tkinter` 期望一个

图像对象的任何地方使用；最后，在子窗口中创建一个 `Label` 控件，将图像显示在 `Label` 中，在子窗口中进入事件循环。

3) 对彩色图像进行智能分类：这个功能的实现需要创建 `predict` 函数，同样不需要传入参数。首先，利用 `Keras` 下的 `models` 模块中的 `load_model()`方法加载出之前利用深度学习网络训练出的用于糖尿病视网膜病变智能分类的模型；然后利用 `cv2` 中的 `imread()`方法进行读取选择的图像，利用 `resize()`方法对图像的尺寸进行调整，利用 `NumPy` 中的 `array()`方法将其转化为数组；最后利用模型对处理后的图像信息进行预测，得到分类结果。

4) 对分类结果进行显示：这个功能需要创建一个 `Label`，将分类的结果以用户的角度展示在这个 `Label` 中，`Label` 的具体实现上面已经提到，这里不再赘述。

### 3.3 本章小结

本章主要从系统开发的角度对整个系统的开发流程进行了详细的描述，其中包括需求分析、概要设计、详细设计。需求分析部分使用了用例图简洁清晰地展示了系统与用户之间的交互关系以及系统需要实现的功能模块。概要设计部分以 `IPO` 图的形式直观简要地设计了系统中每个子功能模块的输入、输出和中间大致的处理步骤。详细设计部分详细说明了系统实现的搭建环境、系统的界面和功能设计，并且详细阐述了界面设计和功能设计中所需要的模块、接口、方法以及控件等。

## 第4章 系统运行与测试

这一章主要介绍系统的界面和对系统运行的测试，利用简单的示例来对系统的界面和功能进行详细的说明。

### 4.1 系统界面

本系统的界面部分主要由两个部分组成，主界面（见图 4-1）和图像可视化的子界面（见图 4-2）。从主界面中可以看到，系统界面由两部分构成：用户操作部分和信息显示部分。子界面是用户在点击显示图像按钮之后才会弹出。

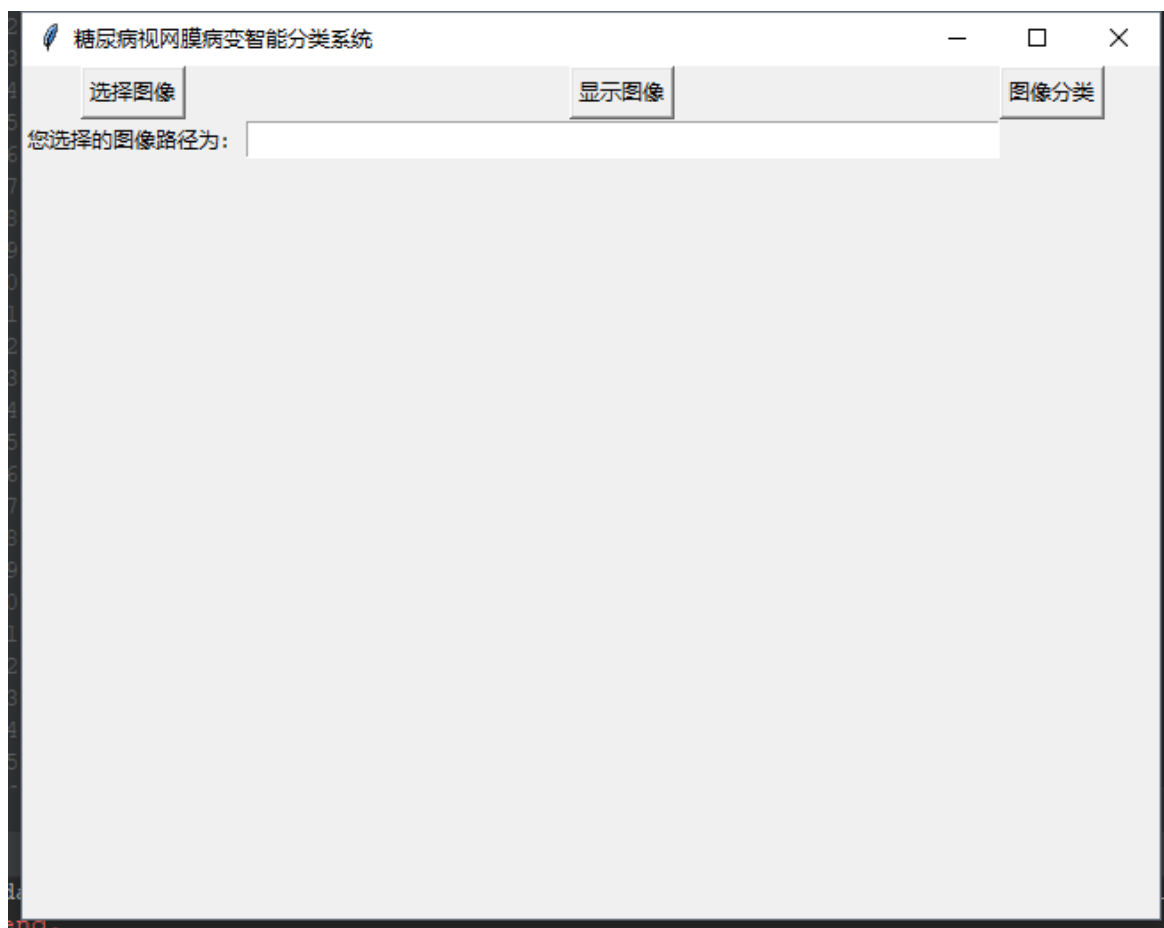


图 4-1 系统主界面

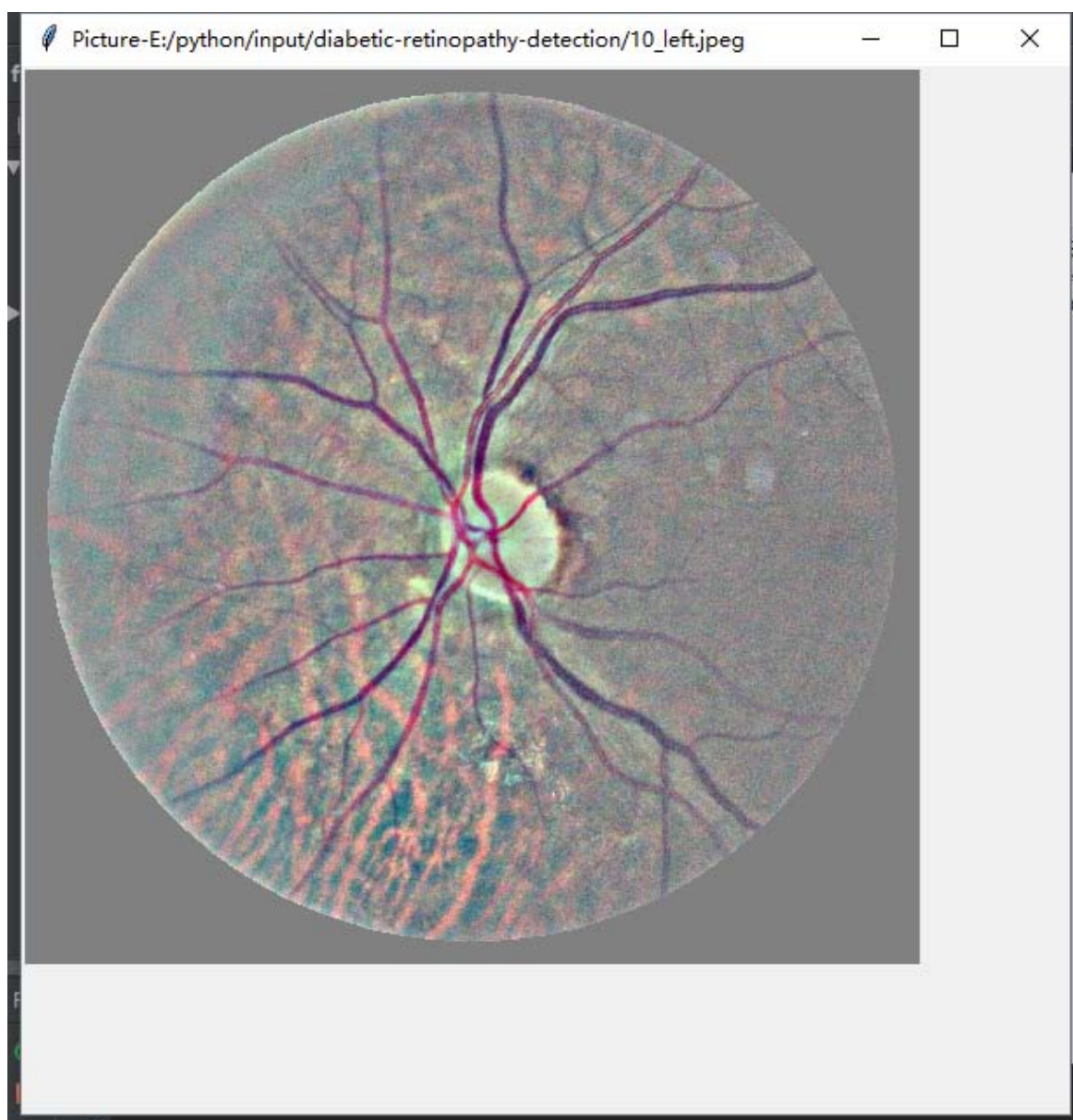


图 4-2 图像显示子界面

## 4.2 系统测试运行

从上一节的界面中，可以看到系统共实现了三个功能：选择图像、显示图像和图像分类并显示结果。下面，对这三个功能分别进行测试，查看功能是否正常实现。

### 4.2.1 “选择图像”功能

点击“选择图像”按钮会弹出“选择文件”窗口，可从本地磁盘下选择用户需

要的文件，见图 4-3。

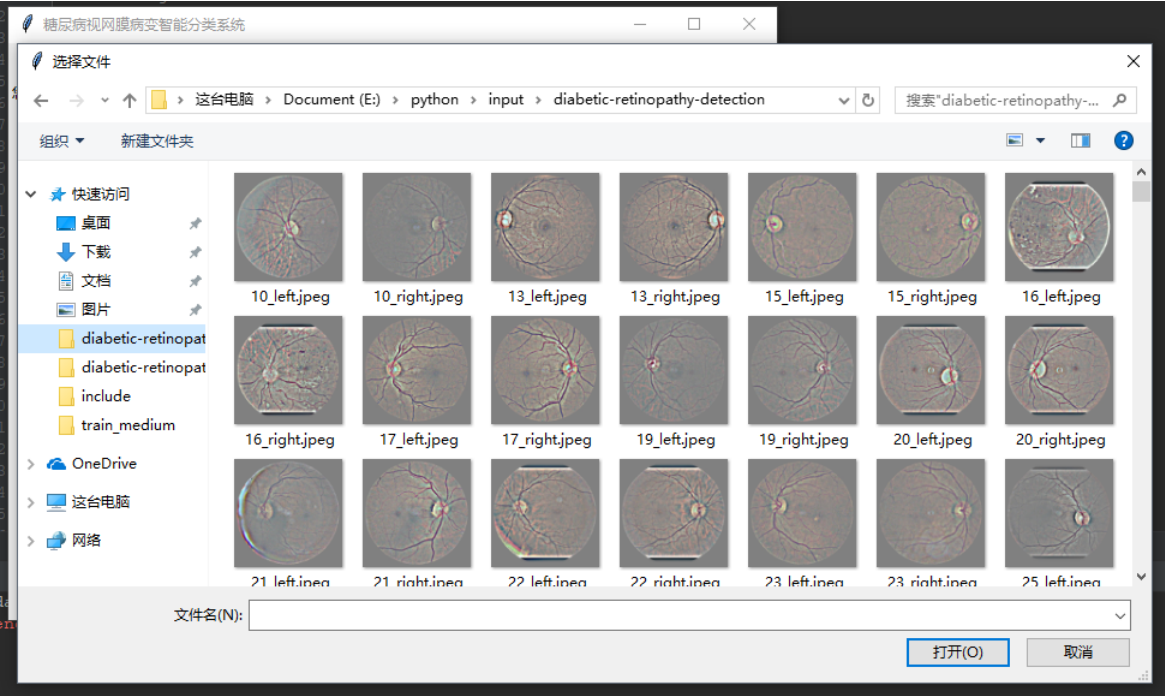


图 4-3 “选择文件”界面

在选择文件之后，可以看到系统主界面中显示了用户选择的文件路径，见图 4-4。

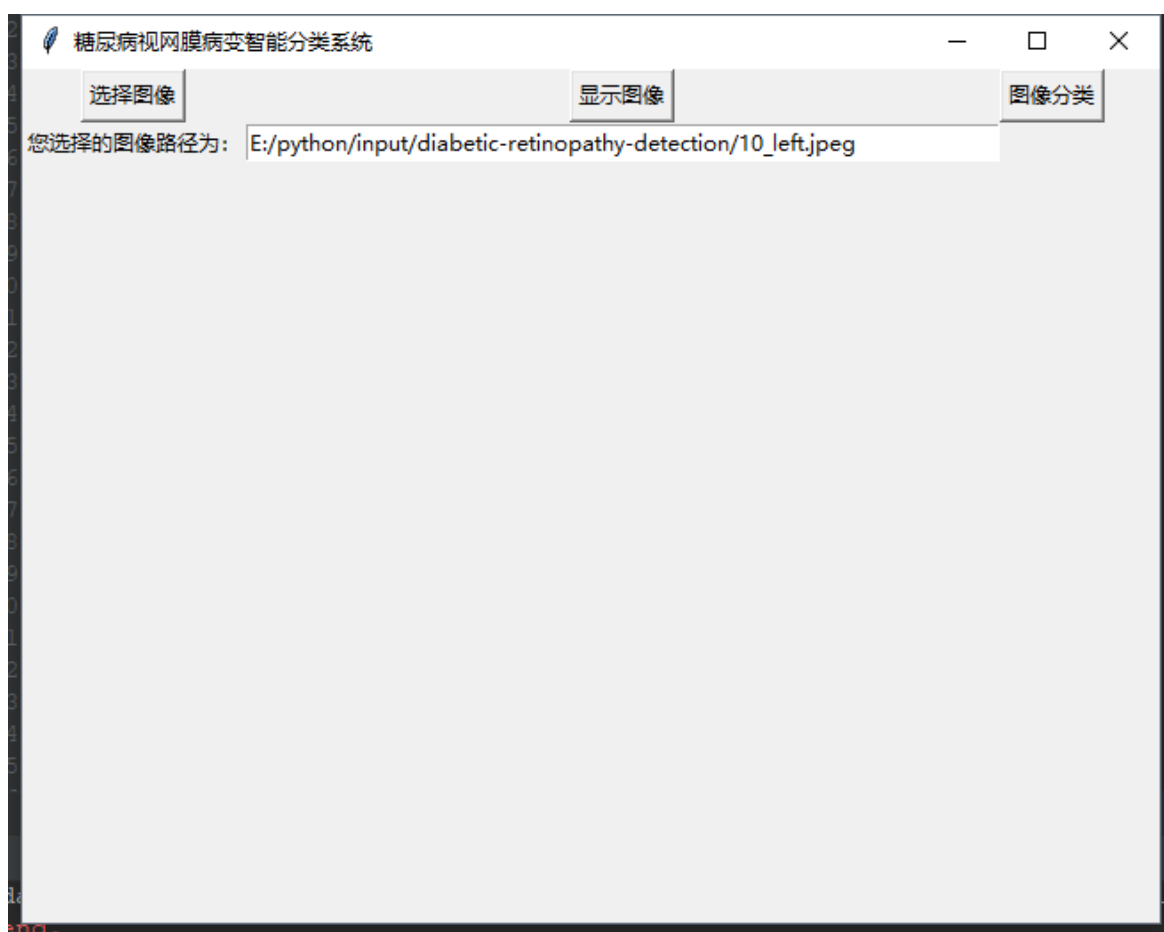


图 4-4 选择文件后的系统主界面

#### 4.2.2 “显示图像”功能

点击“显示图像”按钮后，会弹出显示图像的子界面，界面中显示了用户选择的图像，见图 4-5。



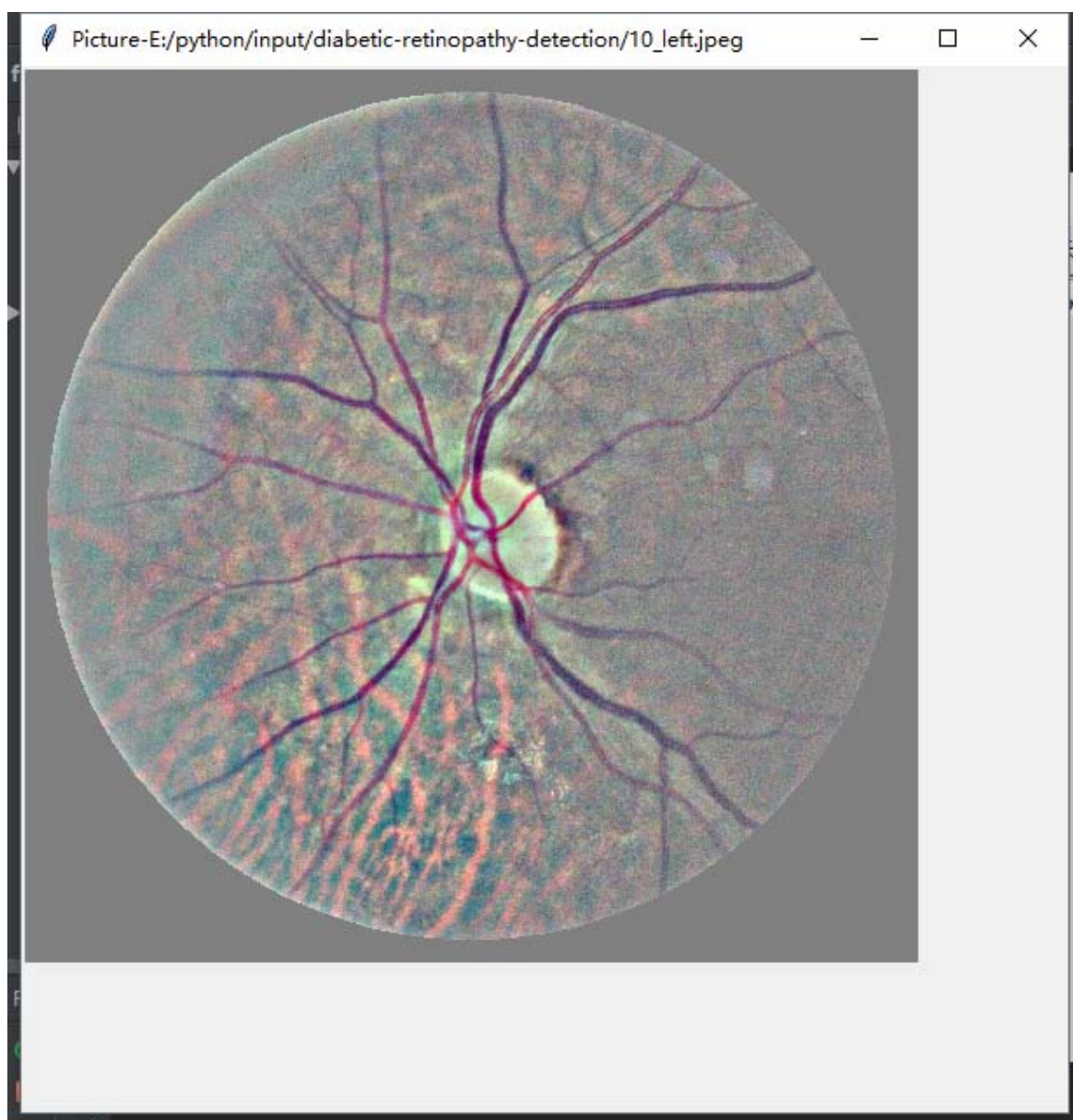


图 4-5 “显示图像”功能

#### 4.2.3 “图像分类”功能

点击“图像分类”按钮，可以发现在几秒钟后系统的主界面中显示了图像的糖尿病视网膜病变的严重程度，见图 4-6。

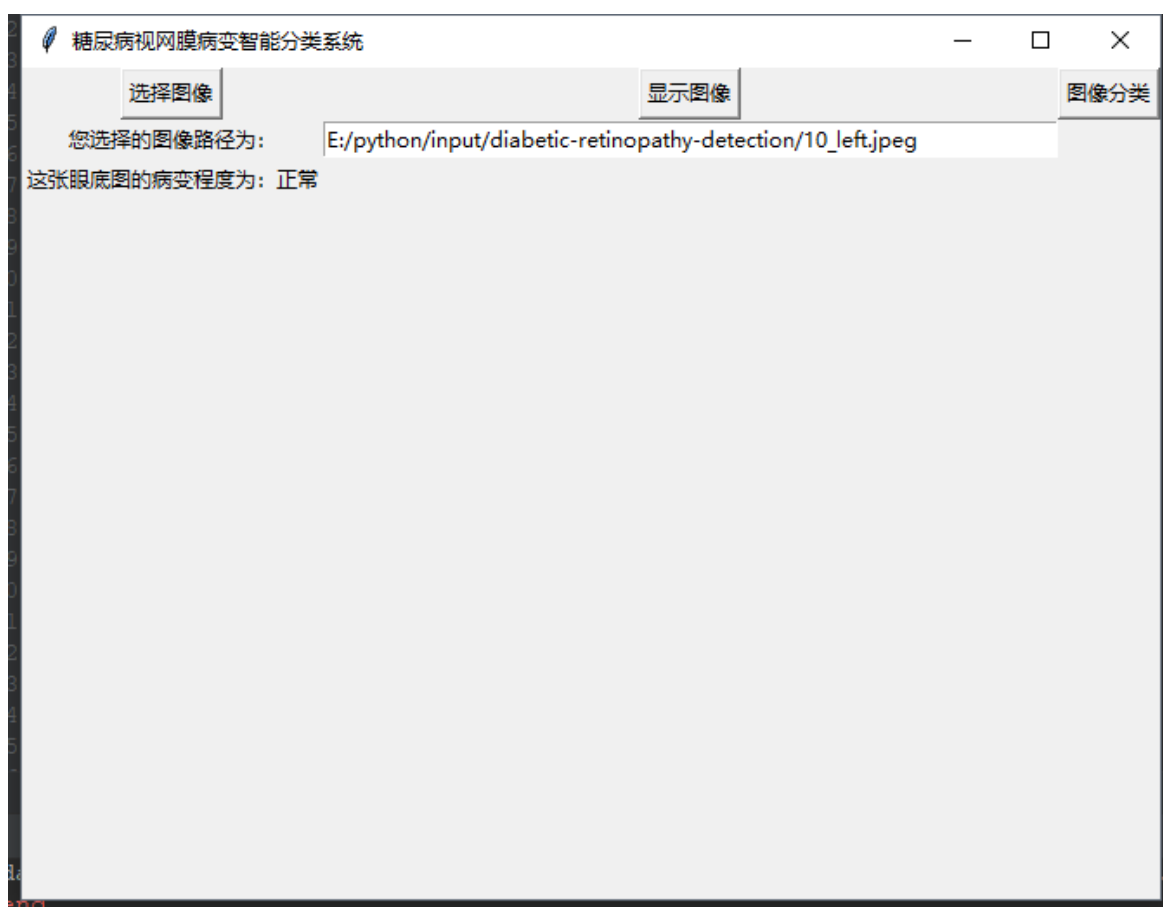


图 4-6 “图像分类”

### 4.3 本章小结

本章主要对系统的界面进行介绍，对系统的功能进行简单的测试，用详细的图文说明解释系统实现了要求的三个功能：选择图像、显示图像和图像分类。但是系统的图像分类部分需要的时间比较多，响应时间有些长，可能会稍微超出用户忍受的时间限制。

## 结 论

彩色眼底图像是医学领域诊断眼科疾病非常重要的来源，同样也是糖尿病视网膜病变的重要检测指标。基于人工智能技术的飞速发展，本糖尿病视网膜病变智能分类系统是在深度学习的基础上实现的，利用深度学习框架TensorFlow、Keras和深度学习网络Inception V3以及注意机制，对大量的数据进行训练，然后获取性能较好的模型。

这个糖尿病视网膜病变智能分类系统主要实现了眼底图像的获取、对眼底图像的显示、对眼底图像的智能分类功能，其中对眼底图像的智能分类功能部分主要采用了深度学习算法训练的模型。整个系统界面简洁，操作简单易懂，虽然实现的功能比较单薄，但是需求中提到的功能也已经实现。然而，由于在临床上对彩色眼底图像进行分类的要求极高的特点，这个系统其实还有很大的缺陷，主要是训练出的模型的分类精度比较低。

在整个系统实现的三个多月中，因为实现系统的需要，主要学习了Python语言的一些基础语法，研究深度学习所需的一些网络框架，学习Python中处理图像的一些函数，了解并学习NumPy、Matplotlib、Pandas等基础软件包中的常用函数，对卷积神经网络的发展有了进一步的了解，还学习了Python中创建前端界面的软件包Tkinter。在这个过程中，将学习和实践结合到一起，学习的知识用到实践中不只艰难，而且有一些成就感。

由于当下专业知识的缺乏、部分地区医疗资源的短缺，糖尿病视网膜病变智能分类系统有一定的实际意义，创建这种专家系统能够减少这些外在因素对糖尿病视网膜病变的病患的治疗的影响。但是，如果应用在临床疾病诊断中，这个系统就被要求必须有极高的准确率，目前的系统还未达到这样的要求。我认为主要的原因有：时间太短，开发人员对深度学习中对图像的分类方向还没有深入的了解。其中对网络的改进有很多方面，可以对图像的预处理可以做到更好、选择更好的深度学习网络、在训练过程中增加数据集、改进一些参数如卷积核的大小等。这个系统的产生有一定的积极意义，但是也需要更多的改进进行功能的提升。

## 参考文献

- [1] 严律南. 人工智能在医学领域应用的现状与展望[J]. 中国普外基础与临床杂志, 2018, 25(5): 513-514.
- [2] 李琼, 柏正尧, 刘莹芳. 糖尿病性视网膜图像的深度学习分类方法[J]. 中国图象图形学报, 2018, 23(10): 1594-1603.
- [3] 张德彪. 基于深度学习的糖尿病视网膜病变分类和病变检测方法的研究[D]. 哈尔滨: 哈尔滨工业大学, 2017.
- [4] Gardner G., Keating D., Williamson T., Elliott A. Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool[J]. British journal of Ophthalmology, 1996, 80(11): 940-944.
- [5] Roychowdhury S, Koozekanani D D, Parhi K K. Dream: Diabetic retinopathy analysis using machine learning[J]. IEEE journal of biomedical and health informatics, 2014, 18(5): 1717-1728.
- [6] Gulshan V, Peng L, Coram M, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs[J]. Jama, 2016, 316(22): 2402-2410.
- [7] Nayak J, Bhat P S, Acharya R, et al. Automated identification of diabetic retinopathy stages using digital fundus images[J]. Journal of medical systems, 2008, 32(2): 107-115.
- [8] Carson Lam D Y, Guo M, Lindsey T. Automated detection of diabetic retinopathy using deep learning[J]. AMIA Summits on Translational Science Proceedings, 2018, 2017: 147.
- [9] Pratt H, Coenen F, Broadbent D M, et al. Convolutional neural networks for diabetic retinopathy[J]. Procedia Computer Science, 2016, 90: 200-205.
- [10] Sayres R., Taly A., Rahimy E., Blumer K., Coz D., Hammel N., Krause J, Webster D.R.. Using a Deep Learning Algorithm and Integrated Gradients Explanation to Assist Grading for Diabetic Retinopathy[J]. Ophthalmology, 2019, 126(4):552-564.
- [11] 林蔚. 基于 K 近邻图区域合并的视网膜硬性渗出自动检测研究[D]. 上海: 上海

交通大学生物医学工程系, 2008.

- [12]高玮玮, 沈建新, 王玉亮, 等. 免散瞳眼底图像中微血管瘤的高效自动检测[J].  
中国生物医学工程学报, 2012, 31(6): 839-845

## 致 谢

时间飞逝，转眼间大学四年生活就要结束。写下毕业论文最后这一章的时候，才意识到时间是有多么快地离开，我们本身却不能感知到这种飞速。回想大学四年，自己在这段求学路上学到了很多，感谢在这段路上遇到相互陪伴的人、互相鼓励的人。

回想毕业设计这段时间，这段为大学时光画上句号的时光，对我来说，其中的记忆有辛苦还有更多的感谢。从选择毕业设计课题到最终完成毕业设计论文，我的指导老师宋红老师都悉心地给予了我很多帮助。选择完课题之后，老师就详细地为我介绍了课题的内容，告诉我要学习这些没有接触过的知识。开题之前，老师为我寻找课题相关的论文，然后还帮我修改开题报告。后来整个毕业设计的过程中，老师一直督促我并帮我一起想办法、解决问题。感谢宋红老师，如此我才能够顺利地完成毕业设计。在这个过程中，还要感谢实验室的直系学姐何晓颖，当我遇到问题时，她总是很有耐心地告诉我很多，还在我慌乱不安的时候安慰我；感谢实验室的学长王棋，明明自己还要学习，也不吝啬时间为我寻找和课题相关的论文。最后，还是要感谢一下自己，虽然没有很大的进步，但至少我没有放弃，希望自己在以后的硕士研究生阶段能够紧跟周围人的脚步，多学习，多充电。

大学四年，我遇到了很多优秀的同学，在他们身上我看到了很多优秀的专业素养，一定要善于动手、善于思考，感谢他们；感谢大学四年一直组队做各种项目的小伙伴们，谢谢大家的并肩作战和相互鼓励；感谢大学四年对我们悉心教导的老师们，从他们身上看到了对学生的负责、对工作的认真对待；最后，感谢父母，在大学期间的我时常颓废，是他们一直在鼓励我，告诉我要坚持，要努力，谢谢！