摘要

本项目为人工智能原理课程的大作业，最终效果为实现对于图片中的猫狗识别。项目使用Keras库和VGG16模型实现卷积神经网络，数据集基于Kaggle网站竞赛项目提供的数据集进行简化。

关键词：人工智能 图像分类 Keras VGG16 CNN

目录

[一、需求分析 3](#_Toc91967346)

[1.任务、目标介绍 3](#_Toc91967347)

[2.研究进展 3](#_Toc91967348)

[3.存在的问题、缺陷 3](#_Toc91967349)

[4.相关技术介绍 4](#_Toc91967350)

[卷积神经网络（Convolutional Neural Networks） 4](#_Toc91967351)

[VGGNet（Very Deep Convolutional Networks） 5](#_Toc91967352)

[TensorFlow 5](#_Toc91967353)

[二、功能设计 6](#_Toc91967354)

[1.设计思路分析 6](#_Toc91967355)

[2.方案使用技术原理分析 6](#_Toc91967356)

[Python 6](#_Toc91967357)

[Keras 6](#_Toc91967358)

[VGG16 7](#_Toc91967359)

[3.实现功能设计 7](#_Toc91967360)

[三、实验设计 8](#_Toc91967361)

[1.仿真数据集 8](#_Toc91967362)

[2.仿真流程设计 8](#_Toc91967363)

[3.代码实现情况 8](#_Toc91967364)

[模型建立与训练（train.py） 8](#_Toc91967365)

[模型测试（test.py） 9](#_Toc91967366)

[4.仿真结果展示 9](#_Toc91967367)

[四、方案分析评价 10](#_Toc91967368)

[1.项目总结 10](#_Toc91967369)

[2.心得体会 10](#_Toc91967370)

[3.参考文献 11](#_Toc91967371)

# 一、需求分析

## 1.任务、目标介绍

图像相比文字能够提供更加生动、容易理解及更具艺术感的信息，是人们转递与交换信息的重要来源。

图像分类是根据图像的语义信息将不同类别图像区分开来，是计算机视觉中重要的基本问题，也是图像检测、图像分割、物体跟踪、行为分析等其他高层视觉任务的基础。图像分类在很多领域有广泛应用，包括安防领域的人脸识别和智能视频分析等，交通领域的交通场景识别，互联网领域基于内容的图像检索和相册自动归类，医学领域的图像识别等。

一般来说，图像分类通过手工特征或特征学习方法对整个图像进行全部描述，然后使用分类器判别物体类别，因此如何提取图像的特征至关重要。在深度学习算法之前使用较多的是基于词袋(Bag of Words)模型的物体分类方法。词袋方法从自然语言处理中引入，即一句话可以用一个装了词的袋子表示其特征，袋子中的词为句子中的单词、短语或字。对于图像而言，词袋方法需要构建字典。最简单的词袋模型框架可以设计为底层特征抽取、特征编码、分类器设计三个过程。

猫狗识别是一类经典的图像二分类问题，通过模型的训练，实现目标的最终效果应当是能够分辨出任意给出的猫或狗的照片。

## 2.研究进展

基于深度学习的图像分类方法，可以通过有监督或无监督的方式学习层次化的特征描述，从而取代了手工设计或选择图像特征的工作。深度学习模型中的卷积神经网络(Convolution Neural Network, CNN)近年来在图像领域取得了惊人的成绩，CNN直接利用图像像素信息作为输入，最大程度上保留了输入图像的所有信息，通过卷积操作进行特征的提取和高层抽象，模型输出直接是图像识别的结果。这种基于"输入-输出"直接端到端的学习方法取得了非常好的效果，得到了广泛的应用。

## 3.存在的问题、缺陷

深度学习存在以下一些缺点：

1. 计算量大，便携性差：

深度学习需要大量的数据很大量的算力，所以成本很高。并且现在很多应用还不适合在移动设备上使用。目前已经有很多公司和团队在研发针对便携设备的芯片。这个问题未来会得到解决。

1. 硬件需求高：

深度学习对算力要求很高，普通的 CPU 已经无法满足深度学习的要求。主流的算力都是使用 GPU 和 TPU，所以对于硬件的要求很高，成本也很高。

1. 模型设计复杂：

深度学习的模型设计非常复杂，需要投入大量的人力物力和时间来开发新的算法和模型。大部分人只能使用现成的模型。

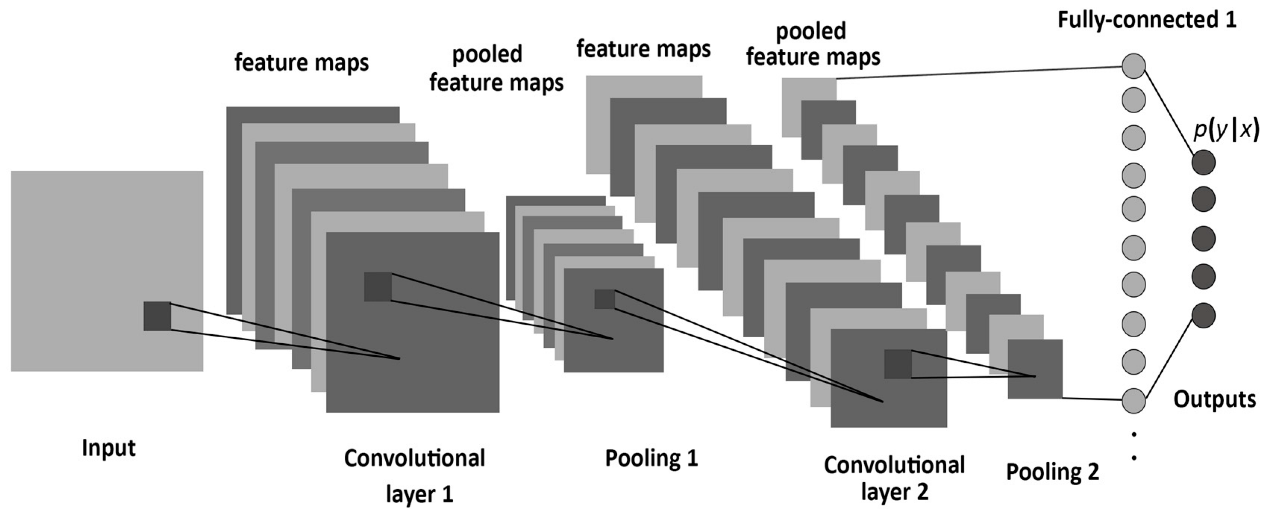
1. 没有"人性"，容易存在偏见：

由于深度学习依赖数据，并且可解释性不高。在训练数据不平衡的情况下会出现性别歧视、种族歧视等问题。

## 4.相关技术介绍

### 卷积神经网络（Convolutional Neural Networks）

这是一类包含卷积计算且具有深度结构的前馈神经网络，是深度学习的代表算法之一。



**CNN 的价值：**

1. 能够将大数据量的图片有效的降维成小数据量(并不影响结果)
2. 能够保留图片的特征，类似人类的视觉原理

**CNN 的基本原理：**

1. 卷积层 – 主要作用是保留图片的特征
2. 池化层 – 主要作用是把数据降维，可以有效的避免过拟合
3. 全连接层 – 根据不同任务输出我们想要的结果

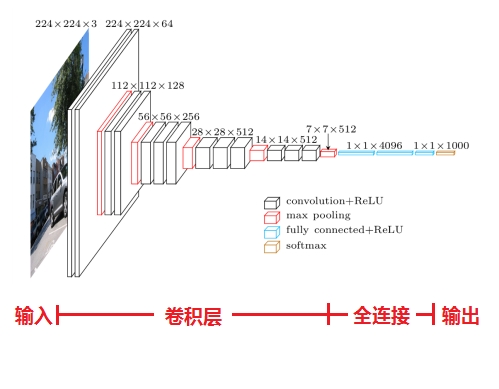
**CNN 的实际应用：**

1. 图片分类、检索
2. 目标定位检测
3. 目标分割
4. 人脸识别
5. 骨骼识别

### VGGNet（Very Deep Convolutional Networks）

牛津大学计算机视觉组（Visual Geometry Group）和 Google DeepMind 公司的研究员于2014年一起研发出的新的深度卷积神经网络，并取得了ILSVRC2014比赛分类项目的第二名和定位项目的第一名。

VGGNet探索了卷积神经网络的深度与其性能之间的关系，成功地构筑了16~19层深的卷积神经网络，证明了增加网络的深度能够在一定程度上影响网络最终的性能，使错误率大幅下降，同时拓展性又很强，迁移到其它图片数据上的泛化性也非常好。到目前为止，VGG仍然被用来提取图像特征。VGGNet可以看成是加深版本的AlexNet，都是由卷积层、全连接层两大部分构成。



### TensorFlow

TensorFlow是一个基于数据流编程的开源符号数学系统，被广泛应用于各种感知和语言理解任务的机器学习，其前身是谷歌的神经网络算法库DistBelief。Tensorflow拥有多层级结构，可部署于各类服务器、PC终端和网页并支持GPU和TPU高性能数值计算，被广泛应用于谷歌内部的产品开发和各领域的科学研究。

TensorFlow最初由谷歌大脑团队开发，用于Google的研究和生产，于2015年11月9日在Apache 2.0开源许可证下发布。

# 二、功能设计

## 1.设计思路分析

为实现猫狗二分类，首先应当找到一个适合的数据集，数据集应当有足够数量的猫和狗的图片。在已有数据集的基础上，考虑可以使用 CNN 算法进行模型的建立，这一过程中还有编程语言以及机器学习库的选择。在建立模型的过程中，还应当对于 accuracy 和 loss 等参数有着充分的评估和考量。

## 2.方案使用技术原理分析

本项目使用 Python 语言构建，借助基于 TensorFlow 的 Keras 深度学习库，并且将 VGG16 作为特征提取的模型。

### Python

Python 是一种广泛使用的解释型、高级和通用的编程语言。Python 支持多种编程范型，包括函数式、指令式、反射式、结构化和面向对象编程。它拥有动态类型系统和垃圾回收功能，能够自动管理内存使用，并且其本身拥有一个巨大而广泛的标准库。

在本项目中，主要使用了 os、matplotlib.pyplot、cv2 等库。os 库用于数据集内文件的读取等操作，matplotlib.pyplot 库主要用于将训练过程中的一些参数绘制成图像便于展示，cv2 库用于测试时对图片进行处理。

### Keras

Keras是一个用Python编写的开源神经网络库，旨在快速实现深度神经网络，专注于用户友好、模块化和可扩展性，是ONEIROS（开放式神经电子智能机器人操作系统）项目研究工作的部分产物。Keras包含许多常用神经网络构建块的实现，例如层、目标、激活函数、优化器和一系列工具，可以更轻松地处理图像和文本数据。除标准神经网络外，Keras还支持卷积神经网络和循环神经网络。其他常见的实用公共层支持有Dropout、批量归一化和池化层等。

TensorFlow 环境中已经内置了 Keras，所以可以较为便捷地使用，从而方便地构建训练模型。

### VGG16

本项目使用的是 VGG16，其网络结构含有参数的网络层一共有16层，即13个卷积层，5个池化层，3个全连接层，不包括激活层，输入图像的默认大小是224\*224\*3 。

vgg16网络结构可以划分为6个模块层次加1个输入模块，分别如下：

| **模块** | **各模块的涉及的层次** |
| --- | --- |
| 输入模块 | 224\*224\*3 |
| 第一个模块 | conv3-64 |
| conv3-64 |
| maxpool |
| 第二个模块 | conv3-128 |
| conv3-128 |
| maxpool |
| 第三个模块 | conv3-256 |
| conv3-256 |
| conv3-256 |
| maxpool |
| 第四个模块 | conv3-512 |
| conv3-512 |
| conv3-512 |
| maxpool |
| 第五个模块 | conv-512 |
| conv3-512 |
| conv3-512 |
| maxpool |
| 第六个模块（全连接层和输出层） | FC-4096 （实际上前面需要加一个Flatten层） |
| FC-4096 |
| FC-1000 (负责分类) |
| softmax(输出层函数) |

## 3.实现功能设计

经过训练之后，可以实现较为快速且准确地对于给出的任意的猫狗图片进行分辨这一功能，并且对于训练完成后的权值文件，可以进行迁移训练，便于进行下一步研究。所以在模型的训练之后应当注意模型的保存，在测试时也能够快速地读取。在测试功能时，应当对于模型进行展示，同时展示测试用的图片以及输出结果，便于比对与评价。

# 三、实验设计

## 1.仿真数据集

在 Kaggle 上有个 [Dogs vs. Cats Redux: Kernels Edition](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition) 的竞赛，使用比赛提供的 train.zip 和 test.zip 两个[压缩包](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data)中提供的图片，可以进行模型的训练和测试。train.zip 解压后可以看到是25000张 jpg 图片，其中猫和狗的图片各占一半，按照 cat/dog.序号.jpg 的格式命名。test.zip 解压后可以看到是乱序混排的12500张猫狗照片，按照 序号.jpg 格式命名。数据集比较大训练时间较长，所以在配置 data 文件夹时，选取2000张猫和2000张狗作为训练集，1000张猫和1000张狗作为验证集，500张图片作为测试集。

## 2.仿真流程设计

项目的最终目的是实现输入一张图片，输出图中是猫或狗。那么模型的训练结果就是得到一个对于猫狗的分类器，分解下来为：建立模型，读入数据集，进行模型训练，输出结果并保存模型这样的过程。

## 3.代码实现情况

### 模型建立与训练（train.py）

首先将 VGG16 卷积基实例化，项目中使用的是 vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 文件，不包含dense层，设置 weights="imagenet", include\_top=False，接着使用 models.Sequential() 初始化，再加上全连接层的相关代码，模型就建立完成了。

在指定路径下使用 os.path.join() 读取图片，接着使用 ImageDataGenerator() 和 flow\_from\_directory() 创建训练集和验证集的 batch，同时在每一个批次中对这batch\_size个样本数据进行增强，适时地进行了数据增强处理以增强模型的泛化能力。接着通过 conv\_base.trainable 和 set\_trainable 两个参数的设置，利用冻结的卷积基端到端地训练模型。这里把 set\_trainable 看成是一个开关，最开始 set\_trainable=False ，所以下边的if语句中运行 layer.trainable = False 这个语句，使每个层都不能训练，当它发现 block5\_conv1 层的时候 set\_trainable = True，之后的每个层都运行 layer.trainable = True，也就是都可以训练，最后的结果就是 block5\_conv1 层之前的所有层都被冻结，之后的所有层都可训练。

接着配置模型并拟合，使用 optimizers.RMSprop() 设置学习率等参数，fit\_generator() 设置训练及验证的 epoch 和 step，save() 保存训练的模型，默认将训练后的模型命名为 cats\_and\_dogs.h5 置于文件夹 model 中。

最后，利用 matplotlib.pyplot 库内的函数进行绘图，对于训练过程中的 accuracy和 loss 进行可视化展示。

### 模型测试（test.py）

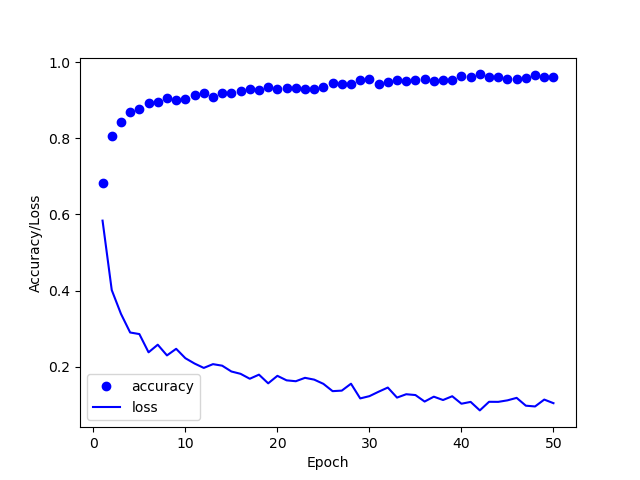
首先使用 load\_model() 函数载入模型，使用 summary() 函数展示模型中的参数。

使用 cv2 库中的函数读取选定的测试图片并进行图片处理，这里使用 resize() 更改图片尺寸，使用 reshape() 转换成模型输入的维度。借助 model.predict() 函数进行预测，其返回值是数值，表示样本属于每一个类别的概率，在多分类中可以使用 numpy.argmax() 方法找到样本以最大概率所属的类别作为样本的预测标签，二分类只需要将得到的结果与0.5进行比较，并将结果转换为 int32 类型，从而得出是否为0（猫）或1（狗）。

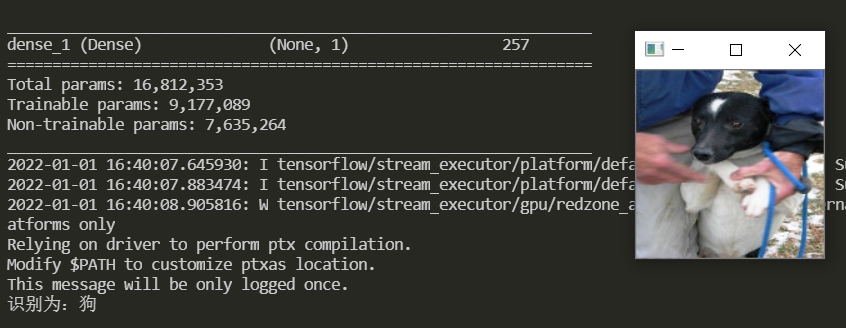
最后将结果输出，同时使用 imshow() 将图片展示，便于比对结果的正确与否。

## 4.仿真结果展示

进行 epoch = 50 的训练，accuracy 和 loss 效果如下：



训练后的模型进行预测的结果如下：



# 四、方案分析评价

## 1.项目总结

经过这次人工智能大作业，我独立完成了一个猫狗分类程序。在这一过程中，我将在人工智能原理课程上的所学，借助实用的编程工具和开源的数据集和数据模型，成功转化成了一个可以真正使用的完整的程序。基于对于 CNN 算法的了解，并且我获取到 VGG16 模型，便可以创建一个复杂又高效的卷积神经模型。在项目构建的过程中，其次重要的便是学会使用 Keras 库中的 API，同时对于像 TensorFlow 等工具实现深度学习的原理进行一定程度的了解，这样才成功地使用了必要的函数实现了最终的功能。在初步完成之后，还应当对于结果进行评估和优化，应当对于模型建立过程中的一些参数进行调整以达到最高效。

## 2.心得体会

在项目前期准备中，我意识到了前人为我们铺平道路是多么伟大，只要稍稍了解一下 CNN 算法，并且利用 VGG 模型，就可以在短短几行代码中创建一个卷积神经模型，我对于程序能够像人一样识别图片的内容而感到惊奇。

对于参数选择可以遵循这样的原则:

1. 开发一个过拟合的模型：(1)添加更多的层数；(2)让每一层变得更大；(3)训练尽量多的次数
2. 调参实现抑制过拟合：(1)增加训练数据；(2)dropout；(3)正则化；(4)图像增强
3. 继续调参直到再次过拟合

一开始，我使用 TensorFlow 内置的方法实现卷积神经模型的建立（例如使用 tf.variable\_scope()，tf.nn.conv2d() 等函数），这样的方法比较麻烦，而且最后自己设置的模型训练后得到的效果并不如预期，所以最后改用了 Keras 库和运用了 VGG16 等模型，得到了较好的效果。所以，在构建模型的过程中，也应当学会借助前人和他人的成功经验，借助封装的库和函数和经典的模型更方便有效地实现自己的需求。

## 3.参考文献

1. TensorFlow - 维基百科，自由的百科全书<https://zh.wikipedia.org/wiki/TensorFlow>
2. Keras - 维基百科，自由的百科全书<https://zh.wikipedia.org/wiki/Keras>
3. 应用 Applications - Keras 中文文档<https://keras.io/zh/applications/#vgg16>
4. 函数式 API - Keras 中文文档<https://keras.io/zh/models/model/>
5. Python 标准库 — Python 3.10.1 文档<https://docs.python.org/zh-cn/3/library/index.html>
6. tensorflow卷积神经网络Model模板\_天上掉下两毛钱-CSDN博客\_卷积神经网络模板<https://blog.csdn.net/qq_39622065/article/details/81149080>
7. 用keras框训练猫狗分类并进行验证\_Candy\_Shang的博客-CSDN博客<https://blog.csdn.net/weixin_41711422/article/details/105460138>
8. 一文看懂深度学习（白话解释+8个优缺点+4个典型算法） - 掘金<https://juejin.cn/post/6844903896700157960>
9. 大话CNN经典模型：VGGNet - 雪饼的个人空间 - OSCHINA - 中文开源技术交流社区<https://my.oschina.net/u/876354/blog/1634322>
10. 经典卷积神经网络---VGG16详解*笼中鸟-CSDN博客*vgg16<https://blog.csdn.net/hgnuxc_1993/article/details/115956774>
11. keras的图像预处理全攻略（二）—— ImageDataGenerator 类\_MIss-Y的博客-CSDN博客\_imagedatagenerator<https://blog.csdn.net/qq_27825451/article/details/90056896>
12. Dropout实现过拟合抑制 – 星空笔记<https://wolove.tech/240.html>
13. tensorflow简介以及与Keras的关系、常用机器学习框架一览*德prince-CSDN博客*tensorflow和keras<https://blog.csdn.net/li528405176/article/details/83857286>
14. Release VGG16, VGG19, and ResNet50 · fchollet/deep-learning-models<https://github.com/fchollet/deep-learning-models/releases/tag/v0.1>
15. 关于VGG模型文件的存放位置\_huatianxue的博客-CSDN博客<https://blog.csdn.net/huatianxue/article/details/106757252>
16. 如何使用Keras fit和fit\_generator（动手教程）*W\_Tortoise的博客-CSDN博客*fit\_generator<https://blog.csdn.net/learning_tortosie/article/details/85243310>
17. gzshan/dogsVScats: 图像二分类问题 猫狗大战 pytorch CNN<https://github.com/gzshan/dogsVScats>
18. 0809zheng/dogs-VS-cats-pytorch: Pytorch实现Kaggle竞赛“猫狗分类”，准确率超过99%。<https://github.com/0809zheng/dogs-VS-cats-pytorch>