

# 用激活地图集探索神经网络的算法研究

周慧慧, 郭 仲, 马汉杰\*

浙江理工大学计算机科学与技术学院, 浙江 杭州

收稿日期: 2022年10月10日; 录用日期: 2022年11月24日; 发布日期: 2022年12月7日

## 摘 要

当今信息时代, 人工智能的应用几乎渗透到了各种领域, 如航空航天图像处理技术的地质探测, 军事目标的侦测制导, 医学上的临床病理诊断, 如今它已经真正地融入到我们的日常生活当中, 虽然机器视觉系统得到了广泛应用, 但是要理解它到底是如何“分辨”这些事物, 将水果A归类为苹果而把水果B归类为梨, 仍然不为我们所知。本文主要以“激活地图集”为核心, 首先从网络上收集带有标签的图片形成数据集和使用已有的数据集, 利用tensorflow自行搭建一个卷积分类神经网络对两类数据集进行训练, 最终从训练模型中得到一个较为准确的分类结果。其次就是地图集的构建, 将输入图像运行到神经网络中的某一层, 然后经过特征提取能够还原图像, 经过图像整合并形成一张地图集。总的来说, 它不仅能够从这些图像中识别出其形状、颜色, 还能够把这些特征结合起来形成特定的场景或者物体。

## 关键词

神经网络, 激活地图集, 图像整合, 特征提取, Tensorflow

# Algorithmic Research on Exploring Neural Networks with Activation Atlases

Huihui Zhou, Zhong Guo, Hanjie Ma\*

School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: Oct. 10<sup>th</sup>, 2022; accepted: Nov. 24<sup>th</sup>, 2022; published: Dec. 7<sup>th</sup>, 2022

## Abstract

In this current information age, machine learning technology application has begun to gradually enter our lives, and the image processing technology is also developing rapidly. The machine vision system has been widely used, such as geological detection of aerospace image processing

\*通讯作者。

technology, detection and guidance of military targets, and clinical pathological diagnosis in medicine, it has been truly integrated into our daily life. However, when we try to understand how it “recognizes” these things, like categorizing fruit A as an apple and fruit B as a pear, it’s still unknown. This article mainly focuses on “activating atlases”. Firstly, it collects tagged images from the network to form a data set or an existing data set. Then, it uses tensorflow technology to build a convolutional neural network to train the two types of image data sets to get a more accurate classification result from the training model. The second is the construction of the atlas, which runs the input image to a certain layer in the neural network, and then through feature extraction can restore the image and form an atlas. In general, it can not only recognize its shape and color from these images, but also combine these features to form a specific scene or object.

## Keywords

Neural Network, Activation Atlas, Image Integration, Feature Extraction, Tensorflow

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 课题背景意义

### 1.1. 研究性质和意义

对于人类来说，可以通过特定的视觉系统和大脑分析来对图像进行解释，通过人眼得到个体的整个图像并且由大脑多个特定区域的神经元“观测”到它的局部结构并且分析得出该个体的类别，但是，如何设计出一套这样的模型结构运用于机器上更好地实现对于自然图像或者个体的识别，依然是目前机器学习比较火热的课题研究[1]。

在生活中，还存在着许多没有被标记的图像，而这些图像在现有的技术下进行搜索和处理还是比较困难，用户也没有办法更快地得到自己需要的图像，但是传统的图像识别算法效率对于现在的需求满意程度来讲不尽人意，一张图像的识别往往需要在上采取多个特征值，但是如何采样特征值，设计隐藏层分类模型和使用什么样的激活函数是当前面临的重要且复杂的问题。利用激活地图集技术，在多种环境下提出一套对自然图像的识别的算法研究是我们对深度网络学习的另一大拓展。

本次研究主要分为两个部分，第一个部分是分类卷积神经网络的构建，通过 tensorflow 搭建神经网络并对该网络进行图片训练，最终得到分类模型并利用模型进行分类；第二个部分则是地图集的构建，通过特征提取等方式，最终得到一张由特征图形成的图集。

### 1.2. 深度神经网络处理图像研究状况

深度神经网络是机器学习中的重要技术，对于深度学习的学习对于现代机器学习和人工智能发展起到了重要的作用。通过让机器处理大量图像数据得到特征，人为地构造出多层次的神经网络层，对每一层进行进行训练，从而建立起一个多层神经网络，对输入自然图像进行处理分析以及判断其所属类别[2]。

#### 1.2.1. 神经网络发展历程

神经网络是由多个神经元组合而成的用于数据处理的一类数学模型，由 1958 年，Warren McCulloch 和 Walter Pitts 提出了一种叫做 MP 的神经元模型[3]，由于神经元非常简单，只能处理一些线性函数的操

作，而其他相对于比较复杂的函数却没有任何办法，比如，异或操作没有办法在该神经元模型上运行，所以也可称为“单层感知机”，之后，由 Rosenblatt 提出了一种简单的神经网络，其仅仅只有两层神经元，用于简单的图像学习，并命名为“感知器”[4]。上世纪八十年代，“多层感知机”的概念首次被提了出来，Hinton、Rumelhart 等人提出该多层感知机内包含多个隐藏层[5]，输入数据通过一层的神经元响应，最终利用激活函数将所有数据进行拟合，即人类可以看到并理解的一类函数。1986年，Rumelhart 等提出人工神经网络的反向传播算法(Back propagation, BP) [6]，掀起了神经网络在机器学习中的研究热潮。一个典型的神经网络结构通常包含三个部分，输入层，隐藏层和输出层，输入的特征向量通过隐藏层的筛选分类，在输出层得到分类结果。在训练算法上使用 BP (back propagation)神经网络为每个神经网络层进行训练[7]。多个隐藏层为解决早期离散函数束缚的问题开辟了一条新的出路。

### 1.2.2. 梯度消失及问题解决

梯度消失问题是分类神经网络中经常遇到的一种问题，一个分类比较准确的神经网络层数会尽可能的深，只要神经元和隐藏层足够的多，该神经网络就可以任意拟合出适应输入数据的函数[8]，在神经网络使用反向传播算法(BP 算法)时，即使用 sigmoid 激活函数来对所有数据进行一个拟合的过程，而 sigmoid 函数存在一个最大导数为 0.25，即下层网络所能接收到上层网络传递过来的训练信号仅仅只有上层网络训练信号的 0.25，随着网络层逐渐下降，后面网络层接收的训练信号越来越差，其最终的结果也相比较准确率更低。从而在数据分析分类上与原先输入的特征相比，出现极大的偏差值。若要解决这种梯度消失问题，可有以下四种方法：

1) 采用其他激活函数，如 Maxout、ReLU、Leaky ReLU 函数等。即  $f(x) = \max(0, x)$ ，这就使得正数一侧永远为 1，负数一侧永远为 0，相比较于 sigmoid 激活函数，它的收敛速度更快，更不饱和，向下一神经网络层传递的特征向量误差更小[9]，但是由于其学习速率更快，在大量数据需要学习的时候，ReLU 的压缩能力比不上 sigmoid 激活函数，不保证数据幅度上的问题。

2) 权值初始化。即 xavier 初始化方法，这种方法将参数(主要是 weight 和 biases)以均匀分布的形式在  $[-\sqrt{6/(m+n)}, \sqrt{6/(m+n)}]$  区间中进行初始化，区间公式中  $m$  是当前隐藏层的输入维度， $n$  表示当前隐藏层的输出维度，其最终的目的也是通过快速收敛的方式来减少向下传参的偏差。

3) 调整网络结构。即可改变神经网络结构来实现，比如 Highway Network，其主要解决的问题当梯度传递回较浅层时，该参数对较浅层的权值分配影响很小，Highway Network 通过增加一个门函数，控制神经网络内部信息流动，设置其函数  $T(x) = 0.3$ ，即当特征向量向前传播时，有百分之三十被激活，其它直接进入下一层。或者构建残差网络结构(ResNet)，其具体会在下一节中详细介绍。

### 1.2.3. 三类神经网络模型

随着神经网络技术的不断提高，神经网络模型也逐渐增多，下面将介绍三类常用搭建的神经网络。

1) 全连接神经网络(FNN) 2) 卷积神经网络(CNN) 3) 循环神经网络(RNN) [10]。其中全连接神经网络全部使用 ReLU 或者 Maxout 等激活函数来替代 sigmoid 函数，该网络比较特殊的就是能够从一幅图像或者学习数据中的各个方面提取多个特征向量，为预测和识别个体提供了更多的有效参数参考[11]，但同时这也容易导致参数数量膨胀，每一层的训练量难度会增加很多。卷积神经网络(CNN)在图像处理领域内发挥着重要的作用，它也是神经网络的典型结构之一，三个基本组成部分，前两个分别为输入层，隐藏层。该隐藏层中又包括着许多卷积层和池化层，主要用于特征向量分析和特征提取[12]，第三个部分是全连接层，主要是压缩前面隐藏层提取的特征向量和维度。CNN 模型限制了参数个数，并且从局部结构中挖掘特征向量，相比较于 FNN 网络模型有效地减少了数据量。循环神经网络(RNN)是在对样本表现在时间顺序上的处理，比如在自然语言处理或者语音处理方面具有重大的作用[13]。CNN 的作用原理是在某一层

输入的结果在下一个时间段会重新作用于自身，用于下一时间段的输入参数，即下一时间段这两层都会以该输出结果作为参数进行输入[14]。

#### 1.2.4. 深度卷积神经网络发展

卷积神经网络是典型的神经网络结构之一，它是一种在图像处理上运用比较广泛的神经网络，能够自动识别二维图像并且提取图像各个方面的特征。发展至今，卷积神经网络的可识别性多层次性以及可训练性还是作为最典型的体系结构之一[15]。它典型的几种结构表现如下：

1) AlexNet 是最早出现的神经网络之一[16]，相比较简单的卷积神经网络，它的神经网络层数更多，每一层的参数也更多，并且采用的新的激活函数 ReLu。该网络共有 5 个卷积层，3 个全连接层用于输出类别。卷积层中分为两个团体进行单独计算，这有利于减少计算量和 GPU 并行化。同时，该神经网络中还加入了 dropout 层，使得网络层中的某些神经元停止活动，以达到减轻拟合的作用[17]。

2) ZFNet 该网络在 AlexNet 网络上进一步改进，通过建立一个反卷积网络，将卷积网络层的一些活动进行可视化，例如将分析卷积网络的效果以及训练过程中特征如何一步步被提取演化，从而将每一层的作用给展现出来[18]。

3) GoogleNet 网络由 Google 在 2014 年提出，同时该网络也是“激活地图集”这种技术所使用的神经网络模型(Inception-V1)。该网络在前者的基础之上，增大了网络的规模和深度，但是这样的改动会使得函数过度拟合，同时还会使得训练量增加[19]。Google 提出的 Inception 网络对图像数据会进行多方面降维处理，即把不同大小尺寸的卷积核以及池化层整合成一个 Inception 模块，减少了网络中需要设定的参数数量[20]。

4) VGG 网络是牛津大学视觉组于 2014 年展现于世，并在 ILSVRC2014 比赛分类任务中取得第二名(第一名为 GoogleNet)。由于 VGGNet 整个卷积网络结构都使用了同样大小的卷积核和池化尺寸，其拓展性和迁移的广泛性表现的非常不错。

## 2. 工作内容和问题解决

### 2.1. 工作内容简介

在介绍完上述各类神经网络模型后，本文将所需要研究的内容分为三个工作内容部进行。第一个工作内容为数据集制作，从网络上寻找已有的数据图片集或者带有标签类的图片，并将其分为训练集、验证集以及测试集三部分；第二个工作内容为用 tensorflow 搭建分类神经网络，包括卷积层、采样层的设计以及各个参数的初始化；第三个工作内容为提取图片特征还原图像并制成地图集；内容流程图如图 1 所示。

### 2.2. 拟解决的问题

在进行设计时，首先在进行任何工作时，一个训练较为良好的神经网络是能够将该设计进行下去的前提，包括数据集的制作，神经网络层的设计以及最后运行神经网络层的特征提取操作，都与神经网络挂钩。以下是考虑解决的几个问题：

1) 数据集大小问题，若干张图片提取出来的激活向量是非常大的，可能产生某些高维的激活向量，如何将这些激活向量压缩来减少训练参数也是训练神经网络层的一个关键；2) 梯度消失问题，在使用特定的激活函数时如 sigmoid 函数，后面隐藏层接收到数据训练信号更少，学习效率更低，最终分类准确率下降的问题；3) 激活函数问题，面对不同的数据训练量和不同的神经网络隐藏层，选择的激活函数也有较大的区别。激活函数和设计的神经网络隐藏层数能够直接影响该神经网络性能的好坏。

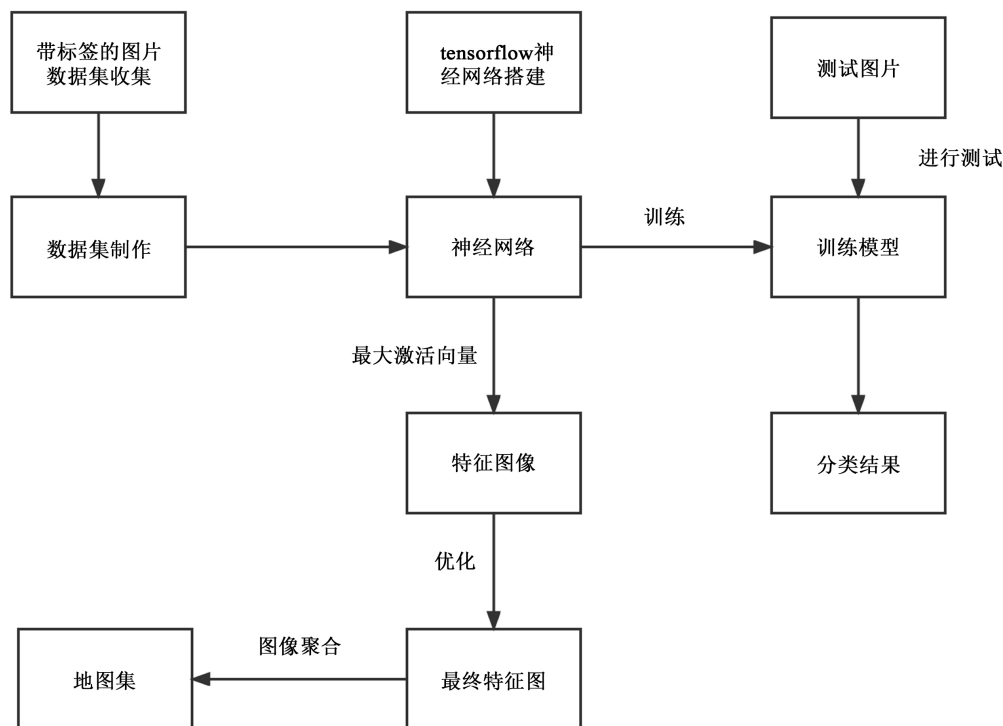


Figure 1. Work content flow chart  
图 1. 工作内容流程图

### 3. 数据集处理

#### 3.1. 数据集制作

数据集收集包括数据集分类以及数据集的大小，它决定了该深度神经网络将学习一种什么样的数据参数以及训练规模的大小。该实现了两类图集的识别分类，第一个是世界各地习俗图集，共有 20 个分类，大概 1000 多幅图片，所用图片为网上带有习俗特定标签的图片收集整理而得到，另外每个在百度百科上搜集关于各个分类习俗的介绍，并在最后的可视化界面上展示。第二个分类为花卉图集，该数据集共有 17 个分类，每种花卉共有 80 副图像，共有 1260 张图片，该图片集是从 CSDN 上下载的花卉图片集，其地址为：[https://www.iteye.com/resource/weixin\\_42458727-10813759](https://www.iteye.com/resource/weixin_42458727-10813759)。

在数据集制作方面，尝试使用 scikit-image 库来读取图片，但是图片读取速度的效果并不太理想，随后采用 Opencv 即 CV2 库来处理图像，读取图片速度相较于 scikit-image 更快，在读取之前，先将所读取的图片利用 resize()方法格式化为统一大小，以便后续神经网络的输入读取数据集时保证不出差错。两个库读取之后的数据格式都是 unit8，图片数据类型为 numpy array，但是 cv2 读取的图片时其输出格式为[B, G, R]格式而不是 scikit-image 的[R, G, B]图像格式，所以在转化为数据集之前，须先使用 cv2.merge()将图片 BGR 格式转化为 RGB 格式，在读取完一类图片之后，利用 os 库中的 Listdir()将该分类图片的标签填充到数据集当中制作数据集标签。

#### 3.2. 数据分布处理

在处理 1000 多个特征的时候，一个个去分析每个变量显然不太可行，这会让我们浪费太多的时间，我们有一种更加方便的方法来处理这些高维数据——降维技术。使用降维技术可以在不用丢失太多信息的情况下还能够减少数据集中的特征数，具体的一些降维技术可见图 2。



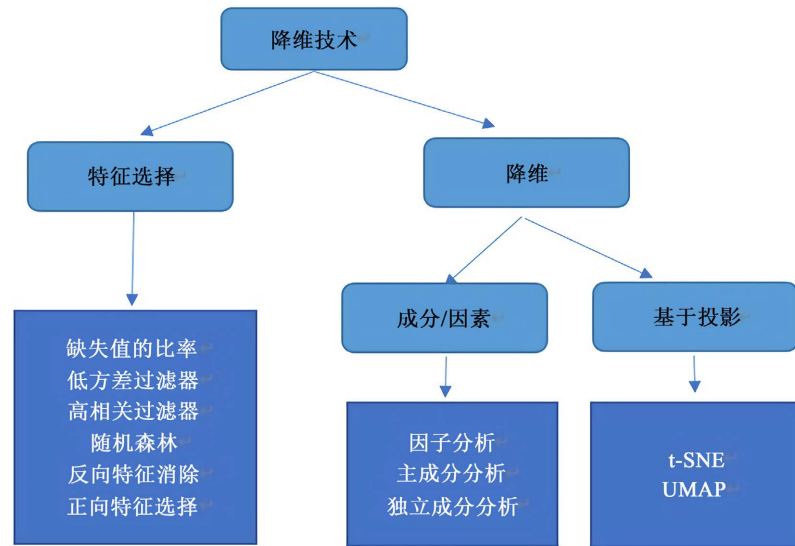


Figure 2. Dimensionality reduction method  
图 2. 降维方法处理

在该数据梳理过程中，我们直接通过降维技术来处理数据的分布，以较低纬度的数据点来表示高纬度的数据，t-分布式随机领域嵌入(t-SNE)方法和 UMAP 相比较基于成分/因素的几种方法处理的数据集更大、处理速度和在保留全局结构上显得更为出色，同时，在该项目需要用到的投影处理也更为合适。T-SNE 和 UMAP 通过计算高维空间和低维空间中点的相似度来判断数据点之间的距离并表现出相似性，t-SNE 在处理大型数据集时相比较 umap 更加完善，由于所制作的数据集不算太大，使用降维技术为 umap 来处理看起来更为合适，具体的两种降维技术的分布都可以在下图体现出来，图 3 为 UMAP 处理数据的分布，图 4 为 t-sne 处理的数据分布。

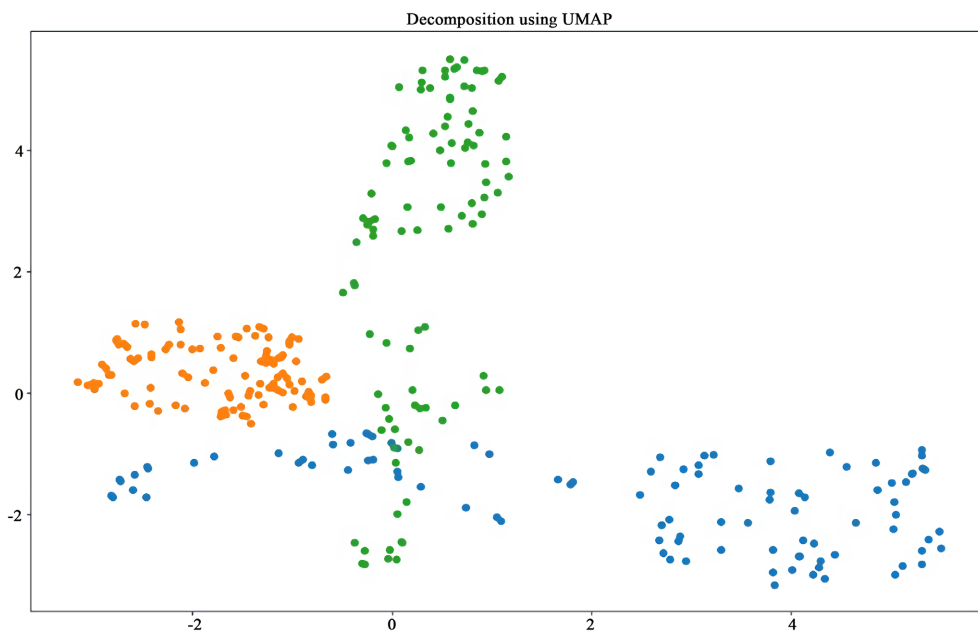


Figure 3. Data distribution in UMAP  
图 3. UMAP 数据分布

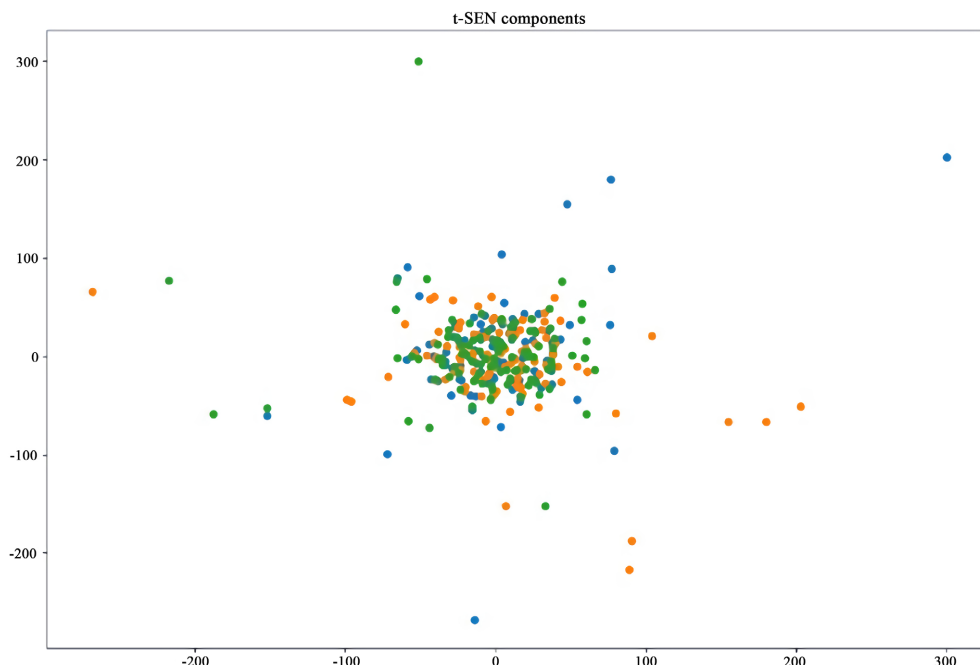


Figure 4. Data distribution in t-sne  
图 4. t-sne 数据分布

Umap 确定使用的相邻点数量为 5,  $\text{min\_dist}$  为 0.05 ( $\text{min\_dist}$  表示的值越大时, 嵌入点的分布越均匀)。具体可观察以下 UMAP 和 T-SNE 分布代码:

```
umap_data = umap.UMAP(n_neighbors = 5, min_dist = 0.05, n_components =
3).fit_transform(df[feat_cols][:6000].values)
tsne_data = TSNE(n_components = 3, n_iter = 300).fit_transform(df[feat_cols][:6000].values)
```

从图 3、图 4 可以看出, 在使用 t-SNE 将低维数据可视化时, 大部分的数据点都集中在一起, 说明低维数据点之间的相关性强, 而比较于 UMAP 的数据点分析时, 其数据点较为均匀, 说明各个低维数据点的相关性不强即其分类结果在二位平面上较为明显。同时, 在该小型数据集上, UMAP 在降维上比 t-SNE 表现的要好, 具体表现在其反映的高维结构更加全面以及其降维的运行速度要更快。

## 4. 搭建神经网络

### 4.1. 神经网络简述

神经网络技术是机器学习当中一门非常重要的技术, 深度学习的学习对现代军事、医疗甚至渗透到我们日常生活中都极为重要。通过让机器处理大量图像数据得到特征, 人为地构造出多层次的神经网络层, 对每一层进行进行训练, 从而建立起一个多层神经网络, 对输入自然图像进行处理分析以及判断其所属类别在实现使用激活地图集探索神经网络时, 首先需先搭建一个能够进行学习的神经网络。对于三类神经网络的模型选择上, 在该设计中, 需要对图像进行分类识别, 故选择卷积神经网络, 其主要优点有以下:

- 1) 不需要对图像数据做太多的处理, 利用 cv2 或者其他图像处理库将输入图像固定到统一大小即可。
- 2) 有特定且精细的特征提取方式(每一层都有卷积核提取特征且有池化层压缩数据和减少过拟合)。
- 3) 在活体识别或者情景识别上适用性较强。

## 4.2. 神经网络模型概要设计

所用卷积神经网络由 TensorFlow 自搭建的七层网络。前四层每一层包含一个卷积层和激励函数 Relu 以及一个池化层，池化层界于每层网络之间，每个池化的结果作为下一个网络层的输入，持续往下不断卷积，共计八个卷积层和池化层，最后计算结果 logits 并且得到神经网络训练模型。

## 4.3. 神经网络参数设计

在构建一个能够学习的神经网络之前，需要确定输入参数的使用(包括区间和数量)，利用 tensorflow.placeholder()构造一个输入参数 x 以便承接图片输入的数据，指定其数据类型为 tf.float32，数据格式为[100, 100, 3] (100 × 100 × 3 的三维矩阵)，分别与图片的长宽(100 × 100)以及通道数(3)相对应，其目的是为了将图片中的每一个像素点以数据的形式进行训练。同时，利用 placeholder()构造一个参数 y，用于计算网络训练的误差，数据类型为 tf.int32，数据格式不定，随训练参数而改变。

在每一层神经网络当中，设置两个基本参数 weight 和 biases，主要是为了减少真实值和预测值之间的偏差。

以第一层卷积层为例，包含卷积、激活以及池化三个过程，从输入图像数据开始，通过卷积核大小为 5 × 5 不断移动提取特征数据，再交由激活函数 Relu 对数据进行非线性拟合，由池化函数对上述拟合数据进行最大化池化，最终池化结果作为下一层输入数据，具体如下所示：

Weight 值以卷积核的大小向量表示，为[5, 5, 3, 32]，在图像上移动的卷积长宽为 5 × 5，通道数以输入的彩色图为 3，数量为 32 个。利用 tf.truncated\_normal\_initializer()函数来初始化参数，即以中间对称，两端截断的正态分布中选取一个随机值，假若选取的随机值比平均值大于两个标准偏差，则会被丢弃并且重新选择一个新的随机值，直到符合条件。

偏差 biases 则利用 tf.constant\_initializer()来生成一个 32 维的张量常量，与卷积之后的数量相对应。

在进行完卷积计算后采用 relu 激活函数并将 biases 与卷积计算结果 conv1 进行向量相加。

即

$$\text{Value}(\text{relu1}) = \text{Value}(\text{biases}) + \text{Value}(\text{conv1}) \quad (3-1)$$

最后调用 tf.nn.max\_pool()函数将上述激活函数值进行最大值池化，且池化窗口与滑动步长一一对应为[1, 2, 2, 1]。

经历该层的结果为由初始图像的 100 × 100 × 3 输入结果为 50 × 50 × 32。

在剩下的三个卷积层以其上一层的输出结果作为输入张量，相同的步骤逐渐向下计算，在全连接层当中，利用 relu 激活函数将该层的 weight 值与从第四层卷积层最后得到的池化结果 reshape (6 × 6 × 128) 之后进行矩阵相乘，fc1\_relu 表示第一层全连接层的激活函数值，fc1\_weight 为第一层的权重，同下类似：

$$\text{Value}(\text{fc1\_Relu}) = \text{Value}(\text{fc1\_weight}) * \text{Value}(\text{Reshape}(\text{pool4})) + \text{Value}(\text{fc1\_biases}) \quad (3-2)$$

同时在该全连接层中采用 dropout 层来减少数据的拟合和过拟合程度以保存模型最好的预测效率。其从整个神经网络层返回的结果为：

$$\text{Logit} = \text{Value}(\text{fc2\_relu6}) * \text{Value}(\text{fc3\_weight}) + \text{Value}(\text{fc3\_biases}) \quad (3-3)$$

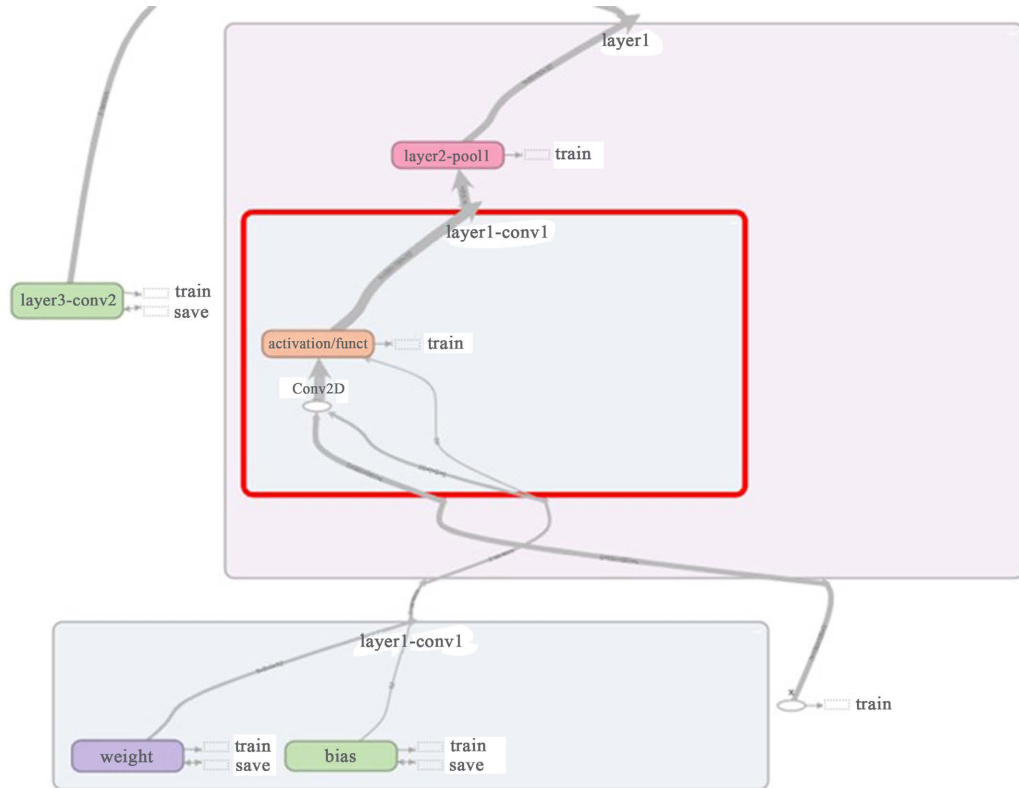
关于卷积层内部参数及设计如下图 5 所示，全连接层内部参数及设计如下图 6 所示。

## 4.4. 损失函数构建以及误差分析

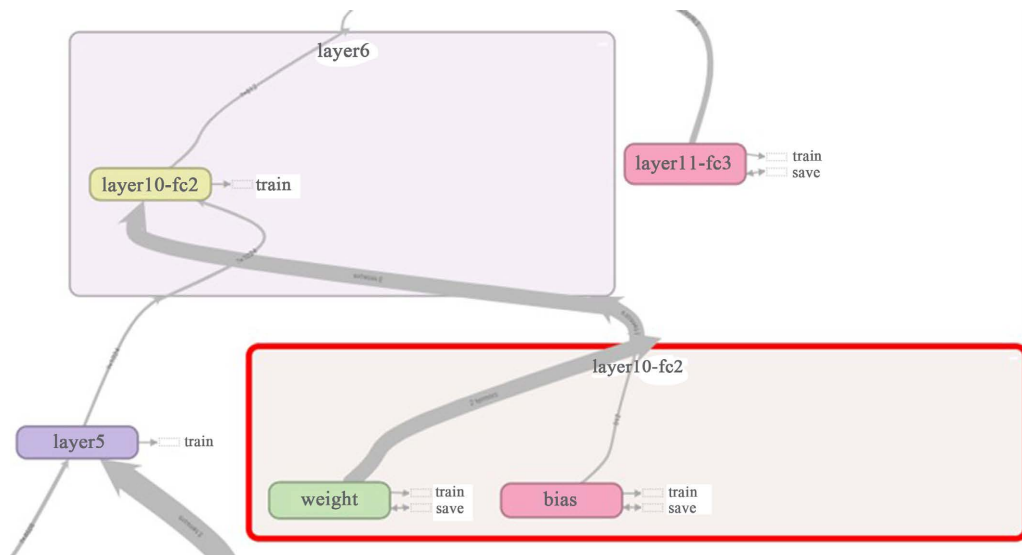
在机器学习算法当中，对于卷积神经网络损失函数一般采用随机梯度下降的算法来实现的，在该神



神经网络中，使用 `tf.nn.sparse_softmax_cross_entropy_with_logits()` 函数来构造损失函数，其包含两个过程，第一个为计算 softmax 部分，第二个为计算其交叉熵部分。在进行图像分类处理时，softmax 是我们常用的分类器之一，它将数据集中的每个分类的输出分量进行归一化，即所有输出分量之和为 1。



**Figure 5.** Convolutional layer internal parameters and design  
**图 5.** 卷积层内部参数及设计



**Figure 6.** Internal parameters and design of fully connected layer  
**图 6.** 全连接层内部参数及设计

第二个部分为计算交叉熵，即分类标签(label)和神经网络结果 logits 进行交叉的乘积。即

$$H_y(y) = -\sum y_i \log(y_i) \quad (3-4)$$

确定好损失函数后，为了将网络模型中的网络参数和训练参数更好的逼近或者达到最优值，一个更适应神经网络的优化器往往是我们最好的选择，它能够将损失函数的值进行最小化处理。在进行优化时，选择 Adam 优化器，学习率默认设为 0.01，过高则会导致过度拟合，过低则会出现拟合度不够的情况，从而在模型训练结果上出现偏差，具体可以根据实际训练精准度进行调整。Adam 在实现小型数据集训练上，由于其实现简单高效的特点更能符合当前网络的构建。

在进行训练时，将整个数据集划分为两个部分，80%的训练图像集以及 20%的验证图像集。训练数据集在该网络模型中的迭代次数为 10 次，并且构造一个函数以在输入数据中利用 shuffle()函数来随机选取一批数据，该批次数据大小为 64，并且将最后的训练模型结果保存至文件夹内，训练模型最大保存个数为 10 个，但在利用模型进行测试时，本次只选择保存 5 个模型，分别是第 2 个、第 4 个、第 6 个、第 8 个和第 10 个模型，以确保在进行分类时不会由于模型过于偏向随机批次训练而产生更大的误差，最终产生的训练和验证误差如图 7 所示。

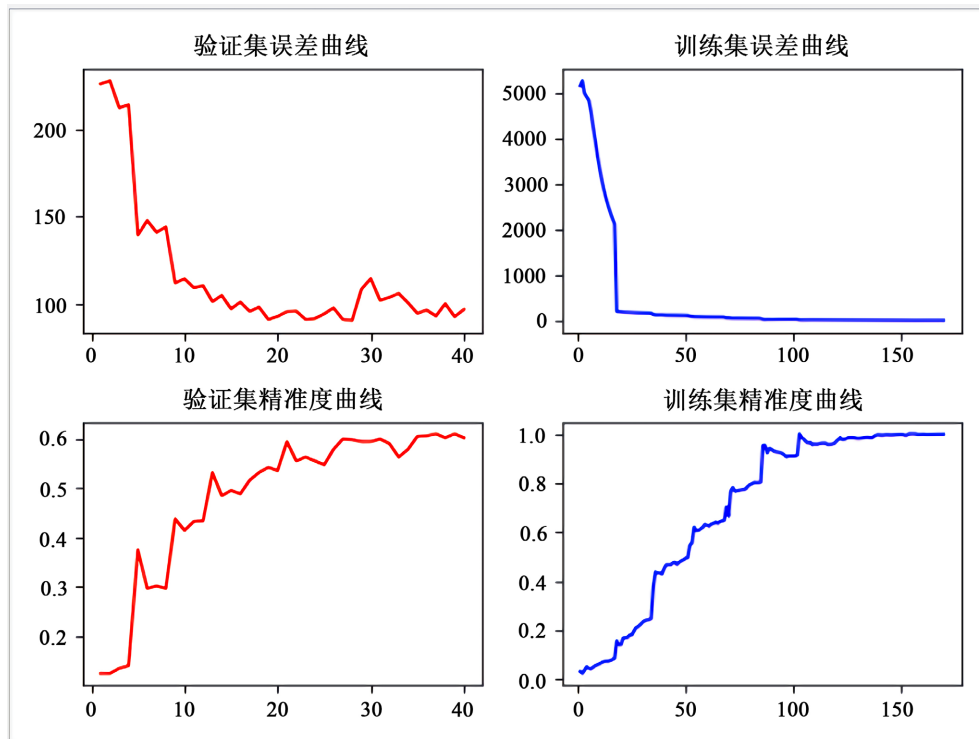


Figure 7. Training/validation error and accuracy curves

图 7. 训练集/验证集误差和精准度曲线

#### 4.5. 分类效果实现

本次设计中，从所选图片除了显示其分类结果外，同时也有一段对于该分类结果的介绍。举个例子，在习俗分类结果中，除了显示其分类，会有一段关于该习俗的介绍，同样的，对于花卉分类介绍亦是如此(以下介绍仅以花卉分类作为讨论)。此分类效果包含两个部分，第一个部分为选择测试图片，第二部分调用接口用于显示图片分类以及介绍，具体演示如图 8 所示。

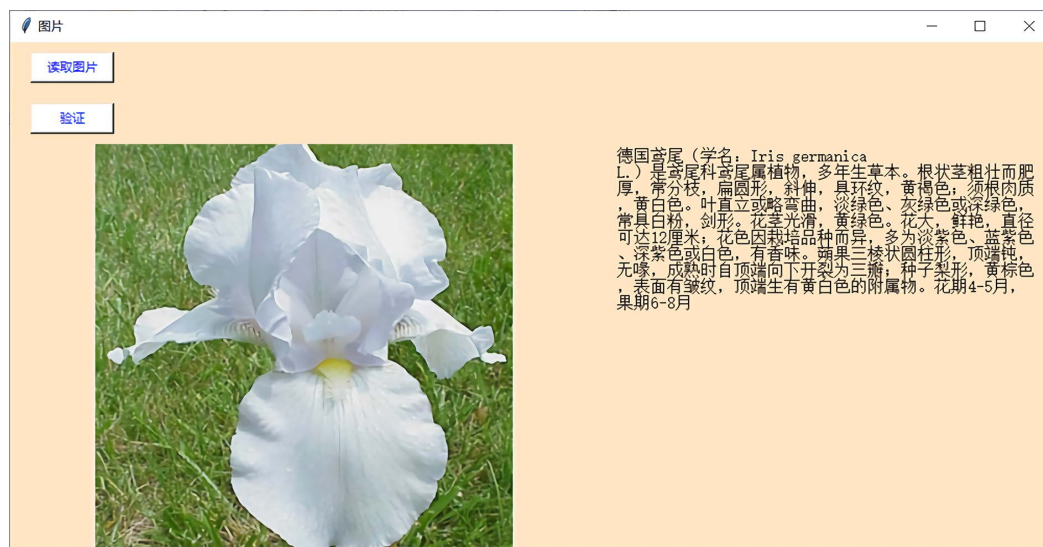


Figure 8. Classification effect demonstration

图 8. 分类效果演示图

该界面利用 tkinter 进行简单的绘制而成，主要由四个部分组成。两个功能按钮，“读取图片”和“验证”，两个 label，分别用于存放图片以及分类介绍。首先定义 flower\_dict[] 和 Introduction\_dict[] 数组分别用于存放花卉类型以及花卉介绍，且其在数组中的位置一一对应。在点击选取图片按钮时，暂时保存其图片的选择路径并且将其展示在界面上，同时调用 test 文件中接口 read\_img() 将该图片转化为 numpy 数组类型以便后续的矩阵预测分析。在点击验证按钮时，调用 test 文件中的 result() 接口，该接口主要功能是加载训练模型，同时从整个模型图中获取到输入变量 x，以选择测试图片数据 feed 输入变量 x，经过 session.run() 计算出预测矩阵，同时以该矩阵每一行的最大索引来预测出其分类结果，另一方面，通过该分类结果来匹配 introduction\_dict[] 中的介绍类容并展示在分类界面上。

## 5. 地图集制作

### 5.1. 地图集简介

在进行地图集制作之前，首先得知道在进行特征训练的时候如何观察个体神经元的反应。我们知道，在神经网络中，单个神经元不能直接进行工作，所以我们需要对网络层中的神经元进行简单的组合并将它们所激活的那部分特征进行可视化组合起来，从而能够可视化我们寻求的网络的全局视图。作为一个类比，我们的汉字，如果单单只有一个字，或许让我们难以理解它所表达的意思，但如果我们将这些汉字进行组合成一个词语，一段话，能够让我们更能清楚的观察理解到它的具体信息。此时，激活地图集显示多个神经元之间的组合能够为我们提供一个更大更全面的图画。

每一幅图像的特征都能够在最后的地图集当中寻找到对应的模块图，如图 9 所示。

### 5.2. 单一特征图可视化

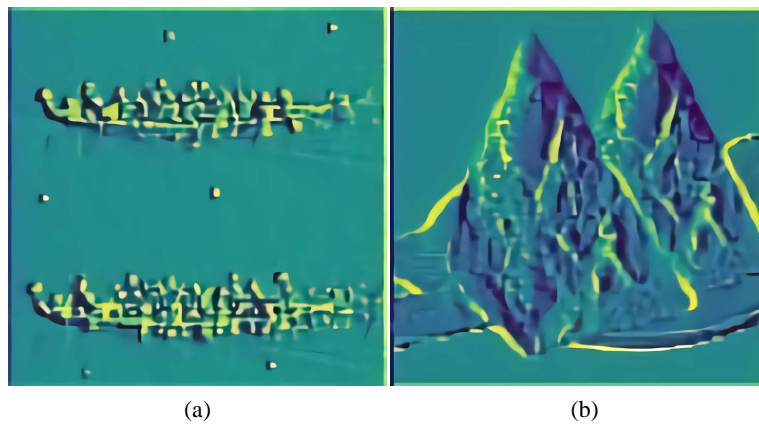
首先，我们从单个图像开始，为了看到该神经网络中的图像，需要将输入图像运行到所需要的网络层(该地图制作选取的也是第二个卷积层可视化图)，根据前面卷积核的大小，我们可以知道第二层的激活形状为 [1, 25, 25, 64]，也就是说，在第二层网络层中其激活矢量会有 64 个，我们从这每一个激活矢量通过迭代优化过程产生一副激活图像，为了从该层选择神经元响应最为强烈的矢量，即 64 个激活矢量中找寻最大激活矢量然后对该矢量优化还原成一幅图像。



**Figure 9.** Map module diagram  
**图 9.** 地图模块图

具体关键代码以及产生的特征图如图 10 所示:

```
max_activation = 0
for i in range(64):
    #选择该层中最大激活向量
    if max_activation < np.array(layer_activation[0, :, :, i]).max():
        feature_image = layer_activation[0, :, :, i]
    #优化所提取的特征
    feature_image -= feature_image.mean()
    feature_image /= feature_image.std()
    feature_image *= 64
    feature_image += 128
    feature_image = np.clip(feature_image, 0, 255).astype('uint8')
```



**Figure 10.** The second layer convolutional layer feature map  
**图 10.** 第二层卷积层特征图



图 9、图 10 两幅特征图是从端午节习俗分类中在第二层的最大激活向量特征图，从图上我们还可以看出划船龙舟比赛以及粽子的轮廓，但是也可以看出，特征图上的比较光亮的地方比如龙舟以及粽叶边缘的轮廓随着网络的逐渐向下，其特征显示更为明显，而其它地方则会显得愈加模糊，最终，我们可以对该图进行迭代优化还原出其更加美观的图像。

### 5.3. 图像聚合

从上述单一特征图可以简单看出网络中神经元对单一图像的激活状态，当我们将单个分类结果图像或者所有分类结果图像上进行组合，从而形成一张庞大的地图集，可以更好地帮助我们理解各个神经元之间组合产生的激活。我们可以做一个设定，即在上述单一特征图处理时，我们先将同类别图像进行特征处理，然后绘制一个 cell 网格，将已经处理好的图像嵌入该网格内，网格数量由该分类中特征图数量确定，在完成所有网格图像嵌入后，对于不同类的最大激活特征图，比较其向量的相似程度，相似程度高的分类则邻近存放，相似程度较为低的则隔类存放，从我们的角度上看，款冬和向日葵从其花盘，花色以及花瓣，两者样子很是相像，在最终的地图集上，这两个分类是邻近存放的。当然，在最终分类合并和相似分类合并时，尽管某些花卉的相似度很低，也有可能是邻近存放的。具体流程图如图 11。相似类的隔离如图 12 所示，图上分别为蒲公英，款冬黄花九轮草和黄水仙四种花卉，不同类的聚合如图 13 所示，分别有番红花，角堇，款冬和黄花九轮草组成。

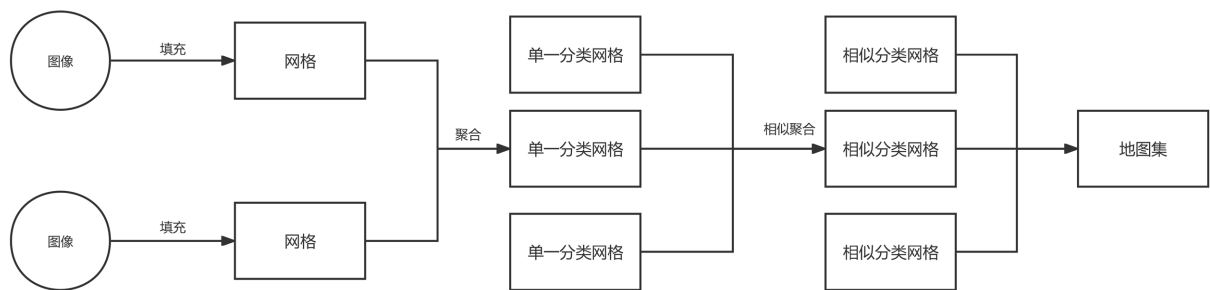


Figure 11. Image aggregation flowchart

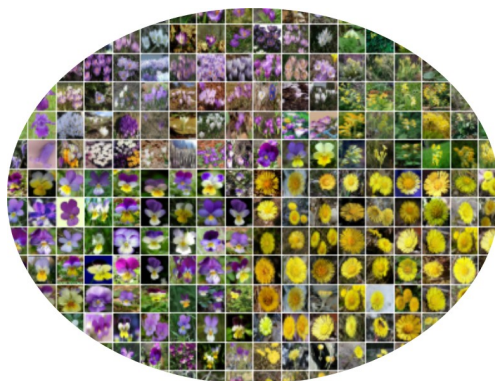
图 11. 图像聚合流程图



Figure 12. Similar class isolation

图 12. 相似类隔离





**Figure 13.** Dissimilarity class  
**图 13.** 不同类隔离

## 6. 总结与展望

### 6.1. 总结

机器学习模型通常被我们认作为“黑盒子”，它可以自动帮助我们执行任务并且输出一个结果，但是我们并不理解其中的工作过程。激活地图集这种技术的提出为我们提供了对卷积神经网络的一个全新的理解和融入，透过隐藏层，它提供了人可以理解和解释的一种全局的概念和概观，我们可以更好地了解这些复杂系统的内部工作原理，同时还可以有一种新的方法来处理图像。这种方法技术揭示了图像学习的字符组合表，它不仅仅是单一神经元所能表现出来的，更多的是与它相关联且能组合的神经元一起工作所产生的结果。在本文中所展现的分类网络层数只有简单的七层，相比较于 VGG16 网络以及其他通用卷积神经网络在分类效果上可能略显不足，但是其需要的运行环境也会尽可能降低，在运行迭代次数不超过 100 次的情况下，其在 CPU 环境下运行也都是可行的，要求的时间也会随迭代次数增加而倍增。

### 6.2. 展望

激活地图集技术可以为我们以后提供一个更加简便的从大型图像数据集中查询图像类型的方法，与我们利用单词来从文本中的查询方式类似，比如在我们需要在图像中寻找狗的图像时，由于人类的语言通常并不适合描述特定的视觉图像特征，但随着从地图集中哺乳动物、犬类等一步步往下细分，或者从特征上分析，比如狗的鼻子那一类分布的图集中，便能够快速的找到所需要的特征图集。

## 参考文献

- [1] 常亮, 邓小明, 周明全, 等. 图像理解中的卷积神经网络[J]. 自动化学报, 2016, 42(9): 1300-1312.
- [2] Buchheim, A., Erk, S., George, C., *et al.* (2016) Neural Response during the Activation of the Attachment System in Patients with Borderline Personality Disorder: An fMRI Study. *Frontiers in Human Neuroscience*, **10**, Article No. 389. <https://doi.org/10.3389/fnhum.2016.00389>
- [3] Jia, Z., Lin, S., Qi, C.R., *et al.* (2018) Exploring Hidden Dimensions in Parallelizing Convolutional Neural Networks.
- [4] Shaked, A. and Wolf, L. (2016) Improved Stereo Matching with Constant Highway Networks and Reflective Confidence Learning. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 4641-4650. <https://doi.org/10.1109/CVPR.2017.730>
- [5] Nikhil, K. (2017) *Deep Learning with Python*. Springer, New York, 117-119. <https://doi.org/10.1007/978-1-4842-2766-4>
- [6] Zhao, Y., Dong, Q., Chen, H., *et al.* (2017) Constructing Fine-Granularity Functional Brain Network Atlases via Deep Convolutional Autoencoder. *Medical Image Analysis*, **42**, 200-211. <https://doi.org/10.1016/j.media.2017.08.005>
- [7] 刑珍珠. 卷积神经网络在图像处理中的应用研究[J]. 软件工程, 2019, 22(6): 5-7.

- 
- [8] 王瑞. 基于卷积神经网络的图像识别[D]: [硕士学位论文]. 开封: 河南大学, 2015.
- [9] 段萌. 基于卷积神经网络的图像分类技术研究[D]: [硕士学位论文]. 湘潭: 湘潭大学, 2017.
- [10] 杨子贤. 基于深度学习卷积神经网络图像识别技术的研究与应用[J]. 中国设备工程, 2018(23): 146-149.
- [11] 孙康, 李千目, 李德强. 基于级联卷积神经网络的人脸检测算法[J]. 南京理工大学学报, 2018, 42(1): 40-47.
- [12] Carter, S., Armstrong, Z., Schubert, L., *et al.* (2019) Exploring Neural Networks with Activation Atlases. <https://doi.org/10.23915/distill.00015>
- [13] Dwivedi, Y.K., Hughes, L., Ismagilova, E., *et al.* (2019) Artificial Intelligence (AI): Multidisciplinary Perspectives on Emerging Challenges, Opportunities, and Agenda for Research, Practice and Policy. *International Journal of Information Management*, 2019, Article ID: 101994. <https://doi.org/10.1016/j.ijinfomgt.2019.08.002>
- [14] 纪守领, 李进锋, 杜天宇, 等. 机器学习模型可解释性方法, 应用与安全研究综述[J]. 计算机研究与发展, 2019, 56(10): 2071-2096.
- [15] Tadeusiewicz, R., Chaki, R. and Chaki, N. (2014) Exploring Neural Networks with C. CRC Press, Boca Raton.
- [16] 石琪. 基于卷积神经网络图像分类优化算法的研究与验证[D]: [硕士学位论文]. 北京: 北京交通大学, 2017.
- [17] 宋光慧. 基于迁移学习与深度卷积特征的图像标注方法研究[D]: [博士学位论文]. 杭州: 浙江大学, 2017.
- [18] 何鹏程. 改进的卷积神经网络模型及其应用研究[D]: [硕士学位论文]. 大连: 大连理工大学, 2015.
- [19] 谢宝剑. 基于卷积神经网络的图像分类方法研究[D]: [硕士学位论文]. 合肥: 合肥工业大学, 2015.
- [20] 许可. 卷积神经网络在图像识别上的应用的研究[D]: [硕士学位论文]. 杭州: 浙江大学, 2012.