



**HUNAN UNIVERSITY**

数据挖掘方法及应用

|  |  |
| --- | --- |
| 研究题目： | 基于深度学习的游戏皮肤颜色与定价关系研究 |
| 学生姓名： | 高清诠 刘韩 张鑫 |
| 专业班级： | 2023级管理科学与工程 |
| 学院名称： | 工商管理学院 |
| 指导老师： | 田博士 |

2024年6月10日

# 小组分工

高清诠：数据爬取，文档撰写，复盘总结

刘韩：数据爬取，图片信息导入，卷积神经网络构建

张鑫：数据细节优化，卷积神经网络优化，定制价格标签

目 录

[小组分工 I](#_Toc22899)

[插图索引 III](#_Toc28612)

[插表索引 IV](#_Toc4530)

[一、绪论 1](#_Toc32481)

[（一）研究背景 1](#_Toc4773)

[（二）数据概述 1](#_Toc7375)

[（三）研究框架 1](#_Toc18910)

[二、数据处理 2](#_Toc19513)

[（一）数据爬取 2](#_Toc9217)

[1. 相关库介绍 2](#_Toc19922)

[2. 相关库作用解析 2](#_Toc5075)

[3. 数据查找与初步爬取 3](#_Toc12961)

[（二）数据调整 6](#_Toc4660)

[1. 皮肤与对应英雄文件夹一致 6](#_Toc21555)

[2. 皮肤图片大小一致 7](#_Toc28159)

[3. 皮肤定价标签化 7](#_Toc15216)

[三、基于卷积神经网络的皮肤颜色与定价关联性数据分析 8](#_Toc5132)

[（一）卷积神经网络介绍 8](#_Toc21276)

[（二）卷积神经网络结果分析 9](#_Toc22580)

[四、项目总结 11](#_Toc27457)

# 插图索引

[图1 数据来源官网 4](#_Toc5693)

[图2 API接口查找步骤图 4](#_Toc5207)

[图3 异步编程asyncio.Semaphore函数使用流程图 5](#_Toc24620)

[图4 英雄皮肤爬取流程图 5](#_Toc18229)

[图5 英雄皮肤存储流程图 6](#_Toc7087)

[图6 英雄皮肤图片格式设置图 7](#_Toc1517)

[图7 卷积神经网络基础结构图 8](#_Toc16643)

[图8 测试集与预测集划分图 9](#_Toc28642)

[图9 卷积神经网络代码图 10](#_Toc20155)

[图10 卷积神经网络Epoch-Accuracy图 10](#_Toc31100)

# 插表索引

[表1 相关包的介绍 2](#_Toc25156)

[表2 皮肤价格标签值表 7](#_Toc29421)

基于深度学习的游戏皮肤颜色与定价关系研究

一、绪论

## （一）研究背景

根据对课上提供文献进行研读后，本小组选择了《Color and engagement in touristic Instagram pictures: A machine learning approach》这篇文献，并运用其中的机器学习方法，对原文进行了代码复现，分析其实际意义。选用文献主要研究了颜色在识别游客体验和影响游客情绪方面起到的作用。作者通过使用机器学习方法对Instagram上的旅游照片进行分类，并分析了不同照片颜色和用户参与度之间的关系。研究结果表明，在以自然风光、高端美食和神圣建筑为特色的照片中出现蓝色有助于用户参与。红/橙配色方案增强了关于当地美食和氛围的照片，而紫色和暖色的共存对于以城市景观和室内设计为特色的照片至关重要。可以发现该文献的实际意义在于营销人员通过色彩特点进行推销，从而促进旅游业发展。

基于上述背景，本项目对知名腾讯游戏《英雄联盟》中皮肤色彩与皮肤定价的关系进行分析，针对不同颜色与皮肤定价，找到他们之间的关系，为营销人员与美工设计人员进行皮肤设计与营销提供相应参考。

## （二）数据概述

本项目使用的数据为《英雄联盟》游戏皮肤图片与皮肤定价，获取途径为腾讯游戏道具城官网，获取方式为Python进行爬虫处理，并将爬取后的图片及定价数据进行处理，之后进行数据相关性的分析。

## （三）研究框架

本文基于数据爬取，运用Python对《英雄联盟》皮肤颜色与皮肤定价相关性进行分析，分析步骤如下：

1. 下载并安装所需的库，对软件进行熟悉，创建深度学习环境；

2. 进行数据爬取操作学习并运用异步编程爬取图片数据，为下一步工作做准备；

3. 对爬取的图片数据进行调整，图片格式调整为一致，并对各皮肤价格根据价格区间进行打标签处理，并采用深度学习卷积神经网络，对《英雄联盟》皮肤颜色与皮肤定价关联性进行分析；

4. 总结并提出相关建议，回顾数据处理与课程设计中的不足，为之后的科研任务打好基础。

# 二、数据处理

## （一）数据爬取

### 1. 相关库介绍

在开始数据爬起前，需要导入Python相关库并配置深度学习环境方便进行数据爬取，本项目使用的Python库如下表所示：

表1 相关包的介绍

|  |  |
| --- | --- |
| 包名称 | 包的作用 |
| AsyncIO | 提供了一个高性能的事件循环和异步编程模型 |
| OS | 提供了一种直接与操作系统交互的方式 |
| ClientSession | 用于在异步环境中发送 HTTP 请求 |

### 相关库作用解析

AsyncIO 库主要用于以下几个方面:

1. 网络编程:AsyncIO 提供了一个高效的网络 I/O 模型,可用于编写高性能的网络服务器和客户端,如 Web 服务器、RPC 框架等。

(2) 异步 I/O:AsyncIO 可以帮助你编写高效的异步 I/O 代码,比如与数据库、文件系统、外部 API 等进行交互。

(3) 并发编程:AsyncIO 提供了一种基于协程的并发编程方式,可以帮助编程者编写出更加简洁、易维护的并发代码。

(4) 事件驱动编程:AsyncIO 的事件循环机制允许你编写事件驱动的应用程序,这在需要处理大量并发事件的场景下非常有用。

(5) 微服务和分布式系统:由于 AsyncIO 的高性能和可扩展性特点,它可以用于构建微服务架构和分布式系统。

总的来说,AsyncIO 是 Python 中一个非常强大和灵活的并发编程库,它可以帮助你编写出高性能、可扩展的 Python 应用程序。它广泛应用于网络编程、I/O 密集型应用、微服务架构等场景。

os库是Python中用于与操作系统交互的标准库之一,它提供了一系列与操作系统相关的功能和接口。os库的主要用处包括以下几个方面:

1. 文件和目录操作:包括创建、删除、重命名文件和目录、获取文件和目录的属性信息、遍历目录结构以及设置和获取当前工作目录。
2. 进程管理:包括创建、终止和管理进程、获取进程ID和父进程ID、等待进程终止并获取退出状态。
3. 环境变量管理:包括获取和设置环境变量、操作系统相关的路径分隔符和换行符。
4. 系统相关操作:包括获取操作系统名称和版本信息、执行系统命令并获取输出、访问和修改系统权限。
5. 跨平台支持:os模块提供了许多跨平台的函数和常量,使得代码可以在不同操作系统上运行。

总的来说,os模块提供了一种直接与操作系统交互的方式,使得Python程序能够完成许多系统级别的任务,如文件管理、进程控制、环境配置等。这在系统编程、自动化脚本等场景下非常有用。同时,它还帮助Python程序实现了良好的跨平台性。

ClientSession是aiohttp库中的一个核心组件,它用于在异步环境中发送HTTP请求。

ClientSession的主要作用包括:

1. 建立HTTP会话：ClientSession提供了一个持久的HTTP会话,可以在多个请求之间共享cookies、headers等信息。这样可以减少重复设置这些信息的开销。
2. 管理连接池：ClientSession会管理底层的TCP连接,使用连接池来避免频繁的连接/断开开销。这对提高HTTP请求的性能很重要。
3. 支持异步：ClientSession是基于asyncio实现的,可以与其他异步操作无缝集成,方便编写高性能的异步HTTP客户端程序。
4. 提供丰富的API：ClientSession提供了各种HTTP方法(GET、POST、PUT等)的API,以及对cookies、headers、身份验证等的支持。

### 3. 数据查找与初步爬取

首先对数据进行查找，并进行数据接口检视，数据来源网站如下图所示：

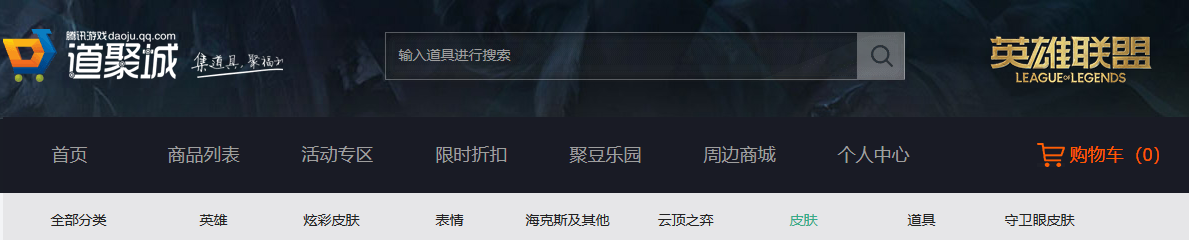


图1 数据来源官网

运用def函数定义英雄列表，即defhero\_list()，使用该步骤进入英雄联盟官网的英雄链接https://lol.qq.com/data/info-heros.shtml，发现内容并不是储存在静态网页中，通过查看元素，找寻到了接口https://game.gtimg.cn/images/lol/act/img/js/heroList/hero\_list.js，操作如图2所示：

1e845816fa3eba5e27dc4e4b6538103

图2 API接口查找步骤图

之后运用def函数进行各英雄数据异步任务建立，即defrun()，使用该函数遍历英雄列表，为每个英雄创建一个下载英雄皮肤的异步任务，并储存在项目总目录下，这样做的目的是将每个英雄的数据分类爬取，为之后数据分析奠基。“asyncio.Semaphore(30)”这段代码用于限制并发请求的数量,这在编写异步网络爬虫程序时非常有用。

在异步编程中,我们可以同时发起大量的网络请求,以提高整体的执行效率。但是如果直接发起大量请求,可能会给服务器造成负担,甚至导致网站访问被屏蔽。为了避免这种情况,我们可以使用“asyncio.Semaphore”来限制并发请求的数量。“asyncio.Semaphore(30)”的作用就是创建一个最多允许30个并发请求的信号量对象。具体工作原理如下:

(1) 当一个请求开始时,会先尝试获取信号量。

(2) 如果当前并发请求数量未达到上限(30),则该请求可以立即获取信号量并继续执行。

(3) 如果当前并发请求数量已达到上限,则该请求会被挂起,直到有其他请求释放了信号量。

(4) 当请求执行完毕时,会释放占用的信号量,让其他挂起的请求有机会获取。

代码流程如图3所示：



图3 异步编程asyncio.Semaphore函数使用流程图

因此，使用该函数可以有效控制并发请求的数量,避免给服务器造成过大压力。在爬虫过程中，对于需要长时间处理的请求,可以释放信号量供其他请求使用,提高整体的执行效率。同时可以根据服务器的承受能力,灵活调整并发请求的上限。

根据英雄爬取结果，对各个英雄皮肤进行进一步的爬取并储存在英雄对应的独立文件夹中，这样做可以减少后续数据重复删除、数据缺漏查找时的时间消耗，依然使用def函数进行操作，这里定义def skins\_downloader()，找到皮肤对应接口并请求到对应英雄的url地址，在各个英雄的对应路径下进行遍历爬取操作。如图4所示：



图4 英雄皮肤爬取流程图

这里处理的难点在于对皮肤图片的命名，这里运用“format()”是 Python 中字符串格式化的一种方法,它可以用来将值插入到字符串中。其主要作用如下：

1. 替换占位符，在实际操作中，可以在字符串中使用花括号“{}”作为占位符,然后在“format()”方法中传入相应的值,这些值将会替换掉这些占位符。

(2) 控制格式，可以在占位符中添加格式化选项,例如对齐方式、数字精度等,以控制值的显示方式。

(3) 按索引或关键字引用，占位符可以通过位置索引或关键字参数来引用对应的值。

总之,“format()”方法提供了一种灵活的方式来格式化字符串,在实际编程中能够将动态值插入到文本中,并控制它们的显示方式。这在需要构建动态文本输出时非常有用。

在实际的爬取过程中，受网络稳定性、服务器连接稳定性等因素的影响，英雄数据会在爬取约10次左右可以得到完整结果。

## （二）数据调整

### 1. 皮肤与对应英雄文件夹一致

在数据爬取之后，如何将爬取的皮肤与对应的英雄信息进行对应，需要进行二次处理，从而确保皮肤与对应英雄信息一致的同时去除重复值，因在数据爬取过程中，存在部分皮肤因实际版本原因导致图片重复，这里进行去除操作。

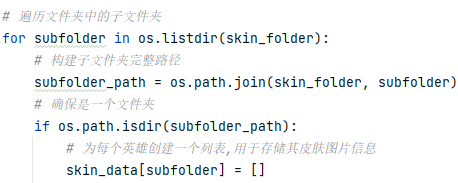


图5 英雄皮肤存储流程图

这里根据上述操作，将各个英雄皮肤存储至对应的英雄命名文件夹内，并遍历子文件夹中的图片文件以确保储存的文件夹是正确的。

### 2. 皮肤图片大小一致

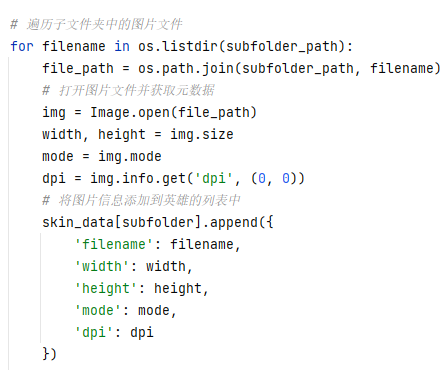


图6 英雄皮肤图片格式设置图

为方便后续数据分析，这里对各个皮肤图片格式进行定义，将各图片格式保持一致，从而便利于后续数据分析。

### 3. 皮肤定价标签化

上述处理将各英雄皮肤图片进行分类储存，之后需要对各个皮肤的定价进行处理。Joanne Yu等人在对于照片颜色对人的吸引力研究中，将吸引力定义为价格，即人们看到该颜色愿意去该场景消费的金额，这里作者将消费金额的区间进行了界定，对于各个区间的消费金额打上标签，方便之后进行数据分析。因此本文沿用该文献中对于价格、金额的处理办法，将英雄皮肤售价数据进行爬取导出后，对售价进行标签，下表为各个皮肤价位的标签值：

表2 皮肤价格标签值表

|  |  |
| --- | --- |
| 价格区间 | 标签值 |
| 35及以下 | 0 |
| 36-58 | 1 |
| 59-78 | 2 |
| 79-88 | 3 |
| 89-98 | 4 |
| 99-108 | 5 |
| 109-129 | 6 |
| 130及以上 | 7 |

# 三、基于卷积神经网络的皮肤颜色与定价关联性数据分析

## （一）卷积神经网络介绍

卷积神经网络（Convolutional Neural Network，CNN）是一种常用于图像识别、语音识别等领域的深度学习模型。CNN的核心思想是通过卷积操作来提取图像等数据的特征，从而实现对数据的分类、识别等任务。

CNN的基本结构由卷积层、池化层和全连接层组成。其中，卷积层是CNN的核心，它通过滑动一个卷积核在输入数据上进行卷积操作，从而提取出数据的局部特征。卷积操作可以看作是一种特殊的加权求和操作，其中卷积核中的权重参数是通过训练学习得到的。池化层用于对卷积层输出的特征图进行降维处理，从而减少模型参数和计算量。常用的池化方式有最大池化和平均池化两种。全连接层用于将池化层输出的特征向量映射到输出类别上，从而实现对输入数据的分类或识别。下图是卷积神经网络的图解：

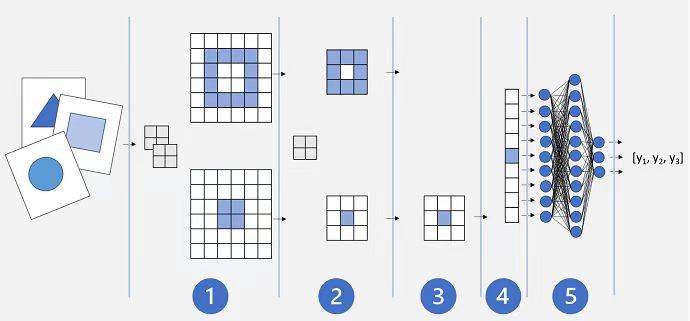


图7 卷积神经网络基础结构图

由图7可知，图像送入卷积层中，在图中，有两个筛选器，因此每个图像会生成两个特征图。特征图传递到池化层，其中 2x2 池化内核将减小特征图的大小。删除层会随机删除某些特征图，以帮助防止过度拟合。平展层利用剩余的特征图数组，并将它们平展为向量。向量元素送入一个完全连接的网络，该网络将生成预测。在本例中，网络是一个分类模型，可预测三个可能的图像类（三角形、正方形和圆圈）的概率。

CNN的优点在于它能够自动学习和提取数据的特征，无需手动进行特征工程。此外，CNN还具有平移不变性、参数共享等特点，使得模型具有较好的泛化能力和鲁棒性。

总之，卷积神经网络是一种基于卷积操作的深度学习模型，它通过自动学习和提取数据的特征来实现对数据的分类、识别等任务。

## （二）卷积神经网络结果分析

根据文献的研读，Ozkul等人强调，颜色为旅游和营销领域的服务场景文学提供了一个前提，因为它在引导一个人的情绪和行为反应方面发挥着关键作用。则在本研究中，小组对于英雄联盟皮肤颜色是否对于销售方定价有影响进行分析，从而在之后的研究中，可以对该思路进行展开，对皮肤颜色是否影响皮肤销量，即消费者行为进行研究，这同样需要运用卷积神经网络进行分析。本文根据数据爬取结果进行数据处理后，共1440条数据，其中测试集与预测集进行1：1划分，即720个测试集与720个预测集，如下图所示：

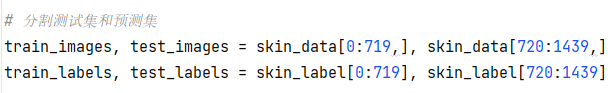


图8 测试集与预测集划分图

根据CNN原理，本文对爬取图像进行处理如下图所示：



图9 卷积神经网络代码图

根据上述代码，得出皮肤颜色与皮肤定价的关联性，通过CNN进行遍历迭代分析，得出结果如下图所示：

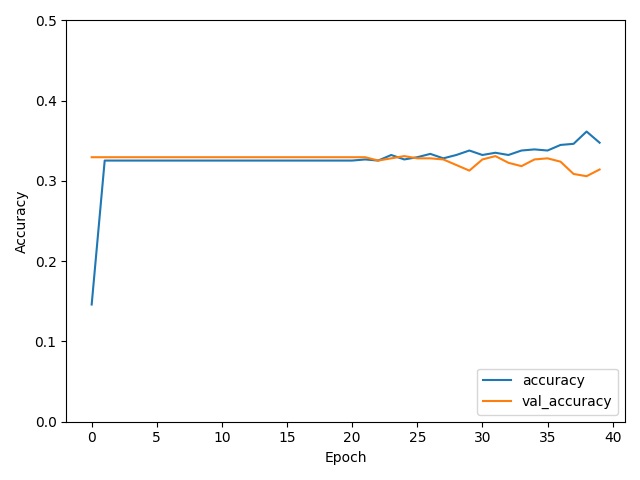


图10 卷积神经网络Epoch-Accuracy图

由图可知，在训练次数为3次左右时训练达到稳定，之后再22次之后开始波动，可以发现在3-22次的曲线是最稳定的，即训练次数为该次数最佳，此时的训练拟合的准确率为31.43%，可知对于皮肤颜色与皮肤定价的预测准确率是较低的，在之后的研究中将根据以下解决方案对准确性进行提升：

1.使用更复杂的模型：可以尝试增加模型的复杂度，例如增加更多的隐藏层或增加隐藏单元的数量。这样可以提供更多的参数来捕捉数据中的复杂关系。

2.调整参数：包括学习率、批次大小和训练轮次等。可以尝试不同的超参数组合，找到最优的设置。可以尝试使用学习率调度器来动态地调整学习率。

3.数据增强：通过在训练集中应用变换（例如旋转、平移和缩放），可以增加数据的多样性，有助于模型更好地泛化到新样本。

4.正则化：可以尝试使用正则化技术（如L1或L2正则化）来限制模型的复杂度，以减少过拟合。

5.模型集成：可以尝试使用集成学习方法，将多个模型的预测结果进行组合，以提高准确性。

# 四、项目总结

本项目主要基于文献研读，根据文献作者提供的对于场景色彩与消费者花费金额之间的关系建立的深度学习分析方法进行沿用，从数据爬取、数据处理到数据分析形成了完整的体系，根据分析结果对于消费者购买皮肤及营销人员指定皮肤价格有一定的参考意义。在模型训练过程中，小组成员学习到了深度学习的卷积神经网络方法对图像进行处理，不足之处在于对于模型的训练仍存在欠缺，在后续的研究与科研生活中，将持续精进深度学习有关内容，实现自我突破。