

**《专业综合实践II》课程考核**

|  |  |
| --- | --- |
| **题 目：** | 基于SVM、KNN、决策树实现垃圾图像识别 |
| **学 院：** | 数据科学学院 |
| **专 业：** | 数据科学与大数据技术 |
| **年级班别：** | 2021级数据科学与大数据技术（1）班 |
| **学 号：** | 421470104 |
| **学生姓名**： | 陈嘉宁 |

**提交日期：** 2024 **年** 6 月

**目 录**

[封 面](#_Toc384051565) 1

[目 录](#_Toc384051565) 2

一、[概要](#_Toc384051566) 3

1.1研究背景和研究意义

1.1.1研究背景

1.1.2研究意义

二、[数据描述](#_Toc384051566) 4

2.1垃圾图像数据来源

[三、数据预处理 5](#_Toc384051567)

3.1数据样本分析

3.2数据集长宽分析 6

3.3数据增强处理 7

3.4数据集平衡处理

3.5求出图像均值和方差

3.6数据预处理结果

[四、图像分类算法 8](#_Toc384051568)

4.1 SVM(支持向量机)

4.1.1 SVM的实现原理

4.2 KNN(K最近邻)

4.2.1 KNN的实现原理

4.2.2 KNN的具体实现步骤

4.3 决策树模型

4.3.1 决策树的实现原理

[五、垃圾图像分类算法的实现与结果分析 9](#_Toc384051569)

5.1模型实现步骤

5.2结果分析 [1](#_Toc384051568)0

[六、结论 1](#_Toc384051568)2

# 1概要

在科技发达、智能时代中，深度学习、机器学习以及人工智能成为了高频词。它们看似深不可测，但是又离不开我们的生活。深度学习和机器学习是一种技术、而人工智能一种是一种体现。在这种人工智能急剧膨胀的形势下，深度学习与机器学习成了垃圾分类的主要推力。众所周知，垃圾是人类既厌恶又无法摆脱的物体，而垃圾则是铺天盖地层出叠见地出现在我们地视野中，解决垃圾问题给全球带了巨大的挑战。想要有效处理垃圾，垃圾分类是必然的结果，然而垃圾分类过程又是一件既繁琐又耗时的事，而且使用人工进行垃圾分类它不仅需要耗费大量的人工而且它还会大大降低准确率。这时人工智能、深度学习就起了重要的作用。

本文正是研究深度学习算法的垃圾分类图像识别。论述多种深度学习算法及网络结构的图像识别处理原理，分析深度学习在图像识别中的突出优势，并且提出垃圾分类在现实社会中面临的问题与挑战。在综合了解研究后，深入探讨使用深度学习算法的卷积神经网络，在大量的有效图像数据集的训练过程中是如何增加

种全新的隐藏层，并且使用这种增加卷积层的方法来得出更高层次的特征提

而让机器自动提取特征来实现图像的识别。

关键词： 深度学习 图像识别 垃圾分类 机器学习 人工智能

## 1.1研究背景和研究意义

### 1.1.1研究背景

随着社会经济的快速发展，人们的生产生活产生了大量的垃圾，垃圾分类已经成为环境保护的重要环节之一。然而，传统的垃圾分类方法往往依赖于人工操作，不仅效率低下，而且容易出错。因此，利用图像识别技术开发垃圾分类系统具有重要的实际意义。

1.1.2研究意义

提高垃圾分类效率：基于图像识别技术的垃圾分类系统能够自动识别垃圾类别，大大提高垃圾分类的效率，减少人工分拣的工作量。 提升垃圾分类准确率：通过综合应用多种分类算法，系统能够更准确地识别各种类型的垃圾，降低错误率。 促进环保意识：该系统不仅可以提供垃圾分类结果，还能为用户提供分类建议和环保知识，有助于提高公众的环保意识和参与度。 数据积累和优化：系统后台负责采集和整理垃圾图像数据集，通过数据预处理和特征提取，为后续的垃圾分类研究提供数据支持，有助于持续优化垃圾分类的准确性。 便捷性和普及性：基于Web平台的设计使得系统可以随时随地被访问和使用，增强了垃圾分类的普及性和便利性

# 2数据描述

## 2.1垃圾图像数据来源

本设计的主题是基于深度学习算法的垃圾分类图像识别，所以需要准备的原始数据是各个类型的垃圾图片，根据网上随机下载的垃圾图片，还有小部分垃圾图片数据是收集于自行拍照。把所有收集的数据集合在一起进行分类，主要分为四大类型，其中包括有：厨余垃圾、可回收物、其他垃圾、有害垃圾四大类，每一大类下又分若干类别，并且在本地建立以这四大类型为分类的垃圾图像文件。如下所示：



图2.1 垃圾图像数据文件

# 3数据预处理

## 3.1数据样本分析

分析数据集的样本数量和图像分辨率，并绘制相应的条形图和散点图。

首先遍历数据集中的所有文件，并将每个文件的尺寸（宽度和高度）存储在一个列表中。然后，打印出这个列表，以便查看数据集的样本数量和图像分辨率。

在主程序中，定义了数据集的根目录`dataset\_root\_path`。然后，调用`plot\_resolution`函数来绘制图像分辨率散点图，最后调用`plot\_bar`函数来绘制条形图，运行结果如图：

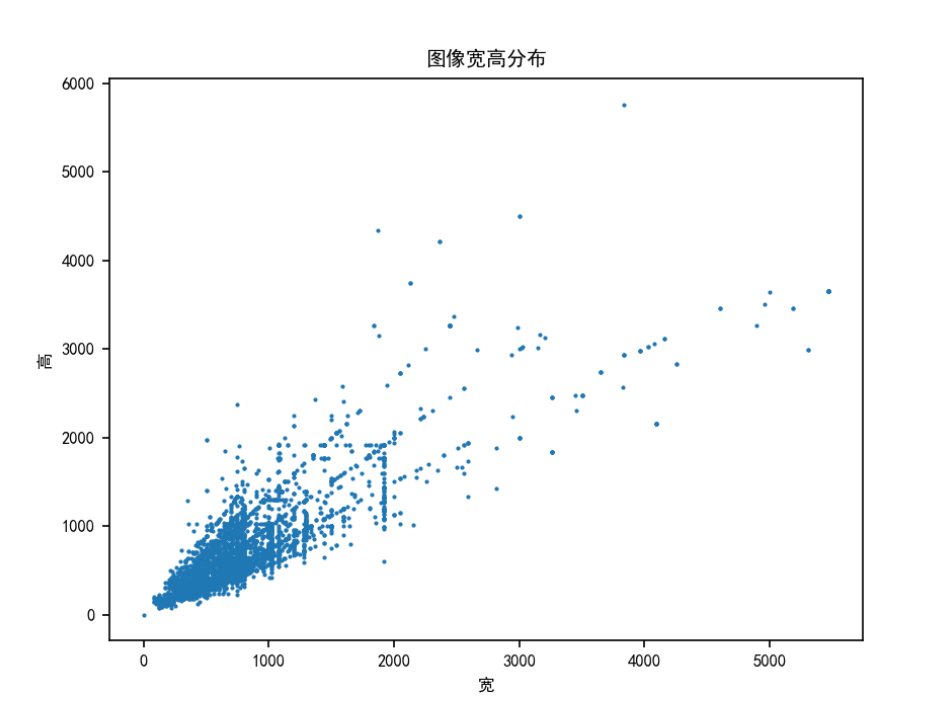


图3.1 原始数据样本数量和分辨率分布情况

由此看出，原始数据集的数据样本过于集中，不同数据之间差异过大，因此，需要对图片数据进行裁剪和处理。

## 3.2数据集长宽分析

首先，通过检查给定文件的长和宽比例和尺寸，将不满足的要求的的图像文件添加到列表delete\_list中，相关代码如图3.2所示：

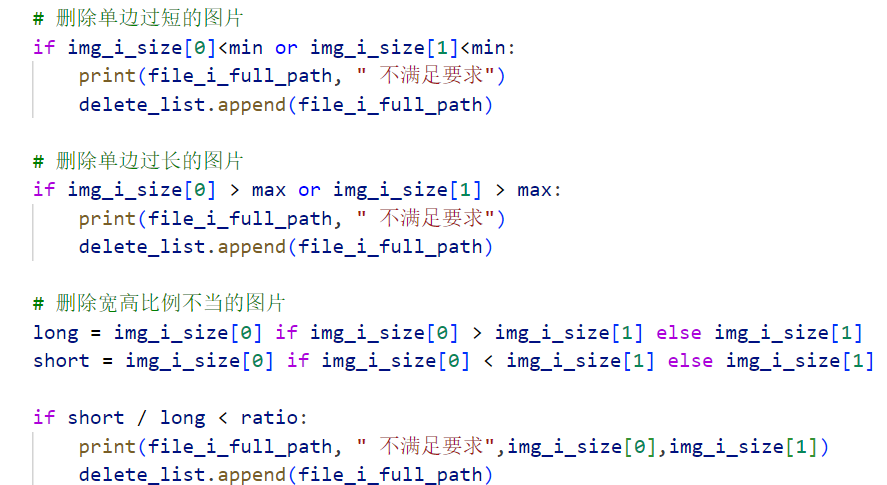


图3.2数据集长宽分析核心代码

## 3.3数据增强处理

使用旋转进行数据增强，通过翻转图像来创建新的训练数据。数据增强是一种提高模型泛化能力的技术，它可以提高模型的鲁棒性，使其在未见过的数据上也能表现良好。主要操作步骤如下：

  在主程序中，遍历指定目录下的所有文件。

对每个文件，首先将其保存到指定的保存目录中。

如果文件数量小于指定阈值，则对图像进行水平翻转和垂直翻转，并将翻转后的图像保存到指定的保存目录中。

如果文件数量等于或大于阈值，则不进行翻转操作。

通过数据增强后，每类数据量都达到了四五百张！

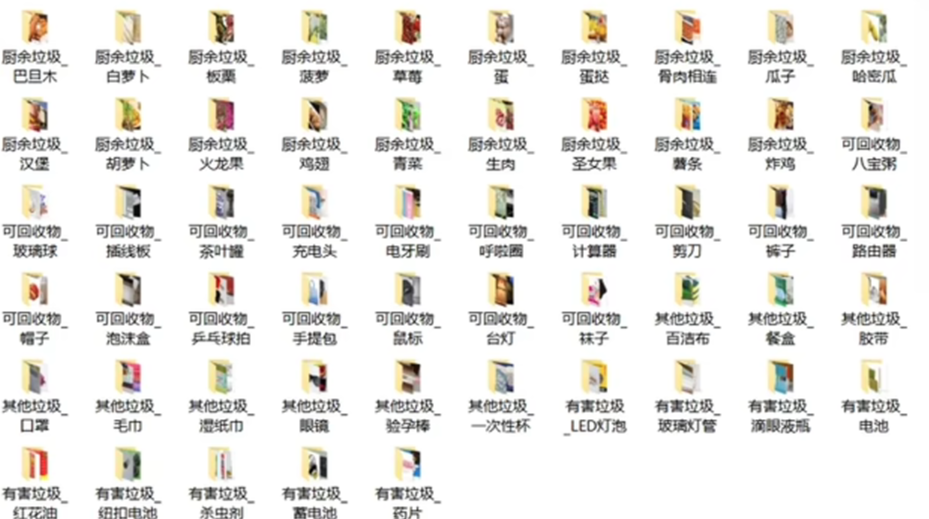


图3.3 图像文件类别图

## 3.4数据集平衡处理

接着对数据集进行数据平衡处理。遍历指定目录下的所有文件，如果文件数量超过阈值，则随机删除一些文件。

实现原理：使用os.walk()遍历指定目录下的所有文件，然后统计文件数量。如果文件数量超过阈值，则创建一个空列表，用于存放需要删除的文件路径。接着，遍历文件列表，将文件路径添加到delete\_list中。最后，对delete\_list进行随机排序，然后从delete\_list中取出超过阈值的文件路径，并逐个删除这些文件。可以用于清理大量文件的数据集，避免占用过多磁盘空间。结果如下：



图3.4 删除超过阈值的文件路径

## 3.5求出图像均值和方差

通过计算给定数据集（在本例中为ImageFolder数据集）的均值和标准差。数据集通常包含大量图像，计算这些统计数据可以帮助我们了解图像数据的分布，从而更好地处理和分析数据。求出数据集图像的均值和方差如图所示：

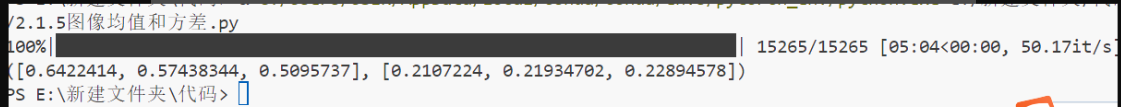


图3.5 图像均值和方差

## 3.6数据预处理结果

通过一系列的操作后，最终得到处理后的数据集样本数量和分布情况：

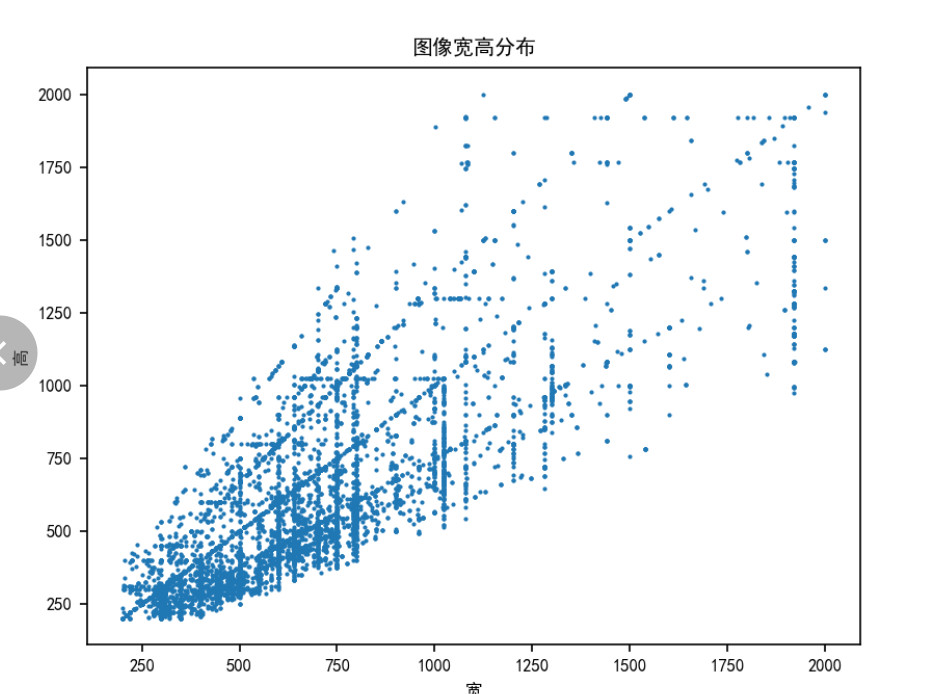


图 3.6 数据预处理后的样本数量和分辨率情况

# 4图像分类算法

## 4.1 SVM(支持向量机)

### 4.1.1 SVM的实现原理

SVM（支持向量机）是一种二分类模型，其基本原理是寻找一个超平面，使得两类数据之间的间隔（也称为边距）最大1。这个超平面被称为最大边距超平面，而最靠近这个超平面的那些数据点被称为支持向量。

SVM的核心思想是通过最大化两类数据之间的间隔来提升模型的泛化能力。在线性可分的情况下，SVM通过找到最大边距超平面实现分类。如果数据是线性不可分的，SVM引入了核技巧和软间隔最大化来解决这个问题。 核技巧是通过将数据从原始空间映射到一个更高维度的特征空间，使得数据在这个高维空间中线性可分。这样，即使原始数据在低维空间中是非线性的，也可以在高维空间中找到一个线性超平面来进行分类。

软间隔最大化则是允许一部分数据点可以处于边距内部或者甚至被错误分类，以实现在大多数数据点上的分类正确。这样做的目的是为了平衡分类的准确性和复杂性，避免过拟合。

总的来说，SVM是一种强大的分类方法，它既可以解决线性可分的问题，也可以解决非线性的问题，而且还可以用于多分类问题。

## 4.2 KNN

### 4.2.1 KNN的实现原理

KNN（K最近邻）算法是一种基本的分类和回归方法，它工作在特征空间上，能够为数据点分配类别标签或预测连续值。其核心思想是对于一个新的数据点，通过计算它与训练集中各个数据点的距离，找出距离它最近的k个数据点，然后根据这k个数据点的多数类别或平均值来预测新数据点的标签或值。

4.2.2KNN的具体实现步骤

具体实现步骤如下： 特征提取：首先，从数据集中提取出对分类有帮助的特征，并将其转换为数值形式，以便后续计算。

计算距离：选择一种距离度量方法（例如欧氏距离、曼哈顿距离等），计算新数据点与训练集中每个数据点之间的距离。

排序选取k个最近邻：按照距离的远近对所有训练数据进行排序，选择距离新数据点最近的k个数据点。

分类决策：对于分类问题，通常根据这k个最近邻的多数类别来确定新数据点的类别；对于回归问题，通常取这k个最近邻的平均值作为新数据点的预测值。

调整参数k：k是一个超参数，可以通过交叉验证等方法来选择最优的k值。

KNN算法的优点是简单直观，对于非线性数据也有很好的分类效果。缺点是对于高维数据，计算距离和寻找最近邻的复杂度较高，并且容易受到噪声数据的影响。

### 4.3决策树模型

### 4.3.1决策树的实现原理

决策树算法分类器的实现原理主要基于树的结构，其中每个节点代表一个特征或属性的测试，而每个分支则代表该测试的可能输出。叶节点则代表最终的类别或决策。

特征选择：决策树首先需要确定哪些特征或属性最适合作为分类的依据。这通常通过计算不同特征的信息增益或基尼指数来完成，目的是找到能够最好地将数据集划分为不同类别的特征。

树的生长：一旦选择了最佳特征，决策树就会根据这个特征的值将数据集划分为两个或更多的子集。然后，对于每个子集，算法会重复这个过程，直到达到预定的条件（例如，所有数据点都属于同一类别，或者没有更多的特征可以用于划分）。

树的剪枝：为了防止过拟合，可能需要对生成的决策树进行剪枝。这通常通过检查删除某些分支或节点后树的准确性是否下降来实现。

分类预测：当决策树建立完成后，可以使用它来对新的数据进行分类。这通常是通过从根节点开始，根据数据的特征值选择相应的分支，然后沿着树移动，直到达到一个叶节点，该叶节点代表的类别就是对新数据的预测。

这种方法的优势在于它的简单性和易于解释。然而，决策树也有其局限性，例如对于某些数据集可能容易产生过拟合，或者在处理具有大量特征的数据时效率较低。

# 5垃圾图像分类算法的实现与结果分析

## 5.1模型实现步骤

以下介绍垃圾图像分类算法的实现过程：以SVM为例

①对图像进行预处理。

②读取图片，并分别对图像进行灰度直方图特征提取和SIFT特征提取，并将特征提取处理后的图像拉伸为一维向量。

③建立训练集，将其代入 SVM 算法中进行训练生成一个模型，最后将测试集代入模型中进行分类预测。

④将分类结果与实际值进行对比，得到准确率、召回率等以及混淆矩阵，并对试验结果进行对比讨论。

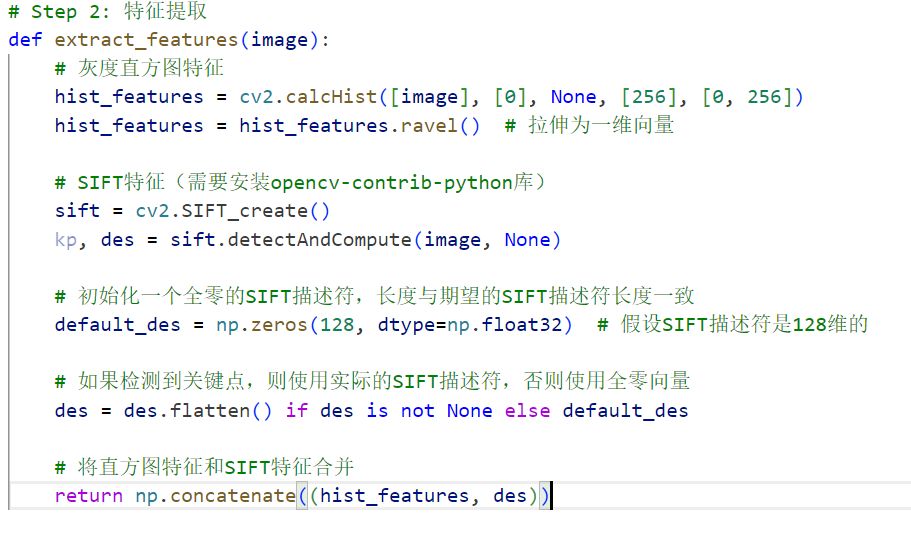


图5.1 特征提取核心代码图



图5.2模型训练核心代码图

## 5.2结果分析

使用SVM算法进行图像分类，得到以下结果：

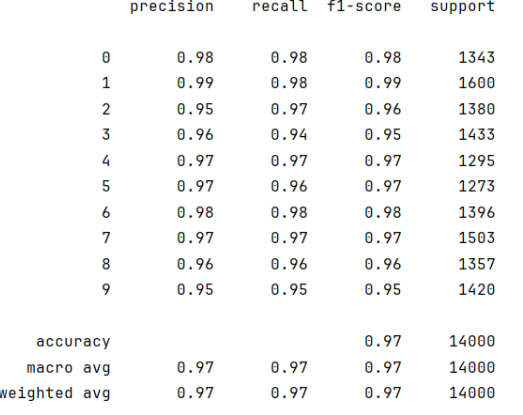


图5.2.1 SVM准确率、召回率、F1指标

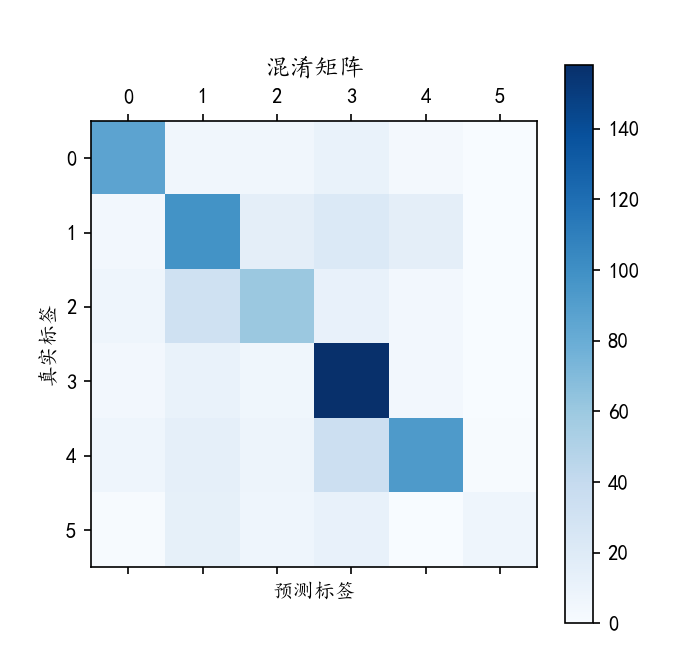


图5.2.2 混淆矩阵

这个混淆矩阵清晰地展示了分类模型在不同标签上的预测性能。以下是对该混淆矩阵的详细分析：

首先，从标签分布来看，矩阵的行代表实际标签，列代表预测标签，涵盖了从标签0到标签4的五个不同类别。

在模型性能方面，标签0和标签1的样本几乎都被正确分类，显示出模型在这两个类别上的出色表现。然而，标签2、标签3和标签4的样本存在不同程度的误分类情况。特别是标签4，其样本被误分类到其他标签的情况较多，表明模型在这个类别上的性能有待提升。

进一步观察误分类情况，我们发现标签4的样本主要被误分类为标签3、标签2和标签0，这提示我们可能需要针对这些类别之间的混淆进行更细致的分析和调整。同样，标签3和标签2的样本也存在一定的误分类现象，但相对较轻。

在颜色表示方面，矩阵中的颜色深浅反映了预测的置信度或概率。深蓝色表示高置信度的正确分类，而浅蓝色或白色则可能表示低置信度或错误分类。通过观察颜色的分布，我们可以进一步识别出哪些类别的预测较为稳定，哪些类别存在较大的不确定性。

接着使用决策树进行图像分类：

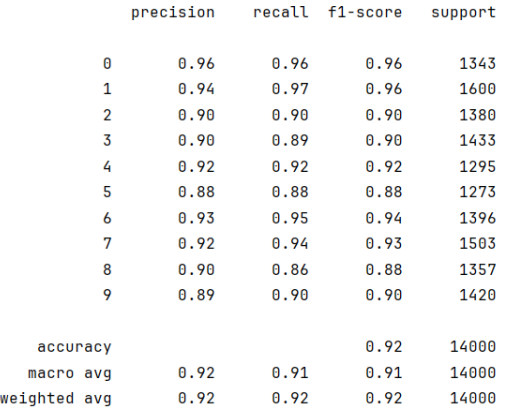


图5.2.3 决策树准确率、召回率、F1指标

使用KNN进行图像分类：

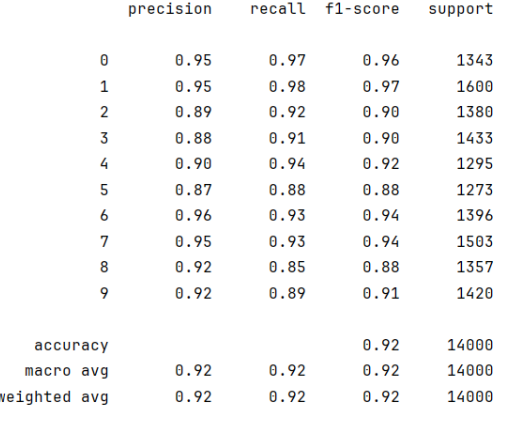


图5.2.4 KNN准确率、召回率、F1指标

对决策树和KNN算法的结果进行分析：从表格中的数据可以看出，不同类别的算法性能存在显著差异。

首先，精度方面，类别0和类别6的精度相对较高，而类别2和类别3的精度则较低。这表明在某些类别上，算法能够较为准确地识别出正例，而在其他类别上则存在较大的误差。

其次，召回率方面，类别3的召回率最高，而类别2和类别5的召回率则相对较低。召回率反映了算法能够正确识别出所有正例的能力，

最后，F1分数是精度和召回率的调和平均数，能够综合反映算法的性能。从表格中可以看出，类别0和类别4的F1分数相对较高，而类别2的F1分数最低。这进一步证实了类别2的算法性能相对较差。

六 结论

通过实验，对三款分类器的分类结果进行比较，得到以下结论：

在对图像分类任务中，我们比较了支持向量机（SVM）、K-最近邻（KNN）和决策树三种分类器的性能。以下是对这三款分类器在图像分类任务上的对比结论：

**支持向量机（SVM）**：

**优点**：SVM在图像分类任务中表现出了较高的准确性。由于其基于边缘最大化的原理，SVM在处理高维数据和非线性问题时具有较好的泛化能力。此外，SVM对于小样本数据也有较好的表现。

**缺点**：SVM对参数的选择较为敏感，如核函数的选择、惩罚系数等，这些参数的调整对分类性能有较大影响。同时，SVM在处理大规模数据集时训练时间较长，且内存消耗较大。

**K-最近邻（KNN）**：

**优点**：KNN算法简单直观，易于实现。在图像分类任务中，KNN能够取得一定的分类精度，特别是对于某些具有明显特征差异的图像类别。

**缺点**：KNN算法对k值的选择较为敏感，不同的k值可能导致分类结果发生显著变化。此外，KNN在处理高维数据时容易受到“维数灾难”的影响，导致分类性能下降。同时，KNN需要存储整个训练数据集，因此在处理大规模数据集时，存储和计算成本较高。

**决策树**：

**优点**：决策树算法易于理解和解释，其分类规则直观明了。在图像分类任务中，决策树能够快速地构建分类模型，并且对于某些具有明显特征差异的图像类别具有较好的分类效果。

**缺点**：决策树在处理连续属性或高维数据时可能变得复杂且容易过拟合。此外，决策树对噪声数据较为敏感，可能导致分类性能下降。为了提高决策树的泛化能力，通常需要进行剪枝操作或使用随机森林等集成学习方法。

综上所述，通过图片展示了算法评估结果的表格，通过对比不同类别的精度、召回率和F1分数，我们可以发现不同算法在不同类别上的性能存在显著差异。为了提高算法的整体性能，我们需要针对性能较差的类别进行进一步的优化和改进。由于分类类别较多，同时数据集的数量有限，各大分类器在分类的效果中并未体现出很高的精准率，决策树和KNN分类的准确率较低。三类算法中，决策树的速度最快。由数据的精度对比可知，在该项目中，SVM的表现最为出色。