基于朴素思想的岩石样本智能识别分类方法

姓名：欧祉辛

学号：20020153

单位：研究生院五大队二十五队

姓名：韩建臣

学号：20020083

单位：研究生院五大队二十五队

摘 要

生存类游戏在近年来的游戏市场中越来越受玩家欢迎，如开放世界类游戏《我的世界（Minecraft）》、末日求生类游戏《饥荒》等。本文通过对网易旗下生存类游戏《明日之后》的游戏设定进行分析和抽象建模，利用A\*（A-Star）算法，在给定的游戏场景中找到最优路线。

对问题一，。

对问题二，。

对问题三，。

最后对本文所建立的模型进行了讨论和分析，综合评价模型。

关键词：A\*（A-Star）算法

1.问题重述

1.1　问题背景

在油气勘探和矿产资源尤其是固体金属矿产资源勘探过程中，岩石样本的识别和分类对于地质分析尤为重要。人们通常利用复杂精密的仪器和设备进行现场采样，并根据岩石的颜色和结构来识别特定的岩石类型，划分不同岩石的范围。这需要大量的时间和精力。目前岩石样本识别的方法主要有重磁、测井、地震、遥感、电磁、地球化学、手标本及薄片分析方法等。

随着大数据的到来，利用机器学习对图像进行数据挖掘的方法建立岩石样本自动识别分类模型是一条新的途径。张野等（2018）[1]采用Inception-v3深度卷积神经网络模型进行花岗岩、千枚岩和角砾岩的学习和识别分类研究；Pengyu Wang et al.（2020）[2]基于TensorFlow建立了岩石图像分析的深度卷积神经网络模型，合并提取并压缩了近80000张岩石图像，最终训练集准确率为98%，测试集准确率为90%；等等。

现有样本数据是采用工业相机在录井现场对岩屑和岩心样本进行拍照，分别在暗箱内拍摄白光和荧光两种相片。白光下拍摄的照片主要用于提取颜色、纹理、粒度等特征来识别岩性，如图1-1所示；荧光下拍摄的照片主要用于识别含油气性，因为石油在紫外线照射下具有发光的特征，通常呈现绿色和黄色，如图1-2所示。



图 1-1　白光灯下拍摄的图片



图 1-2　荧光灯下拍摄的图片

1.2　问题重述

问题一：构建岩石样本岩性智能识别模型。请设计合适的机器学习或深度学习算法，针对数据集rock实现岩石样本岩性智能识别与分类。在数据集rock中，“白光/荧光”标签为“1”的数据是相同白光环境下拍摄的岩石样本图像数据，“白光/荧光”标签为“2”的数据是相同荧光环境下拍摄的岩石样本图像数据。

问题二：计算岩石含油面积百分含量。石油在紫外线照射下具有发光特征，即荧光灯下拍摄的照片中绿色或黄色的部分是含油的。请设计合适的算法计算岩石的含油面积百分含量，即绿色和黄色部分的面积占总岩石面积的百分比。

1.3　数据说明

数据共由两部分组成，分别为数据集rock（见1.3.1）和标签表rock\_label.csv（见1.3.2）。

1.3.1　数据集rock

数据集中包含白光和荧光下拍摄的图像数据，图像名称为“X-Y.bmp”或“X-Y.jpg”。X表示样本编号，编号相同表示为同一个岩石样本拍摄的图像，其中编号在1～321之间的为“.bmp”格式的图像数据，编号在322～350之间的为“.jpg”格式的图像数据，如图1-3所示；Y表示拍摄图像时的光照环境，1表示白光，2表示荧光。

如：“247-1.bmp”表示编号为247的岩石样本在白光下拍摄的图像。

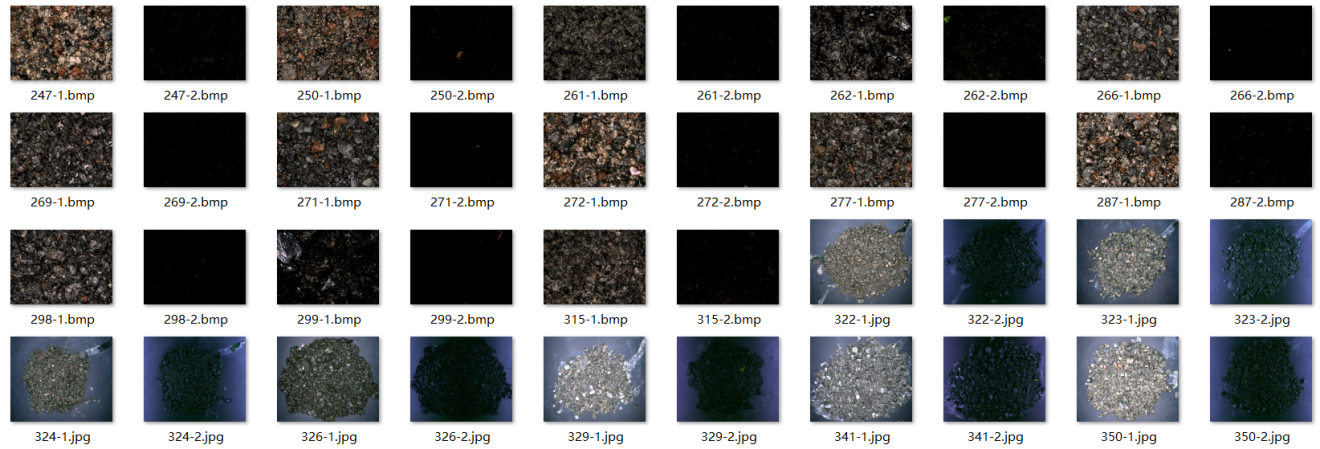


图 1-3　部分岩石样本图像数据集展示

1.3.2　标签表rock\_label.csv

此表格的内容为数据集rock中每个样本的岩性类别，如图1-4所示。标签表中总共涉及到7种岩性类别，分别为：深灰色泥岩、深灰色粉砂质泥岩、浅灰色细砂岩、灰色细砂岩、灰色泥质粉砂岩、灰黑色泥岩、黑色煤。

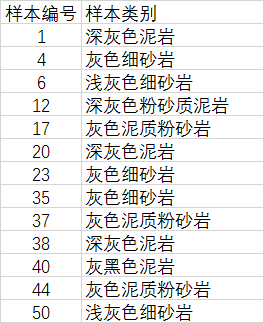


图 1-4　部分样本标签表展示

2.问题分析

2.1　问题一分析

对于问题一，需要在题目给定的白光灯下构建岩石样本岩性智能识别模型，针对数据集rock实现岩石样本岩性智能识别与分类。

从整体思路上分析，本问题建立每个岩石样本的信息库，并将每张岩石样本图片的信息库与岩性类别对应起来，构建出一个样本分类模型，实现岩石样本分类。结合数据集rock进行分析，样本具有数量小（仅70个样本）、种类复杂（多达7种岩性类别）的特点，而现有的许多深度学习、神经网络等分类模型若要达到较高的准确率，需要数以万计的训练样本，因此本问题在具体实现上需要考虑更简单朴素的建模方法。可以将问题抽象为：基于朴素贝叶斯分类的多分类模型。

Scikit-learn(sklearn)是机器学习中常用的第三方模块，对常用的机器学习方法进行了封装，包括回归(Regression)、降维(Dimensionality Reduction)、分类(Classfication)、聚类(Clustering)等方法。Sklearn中包含众多机器学习方法，首先引入需要训练的数据，Sklearn自带部分数据集，也可以通过相应方法进行构造，然后选择相应机器学习方法进行训练，模型训练完成之后便可预测新数据。

本问题中图像的像素颜色可认为是连续分布的，又考虑到自然界中普遍存在的正态分布现象，因此考虑使用sklearn.naive\_bayes机器学习库中的GaussianNB模型（即：高斯朴素贝叶斯分类模型）进行训练与分类。

2.2　问题二分析

对于问题二，需要在题目给定的荧光灯下拍摄的岩石样本图片的条件下，分析该岩石样本的含油量。

OpenCV（Open Source Computer Vision Library）是一个跨平台的计算机视觉库，由英特尔公司发起并参与开发，以BSD许可证授权发行，可以在商业和研究领域中免费使用。OpenCV可用于开发实时的图像处理、计算机视觉以及模式识别程序。该程序库也可以使用英特尔公司的IPP进行加速处理。Pillow是基于PIL（Python Imaging Library，一个强大且方便的Python图像处理库，曾经一度被认为是Python平台事实上的图像处理标准库，但Python 2.7以后不再支持该库）模块fork的一个派生分支，如今已经发展成为比PIL本身更具活力的图像处理库。

本问题需要利用OpenCV开源计算机视觉库和Pillow图像处理库对岩石样本图片进行像素块颜色提取，计算其中包含黄色、绿色像素块的比例，进而推算出岩石样本含油面积百分含量。

3.模型假设与约定

（1）假设不同种类的岩石样本之间在像素的RGB和HSV颜色直方图分布上有明显的差异；

（2）假设在荧光灯下拍摄的岩石样本图片中，除石油外，不存在其他物质会在紫外线照射时发出绿色光和黄色光；

（3）在荧光灯下拍摄的岩石样本图片中，参考HSV的表示方式[H: 色调，S: 饱和度，V: 明度]，将像素颜色筛选范围的上下限分别规定为：[26,43,46]、[77,255,255]。

4.符号说明及名词解释

4.1　符号说明

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | R、G、B三个通道在整张图像中分别对应的平均值 |
|  | Black、Gray、White三个颜色在整张图像中分别对应的像素数量 |
|  | Black、Gray、White三个颜色对应的像素数量在整张图像所有像素中所占百分比 |
|  | 图像特征，共6个，详见5.1.2节特征选取 |
|  | 分类标签，对应7种岩性类别，分别为：深灰色泥岩、深灰色粉砂质泥岩、浅灰色细砂岩、灰色细砂岩、灰色泥质粉砂岩、灰黑色泥岩、黑色煤 |
|  | 在类别为的样本中，特征的均值 |
|  | 在类别为的样本中，特征的标准差 |
| HSVi,j | 图像中第i行第j列像素块的像素值 |

4.2　名词解释

4.2.1　RGB

RGB就是常说的光学三原色，R代表Red（红色），G代表Green（绿色），B代表Blue（蓝色）。自然界中肉眼所能看到的任何色彩都可以由这三种色彩混合叠加而成（通过对红绿蓝三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色），因此也称为加色模式。RGB色彩模式是运用最广的颜色系统之一。RGB色相环如图4-1所示。

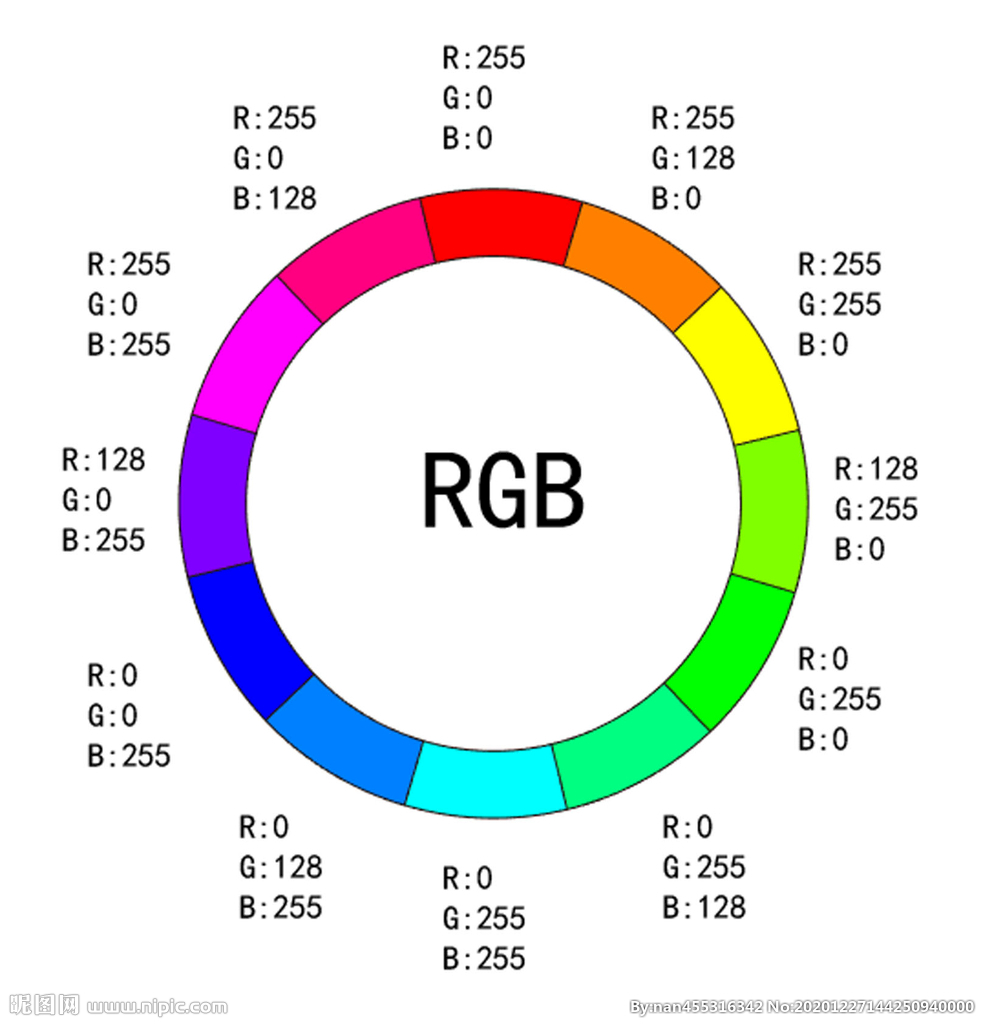


图 4-1　Value（明度）示意图

4.2.2　HSV

HSV(Hue色调, Saturation饱和度, Value明度)是根据颜色的直观特性由A. R. Smith在1978年创建的一种颜色空间, 也称六角锥体模型(Hexcone Model)。

色调H用角度度量，取值范围为0°～360°，从红色开始按逆时针方向计算，红色为0°，绿色为120°,蓝色为240°，如图4-2所示。

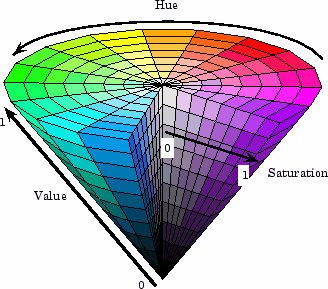


图 4-2　Hue（色调）示意图

饱和度S表示颜色接近光谱色的程度。一种颜色，可以看成是某种光谱色与白色混合的结果。其中光谱色所占的比例愈大，颜色接近光谱色的程度就愈高，颜色的饱和度也就愈高。饱和度高，颜色则深而艳。光谱色的白光成分为0，饱和度达到最高。通常取值范围为0%～100%，值越大，颜色越饱和。

明度V表示颜色明亮的程度。对于光源色，明度值与发光体的光亮度有关；对于物体色，此值和物体的透射比或反射比有关。通常取值范围为0%（黑）到100%（白），如图4-3所示。

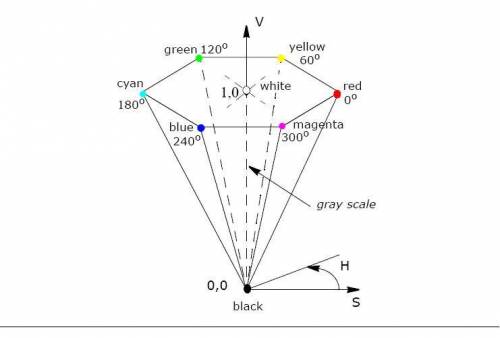


图 4-3　Value（明度）示意图

5.模型建立与求解

5.1　问题一建模与求解

5.1.1　图像预处理

首先，利用PIL库中的Image模块将图像的格式与大小进行处理。格式上，数据集rock中的图像格式有bmp和jpg两种，为方便程序复用与进一步数据处理，将其统一转存为png格式；大小上，可以利用Image的resize功能将图片的像素进行调整（在问题二中，我们使用了原始图片像素与压缩至10%的图片像素，对比分析了两种像素下的对岩石含油量判断的差异）。

随后，将图片的RGB与HSV信息读取出来，遍历所有像素，获得R、G、B三个通道分别在所有像素上的均值，以及HSV颜色空间中，颜色取值分别为黑、灰、白的像素占所有像素的百分比。图像经预处理后，总共获得上述六个维度的特征值。随后，利用sklearn.naive\_bayes中的GaussianNB模型进行建模。

5.1.2　模型建立

（1）特征选取

（2）目标函数：高斯朴素贝叶斯分类模型，是以高斯分布（即正态分布）作为启发式的朴素贝叶斯分类模型，其概率计算公式如下：

其中，特征对应feature1-6；类别对应7种岩性类别，分别为：深灰色泥岩、深灰色粉砂质泥岩、浅灰色细砂岩、灰色细砂岩、灰色泥质粉砂岩、灰黑色泥岩、黑色煤；为：在类别为的样本中，特征的均值；为：在类别为的样本中，特征的标准差。

5.1.3　问题求解

对于问题一，由于其涉及到的标签种类非常多，需要对每张图片进行预处理，得到统计学数据后再进行模型训练与分类。

（1）求解步骤

首先将图像的RGB与HSV颜色空间的信息提取出来，将每个岩石样本的图片像素信息压缩成一个6维数组，把得到的数组作为训练集的Feature数据，对应的岩石样本分类标签作为训练集的Label数据。使用“gaussianNB = GaussianNB()”构建一个高斯朴素贝叶斯分类模型，将Feature数据（即dataset\_X）与Label数据（即dataset\_Y）输入到模型中，使用“gaussianNB.fit(dataset\_X, dataset\_Y)”训练模型。训练完成后，可以使用“gaussianNB.predict(testset\_X)”对模型进行测试，观察给定的图像特征输入能否得到预期的标签分类结果。

（2）求解结果

依据高斯朴素贝叶斯分类模型和本题实际参数，我们首先对全局准确率进行了评估，得到全局准确率为51.428%。

为了进一步分析模型和实验结果，我们又对7种岩石样本分别进行了评估。当测试集为每种岩石样本的前两个时，得到分类结果如表­5-1所示，准确率为78.57%；当测试集为每种岩石样本的所有图片时，得到部分分类结果如表5-2所示，全部结果详见10.2.1,准确率分别为：深灰色泥岩 0.00%、深灰色粉砂质泥岩 66.67%、浅灰色细砂岩 57.89%、灰色细砂岩 100.00%、灰色泥质粉砂岩 77.78%、灰黑色泥岩 57.14%、黑色煤 80.00%。

表 5-1　测试集为每种岩石样本的前两个时所得分类结果和准确率

|  |  |  |
| --- | --- | --- |
| 样本编号 | 正确分类 | 预测分类 |
| 1 | 深灰色泥岩 | 黑色煤 |
| 20 | 深灰色泥岩 | 灰黑色泥岩 |
| 12 | 深灰色粉砂质泥岩 | 深灰色粉砂质泥岩 |
| 62 | 深灰色粉砂质泥岩 | 深灰色粉砂质泥岩 |
| 6 | 浅灰色细砂岩 | 浅灰色细砂岩 |
| 50 | 浅灰色细砂岩 | 浅灰色细砂岩 |
| 4 | 灰色细砂岩 | 灰色细砂岩 |
| 23 | 灰色细砂岩 | 灰色细砂岩 |
| 17 | 灰色泥质粉砂岩 | 灰色泥质粉砂岩 |
| 37 | 灰色泥质粉砂岩 | 灰色泥质粉砂岩 |
| 40 | 灰黑色泥岩 | 灰黑色泥岩 |
| 53 | 灰黑色泥岩 | 黑色煤 |
| 199 | 黑色煤 | 黑色煤 |
| 207 | 黑色煤 | 黑色煤 |
|  | 准确率 | 78.57% |

表 5-2　测试集为每种岩石样本的所有图片时所得部分分类结果和准确率

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 深灰色  粉砂质泥岩 | 12 | 深灰色粉砂质泥岩 |
| 62 | 深灰色粉砂质泥岩 |
| 87 | 深灰色粉砂质泥岩 |
| 137 | 灰色细砂岩 |
| 153 | 灰黑色泥岩 |
| 155 | 深灰色粉砂质泥岩 |
| 261 | 深灰色粉砂质泥岩 |
| 277 | 灰黑色泥岩 |
| 315 | 深灰色粉砂质泥岩 |
|  | 准确率 | 66.67% |

分析数据可发现，深灰色泥岩准确率为0.00%，其误判类别绝大多数为灰黑色泥岩。依据此情况，我们针对深灰色泥岩和灰黑色泥岩进行综合分析，依据高斯朴素贝叶斯分类模型，将训练集设置为深灰色泥岩和灰黑色泥岩，而后再对这两类岩石样本进行分类预测，结果如表5-3所示。

表 5-3　训练集和测试集均为深灰色泥岩与灰黑色泥岩时所得分类结果和准确率

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 深灰色  泥岩 | 1 | 深灰色泥岩 |
| 20 | 灰黑色泥岩 |
| 38 | 灰黑色泥岩 |
| 52 | 深灰色泥岩 |
| 89 | 灰黑色泥岩 |
| 99 | 灰黑色泥岩 |
| 103 | 灰黑色泥岩 |
| 126 | 灰黑色泥岩 |
| 146 | 灰黑色泥岩 |
| 154 | 灰黑色泥岩 |
| 223 | 灰黑色泥岩 |
| 269 | 灰黑色泥岩 |
| 271 | 灰黑色泥岩 |
| 298 | 灰黑色泥岩 |
| 322 | 深灰色泥岩 |
| 324 | 深灰色泥岩 |
| 341 | 深灰色泥岩 |
|  | 准确率 | 29.41% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 灰黑色  泥岩 | 40 | 灰黑色泥岩 |
| 53 | 深灰色泥岩 |
| 123 | 灰黑色泥岩 |
| 144 | 灰黑色泥岩 |
| 179 | 灰黑色泥岩 |
| 197 | 深灰色泥岩 |
| 204 | 灰黑色泥岩 |
|  | 准确率 | 71.43% |

由于这两类岩石样本在颜色上差距较小，对于高斯朴素贝叶斯分类模型而言，精度难以达到要求，很难对这两种岩石样本进行正确分类。

5.2　问题二建模与求解

5.2.1　二值化预处理

利用OpenCV开源计算机视觉库对原始数据集rock中的图片进行二值化预处理，提取图片中颜色为绿色和黄色的像素块，并将其转化为白色，以便后续计算岩石样本含油面积百分含量。

4.2.3一节中简单介绍了不同颜色HSV的取值范围。在OpenCV库中，不同颜色对应的像素值有所变化，如图5-1所示。

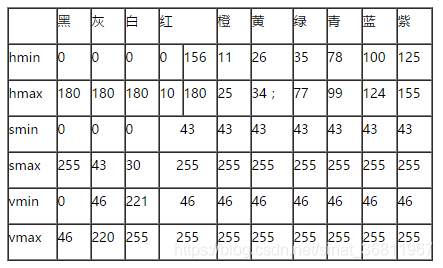


图 5-1　OpenCV中不同颜色对应HSV取值范围

原本输出的 HSV 的取值范围分别是 0-360，0-1，0-1；为了匹配目标数据类型， OpenCV 将每个通道的取值范围都做了修改，变为 0-180，0-255，0-255，这一转换在OpenCV官方文档中有所解释，如图5-2所示。由于在HSV对照表中黄色与绿色相邻，故本文在判断像素块颜色是否符合要求时所取像素值范围为[26,43,46]至[77,255,255]。

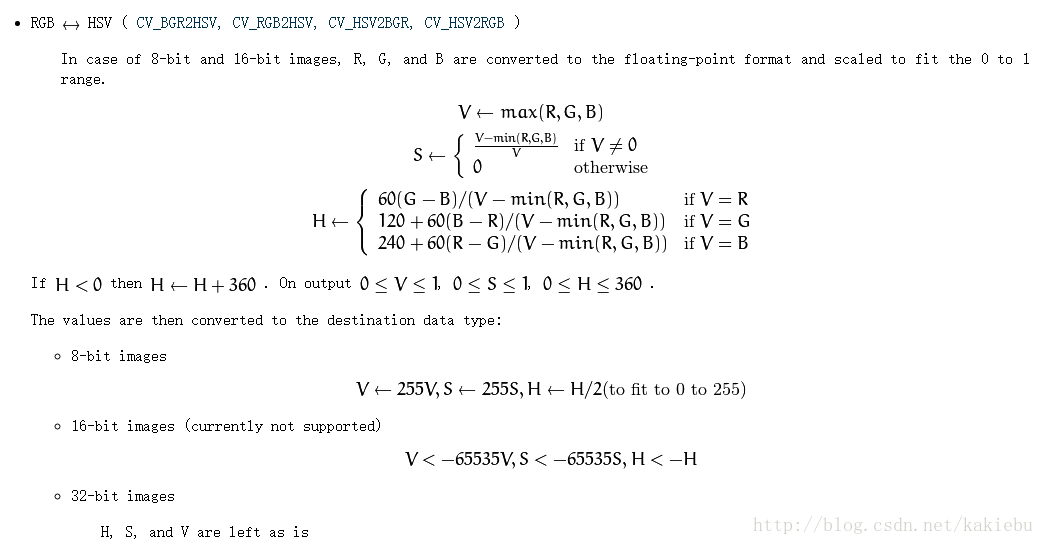


图 5-2　OpenCV官方文档解释

首先利用cvtColor()函数对图片进行颜色空间转换处理，而后利用inRange()函数对图片进行二值化处理。在Python中，inRange()函数共有三个参数，分别为src（需要处理的图片）、lowerb（HSV最小值）和upperb（HSV最大值）。其主要功能为：对于在两个阈值内的像素块（即像素值范围为[lowerb,upperb]）的像素值设置为白色，不在阈值区间内的像素值设置为黑色。二值化预处理结果如图5-3所示。

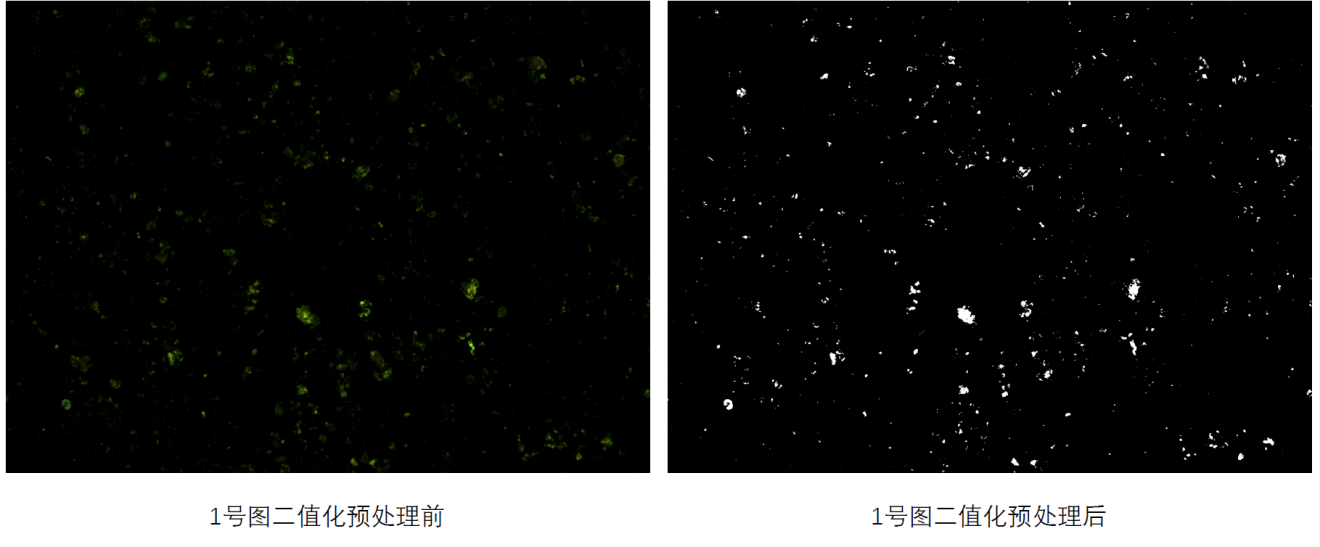


图 5-3　二值化预处理（以1号图为例）

5.2.2　模型建立

（1）决策变量：符合问题求解目标的有效像素块总数ValidPixelBlocks；所求图片像素块总数TotalPixelBlocks。

（2）目标函数：

（3）约束条件：根据问题求解目标，有效像素块的像素值范围和像素块总数需满足

5.2.3　问题求解

对于问题二，由于决策变量较少，根据所建立模型，可采用对每一像素块进行遍历的方法求解。

（1）求解步骤

针对本题所给数据集rock，首先对图片进行二值化预处理，将荧光灯下拍摄的含有多种颜色的照片转化为只有黑白两色，如图5-3所示；而后遍历图片中所有像素块，判断其颜色是否为白色：若是，则计数器+1；若不是，则计数器不变。再计算图片总像素块数量。所给数据集rock共有两种类型图片，包括.bmp格式和.jpg格式，分辨率分别为4096\*3000和2448\*2048。利用Python中numpy扩展程序库中shape()函数可读取图片的维数。shape(0)表示图片水平像素块数量，shape(1)表示图片垂直像素块数量，乘积即为总像素块数。最后，将计数器值与图片像素块总数相除，即可得岩石样本的含油面积百分含量。

（2）求解结果

依据模型计算后得出的部分结果如表5-4所示。全部结果详见附录10.4。

表 5-4　岩石样本含油面积百分含量（部分结果展示，保留小数点后5位）

|  |  |  |
| --- | --- | --- |
| 样本编号 | 样本类别 | 含油面积百分含量 |
| 1 | 深灰色泥岩 | 0.93809% |
| 4 | 灰色细砂岩 | 1.37692% |
| 6 | 浅灰色细砂岩 | 0.03149% |
| 12 | 深灰色粉砂质泥岩 | 0.11325% |
| 17 | 灰色泥质粉砂岩 | 0.02153% |
| 20 | 深灰色泥岩 | 0.02291% |
| 23 | 灰色细砂岩 | 0.12760% |
| 35 | 灰色细砂岩 | 0.94184% |
| 37 | 灰色泥质粉砂岩 | 0.53451% |
| 38 | 深灰色泥岩 | 0.03025% |

6.模型检验与误差分析

6.1　问题一模型检验和误差分析

6.1.1　模型检验

对于问题一，我们采用了高斯朴素贝叶斯分类模型。朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率，并且对小规模的数据表现很好，能个处理多分类任务，适合增量式训练。高斯朴素贝叶斯分类模型鲁棒性强，算法较为简单，对于缺失数据不太敏感，较为适合本题中小数据集分类任务。

6.1.2　误差分析

经分析，问题一的主要误差来源主要在以下两个方面：

（1）模型在训练数据采样上不够均衡，数据集rock的70个样本里，不同的岩性类别在样本数上有着明显差异，样本数最多达19个（浅灰色细砂岩），样本数最少仅4个（灰色细砂岩）。但是考虑到样本总数仅70个，再进行筛选会导致样本数量过少，因此没有对类别不均衡问题进行重采样。这在一定程度上会影响模型的分类效果。

（2）从图像的预处理来看，模型对图像特征提取的维度过于单一，仅考虑了RGB和HSV等颜色方面的特征，而没有加入纹理特征。事实上，对于岩石来说，纹理是用于区分岩性类别的一个非常重要的特征维度。忽视了这一部分的特征信息，在很大程度上降低了模型分类的准确性。

6.2　问题二模型检验和误差分析

6.2.1　模型检验

对于问题二，不存在可对比的岩石样本含油气百分比真实数据，故无法进行数据结果正确性的检验。

通过对图片进行压缩后分析数据可在一定程度上验证模型正确性。当将图片压缩至原分辨率的10%、50%、90%以及100%（不压缩）时，所得到岩石样本含油面积百分含量均不同。且对比所得数据后可发现，绝大多数情况下，压缩图片会导致得到的岩石含油面积百分含量降低。出现上述现象是因为对于油气含量本就较低的岩石样本，压缩图片后，稀疏的绿色和黄色像素块更易损失，导致岩石含油面积百分含量更有可能会降低。

综上所述，本文所建立的朴素岩石油气含量计算模型具有一定的鲁棒性，在不同条件下所得实验结果符合客观规律，即该模型具有合理性，在一定程度上是正确的。

6.2.2　误差分析

对于问题二所建立的朴素岩石油气含量计算模型，我们综合分析后认为，计算时主要误差可能在以下三个方面：

（1）对于荧光灯下拍摄的岩石样本照片，部分自身颜色与绿色和黄色相近的岩石块也被计算在岩石含油面积内，导致实验数据偏高；

（2）由于相机所拍摄的照片分辨率有限（所给数据集中最高的照片分辨率为4096\*3000），由6.2.1可知，计算所得岩石含油面积百分含量会存在一定误差；

（3）在数据集rock中，编号在322及以后的共7张图片为.jpg格式，其拍摄时光照条件与其他.bmp格式的图片不同，导致图片二值化预处理后存在一定误差，测得岩石样本含油面积百分含量普遍偏高，如图6-1所示。

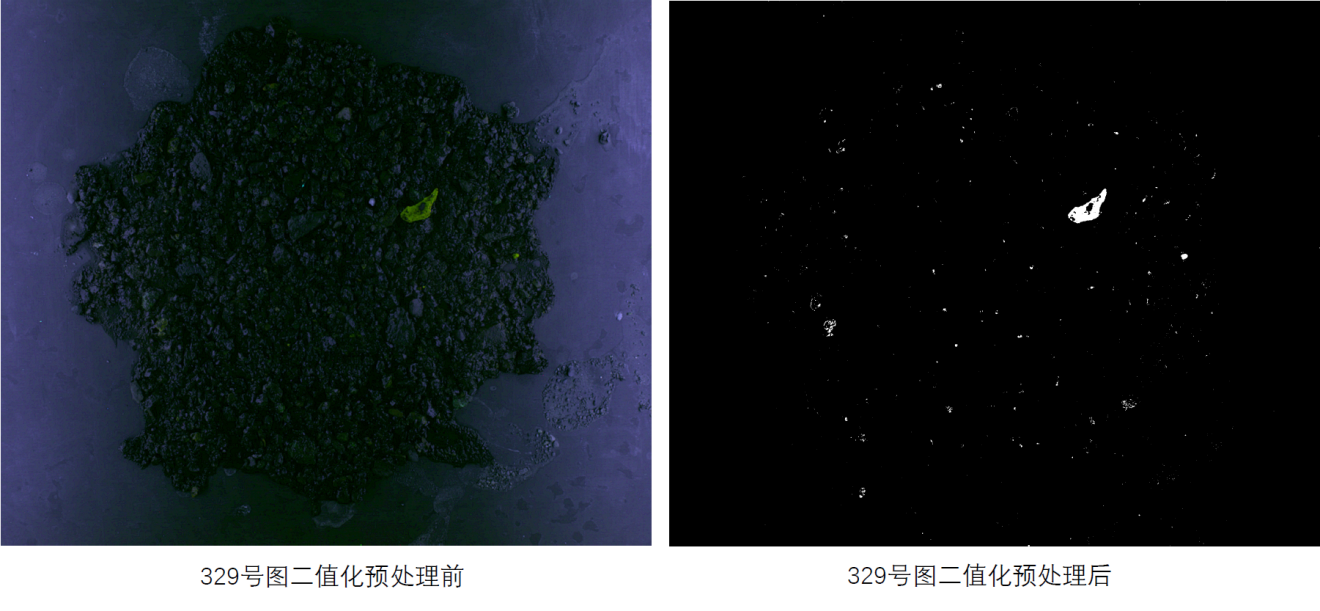


图 6-1　.jpg格式图片二值化预处理（以329号图为例）

7.模型评价

7.1　问题一模型评价

7.1.1　模型优点

考虑到本文问题中“小样本、多分类”的特点，为了尽可能利用小样本得到更好的分类准确性，模型使用了简洁朴素的贝叶斯分类方法。朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率，并且对小规模的数据表现很好，能处理多分类任务。一方面，在样本数量需求上，朴素贝叶斯方法要优于需要大量训练数据的深度学习模型，在每个分类平均只有10个样本的数据集上达到了比较好的效果。另一方面，朴素的贝叶斯方法在模型理解与结果解释上更有优势，从数据处理到模型建立，每个过程都是清晰简洁的，当出现分类误差比较大的现象时，能比较容易的找到可能导致模型偏差的因素，并对模型进行修正。

7.1.2　模型缺点

从训练数据采样上来看，模型并未解决不同类别岩石样本存在的类别不均衡问题。数据集rock的70个样本里，不同的岩性类别在样本数上有着明显差异。类别不均衡问题会影响到模型对训练集的拟合效果，因此在构建分类模型之前，需要对分类不均衡性问题进行处理。但是考虑到样本总数仅70个，再进行筛选会导致样本数量过少，因此没有对类别不均衡问题进行重采样。这在一定程度上会影响模型的分类效果。

从图像的预处理来看，模型对图像特征提取的维度过于单一，仅考虑了RGB和HSV等颜色方面的特征，而没有加入纹理特征。事实上，对于岩石来说，纹理是用于区分岩性类别的一个非常重要的特征维度。本模型为了简化训练集，仅对颜色特征进行了提取。对于纹理特征的考虑在第8节中有简单描述。

7.2　问题二模型评价

7.2.1　模型优点

本文所建立的朴素岩石油气含量计算模型，依据题目所给数据集中荧光灯下拍摄的岩石样本照片自身特征，遍历照片所有像素块，获取相应的像素值，再根据HSV颜色对照表进行颜色比对，判断像素块颜色是否为绿色或黄色，进而计算岩石含油面积百分含量。模型具有合理性和较强鲁棒性，建模思路清晰明了，可为其它模型提供数据支撑。

7.2.2　模型缺点

该模型主要缺点在于其时间复杂度和空间复杂度较大。对于一张分辨率为4096\*3000的.bmp格式图片而言，该模型会获取每个像素点的像素值，即约1200万个像素点的像素值，并以1\*3的矩阵结构存储到文件中。当图片数量较多时，完整运行该模型的开销可能难以接受。

另外，该模型对于拍摄照片时的光照和岩石样本自身的颜色都比较敏感，导致模型泛化性较弱。

8.模型推广与改进

对于岩性分类问题，可以在图像特征提取上增加对纹理的考虑，此处简单增加HOG特征的考虑，以岩石样本1-1为例，能得到图像的纹理图如图8-1所示：

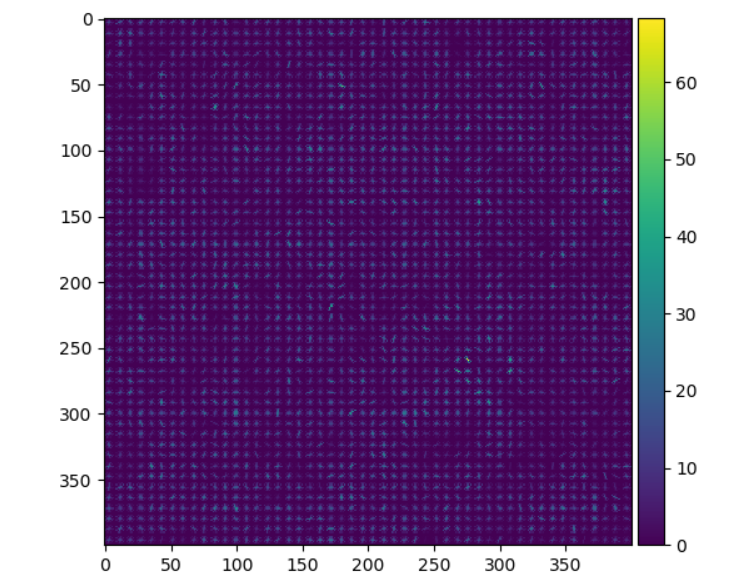


图 8-1　图像提取HOG纹理特征

对于朴素岩石油气含量计算模型，根据7.2.2，为了节约运行时间和空间，可以选择牺牲一定精度，将图片进行压缩，而后获取像素点HSV属性值，再对其进行分析处理。附录10.4.2给出了将图片压缩至原分辨率的10%后所得结果。另外，可引入机器学习方法，对比白光灯和荧光灯下所拍摄照片，减少因岩石样本自身颜色所带来的噪声干扰。

9.参考文献

[1]张野, 李明超, 韩帅. 2018. 基于岩石图像深度学习的岩性自动识别与分类方法. 岩石学报, 34(2): 333-342

[2] Wang, P., Wang, S., Zhu, C. et al. Classification and extent determination of rock slope using deep learning. Geomech. Geophys. Geo-energ. Geo-resour. 6, 33 (2020). https://doi-org-s.nudtproxy.yitlink.com/10.1007/s40948-020-00154-0

10.附录

10.1　问题一程序附录

|  |
| --- |
| 图像处理获取RGB、HSV统计学信息 |
| import os  import matplotlib.pyplot as plt  from PIL import Image  import cv2  import matplotlib.image as mpimg  import numpy as np  # 获取图片的路径  image\_directory = "../Photos/" #"E:/study/master/2021-Spring/Math-Modeling/Task1/Photos/"  image\_paths = os.listdir(image\_directory)  print(image\_paths)  image\_count = len(image\_paths)  RGB\_Ave\_and\_HSV\_Percent = []  for i in range(image\_count):  count = i # 所有照片：count = i | 白光照片：count = i\*2+2 | 荧光照片：count = i\*2+1  if count >= image\_count:  break  image\_path = image\_directory + image\_paths[count]  image\_number = image\_paths[count].split('.')[0]  print(image\_path)  #print(image\_number)  image = Image.open(image\_path)  width = image.size[0] # 获取宽度  height = image.size[1] # 获取高度  #image = image.resize((int(width\*0.1), int(height\*0.1)), Image.ANTIALIAS)  new\_image\_path = image\_directory + 'trans\_' + image\_paths[count] +'.png'  image.save(new\_image\_path)  # 提取图片中的所有像素信息  resized\_image = cv2.imread(new\_image\_path) # RGB  resized\_hsv = cv2.cvtColor(resized\_image, cv2.COLOR\_BGR2HSV) # HSV  width, height = resized\_image.shape[1::-1]  # print(resized\_image.shape) # 输出：height, width, 通道数  print(width, height)  image\_RGB = []  image\_HSV = []  RGB\_R = 0  RGB\_G = 0  RGB\_B = 0  HSV\_Black = 0  HSV\_Grey = 0  HSV\_White = 0  HSV\_Orange = 0  # HSV信息预处理  HSV\_Black\_High = np.array([180,255,46])  HSV\_Black\_Low = np.array([0,0,0])  mask\_HSV\_Black = cv2.inRange(resized\_hsv,lowerb=HSV\_Black\_Low,upperb=HSV\_Black\_High)  #print(mask\_HSV\_Black.shape)    HSV\_Grey\_High = np.array([180,43,220])  HSV\_Grey\_Low = np.array([0,0,46])  mask\_HSV\_Grey = cv2.inRange(resized\_hsv,lowerb=HSV\_Grey\_Low,upperb=HSV\_Grey\_High)    HSV\_White\_High = np.array([180,30,255])  HSV\_White\_Low = np.array([0,0,221])  mask\_HSV\_White = cv2.inRange(resized\_hsv,lowerb=HSV\_White\_Low,upperb=HSV\_White\_High)    HSV\_Orange\_High = np.array([0,43,46])  HSV\_Orange\_Low = np.array([34,255,255])  mask\_HSV\_Orange = cv2.inRange(resized\_hsv,lowerb=HSV\_Orange\_Low,upperb=HSV\_Orange\_High)    #print(width, height)  for y in range(height):  for x in range(width):  image\_RGB.append(resized\_image[y, x]) # array([2, 0, 2], dtype=uint8)  image\_HSV.append(resized\_hsv[y, x])  # RGB信息  image\_RGB\_R, image\_RGB\_G, image\_RGB\_B = resized\_image[y, x]  RGB\_R += image\_RGB\_R  RGB\_G += image\_RGB\_G  RGB\_B += image\_RGB\_B  if mask\_HSV\_Black[y, x] == 255:  HSV\_Black += 1  if mask\_HSV\_Grey[y, x] == 255:  HSV\_Grey += 1  if mask\_HSV\_White[y, x] == 255:  HSV\_White += 1  if mask\_HSV\_Orange[y, x] == 255:  HSV\_Orange += 1    RGB\_R\_Average = RGB\_R / (width\*height)  RGB\_G\_Average = RGB\_G / (width\*height)  RGB\_B\_Average = RGB\_B / (width\*height)  HSV\_Black\_Per = HSV\_Black / (width\*height)  HSV\_Grey\_Per = HSV\_Grey / (width\*height)  HSV\_White\_Per = HSV\_White / (width\*height)  HSV\_Orange\_Per = HSV\_Orange / (width\*height)  RGB\_Ave\_and\_HSV\_Percent.append([image\_number, RGB\_R\_Average, RGB\_G\_Average, RGB\_B\_Average, HSV\_Black\_Per, HSV\_Grey\_Per, HSV\_White\_Per, HSV\_Orange\_Per])  #print(RGB\_Ave\_and\_HSV\_Percent)  #print(image\_RGB)  #print(image\_HSV)  """  # 存储图片的RGB信息  RGB\_txt\_name = '../Train\_Labels/pixels/'+image\_number+'\_RGB.txt'  RGB\_write\_try = open(RGB\_txt\_name, mode='w')  for i in range(len(image\_RGB)):  #print(image\_RGB[i])  s = str(image\_RGB[i]).replace('[','').replace(']','') #去除[]  s = s.replace("'",'').replace(',','') +'\n'  #print(s)  RGB\_write\_try.write(s)  RGB\_write\_try.close()  # 存储图片的HSV信息  HSV\_txt\_name = '../Train\_Labels/pixels/'+image\_number+'\_HSV.txt'  HSV\_write\_try = open(HSV\_txt\_name, mode='w')  for i in range(len(image\_HSV)):  #print(image\_HSV[i])  s = str(image\_HSV[i]).replace('[','').replace(']','') #去除[]  s = s.replace("'",'').replace(',','') +'\n'  #print(s)  HSV\_write\_try.write(s)  HSV\_write\_try.close()  """  # 存储图片的RGB、HSV统计方面的信息  Deal\_txt\_name = '../Train\_Labels/pixels/RGB\_Average\_and\_HSV\_Percent.txt'  Deal\_write\_try = open(Deal\_txt\_name, mode='w')  for i in range(len(RGB\_Ave\_and\_HSV\_Percent)):  #print(RGB\_Ave\_and\_HSV\_Percent[i])  s = str(RGB\_Ave\_and\_HSV\_Percent[i]).replace('[','').replace(']','') #去除[]  s = s.replace("'",'').replace(',','') +'\n'  #print(s)  Deal\_write\_try.write(s)  Deal\_write\_try.close() |

|  |
| --- |
| 建立高斯朴素贝叶斯模型并用于岩石分类 |
| import pandas as pd  import os  import numpy as np  import copy  #import sklearn.naive\_bayes.GaussianNB  from sklearn.naive\_bayes import GaussianNB  #from sklearn.naive\_bayes import GaussianNB  #打开文件  file\_path = '../Train\_Labels/' # 'E:/study/master/2021-Spring/Math-Modeling/Task1/Train\_Labels/'  file\_name = 'rock\_label.csv' # 预处理：改成英文版的csv文件了  train\_data = pd.read\_csv(file\_path+file\_name, encoding='GBK')  train\_data.iloc[:,0].astype(int)  #train\_data.iloc[:,0].astype(str)  train\_data['RGB\_R'] = None  train\_data['RGB\_G'] = None  train\_data['RGB\_B'] = None  train\_data['HSV\_Black'] = None  train\_data['HSV\_Grey'] = None  train\_data['HSV\_White'] = None  #train\_data['HSV\_Orange'] = None  #print(train\_data)  # 读RGB数组信息，得到contents  txt\_path = file\_path + 'RGB\_Average\_and\_HSV\_Percent.txt'  RGB\_read\_try = open(txt\_path, mode='r')  contents = RGB\_read\_try.readlines()  RGB\_read\_try.close()  col\_index\_RGB\_R = train\_data.columns.get\_loc('RGB\_R')  col\_index\_RGB\_G = train\_data.columns.get\_loc('RGB\_G')  col\_index\_RGB\_B = train\_data.columns.get\_loc('RGB\_B')  col\_index\_HSV\_Black = train\_data.columns.get\_loc('HSV\_Black')  col\_index\_HSV\_Grey = train\_data.columns.get\_loc('HSV\_Grey')  col\_index\_HSV\_White = train\_data.columns.get\_loc('HSV\_White')  #col\_index\_HSV\_Orange = train\_data.columns.get\_loc('HSV\_Orange')  for s in contents:  '''  s = s.replace('[','').replace(']','') #去除[]  s = s.replace(' ','').replace('\n','').replace('\'','')  s = s.split(',')  '''  s = s.split(' ')  image\_number, image\_type = s[0].split('-')  if image\_type == '2':  continue    #print(s)  row\_index = train\_data['Sample\_Number'].values.tolist().index(int(image\_number))  # 修改值为RGB、HSV的统计值  train\_data.iloc[row\_index, col\_index\_RGB\_R] = float(s[1])  train\_data.iloc[row\_index, col\_index\_RGB\_G] = float(s[2])  train\_data.iloc[row\_index, col\_index\_RGB\_B] = float(s[3])  train\_data.iloc[row\_index, col\_index\_HSV\_Black] = float(s[4])  train\_data.iloc[row\_index, col\_index\_HSV\_Grey] = float(s[5])  train\_data.iloc[row\_index, col\_index\_HSV\_White] = float(s[6])  #train\_data.iloc[row\_index, col\_index\_HSV\_Orange] = float(s[7])  #print(train\_data.iloc[row\_index, :])  #break  #print(train\_data)  #print('\*\*\*\*\*\*\*\*\*\* Train \*\*\*\*\*\*\*\*\*\*')  dataset = train\_data  dataset\_X = dataset.iloc[:,2:].values  #print(dataset\_X)  dataset\_Y = dataset.iloc[:,1].values  #print(dataset\_Y)  # 高斯贝叶斯分类方法  gaussianNB = GaussianNB()  gaussianNB.fit(dataset\_X,dataset\_Y)  #print('\*\*\*\*\*\*\*\*\*\* Train \*\*\*\*\*\*\*\*\*\*\n\n')  print('\n\*\*\*\*\*\*\*\*\*\* Test \*\*\*\*\*\*\*\*\*\*')  selection = 1  if selection == 0:  # 每种岩石选择前2个编号进行测试  test\_x1 = train\_data['Sample\_Number'].values.tolist().index(1)  test\_x2 = train\_data['Sample\_Number'].values.tolist().index(20)  test\_x3 = train\_data['Sample\_Number'].values.tolist().index(12)  test\_x4 = train\_data['Sample\_Number'].values.tolist().index(62)  test\_x5 = train\_data['Sample\_Number'].values.tolist().index(6)  test\_x6 = train\_data['Sample\_Number'].values.tolist().index(50)  test\_x7 = train\_data['Sample\_Number'].values.tolist().index(4)  test\_x8 = train\_data['Sample\_Number'].values.tolist().index(23)  test\_x9 = train\_data['Sample\_Number'].values.tolist().index(17)  test\_x10 = train\_data['Sample\_Number'].values.tolist().index(37)  test\_x11 = train\_data['Sample\_Number'].values.tolist().index(40)  test\_x12 = train\_data['Sample\_Number'].values.tolist().index(53)  test\_x13 = train\_data['Sample\_Number'].values.tolist().index(199)  test\_x14 = train\_data['Sample\_Number'].values.tolist().index(207)  testset\_X = dataset.iloc[[test\_x1,test\_x2,test\_x3,test\_x4,test\_x5,test\_x6,test\_x7,test\_x8,test\_x9,test\_x10,test\_x11,test\_x12,test\_x13,test\_x14],2:].values  #print(testset\_X)  testset\_Y = dataset.iloc[[test\_x1,test\_x2,test\_x3,test\_x4,test\_x5,test\_x6,test\_x7,test\_x8,test\_x9,test\_x10,test\_x11,test\_x12,test\_x13,test\_x14],1].values  print('correct: ',testset\_Y)  testset\_y = gaussianNB.predict(testset\_X)  print('predict: ',testset\_y)  accuracy = gaussianNB.score(testset\_X, testset\_Y)  print('accuracy: ',accuracy)  elif selection == 1:  # 给定某个标签输出预测结果，以及所有数据的整体准确率  accuracy = gaussianNB.score(dataset\_X, dataset\_Y)  print('accuracy: ',accuracy)  test\_sample\_number = 326  print('\ntest\_sample: ',test\_sample\_number)  test\_index = train\_data['Sample\_Number'].values.tolist().index(test\_sample\_number)  #print(test\_index)  test\_x = dataset.iloc[test\_index,2:].values  print('data: ',test\_x)  test\_y = dataset.iloc[test\_index,1]  print('correct: ',test\_y)  test\_y\_1 = gaussianNB.predict(test\_x.reshape(1,-1))  print('predict: ',test\_y\_1)  test\_y\_2 = gaussianNB.predict\_proba(test\_x.reshape(1,-1))  print('predict\_proba: ',test\_y\_2)  elif selection == 2:  # 输出所有'.jpg'格式的图片的准确率  jpg\_index = train\_data['Sample\_Number'].values.tolist().index(322)  testset\_X = dataset.iloc[jpg\_index:,2:].values  #print(testset\_X)  testset\_Y = dataset.iloc[jpg\_index:,1].values  print('correct: ',testset\_Y)  testset\_y = gaussianNB.predict(testset\_X)  print('predict: ',testset\_y)  accuracy = gaussianNB.score(testset\_X, testset\_Y)  print('accuracy: ',accuracy)  elif selection == 3:  # 输出每种岩石对应的所有图片的准确率  stone\_types = ['Dark\_Gray\_Mudstone', 'Dark\_Gray\_Silt\_Mudstone', 'Light\_Gray\_Packsand', 'Gray\_Packsand', 'Gray\_Pelitic\_Siltstone', 'Gray\_Black\_Mudstone', 'Black\_Coal']  for stone\_type in stone\_types:  #Dark\_Gray\_Mudstone-深灰色泥岩  stone\_type\_index = train\_data[(train\_data['Sample\_Category'].isin([stone\_type]))].index.tolist()  print('\nindex[+2]: ',stone\_type\_index)  testset\_X = dataset.iloc[stone\_type\_index,2:].values  #print(testset\_X)  testset\_Y = dataset.iloc[stone\_type\_index,1].values  #print('correct: ',testset\_Y)  print(stone\_type)  testset\_y = gaussianNB.predict(testset\_X)  print('predict: ',testset\_y)  accuracy = gaussianNB.score(testset\_X, testset\_Y)  print('accuracy: ',accuracy)    '''  print("Dark\_Gray\_Silt\_Mudstone-深灰色粉砂质泥岩\  Light\_Gray\_Packsand-浅灰色细砂岩\  Gray\_Packsand-灰色细砂岩\  Gray\_Pelitic\_Siltstone-灰色泥质粉砂岩\  Gray\_Black\_Mudstone-灰黑色泥岩\  Black\_Coal-黑色煤")  '''  print('\*\*\*\*\*\*\*\*\*\* Test \*\*\*\*\*\*\*\*\*\*') |

|  |
| --- |
| 图像处理提取HOG信息 |
| import os  import matplotlib.pyplot as plt  from PIL import Image  import cv2  import matplotlib.image as mpimg  import numpy as np  from skimage.feature import hog  from skimage import io  # 获取图片的路径  image\_directory = "../Photos/" #"E:/study/master/2021-Spring/Math-Modeling/Task1/Photos/"  image\_paths = os.listdir(image\_directory)  print(image\_paths)  image\_count = len(image\_paths)  for i in range(image\_count):  #print(i)  count = i\*2 # 白光照片：count = i\*2 | 荧光照片：count = i\*2+1  if count >= image\_count:  break  #print(count)  image\_path = image\_directory + image\_paths[count]  image\_number = image\_paths[count].split('.')[0]  print(image\_path)  #print(image\_number)  image = Image.open(image\_path)  width = image.size[0] # 获取宽度  height = image.size[1] # 获取高度  image = image.resize((400, 400), Image.ANTIALIAS) #image = image.resize((int(width\*0.1), int(height\*0.1)), Image.ANTIALIAS)  new\_image\_path = image\_directory + 'try' + image\_paths[count] +'.png'  image.save(new\_image\_path)  resized\_image = cv2.cvtColor(cv2.imread(new\_image\_path), cv2.COLOR\_BGR2GRAY)  #print(resized\_image.shape)  HOG, hog\_image = hog(resized\_image, orientations=9, pixels\_per\_cell=(8, 8), cells\_per\_block=(8, 8), block\_norm='L2-Hys',visualize=True)  #print(HOG)  HOG\_txt\_name = '../Train\_Labels/hog/'+image\_number+'\_HOG.txt'  HOG\_write\_try = open(HOG\_txt\_name, mode='w')  #HOG\_write\_try.write(HOG)  np.savetxt(HOG\_txt\_name,HOG,fmt='%.10f',delimiter=None)  HOG\_write\_try.close()  io.imshow(hog\_image)  io.show() |

10.2　问题一结果附录

10.2.1　每种岩石样本分别计算准确率

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 深灰色  泥岩 | 1 | 黑色煤 |
| 20 | 灰黑色泥岩 |
| 38 | 灰色泥质粉砂岩 |
| 52 | 黑色煤 |
| 89 | 灰黑色泥岩 |
| 99 | 灰黑色泥岩 |
| 103 | 灰黑色泥岩 |
| 126 | 灰黑色泥岩 |
| 146 | 灰黑色泥岩 |
| 154 | 灰黑色泥岩 |
| 223 | 深灰色粉砂质泥岩 |
| 269 | 灰黑色泥岩 |
| 271 | 灰黑色泥岩 |
| 298 | 灰黑色泥岩 |
| 322 | 浅灰色细砂岩 |
| 324 | 浅灰色细砂岩 |
| 341 | 浅灰色细砂岩 |
|  | 准确率 | 0.00% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 深灰色  粉砂质泥岩 | 12 | 深灰色粉砂质泥岩 |
| 62 | 深灰色粉砂质泥岩 |
| 87 | 深灰色粉砂质泥岩 |
| 137 | 灰色细砂岩 |
| 153 | 灰黑色泥岩 |
| 155 | 深灰色粉砂质泥岩 |
| 261 | 深灰色粉砂质泥岩 |
| 277 | 灰黑色泥岩 |
| 315 | 深灰色粉砂质泥岩 |
|  | 准确率 | 66.67% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 浅灰色  细砂岩 | 6 | 浅灰色细砂岩 |
| 50 | 浅灰色细砂岩 |
| 71 | 灰色泥质粉砂岩 |
| 100 | 灰色泥质粉砂岩 |
| 113 | 灰色泥质粉砂岩 |
| 124 | 浅灰色细砂岩 |
| 133 | 灰色泥质粉砂岩 |
| 143 | 灰色泥质粉砂岩 |
| 162 | 浅灰色细砂岩 |
| 169 | 灰色泥质粉砂岩 |
| 210 | 浅灰色细砂岩 |
| 226 | 灰色泥质粉砂岩 |
| 240 | 浅灰色细砂岩 |
| 247 | 浅灰色细砂岩 |
| 272 | 灰色泥质粉砂岩 |
| 287 | 浅灰色细砂岩 |
| 323 | 浅灰色细砂岩 |
| 329 | 浅灰色细砂岩 |
| 350 | 浅灰色细砂岩 |
|  | 准确率 | 57.89% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 灰色  细砂岩 | 4 | 灰色细砂岩 |
| 23 | 灰色细砂岩 |
| 35 | 灰色细砂岩 |
| 122 | 灰色细砂岩 |
|  | 准确率 | 100.00% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 灰色  泥质粉砂岩 | 17 | 灰色泥质粉砂岩 |
| 37 | 灰色泥质粉砂岩 |
| 44 | 灰色泥质粉砂岩 |
| 64 | 灰黑色泥岩 |
| 94 | 灰色泥质粉砂岩 |
| 189 | 灰色泥质粉砂岩 |
| 234 | 浅灰色细砂岩 |
| 250 | 灰色泥质粉砂岩 |
| 266 | 灰色泥质粉砂岩 |
|  | 准确率 | 77.78% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 灰黑色  泥岩 | 40 | 灰黑色泥岩 |
| 53 | 黑色煤 |
| 123 | 灰黑色泥岩 |
| 144 | 灰色泥质粉砂岩 |
| 179 | 灰黑色泥岩 |
| 197 | 浅灰色细砂岩 |
| 204 | 灰黑色泥岩 |
|  | 准确率 | 57.14% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 黑色  煤 | 199 | 黑色煤 |
| 207 | 黑色煤 |
| 262 | 黑色煤 |
| 299 | 黑色煤 |
| 326 | 深灰色泥岩 |
|  | 准确率 | 80.00% |

10.2.2　每种岩石样本取前两个进行测试，计算准确率

|  |  |  |
| --- | --- | --- |
| 样本编号 | 正确分类 | 预测分类 |
| 1 | 深灰色泥岩 | 黑色煤 |
| 20 | 深灰色泥岩 | 灰黑色泥岩 |
| 12 | 深灰色粉砂质泥岩 | 深灰色粉砂质泥岩 |
| 62 | 深灰色粉砂质泥岩 | 深灰色粉砂质泥岩 |
| 6 | 浅灰色细砂岩 | 浅灰色细砂岩 |
| 50 | 浅灰色细砂岩 | 浅灰色细砂岩 |
| 4 | 灰色细砂岩 | 灰色细砂岩 |
| 23 | 灰色细砂岩 | 灰色细砂岩 |
| 17 | 灰色泥质粉砂岩 | 灰色泥质粉砂岩 |
| 37 | 灰色泥质粉砂岩 | 灰色泥质粉砂岩 |
| 40 | 灰黑色泥岩 | 灰黑色泥岩 |
| 53 | 灰黑色泥岩 | 黑色煤 |
| 199 | 黑色煤 | 黑色煤 |
| 207 | 黑色煤 | 黑色煤 |
|  | 准确率 | 78.57% |

10.2.3　只对深灰色泥岩和灰黑色泥岩进行训练并测试，计算准确率

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 深灰色  泥岩 | 1 | 深灰色泥岩 |
| 20 | 灰黑色泥岩 |
| 38 | 灰黑色泥岩 |
| 52 | 深灰色泥岩 |
| 89 | 灰黑色泥岩 |
| 99 | 灰黑色泥岩 |
| 103 | 灰黑色泥岩 |
| 126 | 灰黑色泥岩 |
| 146 | 灰黑色泥岩 |
| 154 | 灰黑色泥岩 |
| 223 | 灰黑色泥岩 |
| 269 | 灰黑色泥岩 |
| 271 | 灰黑色泥岩 |
| 298 | 灰黑色泥岩 |
| 322 | 深灰色泥岩 |
| 324 | 深灰色泥岩 |
| 341 | 深灰色泥岩 |
|  | 准确率 | 29.41% |

|  |  |  |
| --- | --- | --- |
| 样本类别 | 样本编号 | 预测类别 |
| 灰黑色  泥岩 | 40 | 灰黑色泥岩 |
| 53 | 深灰色泥岩 |
| 123 | 灰黑色泥岩 |
| 144 | 灰黑色泥岩 |
| 179 | 灰黑色泥岩 |
| 197 | 深灰色泥岩 |
| 204 | 灰黑色泥岩 |
|  | 准确率 | 71.43% |

10.3　问题二程序附录

|  |
| --- |
| 图像处理及石油含量分析 |
| import os  import matplotlib.pyplot as plt  from PIL import Image  import cv2  import matplotlib.image as mpimg  import numpy as np  # 获取图片的路径  image\_directory = "../Photos/" #"E:/study/master/2021-Spring/Math-Modeling/Task1/Photos/"  image\_paths = os.listdir(image\_directory)  print(image\_paths)  image\_count = len(image\_paths)  for i in range(image\_count):  #print(i)  count = i\*2+1  if count >= image\_count:  break  #print(count)  image\_path = image\_directory + image\_paths[count]  print(image\_path)  # 将图片压缩，并转存为'.png'格式  image = Image.open(image\_path)  width = image.size[0] # 获取宽度  height = image.size[1] # 获取高度  image = image.resize((int(width\*1), int(height\*1)), Image.ANTIALIAS)  new\_image\_path = image\_directory + 'try' + image\_paths[count] +'.png'  image.save(new\_image\_path)  '''  # 显示图片  new\_image = mpimg.imread(new\_image\_path)  print(new\_image.shape)  plt.imshow(new\_image)  plt.axis('off') # 不显示坐标轴  plt.show()  '''    # 提取图片中的指定颜色  src = cv2.imread(new\_image\_path)  cv2.namedWindow("image", cv2.WINDOW\_AUTOSIZE)  cv2.imshow("image", src)  #cv2.waitKey(0)  # 提取图中的绿色部分，mask为白色  hsv = cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  low\_hsv = np.array([26,43,46])  high\_hsv = np.array([77,255,255])  mask = cv2.inRange(hsv,lowerb=low\_hsv,upperb=high\_hsv)  mask\_path = image\_directory + 'try' + image\_paths[count] +'\_deal.png'  cv2.imwrite(mask\_path,mask)  cv2.imshow("test",mask)  cv2.waitKey(0)  cv2.destroyAllWindows()  # 获取白色的像素数量、在整张图上的百分比  mask\_img = cv2.imread(mask\_path)  amount = 0  for x in range(mask\_img.shape[0]):  for y in range(mask\_img.shape[1]):  b, g, r = mask\_img[x, y]  if (b, g, r) == (255,255,255):  amount += 1  print(amount)  amount\_percent = amount / (mask\_img.shape[0] \* mask\_img.shape[1])  print(amount\_percent) |

10.4　问题二结果附录

10.4.1　实验结果（未压缩）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 样本编号 | 样本类别 | 含油面积百分含量 | 样本编号 | 样本类别 | 含油面积百分含量 |
| 1 | 深灰色泥岩 | 0.93809% | 154 | 深灰色泥岩 | 0.01909% |
| 4 | 灰色细砂岩 | 1.37692% | 155 | 深灰色粉砂质泥岩 | 1.04980% |
| 6 | 浅灰色细砂岩 | 0.03149% | 162 | 浅灰色细砂岩 | 0.02388% |
| 12 | 深灰色粉砂质泥岩 | 0.11325% | 169 | 浅灰色细砂岩 | 0.02896% |
| 17 | 灰色泥质粉砂岩 | 0.02153% | 179 | 灰黑色泥岩 | 0.03257% |
| 20 | 深灰色泥岩 | 0.02291% | 189 | 灰色泥质粉砂岩 | 0.07495% |
| 23 | 灰色细砂岩 | 0.12760% | 197 | 灰黑色泥岩 | 0.40933% |
| 35 | 灰色细砂岩 | 0.94184% | 199 | 黑色煤 | 0.15473% |
| 37 | 灰色泥质粉砂岩 | 0.53451% | 204 | 灰黑色泥岩 | 0.04961% |
| 38 | 深灰色泥岩 | 0.03025% | 207 | 黑色煤 | 0.02635% |
| 40 | 灰黑色泥岩 | 0.02684% | 210 | 浅灰色细砂岩 | 0.03182% |
| 44 | 灰色泥质粉砂岩 | 0.02458% | 223 | 深灰色泥岩 | 0.05597% |
| 50 | 浅灰色细砂岩 | 0.04607% | 226 | 浅灰色细砂岩 | 0.02594% |
| 52 | 深灰色泥岩 | 0.51042% | 234 | 灰色泥质粉砂岩 | 0.02271% |
| 53 | 灰黑色泥岩 | 0.99882% | 240 | 浅灰色细砂岩 | 0.03095% |
| 62 | 深灰色粉砂质泥岩 | 0.11632% | 247 | 浅灰色细砂岩 | 0.03801% |
| 64 | 灰色泥质粉砂岩 | 2.99692% | 250 | 灰色泥质粉砂岩 | 0.03276% |
| 71 | 浅灰色细砂岩 | 0.20356% | 261 | 深灰色粉砂质泥岩 | 0.03394% |
| 87 | 深灰色粉砂质泥岩 | 0.10243% | 262 | 黑色煤 | 0.55098% |
| 89 | 深灰色泥岩 | 0.01733% | 266 | 灰色泥质粉砂岩 | 0.06139% |
| 94 | 灰色泥质粉砂岩 | 0.02332% | 269 | 深灰色泥岩 | 0.02348% |
| 99 | 深灰色泥岩 | 0.02448% | 271 | 深灰色泥岩 | 0.03975% |
| 100 | 浅灰色细砂岩 | 0.03184% | 272 | 浅灰色细砂岩 | 0.06158% |
| 103 | 深灰色泥岩 | 0.02488% | 277 | 深灰色粉砂质泥岩 | 0.02521% |
| 113 | 浅灰色细砂岩 | 0.02514% | 287 | 浅灰色细砂岩 | 0.02712% |
| 122 | 灰色细砂岩 | 0.93714% | 298 | 深灰色泥岩 | 0.02872% |
| 123 | 灰黑色泥岩 | 0.13394% | 299 | 黑色煤 | 0.04215% |
| 124 | 浅灰色细砂岩 | 0.03477% | 315 | 深灰色粉砂质泥岩 | 0.09666% |
| 126 | 深灰色泥岩 | 0.03910% | 322 | 深灰色泥岩 | 0.26554% |
| 133 | 浅灰色细砂岩 | 0.47620% | 323 | 浅灰色细砂岩 | 0.77483% |
| 137 | 深灰色粉砂质泥岩 | 1.34727% | 324 | 深灰色泥岩 | 0.52245% |
| 143 | 浅灰色细砂岩 | 0.02630% | 326 | 黑色煤 | 0.01741% |
| 144 | 灰黑色泥岩 | 0.04819% | 329 | 浅灰色细砂岩 | 0.27015% |
| 146 | 深灰色泥岩 | 0.02192% | 341 | 深灰色泥岩 | 0.38293% |
| 153 | 深灰色粉砂质泥岩 | 0.03843% | 350 | 浅灰色细砂岩 | 1.34335% |

10.4.2　实验结果（图片分辨率压缩至10%）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 样本编号 | 样本类别 | 含油面积百分含量 | 样本编号 | 样本类别 | 含油面积百分含量 |
| 1 | 深灰色泥岩 | 0.80929% | 154 | 深灰色泥岩 | 0.00000% |
| 4 | 灰色细砂岩 | 1.20456% | 155 | 深灰色粉砂质泥岩 | 0.89242% |
| 6 | 浅灰色细砂岩 | 0.00570% | 162 | 浅灰色细砂岩 | 0.00000% |
| 12 | 深灰色粉砂质泥岩 | 0.06601% | 169 | 浅灰色细砂岩 | 0.00081% |
| 17 | 灰色泥质粉砂岩 | 0.00000% | 179 | 灰黑色泥岩 | 0.00489% |
| 20 | 深灰色泥岩 | 0.00000% | 189 | 灰色泥质粉砂岩 | 0.04971% |
| 23 | 灰色细砂岩 | 0.05053% | 197 | 灰黑色泥岩 | 0.37327% |
| 35 | 灰色细砂岩 | 0.74898% | 199 | 黑色煤 | 0.09861% |
| 37 | 灰色泥质粉砂岩 | 0.44417% | 204 | 灰黑色泥岩 | 0.01141% |
| 38 | 深灰色泥岩 | 0.01548% | 207 | 黑色煤 | 0.00244% |
| 40 | 灰黑色泥岩 | 0.00081% | 210 | 浅灰色细砂岩 | 0.01059% |
| 44 | 灰色泥质粉砂岩 | 0.00163% | 223 | 深灰色泥岩 | 0.03097% |
| 50 | 浅灰色细砂岩 | 0.02689% | 226 | 浅灰色细砂岩 | 0.00000% |
| 52 | 深灰色泥岩 | 0.42706% | 234 | 灰色泥质粉砂岩 | 0.00244% |
| 53 | 灰黑色泥岩 | 0.75061% | 240 | 浅灰色细砂岩 | 0.00652% |
| 62 | 深灰色粉砂质泥岩 | 0.06846% | 247 | 浅灰色细砂岩 | 0.00489% |
| 64 | 灰色泥质粉砂岩 | 2.77751% | 250 | 灰色泥质粉砂岩 | 0.00163% |
| 71 | 浅灰色细砂岩 | 0.17930% | 261 | 深灰色粉砂质泥岩 | 0.00652% |
| 87 | 深灰色粉砂质泥岩 | 0.05053% | 262 | 黑色煤 | 0.40668% |
| 89 | 深灰色泥岩 | 0.00000% | 266 | 灰色泥质粉砂岩 | 0.04156% |
| 94 | 灰色泥质粉砂岩 | 0.00000% | 269 | 深灰色泥岩 | 0.00000% |
| 99 | 深灰色泥岩 | 0.00163% | 271 | 深灰色泥岩 | 0.02363% |
| 100 | 浅灰色细砂岩 | 0.00000% | 272 | 浅灰色细砂岩 | 0.03178% |
| 103 | 深灰色泥岩 | 0.00000% | 277 | 深灰色粉砂质泥岩 | 0.00000% |
| 113 | 浅灰色细砂岩 | 0.00081% | 287 | 浅灰色细砂岩 | 0.00081% |
| 122 | 灰色细砂岩 | 0.76936% | 298 | 深灰色泥岩 | 0.00570% |
| 123 | 灰黑色泥岩 | 0.11002% | 299 | 黑色煤 | 0.00652% |
| 124 | 浅灰色细砂岩 | 0.00326% | 315 | 深灰色粉砂质泥岩 | 0.05216% |
| 126 | 深灰色泥岩 | 0.01793% | 322 | 深灰色泥岩 | 0.16675% |
| 133 | 浅灰色细砂岩 | 0.42461% | 323 | 浅灰色细砂岩 | 0.28528% |
| 137 | 深灰色粉砂质泥岩 | 1.24368% | 324 | 深灰色泥岩 | 0.36765% |
| 143 | 浅灰色细砂岩 | 0.00163% | 326 | 黑色煤 | 0.00402% |
| 144 | 灰黑色泥岩 | 0.01059% | 329 | 浅灰色细砂岩 | 0.21295% |
| 146 | 深灰色泥岩 | 0.00081% | 341 | 深灰色泥岩 | 0.26318% |
| 153 | 深灰色粉砂质泥岩 | 0.01548% | 350 | 浅灰色细砂岩 | 0.98843% |