

岩石样本智能识别

摘要:

本文在 python 环境下，利用深度学习 keras 框架构建卷积神经网络，并通过迁移学习构建多个分类模型对 315 张白光样本进行 7 类别分类，同时集成各个模型的学习效果，提高模型的泛化能力。使用亮度归一化技术调整显示 315 张荧光图片亮度，再将图片转化为 HSV 色彩空间，提取绿色和黄色亮度进行岩石含油量的计算。

关键词:

深度学习，卷积神经网络，迁移学习，集成学习，亮度归一化，HSV 色彩空间

目录

一、介绍	1
1.1 背景	1
1.2 拟解决问题	1
二、数据集的介绍以及处理	2
2.1 原始数据集	2
2.2 数据集的预处理	3
2.3 训练集，测试集，验证集的划分	5
三、数据增强	7
3.1 使用数据增强原因	7

3.2	数据增强的原理	7
3.3	数据增强的效果展示	8
四、	基于 keras 的模型构建	9
4.1	keras 网络里重要的层次结构	9
4.2	模型的建立	11
4.3	模型的准确率和损失	13
五、	迁移学习	14
5.1	使用迁移学习的原因	14
5.2	迁移学习的原理	14
5.3	现成网络介绍	14
5.4	迁移学习的流程	15
5.5	各种迁移模型的比较	15
六、	集成学习	19
6.1	集成学习的原理	19
6.2	使用集成学习的原因	20
6.3	集成学习的流程	20
七、	模型的评估	20
7.1	集成模型与单个模型的比较	20
7.2	集成模型中每种矿石的准确率	22
八、	模型的使用	22
8.1	无标签数据集的预测	22
8.2	图片处理原理	23

8.3 预测流程-----	23
九、矿石的含油量计算-----	25
9.1 荧光岩石图片的预处理-----	25
9.2 亮度归一化算法-----	25
9.3 对比度的调整-----	27
9.4 颜色识别的算法-----	28
9.5 含油量的计算-----	31
十、总结-----	31
十一、附录-----	32
十二、参考文献-----	32

一、介绍

1.1 背景

对于油气勘探来说，岩石的样本识别是一项必不可少的环节，既基础又相当重要。与此同时，在矿产资源勘探中，特别是在固体金属矿产资源勘探中，岩石的样本识别同样发挥着至关重要的作用。不仅如此，对于地质分析这个领域来说，岩石样本的识别与分类的作用也不可估量。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。而现在计算机视觉已经逐渐兴起，采用图像深度学习的方法构建岩石样本自动识别的分类模型是一条值得尝试的新途径，且具有非常之高的可行性。

1.2 拟解决问题

- 1) 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。
- 2) 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，即荧光灯下拍摄的相片中绿色或黄色部分是含油的，请设计合适的算法计算岩石的含油面积百分含量（绿色和黄色部分的面积占总岩石面积的百分比）。

二、数据集的介绍以及处理

2.1 原始数据集

原始的数据集 rock 包括 730 张岩石样本的图片，其中 315 为岩石的白光图片，用于提取颜色、纹理、粒度等特征识别岩性，标签为 1；剩余 315 张为岩石的荧光图片，用于识别含油气性，因为石油在紫外线照射下具有的发光特征，其中的绿色和黄色部分是含油的部分。从 rock_label.csv 中可以得出数据集 rock 中一共包括七种岩石，分别是浅灰色细砂岩，深灰色粉砂质泥岩，黑色煤，灰色泥质粉砂岩，灰色细砂岩，深灰色泥岩，灰黑色泥岩，其中各种岩石的数量如下图所示：

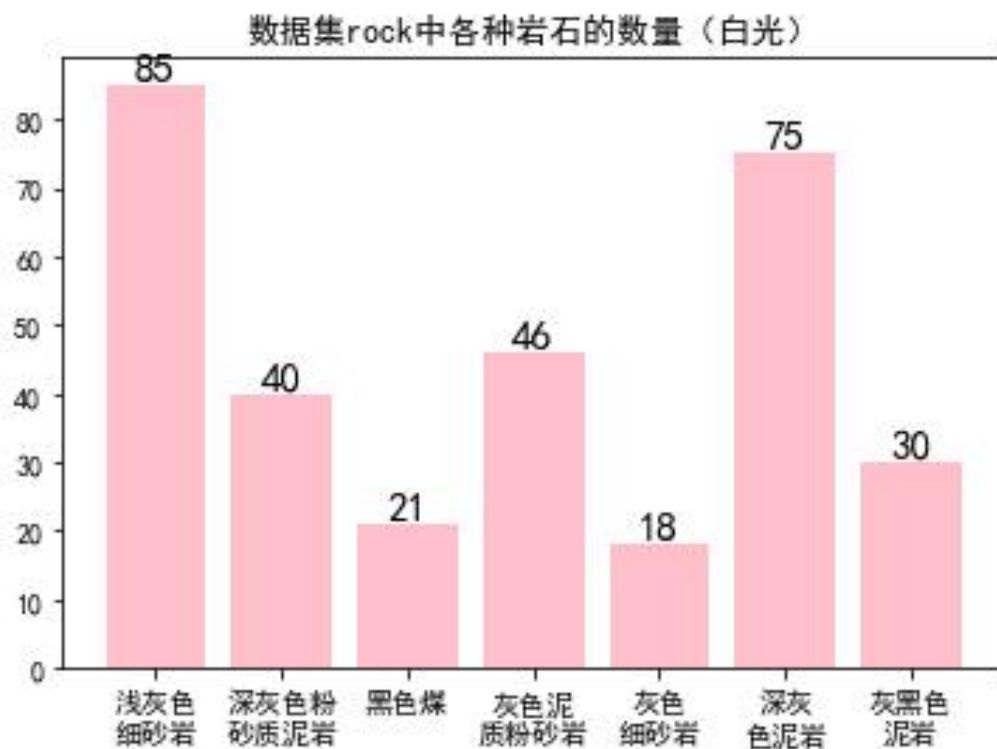


图 2-1-1 各类岩石的数量

2.2 数据集的预处理

由于原始数据集 rock 数据量较少，直接用于训练模型容易过拟合，且 rock 数据集的图片像素较大（ 4096×3000 或 2448×2048 ），可能会导致运行速度慢，效率较低且对机器内存有一定要求。因此我们对这些图片进行分割处理，以达到扩充样本和对图片下采样的目的。图片切割的策略为：将每张图片分为不重复的 12 张像素较小的子图片。在分类任务下，只需要对白光下的岩石图片进行分割，如下图所示，荧光下的岩石图片的预处理将在第 8 讲再介绍。

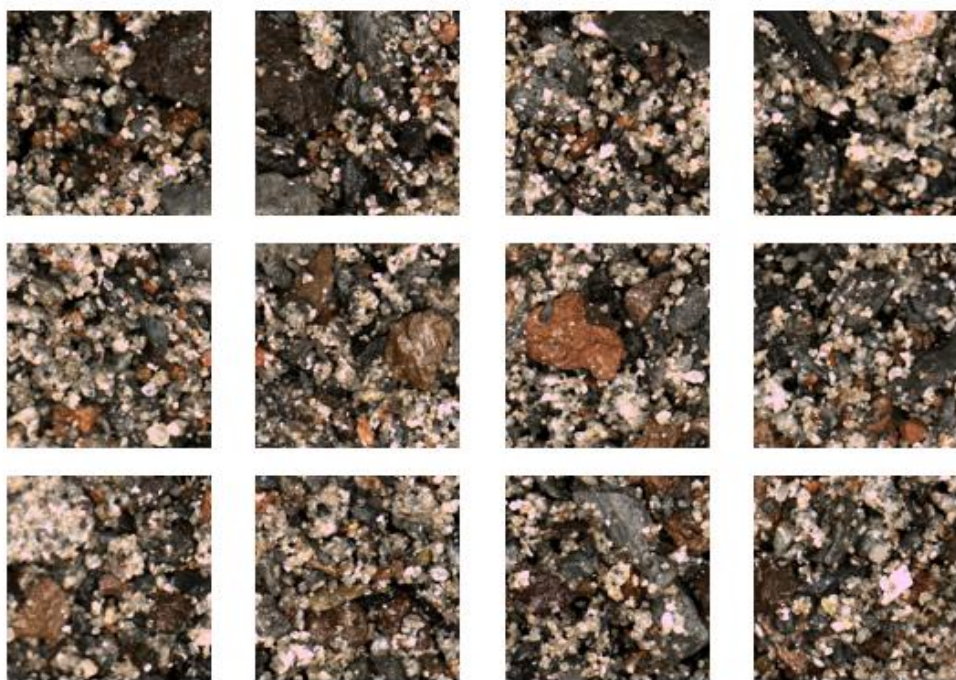


图 2-2-1 白光图像的分割

图像分割完之后我们通过 `rock_label.csv` 给每张图片打上标签，其中每种岩石对应的标签如下图：

岩石种类	标签
浅灰色细砂岩	0
深灰色粉砂质泥岩	1
黑色煤	2
灰色泥质粉砂岩	3
灰色细砂岩	4
深灰色泥岩	5
灰黑色泥岩	6

接下来对这些数据作进一步的处理，由于 `cnn` 模型对数值小的数处理得比较好，因此在读取图片后，对像素除以 255 来进行归一化，然后

转换其数据格式为 float 型以便计算。最后将标签进行 one-hot 化

2.3 训练集，测试集，验证集的划分

数据集 rock 原本含有 315 张岩石的白光图片，经上述方法切割后图片扩充到 3778 张。由于切割方法保证了图片没有重复的部分，我们以 8 : 1 : 1 的比例对训练集，测试集以及验证集进行划分，确保测试集和验证集不会参与到模型训练中。下面的图片分别为扩充后的数据集，训练集，测试集以及验证集中各种岩石的数量示意图：

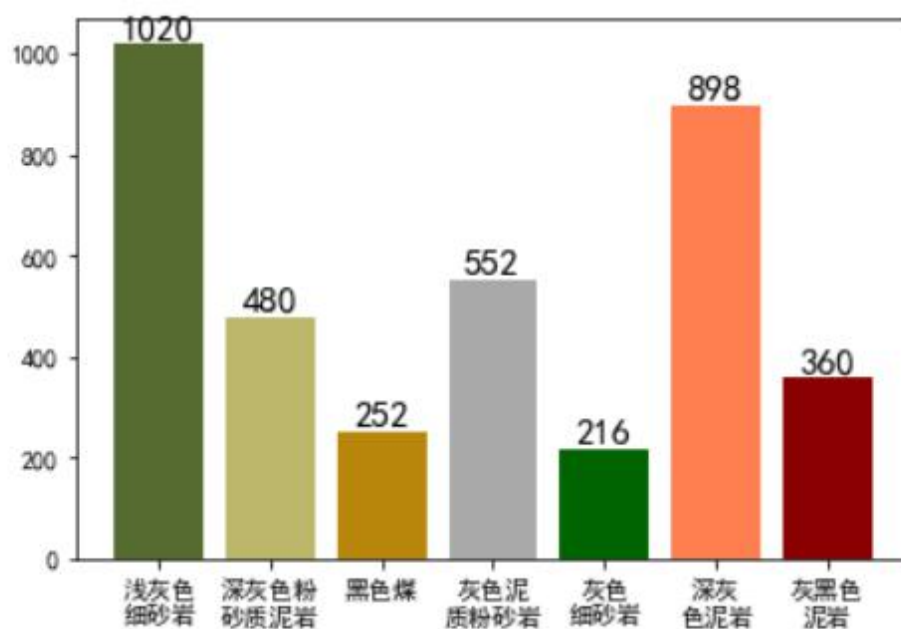


图 2-3-1 扩充后数据集

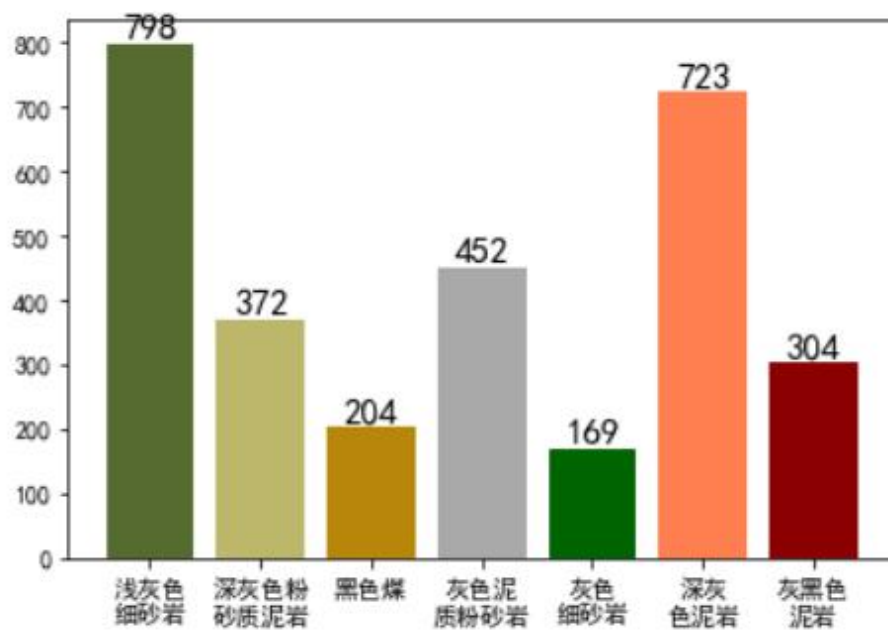


图 2-3-2 训练集

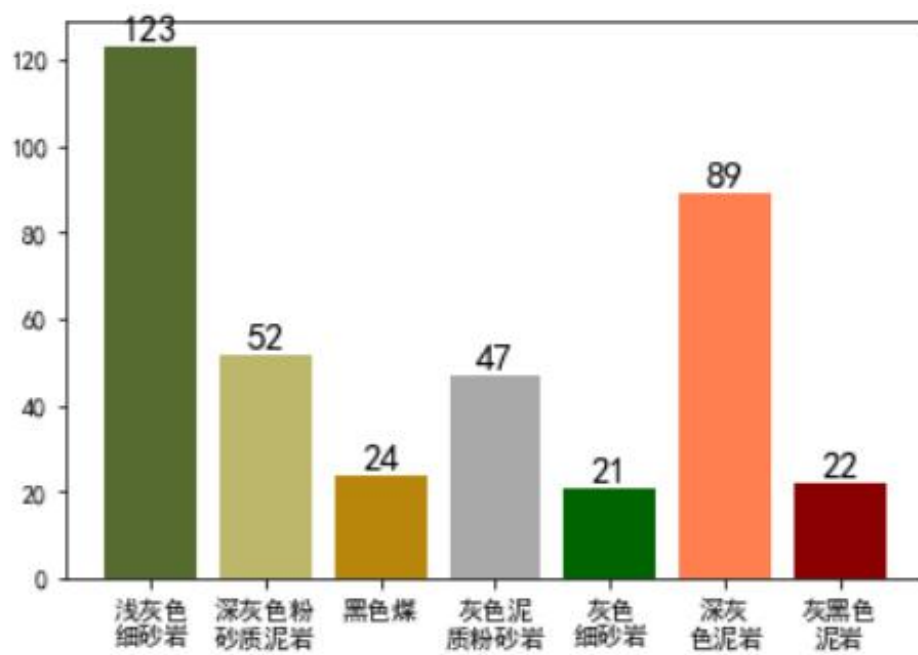


图 2-3-3 测试集

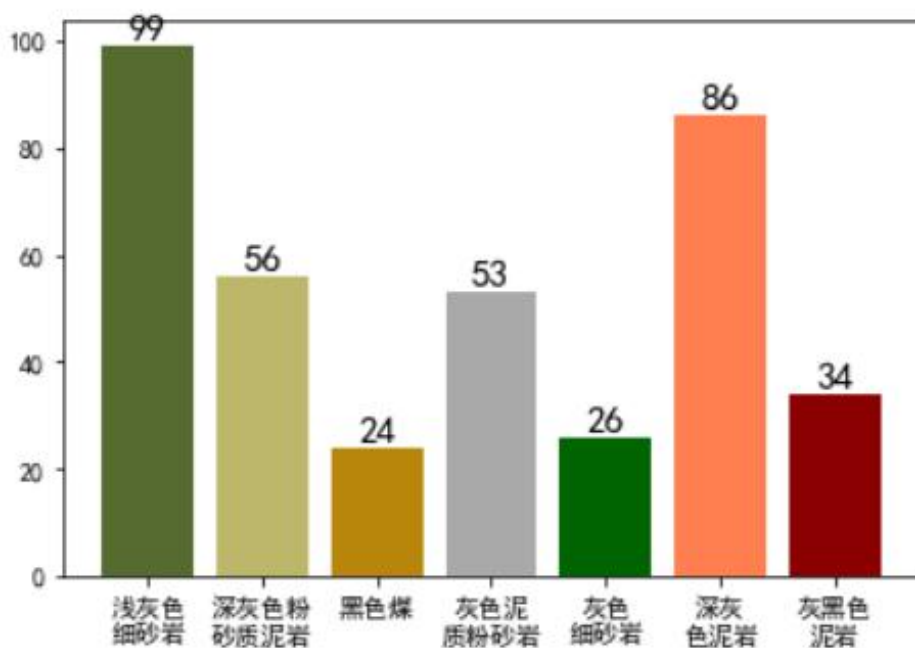


图 2-3-4 验证集

三、数据增强

3.1 使用数据增强的原因

数据增强即图像增强技术，即对图片进行各种操作处理，从而生成数倍于原有图片的图片数据集，可以解决原始图像数据集较少的问题，进一步将数据量扩充。同时，这些增强之后的数据集可以有效地对抗过拟合的问题，让构建出来的模型更加理想。比如对于一块岩石来说，从某个角度拍摄出来的图片可以成功识别出来，但换一个角度拍摄，或拍摄时的光线变暗，都可能导致识别错误，因而数据增强技术就体现出了它的价值和意义，它可以有效提高模型的泛化能力。

3.2 数据增强的原理

数据增强包括对原始图片进行随机翻转，水平平移，竖直平移，随机错切变换，水平翻转，随机放大等操作。其中 python 的深度学习框架 keras 中提供接口 ImageDataGenerator 可以进行数据增强的操作。

该接口里面的 flow 函数接收 numpy 数组和标签为参数, 生成经过数据提升或标准化后的 batch 数据, 并在一个无限循环中不断的返回 batch 数据, 可以一边训练模型一边进行数据增强

3.3 数据增强的效果展示

我们只对训练集进行数据增强处理, 验证集和测试集不作任何操作, 下面几张图为训练集中数据增强的效果图:

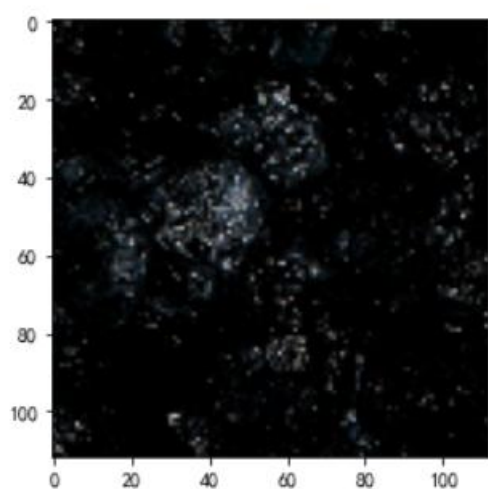


图 3-3-1 数据增强前图片

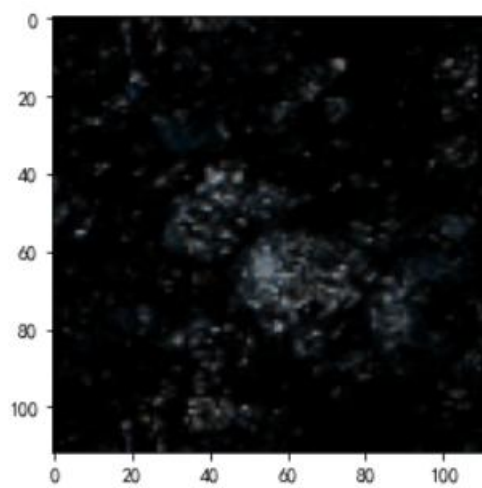
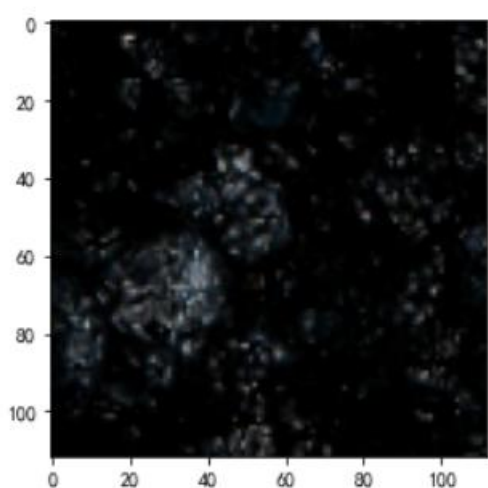


图 3-3-2, 图 3-3-3 数据增强后图片

四、基于 keras 的模型构建

4.1 keras 网络里重要的层次结构

我们构建的模型是基于 keras 的深度学习框架，首先来介绍一下 keras 网络里面的重要的层次结构

Conv2D 卷积层

卷积是一种有效提取图片特征的方法。一般用一个正方形卷积核，遍历图片上的每一个像素点。图片与卷积核重合区域内相对应的每一个像素值，乘卷积核内相对应点的权重，求和再加上偏置后，得到输出图片中的一个像素值

MaxPool2D 池化层

在卷积层中，可以通过调节步长实现特征图的高宽成倍缩小，从而降低网络的参数量，例如，下图左边部分中左上角 2x2 的矩阵中 6 最大，右上角 2x2 的矩阵中 8 最大，左下角 2x2 的矩阵中 3 最大，右下角 2x2 的矩阵中 4 最大，所以得到上图右边部分的结果：6 8 3 4

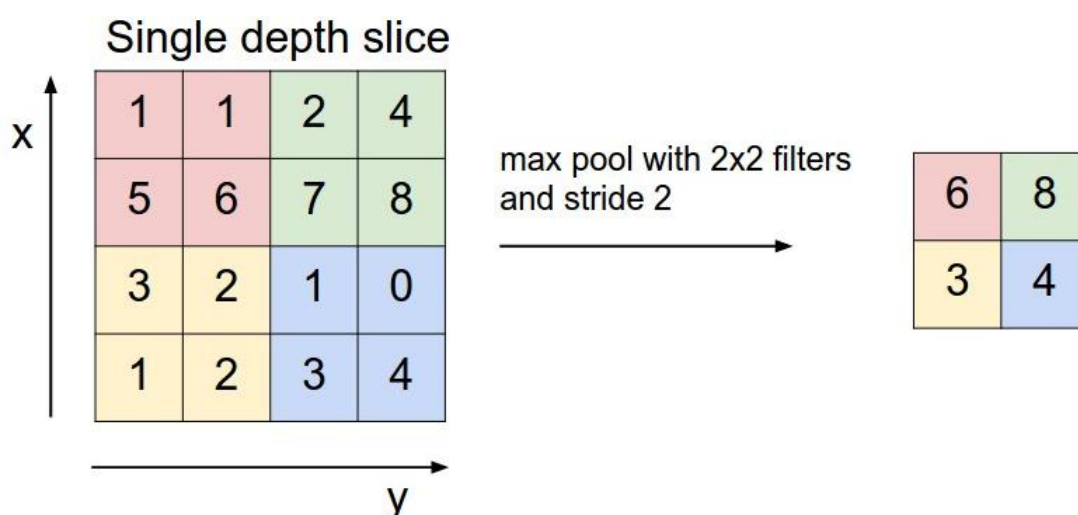


图 4-1-1 最大池化层示意图（图引自 cs231n）[6]

Dropout 层

该层能在训练时能只选用部分神经元进行训练，降低模型拟合速度，对抗过拟合。

Flatten 层

该层用来将输入“压平”，即把多维的输入进行一维化，经常用在从卷积层到全连接层的过渡。

Dense 全连接层

全连接层每个神经元与前后相邻层的每一个神经元都有连接关系，下面是一个简单的全连接层结构示意图：

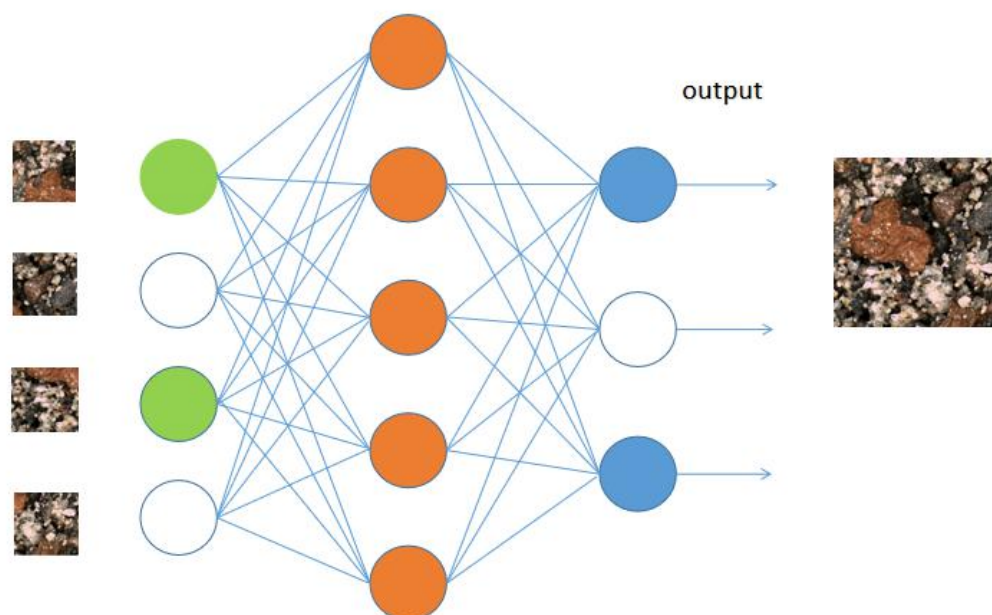


图 4-1-2 全连接层

Relu 激活函数

$$\text{ReLU}(x) = \max(x, 0) = \begin{cases} 0, & x < 0 \\ x, & x > 0 \end{cases}$$

该激活函数属于非线性激活函数，有收敛快的优点，求梯度较简单

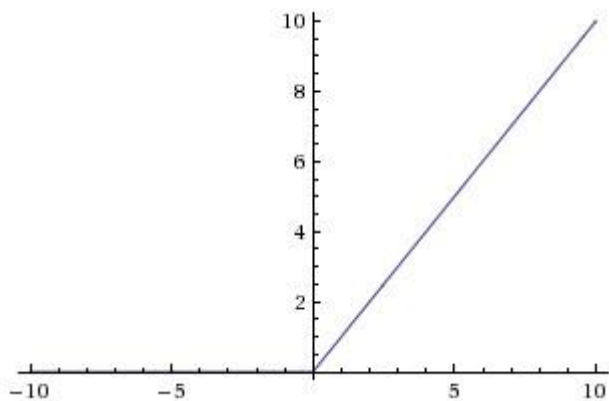


图 4-1-3 ReLU 函数图像

Solfmax 层

该层的功能是把输入映射为 0-1 之间的实数，并且归一化保证和为 1。

Solfmax 的原理如下图所示：

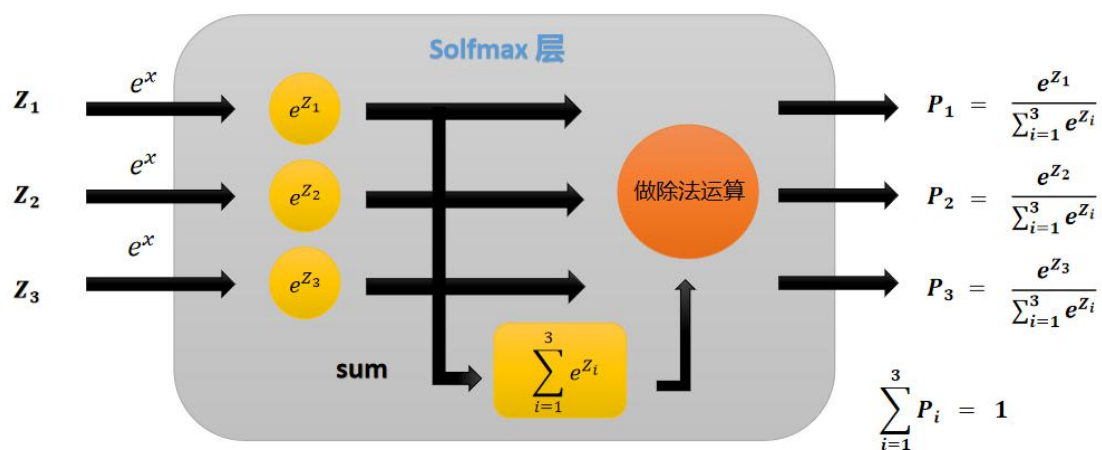


图 4-1-4 Solfmax 函数示意图

4.2 模型的建立

我们构建的模型结构如下图所示：

Layer (type)	Output Shape	Param #
conv2d_188 (Conv2D)	(None, 112, 112, 32)	896
max_pooling2d_8 (MaxPooling2)	(None, 56, 56, 32)	0
conv2d_189 (Conv2D)	(None, 56, 56, 32)	9248
max_pooling2d_9 (MaxPooling2)	(None, 54, 54, 32)	0
conv2d_190 (Conv2D)	(None, 54, 54, 64)	18496
conv2d_191 (Conv2D)	(None, 54, 54, 64)	36928
dropout_3 (Dropout)	(None, 54, 54, 64)	0
max_pooling2d_10 (MaxPooling)	(None, 52, 52, 64)	0
flatten (Flatten)	(None, 173056)	0
dense_6 (Dense)	(None, 128)	22151296
dense_7 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 7)	455

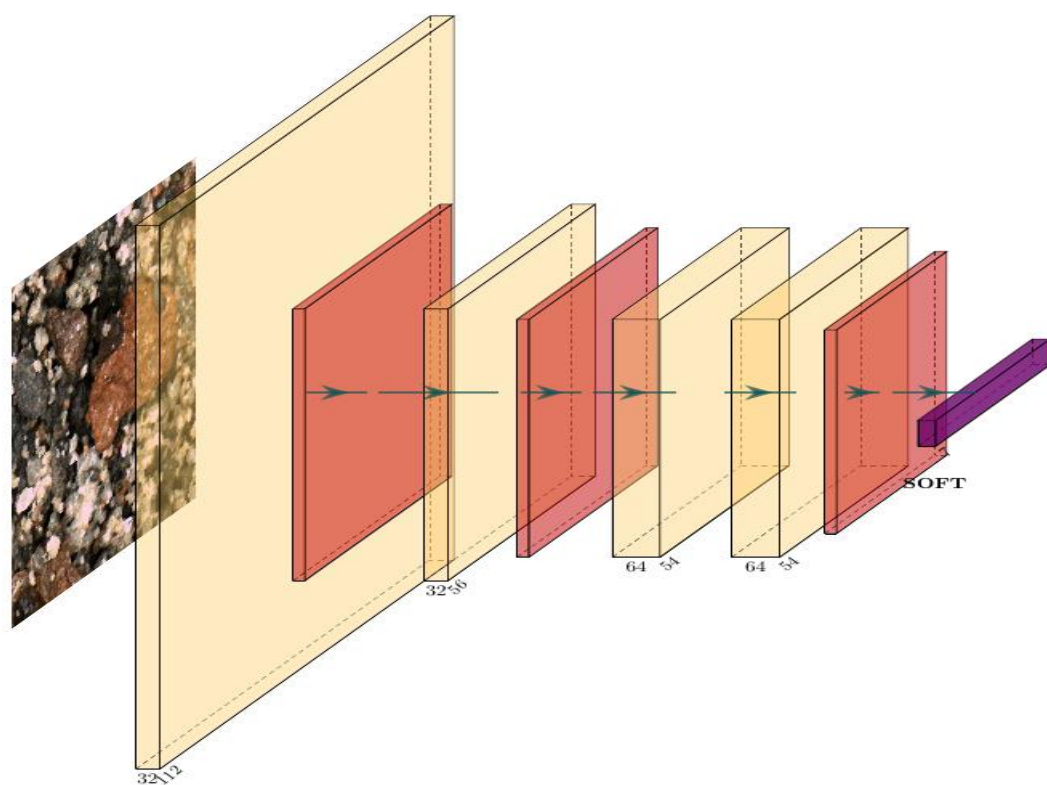


图 4-2-1, 4-2-2 卷积模型示意图

4.3 模型的准确率和损失

在模型迭代了 500 轮后，我们绘制了模型的训练集，验证集的准确率和损失随迭代轮数的变化曲线图，发现训练集的准确率随迭代轮数的增加而增加，但验证集的准确率先增加，然后大概在 100 轮之后趋于平稳；同时训练集的损失随迭代轮数的增加而逐渐下降，但其验证集的损失是先下降，在 100 轮之后又逐渐上升，大概 400 轮之后又逐渐趋于平稳，如下图所示：

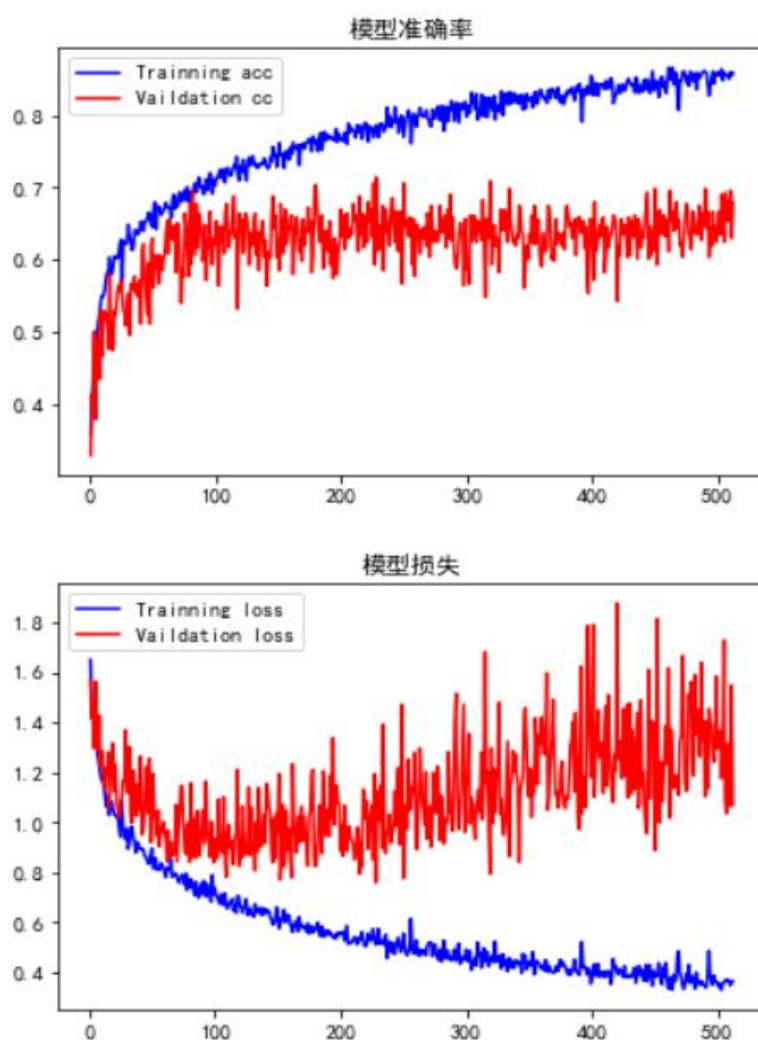


图 4-3-1 模型的准确率和损失值

接下来我们对测试集进行了预测，其测试准确率达到 74.34%，损失

为 0.8743

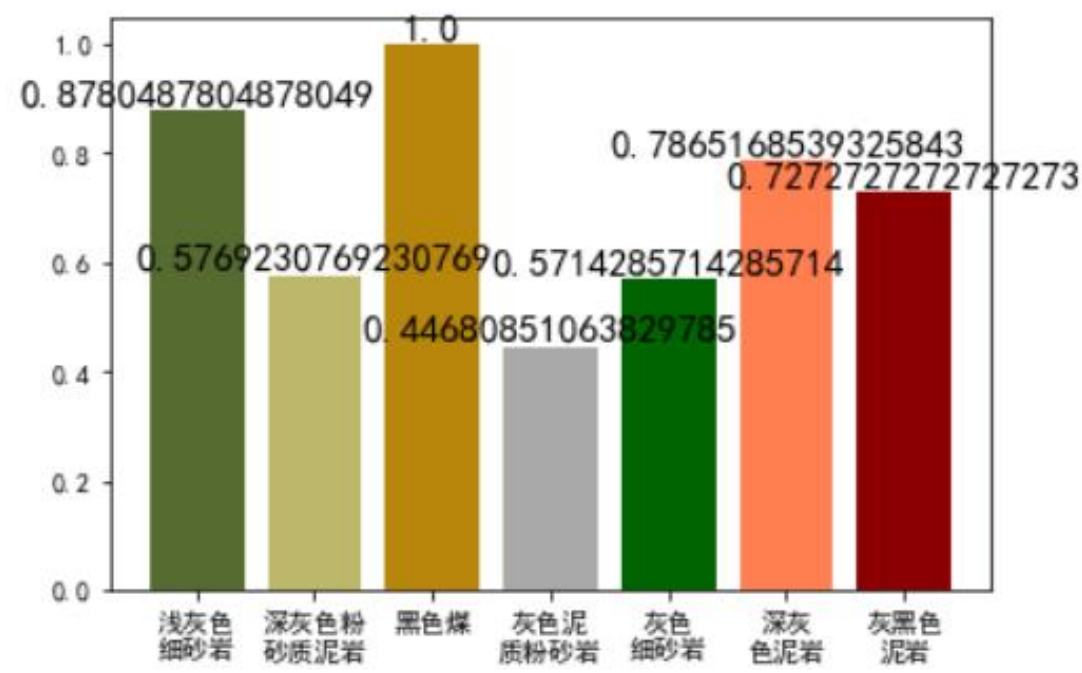


图 4-3-2 测试集中各种岩石的识别准确率

五、迁移学习

5.1 使用迁移学习的原因

在岩石分类问题下，我们通过构建卷积神经网络模型实现对岩石样本的分类，在测试模型过程中，我们发现 keras 的 applications 库中的部分模型在该数据下测试效果较好。例如 Dense121，InceptionV3 模型等，因此我们希望通过迁移学习，进一步提高模型的准确率。

5.2 迁移学习的原理

迁移学习主要研究如何将任务 A 上学习到的知识迁移到任务 B 上，以提高在任务 B 上的泛化性能，对于相似的两个分类任务，可以重用训练效果较好的网络的结构，以便在新任务上获得较好的效果。

5.3 现成网络的介绍

本题所选现成网络为 keras.applications 接口中的 Dense121，Dense169

以及 VGG16。以 Densenet121 模型为例，该模型含有多个卷积层，通过紧密连接减少参数量，在去掉最后一层输出层，并添加 512 个输出通道的全连接层的情况下，该模型共有接近八百万个参数。

5.4 迁移学习的流程

在迁移学习中，使用预训练好的模型，去掉最后一层全连接层，然后追加自定义全连接层和输出节点数为 7 的输出层，使用划分好的训练集，验证集进行训练，使用测试集对模型进行评估。我们对 Dense121，Dense169，VGG16 这三个网络进行了迁移学习，这里我们以 Dense121 为例展示迁移学习的具体流程图：

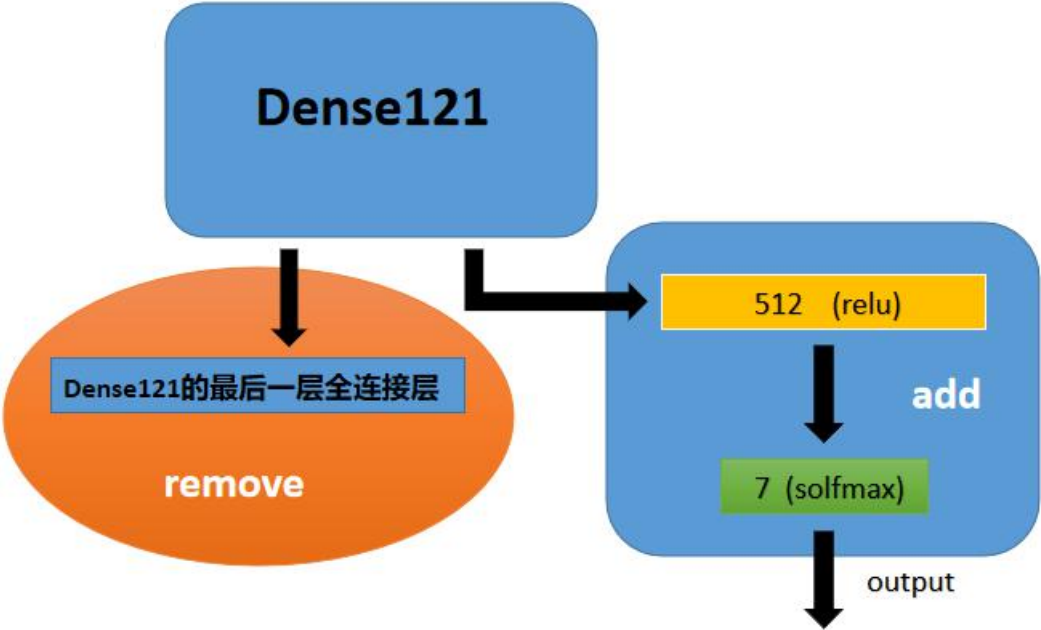


图 5-4-1 通过 Dense121 进行迁移学习的流程图

5.5 各种迁移模型的比较

这三个迁移模型对于各个分类的准确率如下表：

	Dense121	Dense169	VGG16
浅灰色细砂岩	0.8048	0.7479	0.8048

深灰色粉砂质泥岩	0.5576	0.4231	0.3653
黑色煤	0.7916	0.9583	0.8333
灰色泥质粉砂岩	0.1489	0.2340	0.2765
灰色细砂岩	0.7619	0.6190	0.7142
深灰色泥岩	0.7649	0.7865	0.7078
灰黑色泥岩	0.7272	0.5909	0.6363

表 5-5-1 各模型对各类别的分类准确率

接下来是这三个迁移模型的准确率以及损失随迭代轮数的变化曲线：

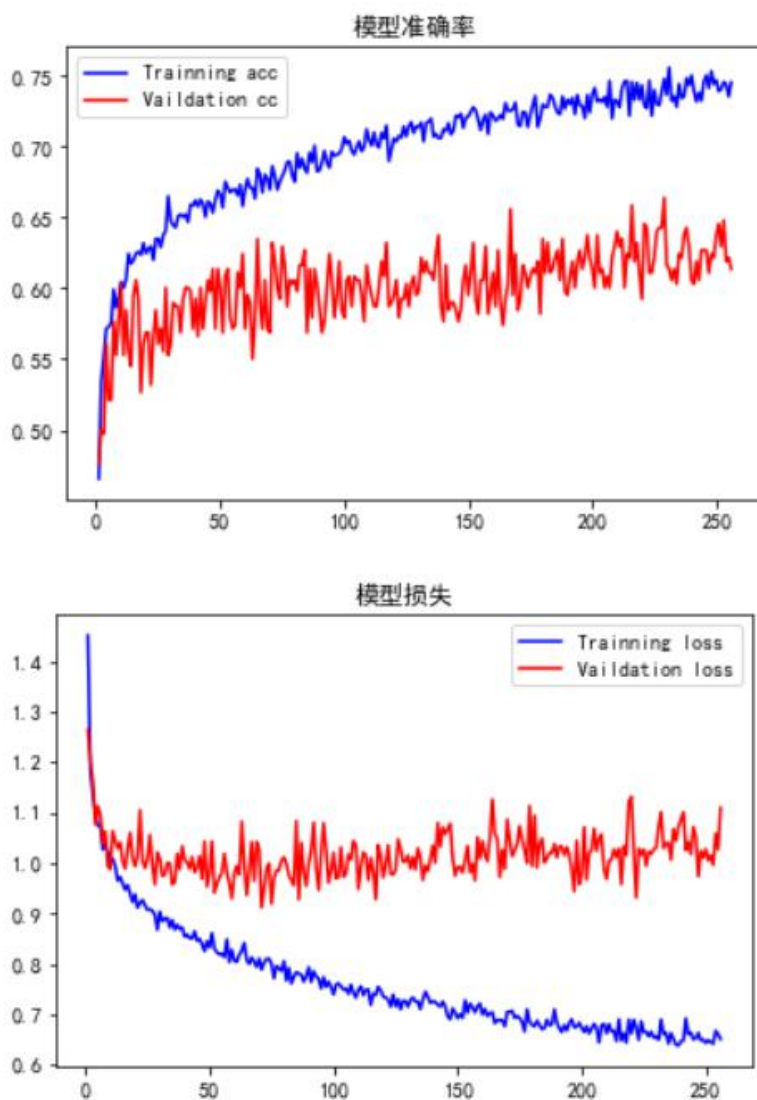


图 5-5-1 Dense121

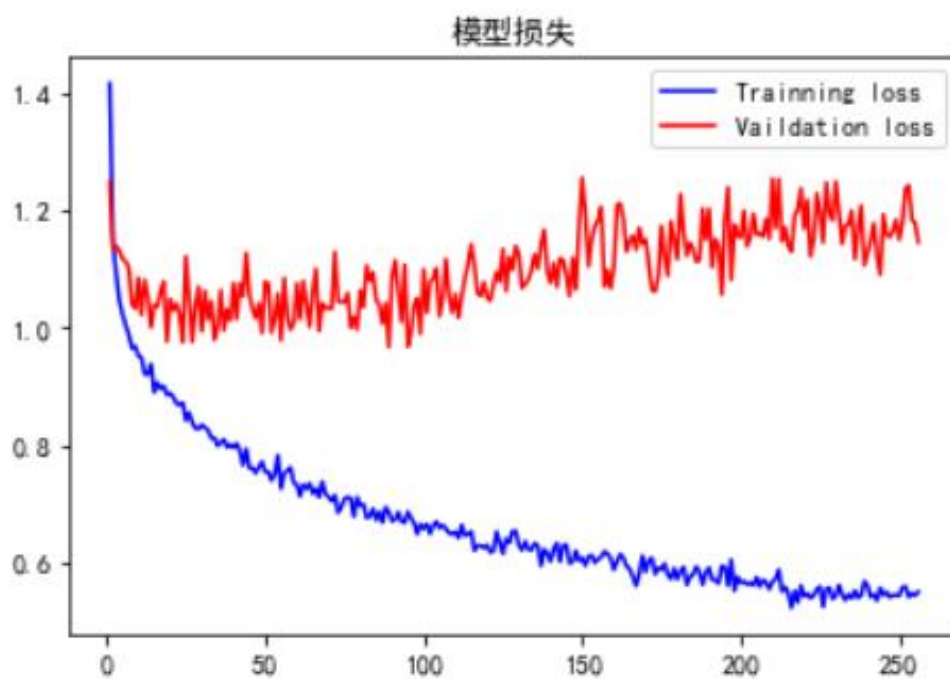
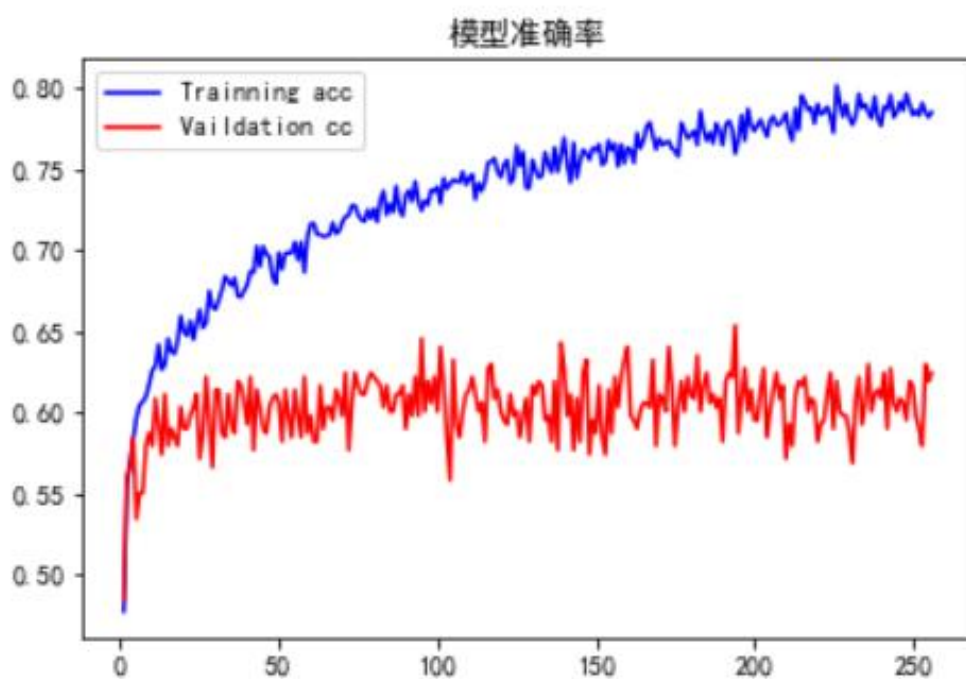


图 5-5-2 Dense169

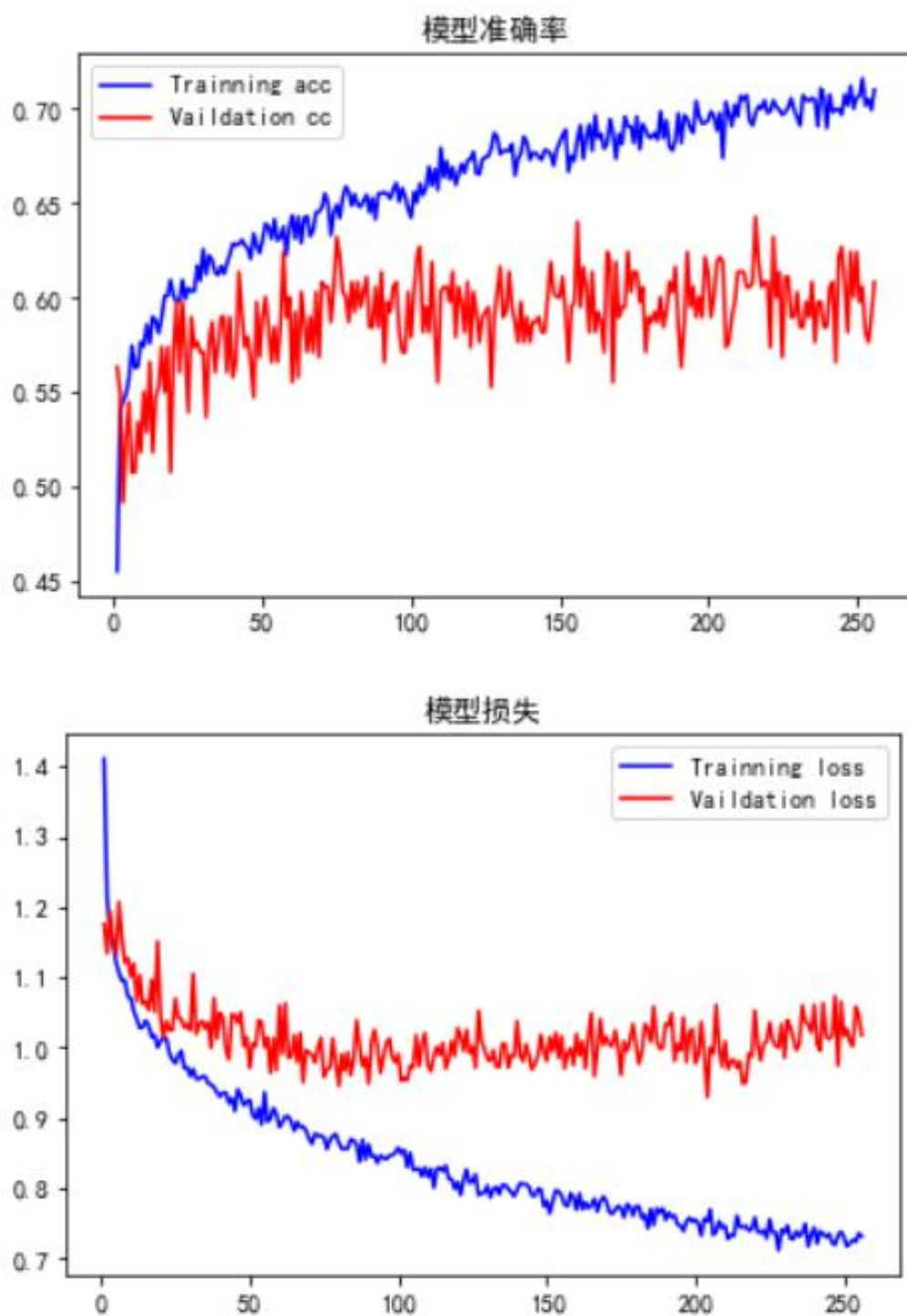


图 5-5-3 VGG16

六、集成学习

6.1 集成学习的原理

集成学习的算法种类很多，其中一种堆叠算法（stacking）相对来说更加通用，性能优秀且容易实现。设已训练的多个模型最后一层全连接层（7 个神经元）激活函数使用 softmax，根据 softmax 的性质，其同一个样本预测的概率求和为 1，那么将多个模型叠加相当于投票，取最大值作为投票结果。为了保持集成后模型的输出仍为 softmax 以及计算损失，我们对求和后的结果进行取均值处理。

具体公式如下：

$$\text{Pred} = \frac{1}{n} \sum_{i=1}^n P_i$$

其中 P_i 为各个分类的预测概率， n 为分类数（在这个模型中取 7）。

在这个算法中，我们将多个已训练的模型针对同一个测试集输入进行预测，而后将预测输出求和取均值作为最终输出。

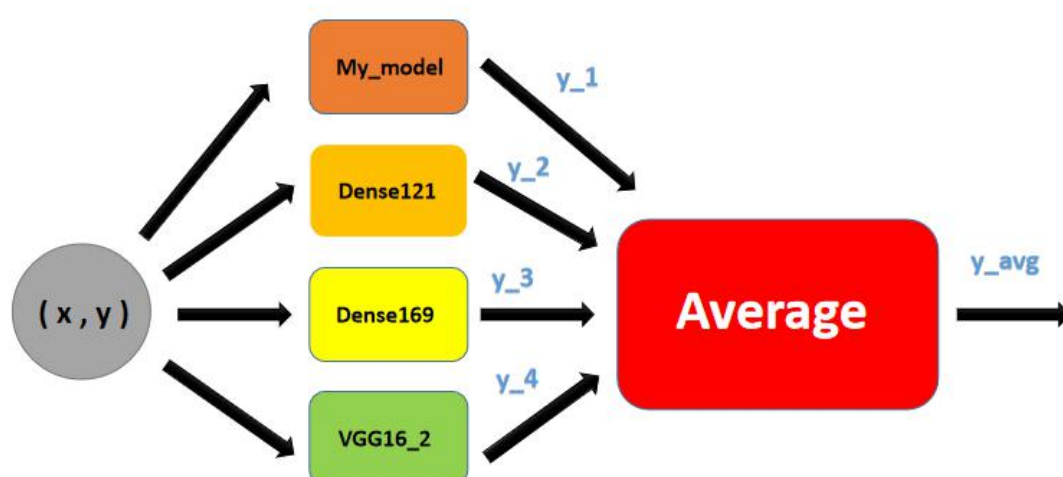


图 6-1-1 采用均值的堆叠算法示意图

6.2 使用集成学习的原因

在使用监督学习算法训练出来的模型对测试集进行预测时，不同神经网络结构的模型对不同分类的准确率是不一样的。例如，假设有总体预测准确率和损失相近的两个模型（记为模型 1 和模型 2），模型 1 对分类 1 的预测准确率是 0.5，模型 2 对分类 1 的预测准确率是 0.8，那么，如果使用合适的集成学习算法，将使得两个模型优势互补，从而集成出来的模型将有优于各个模型的准确度和低于它们的损失。

6.3 集成学习的流程

我们拟采用集成学习中较为通用的算法：堆叠（stacking），首先用不同结构的神经网络，同样规模的训练集（数据增强处理）训练出若干个模型。我们采用迁移学习的 densenet, VGG 等经典模型进行训练。

集成：对于每个训练出来的模型，让其对同一个测试集进行预测，得到每个模型对应的预测概率（one-hot 编码）。然后将每个模型进行叠加，取算术平均值，再取预测值中的最高值对应预测的标签。

七、模型的评估

7.1 集成模型与单个模型的比较

如下图可知，集成学习后的模型有着高于各个弱模型的准确率的以及低于它们的损失。这说明集成的模型能够更加准确稳定地预测样本，且具有更高的泛化能力。同时，集成后的模型识别准确率已高达

77.25% ， 其损失为 0.7078.

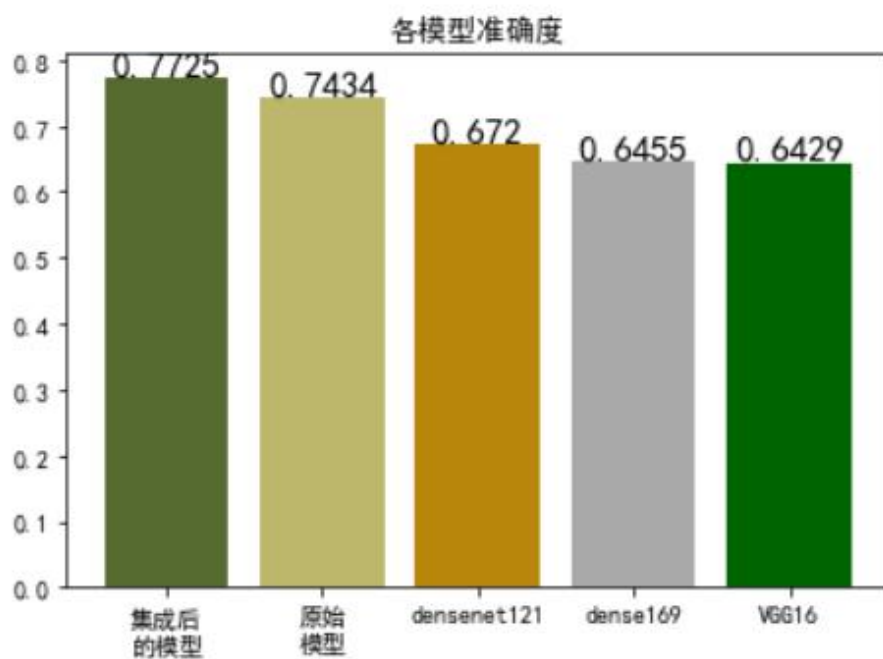


图 7-1-1 各模型准确率

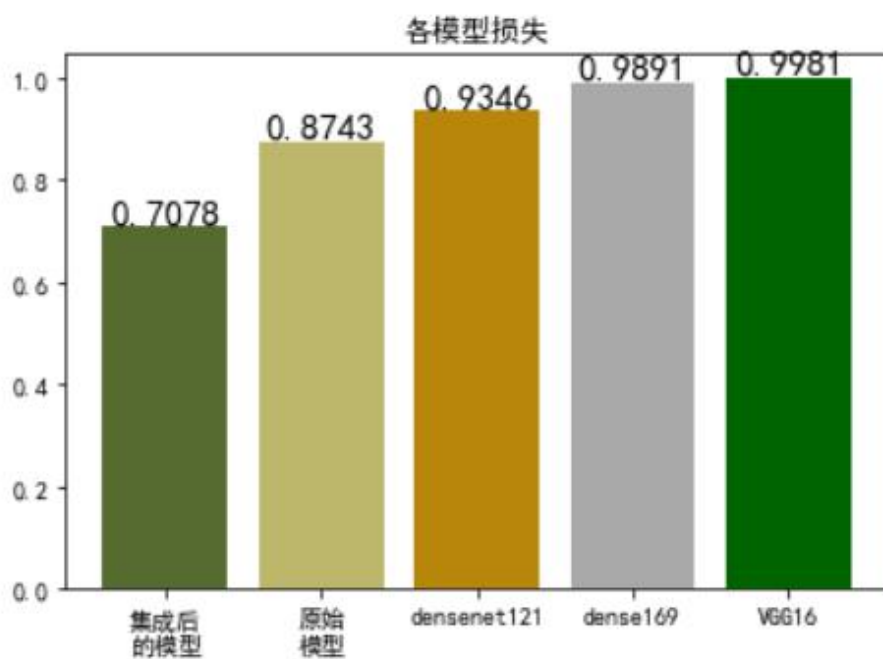
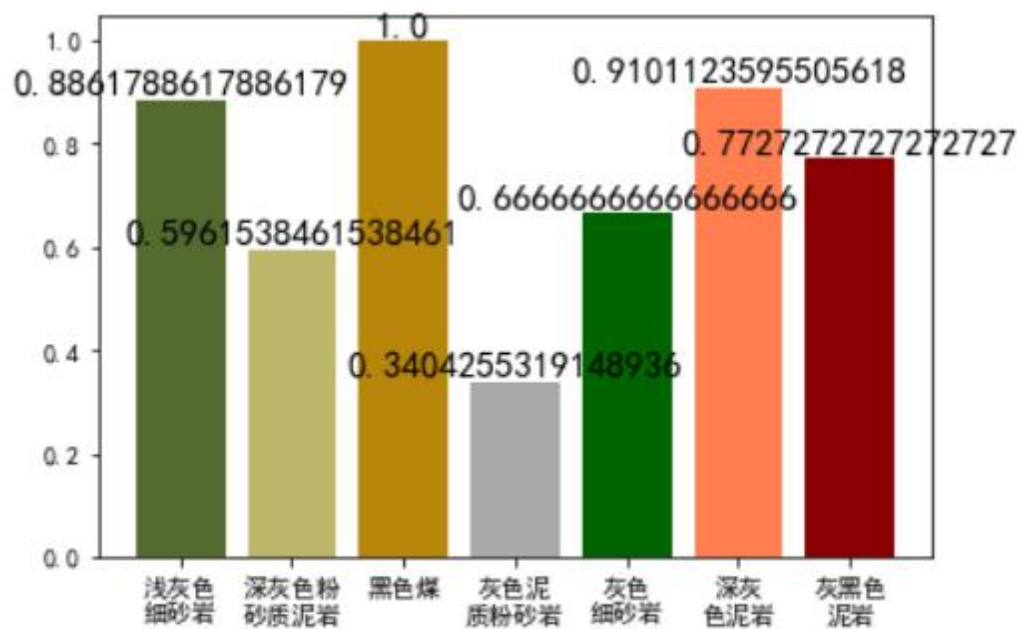


图 7-1-2 各模型损失值

7.2 集成模型中每种矿石的准确率

由下图可知，集成模型对各种岩石的准确率由高到低分别是黑色煤，深灰色泥岩，浅灰色细砂岩，灰黑色泥岩，灰色细砂岩，深灰色粉砂质泥岩，灰色泥质粉砂岩，其中识别率最高的是黑色煤（100%），识别率最低的是灰色泥质粉砂岩（34.04%）。黑色煤识别率高可能是因为其颜色非常的黑，而灰色泥质粉砂岩识别率不高可能是因为其形态或者颜色多样难以提取其特征而导致的。



集成学习后的测试集准确率：0.7724867724867724
集成学习后的损失（交叉熵）：0.7078404502263145

图 7-2-1 集成后对各类别的准确率

八、模型的使用

8.1 无标签数据集的预测

对于将要进行预测的未打标签测试集，为了保持数据处理上的一致性，从而使得模型识别更加稳定，故将对测试集进行不重合的切割分

片处理。

8.2 图片处理原理

该切割识别算法原理类似于集成学习的堆叠法。对于 n 张将要预测的图片（编号 $0, 1, 2, \dots, n-1$ ），每张切割成相同数量以及大小的 m 份正方形尺寸的小图片（从左上到右下依次编号为 $0, 1, 2, \dots, m-1$ ）。对于某些边缘为地面而不是岩石的图片，可能会造成噪音从而干扰预测，故只取中央拍摄到岩石的部分进行切割。

把 n 张图片对应的 m 张切片进行依次预测，得到预测列表，表示每张切片的预测值。如果把切片的位置对应好，那么可以将来自于同一张图片的 m 张切片的所有预测值相加，再取算术平均值，即可视为综合的预测值。取概率最高值对应类别即为预测的类别。

多次识别取均值有助于提高识别的稳定性，模型的鲁棒性，该算法与集成学习共用能够更准确地识别。

8.3 预测流程

按照上述原理进行补充和切割得到 n 张图片对应的 m 张切片。

创建 m 个列表，用于存储每个切片对应的测试集，每个列表含有 n 张图片的切片张量。对于原图测试集中第 i 张图片的第 j 个切片，将其作为张量放到第 j 个测试集列表中的第 i 项。多个存储图片张量的列

表可以视为一个二维表，其中第 j 行第 j 个切片，第 i 列代表第 i 张图片，同一行包含测试集所有的图片其中一个切片，同一列包含同一张图片的所有切片。 (j, i) 代表第 i 张图片第 j 个图片。

如下表（含数字编号是便于显示，空白处为对应位置的切片张量）：

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							

预测时，迭代 m 次，每次把上表中一行的图片张量作为输入，把每个输出的独热编码（含有对每个分类预测的概率的列表）保存到 pre 列表对应的位置中。得到一个 $m \times n$ 二维表，里面含 n 张图片 m 个切片的预测值（数据结构是独热编码列表），二维表结构与上表相同。

将二维表同一行的所有独热编码对应位置相加再取平均，得到一个含 n 个独热编码表的 1 维列表。如下表（空白处为对应位置的切片张量）：

	0	1	2	3	4	5	6
0							

将每个独热编码取概率最高值对应的类型编号，即为预测的岩石样本类型。

九、矿石的含油量计算

9.1 荧光岩石图片的预处理

数据集 rock 中包含两种格式的岩石图片，后缀为 **bmp** 的岩石图片全图为岩石部分，而其荧光下拍摄的岩石图片亮度较低，观察下只有少部分黄色和绿色的荧光，在此前提下，我们使用亮度归一化算法对这种 **bmp** 格式的岩石荧光图片进行预处理，以便凸显出隐藏的黄色和绿色荧光，同时提高含油量的计算准确率。对于后缀为 **jpg** 的图片来说，整张图片不全是岩石部分，因此需要将其岩石部分通过颜色识别算法提取出来，同时由于其荧光的岩石图片相对来说亮度较高，因此不需要进行亮度归一化，直接调整图片的对比度即可。

9.2 亮度归一化算法

该算法是根据灰度、Gamma 归一化亮度[2]

亮度归一化的前后对比（以 2-1.bmp 为例）：

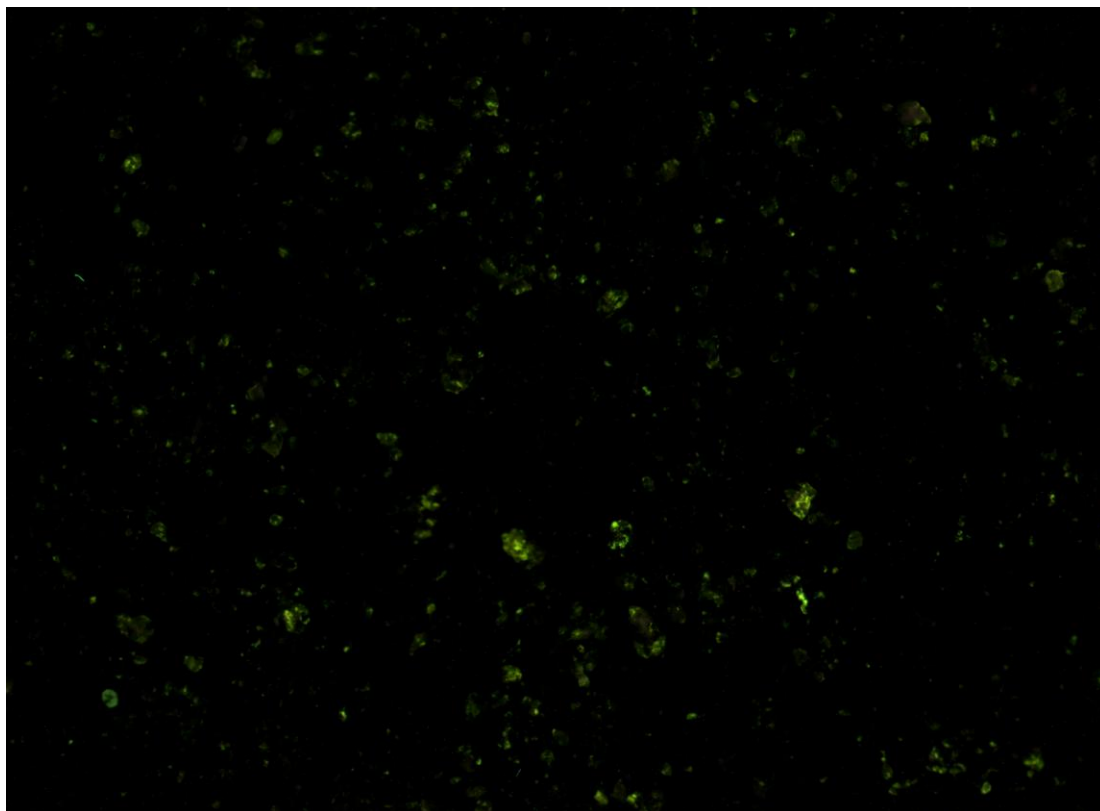


图 9-2-1 亮度归一化前

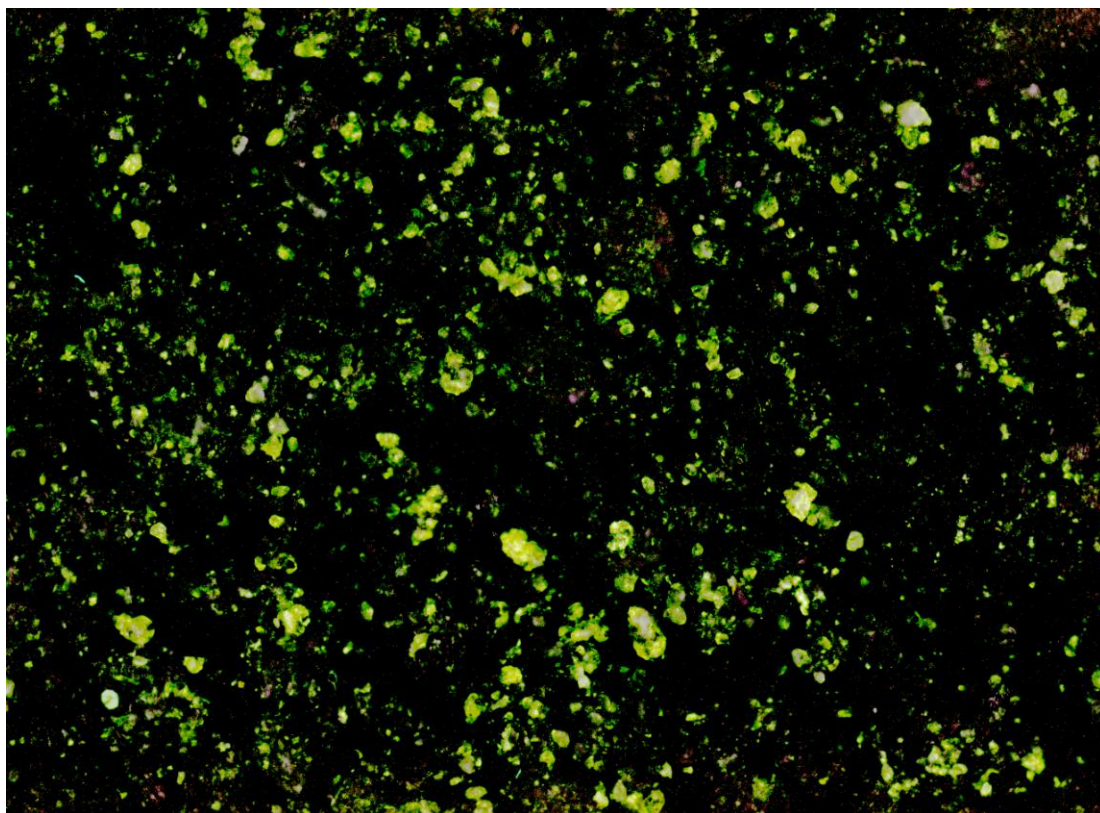


图 9-2-2 亮度归一化后

9.3 对比度的调整

对比度的调整[5]是基于 OpenCv 的 `addWeighted` 函数，该函数一共有 5 个输入，分别是（图像 1，alpha ， 图像 2， beta， gamma）

$$\text{dst} = \text{img}_1 \times \alpha + \text{img}_2 \times \beta + \gamma$$

其输出 `dst`（对比度调整后的图片）是两张图像的线性组合加上一个系数 γ 。其中 α ， β ， γ 三个参数为超参数，图像 `img2` 为与图像 `img1`（原图片）同样规格的全黑图片。通过改变 α 的值就相当于调整图像 `img1` 的对比度。



图 9-3-1 329-2. jpg 原图

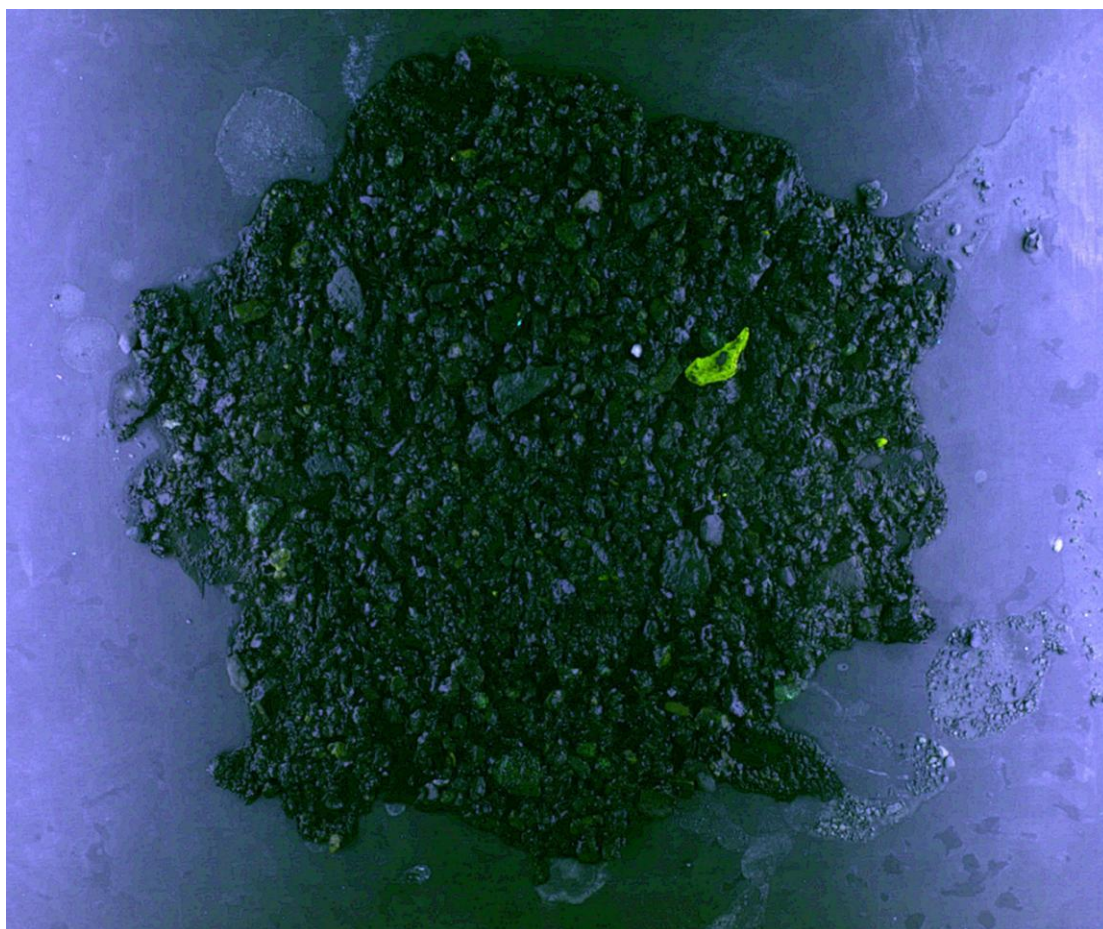


图 9-3-2 329-2. jpg 对比度调整后

9.4 颜色识别的算法

该算法主要是基于 HSV 色彩空间，其中 H 表示色调，用角度度量，取值范围为 $0^{\circ} \sim 180^{\circ}$ ，S 表示饱和度，表示颜色接近光谱色的程度，通常取值范围为 $0 \sim 255$ ，值越大，颜色越饱和，V 表示明度，表示颜色明亮的程度，通常取值范围为 $0 \sim 255$ 。如下表所示：

	黑	灰	白	红		橙	黄	绿	青	蓝	紫
hmin	0	0	0	0	156	11	26	35	78	100	125
hmax	180	180	180	10	180	25	34	77	99	124	155
smin	0	0	0	43		43	43	43	43	43	43
smax	255	43	30	255		255	255	255	255	255	255
vmin	0	46	221	46		46	46	46	46	46	46
vmax	46	220	255	255		255	255	255	255	255	255

表 9-4-1 HSV 色彩空间[4]

本题中我们需要获取的指定颜色有蓝色（jpg 图片中的非岩石部分），绿色和黄色（岩石中的油）。首先我们将图片的色彩空间转为 HSV 色彩空间，接着通过上面的 HSV 参考表，找出需要提取颜色的范围，最后使用 `inRange` 函数进行提取，该函数将目标颜色转变成白色，非目标函数转换成黑色，以达到提取颜色的目的，下面展示颜色提取的效果图：

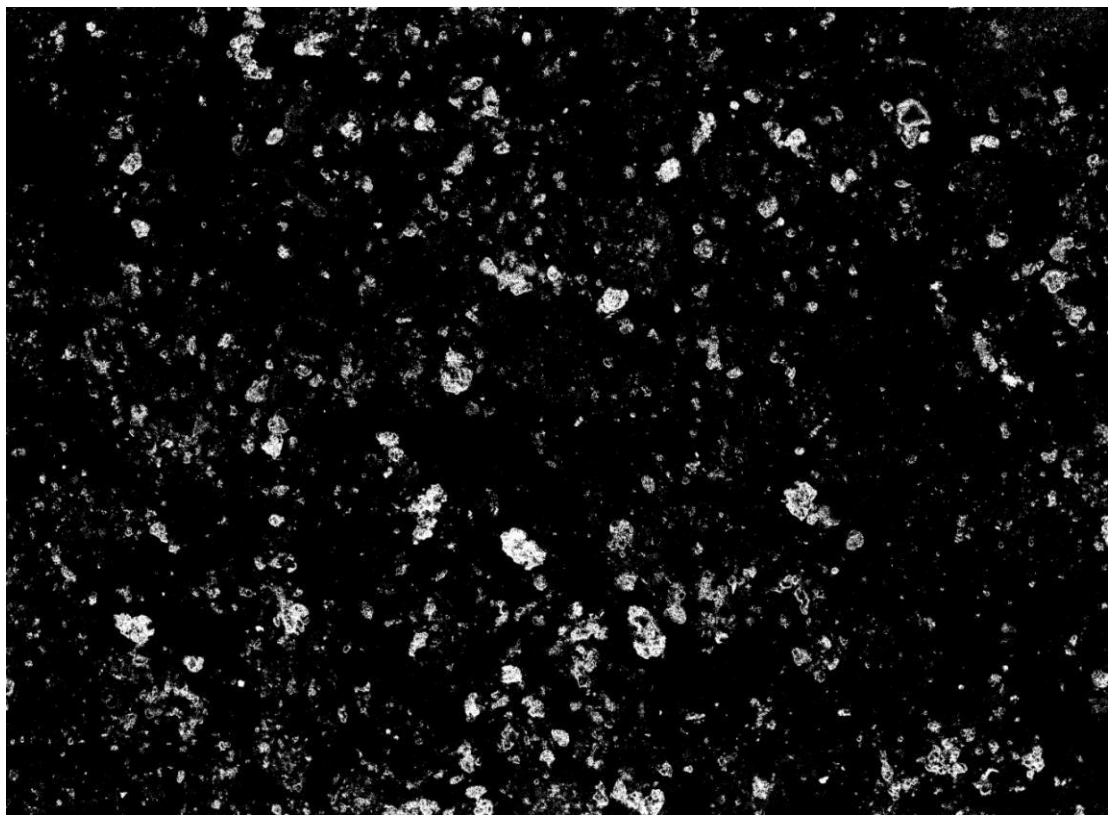


图 9-4-1 提取 2-1.bmp 的绿色部分

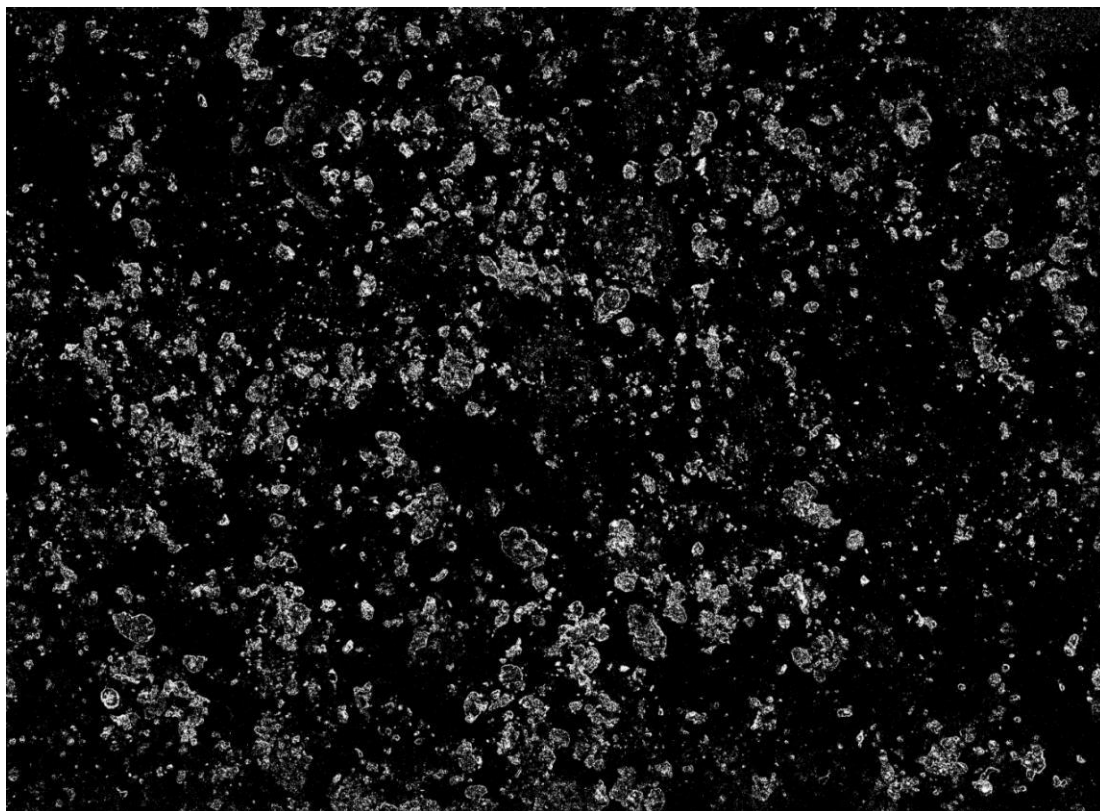


图 9-4-2 提取 2-1.bmp 的黄色部分



图 9-4-3 326-1.jpg 原图

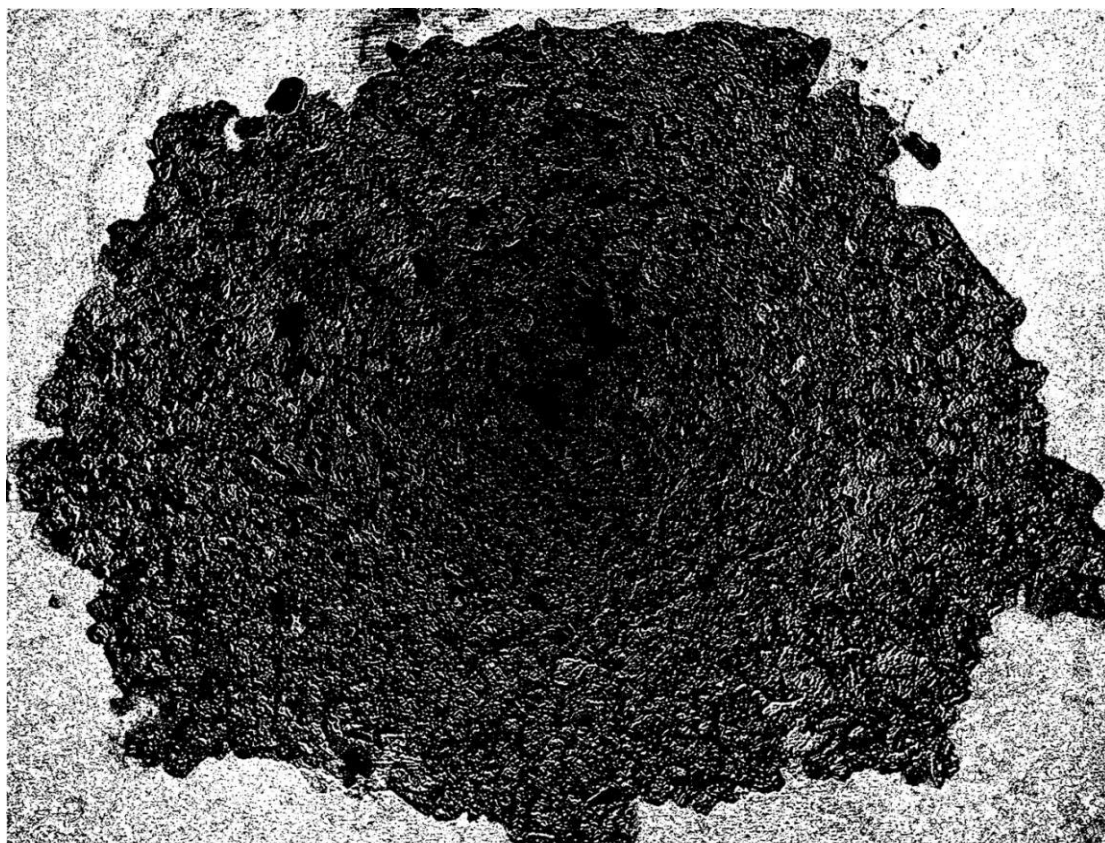


图 9-4-4 提取 326-1.jpg 中的岩石部分（即黑色部分）

9.5 含油量的计算

含油量的计算公式：

$$p = (N + M) / S \times 100\%$$

各参数的含义：

黄色的像素点个数	N
绿色的像素点个数	M
岩石面积	S
含油量	p

十、总结

本文基于深度学习 keras 框架建立了卷积神经网络实现对岩石的

分类,并通过迁移多模型实现集成学习。对图片进行亮度调整,实现含油量的计算。

(1) 在对数据预处理时，使用不重复切片和数据增强扩充数据集，同时保证验证集和测试集的独立。

(2) 对比多个迁移的模型和原模型,按投票法集成,得出的集成结果准确率可达到 77.25% 。

(3) 计算含油量时，先通过亮度归一化调整亮度，再将图片转化为 HSV 色彩空间，提取目标颜色的占比。

十一、附录

1.详细代码见 B 题岩石分类代码.zip（里面包含 jupyter 文件以及 py 文件）

2.第二问的含油量计算结果见 第二问结果.csv

十二、参考文献

[1]张野,李明超,韩帅.基于岩石图像深度学习的岩性自动识别与分类方法[J].岩石学报,2018,34(02):333-342.

[2]多个图片亮度和色度归一化处理_大臉喵愛吃魚的博客-CSDN 博客
[https://blog.csdn.net/qq_36638362/article/details/109311353?](https://blog.csdn.net/qq_36638362/article/details/109311353?ops_request_misc=%257B%2522request%255F)
ops_request_misc=%257B%2522request%255F

[3] 煤等矿物的视觉识别-CNN 卷积神经网络实现_大龙剑神的博客-CSDN 博客

https://blog.csdn.net/qq_42279468/article/details/95631815?
ops request misc=%257B%2522request%255Fid

[4] python+opencv 获取图片中指定颜色的部分_飞天土豆的博客 -CSDN 博客

https://blog.csdn.net/qq_40456669/article/details/93375709?
ops_request_misc=&request_id=&biz_id=102&

[5] [python-opencv 图像的对比度和亮度_、小祖宗的博客-CSDN 博客](#)

https://blog.csdn.net/weixin_41246832/article/details/84260987?ops_request_misc=%257B%2522request%25

[6]池化层详细介绍_Chen_Swan 的博客-CSDN 博客_池化层

https://blog.csdn.net/Chen_Swan/article/details/105486854?o
ps_request_misc=%257B%2522request%255Fid%