Journal of Artificial Intelligence Practice (2022)
Clausius Scientific Press, Canada

DOI: 10.23977/jaip.2022.050208
ISSN 2371-8412 Vol. 5 Num. 2

On Data Analysis and Design and Implementation of Data Preprocessing Scheme Based on Low-quality Rock Datasets

Quan Hao

School of Information Engineering, Wuhan College, Wuhan, 430212, China

Keywords: Deep Learning, Data Preprocessing, Image Processing, Data Augmentation

Abstract: With fast progress of deep learning technology, breakthroughs are achieved in many industries by virtue of efficient artificial intelligence models. In addition, computer hardware is cheaper, which makes it easier to acquire an excellent deep learning model. However, output of a model with good generalization rests with not only powerful hardware computing speed, but also quality of dataset involved in the calculation. Unfortunately, high-quality dataset is probably more expensive than high-end hardware, and this forces deep learning engineers or practitioners to use lower-quality dataset. Anyway, it doesn't mean excellent deep learning programs can't be created by such dataset. In particular, dataset preprocessing is equally important, and even engineers need to spend most of time elaborately formulating preprocessing strategies. This study mainly analyzes data and formulates preprocessing schemes of low-quality rock datasets. It aims to make deep learning programs more efficient and general-purpose at the lowest possible cost.

1. Introduction

1.1 Research Background

Scientists and practitioners in other fields expect that computers can process repetitive tasks, extract key information from pictures, analyze audio data, and assist in medical diagnosis, or scientific research in other disciplines. Practical exploration shows structured tasks can be handled efficiently, but problems difficult to formalize are the bottleneck. By contrast, deep learning refers to a solution that allows computers to learn from experience to solve problems, and helps computers form more complex concepts based on simple ones.

Rocks classification is one basic task of modern geology. It is difficult for computers to directly apply traditional image classification methods, which is because classification standards are set by people. Even same rocks may vary greatly in surface color, texture, and granularity; abstract features in image cannot be captured easily until deep learning technology is introduced to computer vision. With development of deep learning, scholars in computer vision field carry out massive research on identifying rock sample images at home and abroad. To be specific, CHENG Guojian and LIU Ye classify rock images through shallow neural networks and SVM.[1][2]

Attempting to combine cluster segmentation with deep learning, they classify rock images by distinguishing rocks and target pores, as well as extracting features from them. Ideal results are obtained finally, with correct rate of 95%. Moreover, Mariusz Młynarczuk et al., based on slice photos from polarized light microscopes, intelligently classify them through four pattern recognition methods, including nearest neighbor, K-nearest neighbor, nearest mode algorithm and optimal spherical neighborhood. They test effectiveness of these methods in four different color spaces: RGB, CIELab, YIQ and HSV. After research, recognition rate reaches as high as 99.8% by CIELab color space and nearest neighbor classification method[3]. BAI Lin, WEI Xin and others explore recognition of rock slice images based on VGG[4] model. Six types of slice images are involved in neural network training, and recognition rate is 82%. Eventually, they find that under VGG model, sample images with similar components cannot be distinguished easily, such as dolomite and oolitic limestone classified into carbonatite.

1.2 Deep Learning and Dataset

That how to process data is a key link, which can be proved from research process of scholars in rock image recognition. Generally speaking, in deep learning field, data objects to be processed are datasets which can be set of data such as images, text, audio, video, and figures, or a key part of any deep learning program. Reliability of deep learning rests with dataset quality. In fact, most of time is spent on data processing in the development of deep learning programs. Dataset quality is affected by diverse factors, including:

(1) Data size. Insufficient data is one major issue for most deep learning datasets.

(2) Labeling errors and biases. Most of data is labeled by humans, so labeling bias may occur due to human errors or subjective biases.

(3) Different conditions of data collection. Data is collected in different environments, which may result in large differences in display status of same objects.

Massive costs are needed to obtain a really high-quality dataset. To sum up, it is of high research significance to acquire a good deep learning model through low-quality dataset. This study, supported by Pytorch image processing technology, attempts to optimize an open-source rock sample dataset, in order to explore how to help low-quality dataset meet needs of deep learning programs as much as possible.

1.3 Introduction to Data Preprocessing

Data preprocessing means some necessary operations on data before it is officially used. In image data field, major operations include:

(1) Data sampling, including data division and preferential use.

(2) Data conversion, and unify data size.

(3) Data normalization and distribution of unified data.

(4) Data augmentation, dataset expanding

Data preprocessing is critical for both basic data analysis and deep learning development, and it determines application reliability, to a great extent. For programmers, they must properly handle data before writing programs.

2. Dataset Defects Analysis

It is necessary to identify and analyze problems of dataset before formal data processing, in order to make targeted adjustments.

2.1 Image Size

Images of datasets have different sizes, namely 4096×3000 pixel and 2248×2048 pixel respectively. Figure 1 shows size difference of two types of data numbered 160 and 343. In addition, format of image data is bmp, and memory space required for a single image is too large (close to 40M), so image size and data format should be adjusted. In this study, it adopts command line tool ImageMagick and the Resize tool provided by Pytorch framework to process images.

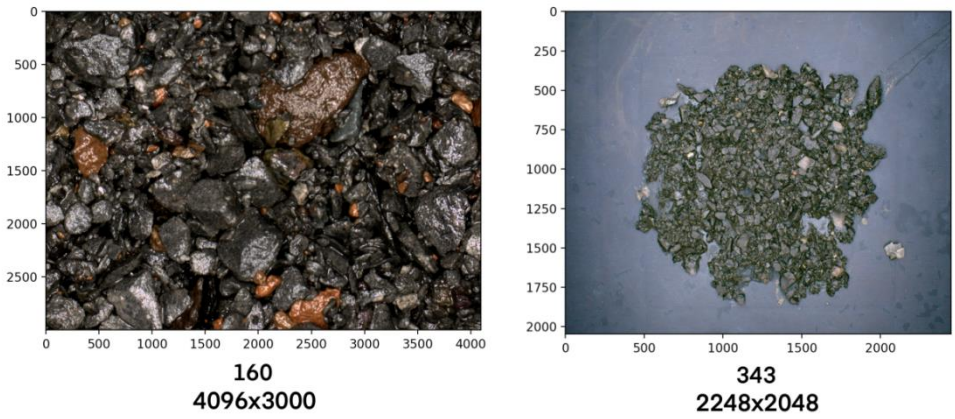

Figure 1: Image Size Difference

2.2 Sample Distribution

Dataset itself contains a small number of images, and different samples are far from same in quantity, as shown in Figure 2. It tells from the figure samples of light gray fine sandstone are the largest, followed by dark gray mudstone, and number of gray fine sandstone and gray black mudstone in the end. Overfitting probably occurs when neural network is trained on an imbalanced dataset. Overfitting means the lack of generalization of model. Specifically, model performs well on the training set, while in image recognition field, it represents high recognition accuracy. Performance is poor on test set, that is, low recognition accuracy. Regularization is a method to cope with overfitting, and data augmentation is one form of regularization technology. This study tries to expand dataset size by data augmentation, including to undersample massive samples, and oversample insufficient samples, aiming to solve data imbalance.

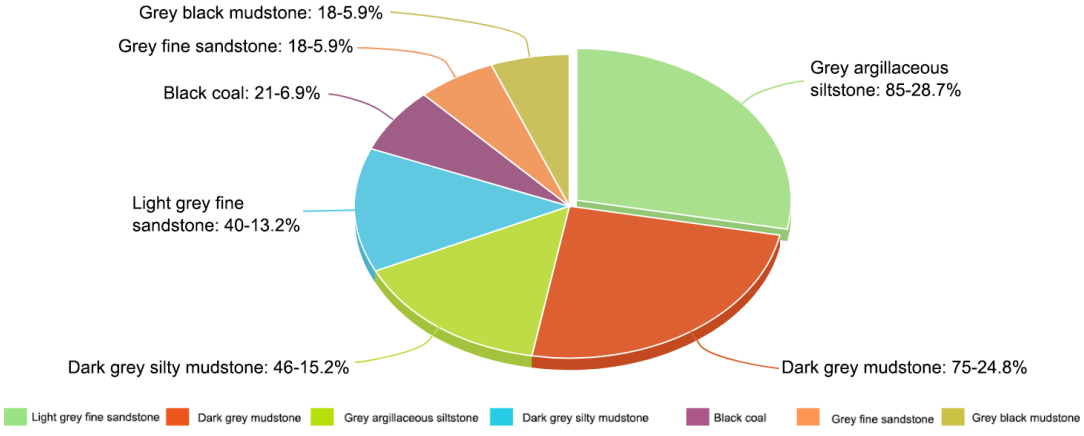

Figure 2: Data Distribution

2.3 Data Labeling

Due to different data collection environments or human errors, images of same rocks are different. Figure 3 describes display difference of data numbered 18 and 21 under different lighting, humidity and other conditions. The left picture has extremely low brightness, and most areas are close to black; while the right one is brighter and can be recognized easily.

Figure 3: Gray Black Mudstone Picture in Different Environments

2.4 Image Noise

In dataset, there are impurities in many images, as shown in Figure 4. These impurities make model identification more difficult.

Figure 4: Impurities in Some Images

3. Data Preprocessing

3.1 Data Normalization Input

Images in the dataset have different sizes and deep learning data network cannot work normally on such a dataset. Therefore, normalization operations are required on data input.

3.1.1 Grayscale

Images should be grayed firstly, in order to abandon partial information in exchange for training efficiency, that is, convert color image to gray scale (Gray Scale). In this study, grayscale is realized by using transforms. Grayscale() function in Pytorch library, and its effect is displayed in Figure 5.

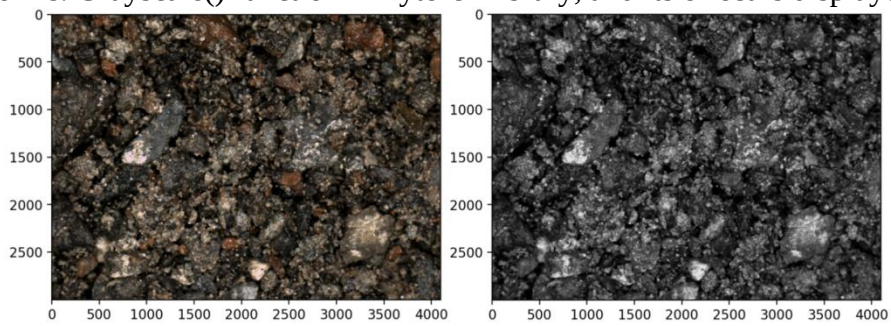

Figure 5: Grayscale of Color Pictures

Images in dataset are different in length and width. Out of consideration on simple compression of subsequent images, an area in image center should be cut by Image in Python image library. The area is calculated according to length and width of the image, with operation code as follows:

```
new_width = min(image.size)
left = (width - new_width)/2
top = (height - new_width)/2
right = (width + new_width)/2
bottom = (height + new_width)/2
image = image.crop((left, top, right, bottom))
```

The effect is shown in Figure 6.

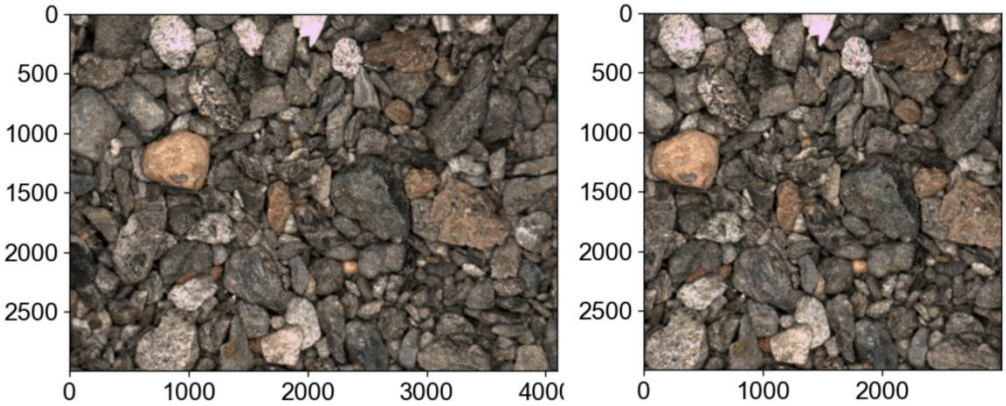

Figure 6: Center Cropping Example

3.1.2 Image Compression

Image formats in dataset are different, and they are generally too big. If a complete image participates in training, higher performance will be required. Accordingly, transforms.Resize((80, 80)) function provided by Pytorch library is used to compress images. The specified parameter of the function is 80, indicating images will be compressed into the size of 80x80. Effect of this function is applied on the basis of grayscale and center cropping, as shown in Figure 7.

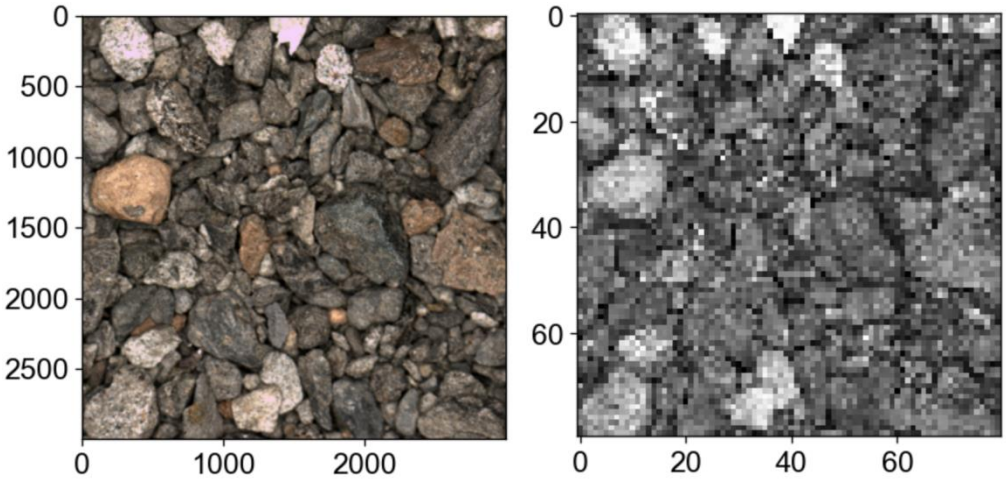

Figure 7: Image Compression Example

3.1.3 Normalization

Normalization is a linear transformation that uniformly maps values of data to an interval or a specific distribution in light of a certain feature. Also, it is also called contrast stretching or histogram stretching. In digital signal processing, it is defined as dynamic range extension. In Pytorch, images are normalized with mean and standard deviation by library function torchvision.transforms.Normalize(). Firstly, it subtracts channel mean from each input channel, then divides the result by channel standard deviation, see (Equation 3.1). Normalization reduces data bias to some extent and helps neural network better obtain features. Normalization in PyTorch is achieved by torchvision.transforms.Normalize() that normalizes images via mean and standard deviation. Its Equation is shown in 3.1, where $input$ and $output$ represent input and output of current channel; $mean$ and std stand for mean and standard deviation of each channel.

$$output = \frac{input - mean}{std} \qquad \text{(Equation 3.1)}$$

Calculate mean and standard deviation of the channel in a picture, to complete normalization effect of picture in Figure 8. The left picture is original, and right one displays effect after normalization.

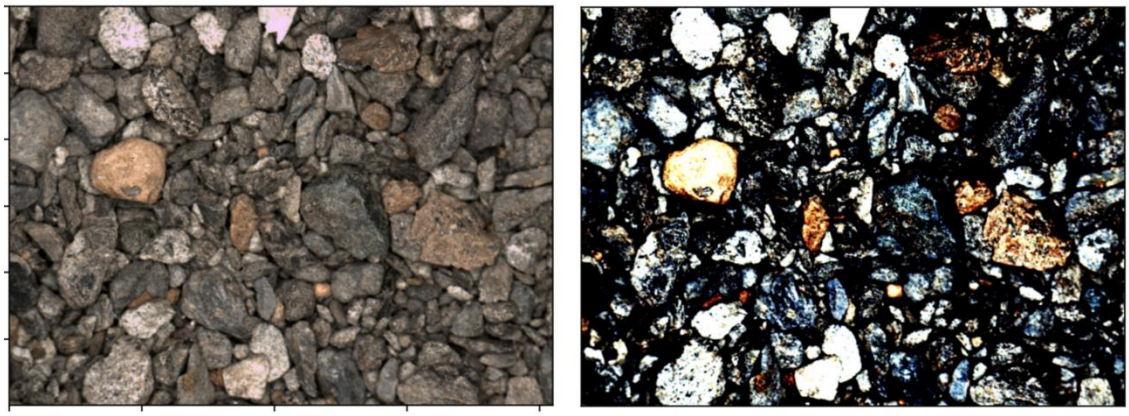

Figure 8: Image Normalization Example

After normalization, pixel distribution of image will change, see Figure 9. At this time, $mean$ and std are 0 and 1 respectively.

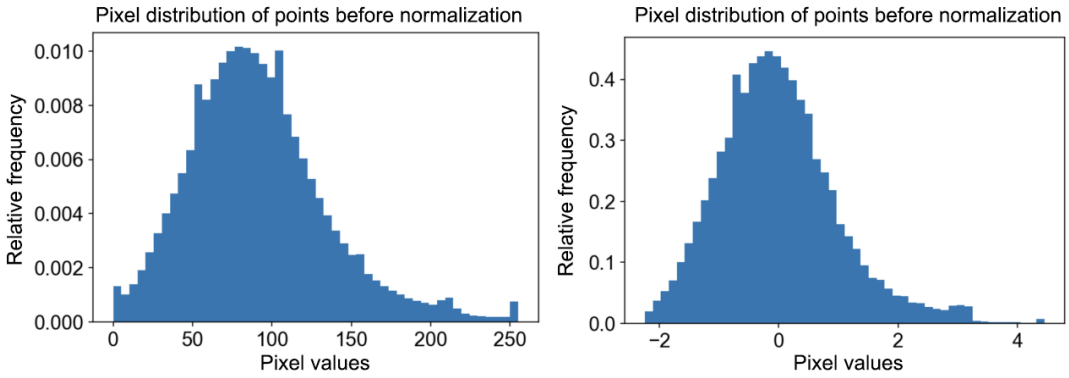

Figure 9: Pixel Distribution Comparison

3.2 Data Augmentation

Training of deep learning model depends on quality, quantity, and relevance of training data. However, insufficient data is the most common challenge in practical neural network training. For network models, it is laborious and expensive for data collection and labeling. But, cost can be reduced by transforming dataset with data augmentation technology.

Data augmentation is a regularization technology that generates new data based on existing data, to expand data amount. New examples are formed to train datasets, to enrich and even dataset in the model. This helps improve generalization ability of machine learning model, suppresses overfitting when neural network model is trained, and expands model generalization.

Since dataset used in experiment is too small, model will experience overfitting, seriously affecting model performance thereby. Consequently, it is urgent to expand dataset by data augmentation technology. Pytorch contributes methods to enhance images in deep learning, including geometric transformation, flipping, color modification, cropping, rotation, noise injection, and random erasure. Figure 10 lists an example of some data augmentation. In actual data augmentation, geometric transformations will be randomly superimposed. For training purpose, training set is expanded to 28,000 images.

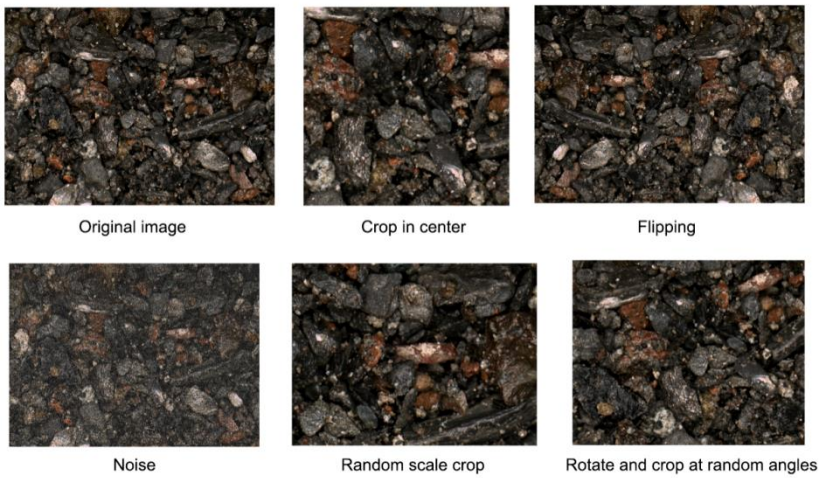

Figure 10: Data Augmentation Example

3.3 Dataset Classification

3.3.1 Training Set

Training set is dataset example in neural network; the network will adjust parameters to fit classifier according to results on training set. In recognition training, learning algorithm consults training set to adjust or learn parameters that are likely to form a good prediction model. Optimization algorithm aims to capture a neural network trained to fit a classifier. At the same time, the network can be effectively applied to new data, or in other words, it boasts of good generalization ability. Other datasets evaluate fitting effect of the model, so as to estimate model accuracy on new data.

For purpose of ensuring sufficient amount of data, images in training set are expanded to 28,000 through data augmentation; samples of 7 rocks are evenly increased to 4000 images respectively.

3.3.2 Validation Set

Validation set evaluates performance of specified model, and it frequently examines performance of model on validation set. Hyperparameters are adjusted slightly in view of the performance. In summary, validation set is set for developers rather than neural network to learn; the data is not involved in the training. As a result, validation set, also known as development set, affects training only indirectly.

With the aid of data augmentation, images in validation set are expanded to 2100, with 300 for each rock.

3.3.3 Test Set

Model performance on test set is mainly used to evaluate results, but this standard shall not be used unless model is trained. Test set is specially prepared. It contains carefully sampled data that describes all possible conditions of model in real work. In a word, test set reflects true level of the model in practical application.

In general, there is no need to augment data in test data, with 35 images in total.

Training set accounts for 92.9% of the dataset, validation set for 7%, and test set for 0.1%, as shown in Figure 11.

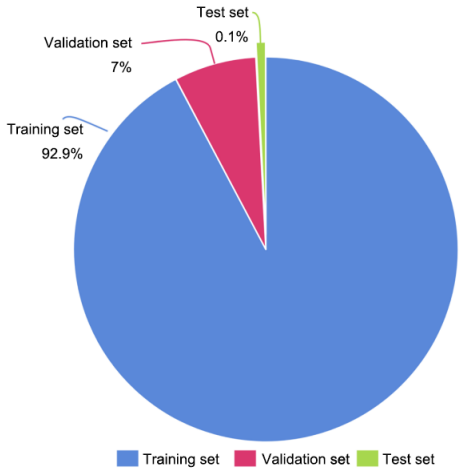

Figure 11: Dataset Composition Diagram

4. Conclusion

Deep learning is able to extract deeper abstract features of data efficiently. In image recognition field, images serve as input of deep learning, and they essentially are pixels represented by data matrix. Through feature extraction of deep learning, location, edge and texture features of images are further displayed. Based on highly abstract features, computers identify objects specified in images. Representation learning aims at simplifying data, ignoring insignificant and useless information, and extracting useful information. Therefore, dataset quality is particularly meaningful.

Dataset refers to a set of data instances with common properties. The quality is decided by accuracy of data annotation, data amount, and data collection environment. A high-quality dataset makes deep learning programs more efficient and keeps deep learning from overfitting. However, the cost is higher for obtaining high-quality deep learning datasets. Under the background, this study, based on image processing technologies, analyzes defects of open-source datasets and data preprocessing schemes on low-quality data sets via image processing algorithms provided by Pytorch. In practical model training, it is hopeful to effectively avoid the fact that deep learning program is affected by low-quality datasets, in order to guarantee accuracy of deep learning programs.

References

[1] Guo Chao, Liu Ye. Research on rock image recognition in multi-color space [J]. Science, Technology and Engineering, 2014 (18): 247-251.

[2] Cheng, Guojian, Wenhui Guo. "Rock images classification by using deep convolution neural network." Journal of Physics: Conference Series. Vol. 887. No. 1. IOP Publishing, 2017.

[3] Młynarczuk, Mariusz, Andrzej Górszczyk, and Bartłomiej Ślipek. "The application of pattern recognition in the automatic classification of microscopic rock images." Computers & Geosciences. 60 (2013): 126-133.

[4] Bai Lin, Wei Xin, Liu Yu, Wu Chongyang, Chen Lihui, Rock thin section image recognition and classification based on VGG model [J]. Geological Bulletin of China, 2019, 38(12):2053-2058.