Stats 302 23-S3
Prof. Schröter
Jingheng Huan

Kaggle Project Report: Sandstone Classification

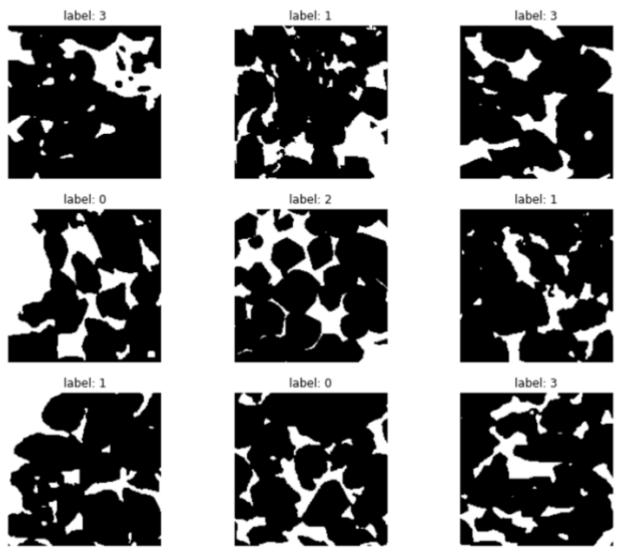
Introduction of Sandstone Classification

Geology tries to comprehend how the Earth's surface and subsurface were formed and evolved. Sand, minerals, and other particles are compressed into layers to form sedimentary rocks known as sandstones. Based on their geological period and formation history, sandstones have various qualities and traits. In this Kaggle project, I want to find out if the geometry of the pore volume inside a matrix of sintered sand grains sufficiently preserves these variations so that one can identify and categorize the kind of sandstone from a two-dimensional photograph of a two-dimensional sandstone slice.

Data Description

Four different varieties of sandstone, including Bentheim (B49), Berea (Br46), Fontainebleau (F57), and Gildehausen, make up the dataset for this Kaggle project (G44). Computed X-ray Tomography (CT) was used to record the raw data as three-dimensional volumes, from which two-dimensional slices with a size of 150 by 150 pixels were then extracted. The test data from the Kaggle website consists of 830 photos, whereas the training data consists of 3320 images and their accompanying labels.

Figure 1: Preview First Nine Training Images of the Sandstone



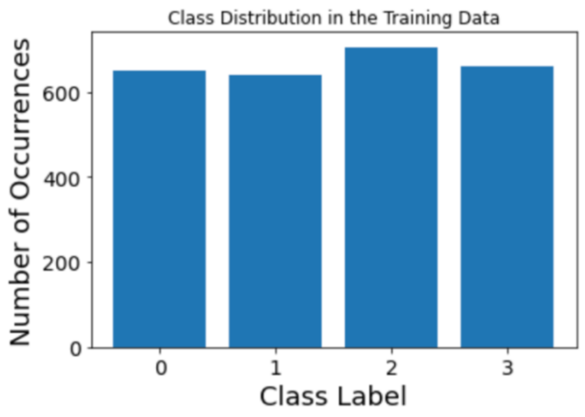
Methodology

In this project, I used a convolutional neural network (CNN) to classify and predict the type of sandstone based on a 2D image of a sandstone slice. The following steps were performed to solve the problem:

1: Load the training data and test data using `np.load()`

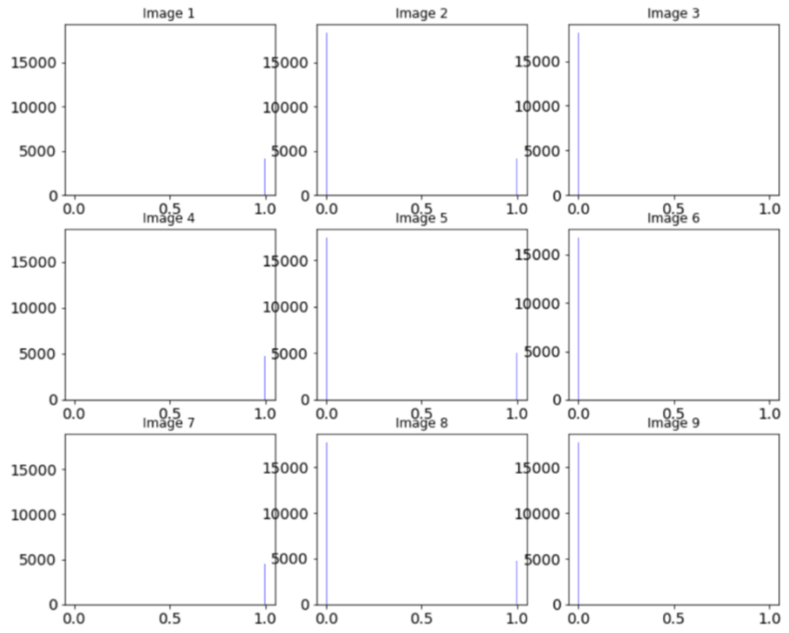
2: Split the training data into training and validation sets using `train_test_split()` and set the `test_size` to be 0.2
3: Analyze the class distribution in the training data by drawing a bar plot below, and I find that the dataset is not imbalanced.

Figure 2: Class Distribution in the Training Data



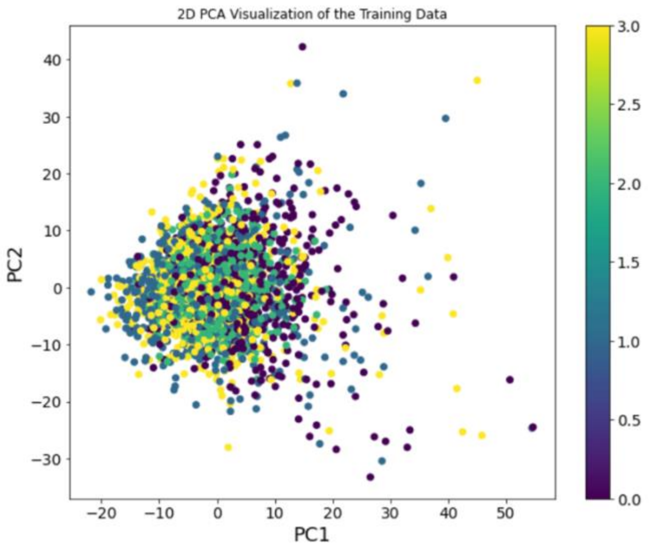
4: Analyze the pixel values of the training images.

Figure 3: The Plot of the Pixel Values of First 9 Training Images



5: Utilize the training data for PCA (Principal Component Analysis) visualization. The PCA visualization, which can be used to see patterns and relationships in the data, is then a plot of the transformed data in this lower-dimensional space. In the instance of this study, a 2D PCA representation of the training data was made to determine whether any distinguishable clusters of data points pertaining to various types of sandstone could be identified.

Figure 4: 2D PCA Visualization of the Training Data



6: Preprocess the training and validation data by using `reshape()` and `to_categorical`
7: Augment the training data by using `ImageDataGenerator` and set the horizontal and vertical flip to be `true` and set the `shear_range` and `zoom_range` to be 0.1.
8: Define the CNN architecture, I create a 4 layers model, and use `categorical_crossentropy` as loss function and set `optimizer` to be `Adam`. Also, I try to use transfer learning by applying VGG-16 model, however, the result is not that ideal so I change it back to CNN model.
9: Compile and train the model, using at least 200 epochs.
10: Make predictions for the test data.
11: Plot the accuracy and loss of the model during the training process.
12: Create a submission csv file and submit it on Kaggle.

Figure 5: Plot of the Training and Test Accuracy of each epoch during Training Process

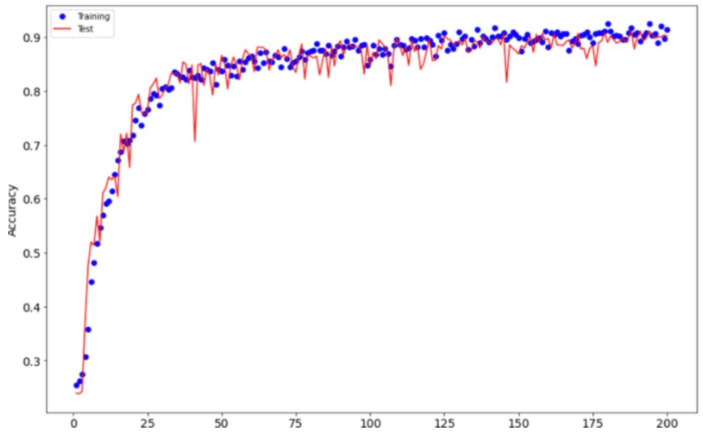
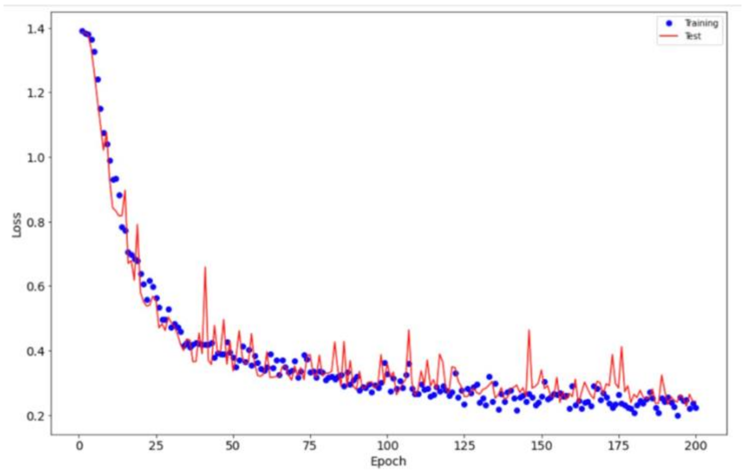


Figure 6: Plot of the Training and Test Loss of each epoch during Training Process



From two plots above, we can find that the test accuracy is extremely close to the training accuracy and it means that my model is not much overfitting.

Results

The validation data showed a maximum 0.9137 accuracy for the model during 200 epochs. As a result of the learning process, the model got better over time, with accuracy rising and loss falling. For 830 photos in the test set, the model was able to accurately identify the kind of sandstone, and the predictions were stored as a CSV file.

Several changes were made to the model during the optimization process to increase its accuracy.

1: The initial Convolutional Neural Network (CNN) model had four layers and a validation accuracy of roughly 0.82. The dropout rate of each layer was altered to enhance the performance of the model even more.

2: It was shown that the validation accuracy significantly increased when the fourth layer's dropout rate was raised over 0.5.

3: The MaxPooling layers' pool size was also changed, although it was discovered that this had little effect on the validation accuracy. However, the validation accuracy suddenly increased to about 0.88 when a GlobalAveragePooling2D layer was added before the Flatten layer.

The results of the optimization procedure demonstrated the model's robustness and lack of overfitting. The number of epochs was extended to 200 after it was discovered that the training accuracy converged slowly. Finally, the accuracy of the result was about 0.924 when the improved model was uploaded to Kaggle.

In summary, the optimization process showed how crucial it is to carefully tweak the model's hyperparameters and the effect they have on performance. The outcomes of this optimization method demonstrate CNNs' potential to handle challenging picture categorization problems.

What I did to demonstrate my ingenuity and critical thinking

It was difficult to estimate the type of sandstone from a 2D image of a sandstone slice without a thorough comprehension of the data and approaches. I employed a deep learning strategy—specifically, a CNN, which has been successfully used to picture classification tasks—to tackle this issue. I used data augmentation to broaden the diversity of the training data and avoid overfitting in order to enhance the model's performance. Besides, I utilized PCA to display the data in two dimensions and learn more about how the data were distributed.

What I did to demonstrate the accuracy and readability of the analysis and code
My well-written, well-commented, and well-organized code makes it simple to follow the actions taken to resolve the issue. My code is written in a simple and straightforward manner, and it is effective and efficient thanks to the usage of libraries and functions from the numpy, sklearn, keras, and matplotlib libraries. In addition to the code, there are visuals and charts that aid in understanding the outcomes and model learning.

Conclusion
As a result, the challenge of categorizing various sandstone varieties based on their two-dimensional images has been successfully solved. A Convolutional Neural Network (CNN) was trained on the training data after the dataset of 3320 photos was divided into a training and validation set. The model learned the characteristics that set the four varieties of sandstones apart and obtained a validation set accuracy of over 89% on average of each epoch. The predictions were saved to a submission file in the appropriate format once the model had been further assessed on the test set.
The outcomes of this experiment show how well CNNs perform tasks involving image categorization and their capacity to identify intricate patterns in the data. The study also emphasizes the significance of feature engineering and how it helps to enhance the effectiveness of machine learning models.
It would be fascinating to investigate other models and methods in the future in an effort to increase the classification's precision. Better outcomes might also follow from including three-dimensional data regarding the sandstones.