**Rock Identification using Deep Convolution Neural Networks & IBM Watson**

**Table of content:**

|  |  |
| --- | --- |
| **1.** | **INTRODUCTION** |
| **2.** | **LITERATURE SURVEY** |
| **3.** | **THEORITICAL ANALYSIS** |
| **4.** | **EXPREIMENTAL INVESTIGATIONS** |
| **5.** | **FLOW CHART** |
| **6.** | **RESULT** |
| **7.** | **ADVANTAGES** |
| **8.** | **APPLICATIONS** |
| **9.** | **CONCLUSION** |
| **10.** | **FUTURE SCOPE** |
| **11.** | **BIBLIOGRAPHY** |
| **12.** | **APPENDIX** |

**1.Introduction**

**Overview:**

Rocks are a fundamental component of Earth. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping. It is a basic part of geological surveying and research, and mineral resources exploration. They contain the raw materials for virtually all modern construction and manufacturing and are thus indispensable to almost all the endeavors of an advanced society. In addition to the direct use of rocks, mining, drilling, and excavating provide the material sources for metals, plastics, and fuels. Natural rock types have a variety of origins and uses. The three major groups of rocks (igneous, sedimentary, and metamorphic) are further divided into sub-types according to various characteristics. It is an important technical skill that must be mastered by students of geoscience.

In this project rock classification we are going to find the type of rock such as 'Blue Calcite', 'Limestone', 'Marble', 'Olivine' and 'Red Crystal' were we are using CNN model to analyze the type of rock. The objective of the project is to build a web application to detect the type of the rock. The model takes input from the user and compares it with the pre trained model and the rock type is classified and showcased on the UI along with its chemical composition.

**Purpose:**

The main aim of this project is to create a model based on rock identification using cnn. In this project we are using convolution neural networks for the accurate prediction. An application is also build which can be interlinked with the model so as to view the result on UI based on the input parameters.

**2. literature Survey**

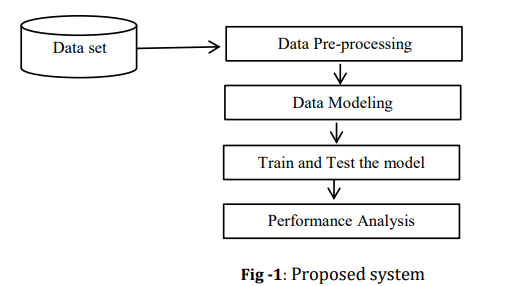
**Existing problem:**

We all know that rocks are a fundamental component of Earth. Rocks have huge variations in their composition physically and chemically. In order to identify the type of rock during mining, agriculture it involves a lot of research and time consuming process. So to know about the type of rock we built a solution using deep convolution neural networks.

**Proposed solution:**

Applying convolution neural networks to rock dataset to identify the type of rock. Machine learning is a method of data analysis which sends instructions (programmable code) to computers so that they can learn from data. Then, based on the learned data, they provide us the predicted results/patterns.

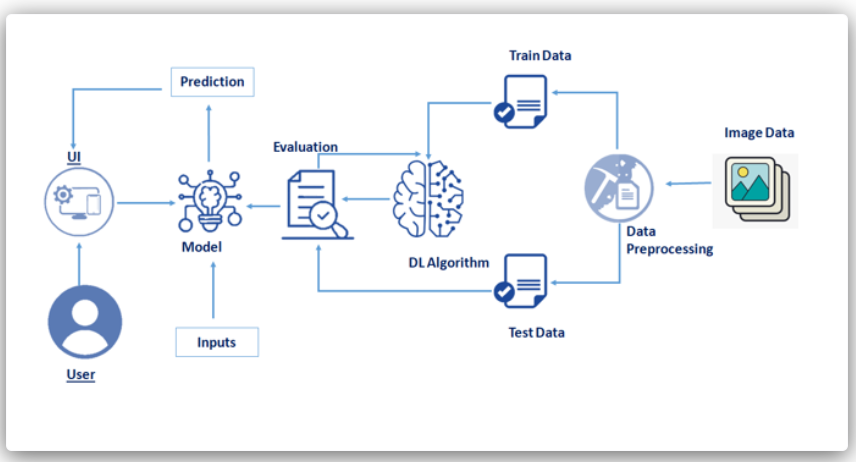
The proposed system was implemented using CNN model to classify the rocks. The steps involved in the proposed system is given in Fig 1.The proposed system has five modules. Viz Data collection, Data Pre-processing, Data Modeling, Train and Test the model and Performance Analysis



* Data Collection.
  + Collect the dataset or Create the dataset
* Data Preprocessing.
  + Import the ImageDataGenerator library
  + Configure ImageDataGenerator class
  + Apply ImageDataGenerator functionality to Trainset and Testset
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Adding Input Layer
  + Adding Hidden Layer
  + Adding Output Layer
  + Configure the Learning Process
  + Training and testing the model
  + Optimize the Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build Python Code

**3.** **Theoretical Analysis**

**Block Diagram:**



**Hardware/Software Designing:**

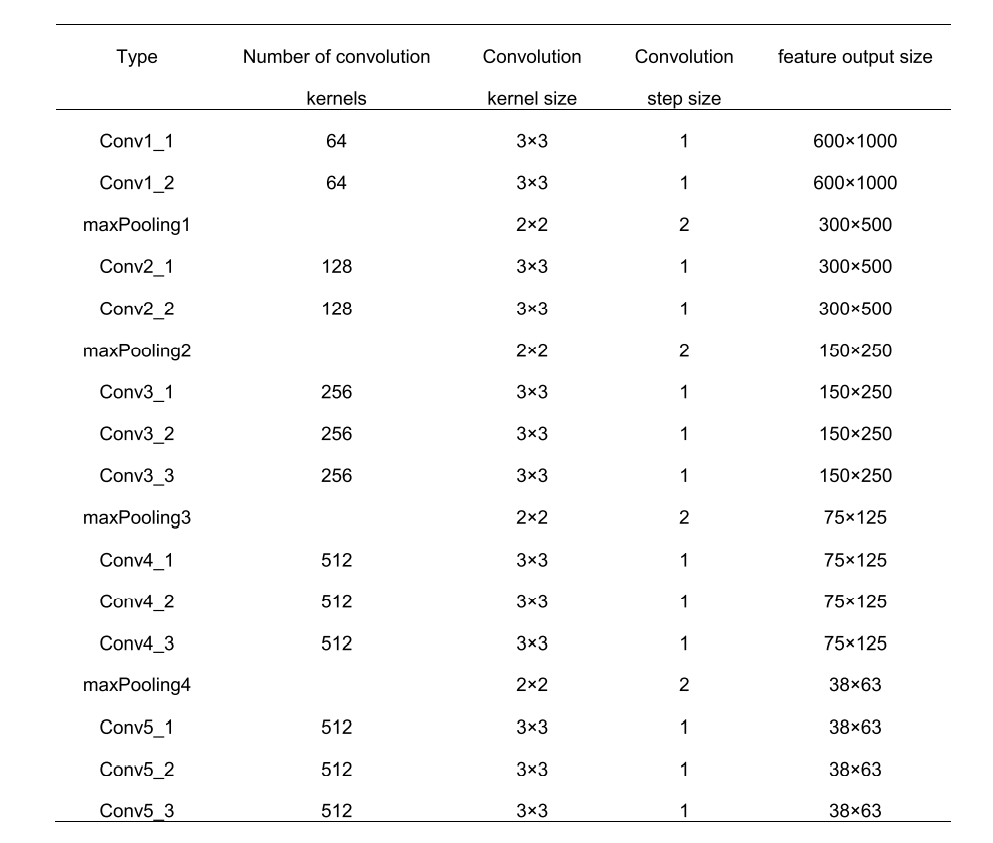
we need basic software skills like python,python web frame works,python for data analysis,python for data visualisation,machine learning for developing this project and to identify the type of rock using cnn and ibm watson.

**4.Experimental Investigations**

**Abstract:**

Fast and reliable identification of rock is important for various applications like geological prospecting, mineral and engineering sciences, physics and other analytical studies. rocks can be classified in a variety of ways, such as visually under a microscope, or by chemical analysis. The main objective of the project is to identify and categorize different types of rocks using computational method. In this project, Convolutional Neural Network(CNN) model is used to classify the rocks. The VGG-16 is one of the most popular CNN models for image classification. The overall performance shows 97% of accuracy for rock classification.

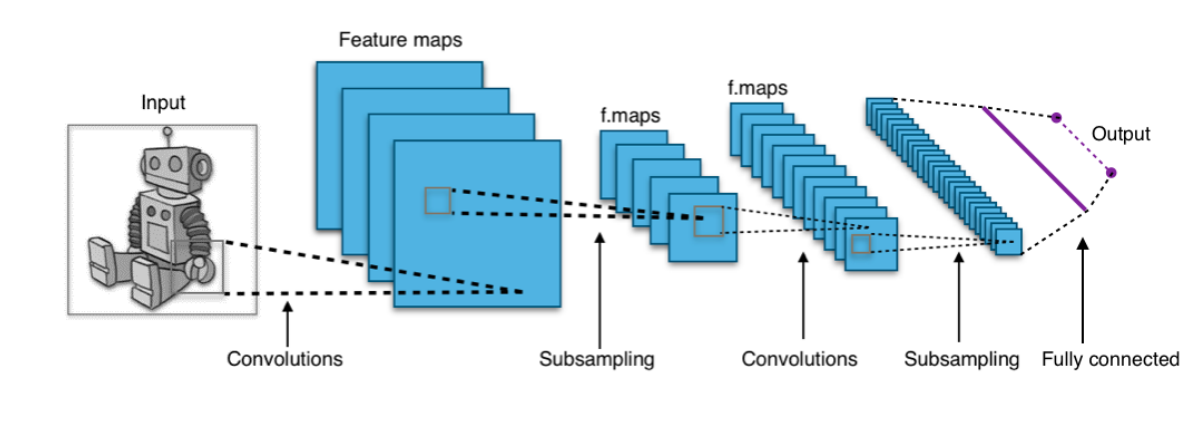
**Simplified VGG16 structural parameters:**

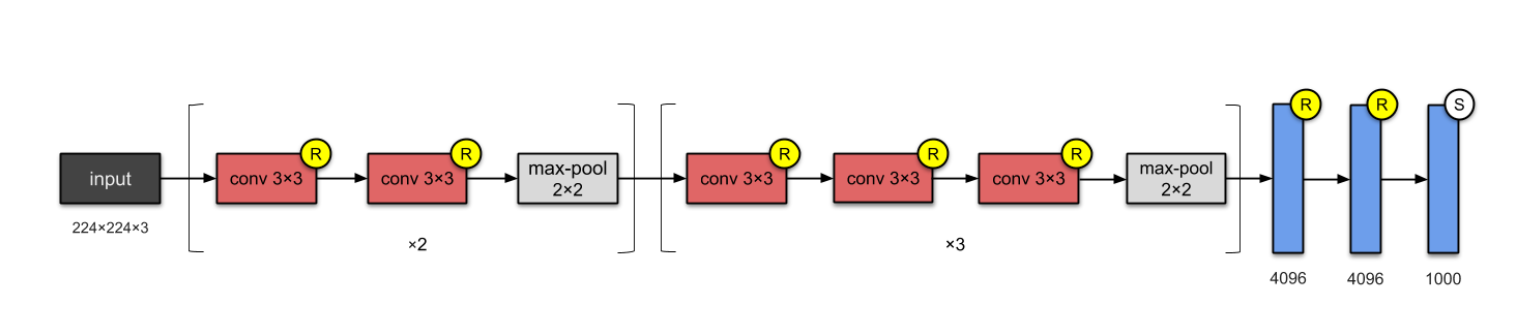
****

**original image data:**

|  |  |  |
| --- | --- | --- |
| Lithology | Class no | Rock sample images |
| Blue calcite | 01 |  |
| Lime stone | 02 |  |
| marble | 03 |  |
| olivine | 04 |  |
| Red crystal | 05 |  |

**5. Flow Chart**



****

**6. Performance Analysis**

Classification report is used to measure the predictions of classification algorithm. This shows classification metrics precision, recall on a per-class basis. The classification accuracy is one of the performance metric used in rock classification. These performance metrics are calculated by using true and false positives, true and false negatives. Precision is a metric that quantifies the number of correct positive rock image predictions made.

Precision = TP / TP + FP.

Recall is a metric that quantifies the number of correct positive rock image predictions made out of all positive rock image predictions made.

Recall = TP / TP + FN.

Accuracy is the number of correct rock image predictions made divided by the total number of rock images predictions made.

Accuracy = (TP+TN)/(TP+TN+FP+FN).

Where, True Positive (TP) is the number of true rock images which are correctly extracted,

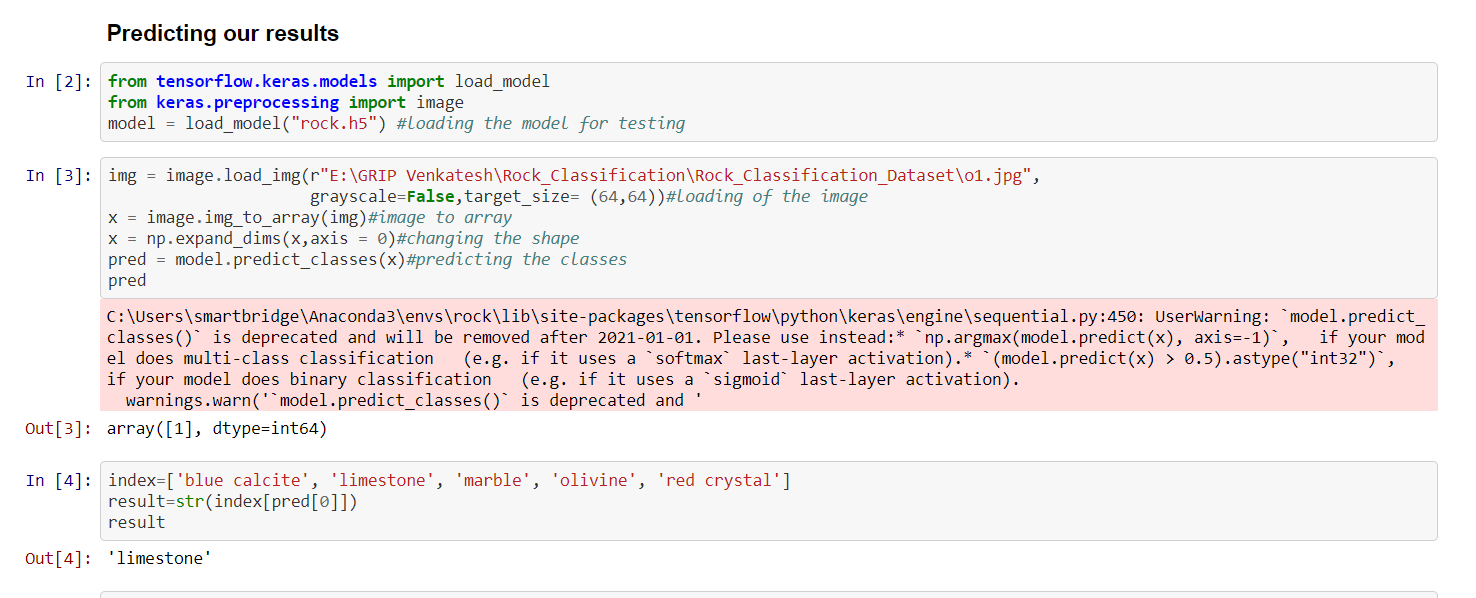
True negatives(TN) is the number of true rock images which are incorrectly identified,

False Positive (FP) is the number of false rock images correctly identified,

False Negative (FN) is the number of false rock images incorrectly identified

**7. Result**

Finally, as a result for this project we will predict the type of rock using deep convolution neural networks. And the result of the random inputs is displayed below with a screenshot.



**8. Advantages**

1.The rock identification method is compared with the traditional method and the method based on rock slice image processing.

2.The presented method can quickly get the recognition results in less than 1 second after taking photos in the field.

3.It is suitable for rapid and accurate recognition of rock lithology under field offline conditions.

**9. Applications**

* Agriculture
* Geology
* Archeology
* Oceanography
* Forestry
* Land cover and land use

**10. Conclusion**

The analysis was carried out on rock identification dataset. The rock classification is done using VGG16 model and performance of the model was analyzed. The classification report shows the precision, recall and f-score values per class basis. The proposed model yields the average precision of 86% and average recall of 89%. The overall performance shows 95% of accuracy for rock classification using VGG16 model. In the future, improvement in the accuracy of classification by using a larger dataset that consists of more number of rock classes may be considered.

**11.Future Scope**

The transfer learning method was used to train the rock recognition model on the PC, and the trained model was deployed on the smartphone.This developed an application program that runs on a smartphone.It also makes up for the defect that the methods based on the image processing and feature extraction of rock thin section cannot recognize the rock quickly and accurately in the field. Geological investigators can quickly and accurately identify rocks by using their smartphones in the field, which is of great help to geological surveys. In the future, this project needs to compare the rock recognition model based on ShuffleNet with more models trained by the lightweight convolutional neural network. In order to improve the accuracy and efficiency of the method, more different kinds of rock training samples were added.This allows the geologist to identify the rock, and, in the process, to learn about its history and the geological environment in which it was formed.

**12.Bibliography**

1. <https://iopscience.iop.org/article/10.1088/1742-6596/887/1/012089/pdf>

2. <https://www.mdpi.com/2227-7390/7/8/755/pdf>

3. <https://ieeexplore.ieee.org/iel7/6287639/6514899/08964384.pdf>

4. <https://www.hindawi.com/journals/misy/2020/7462524/>

5. <https://www.irjet.net/archives/V8/i2/IRJET-V8I2278.pdf>

6. <https://www.sciencedirect.com/science/article/pii/S1674987120300931>

7. <http://www.wcse.org/WCSE_2019_SUMMER/W052.pdf>

**13.Appendix**

**Source code:**

# Detecting Building defects using CNN

### Importing Neccessary Libraries

In [2]:

**import** numpy **as** np*#used for numerical analysis*

**import** tensorflow *#open source used for both ML and DL for computation*

**from** tensorflow.keras.models **import** Sequential *#it is a plain stack of layers*

**from** tensorflow.keras **import** layers *#A layer consists of a tensor-in tensor-out computation function*

*#Dense layer is the regular deeply connected neural network layer*

**from** tensorflow.keras.layers **import** Dense,Flatten

*#Faltten-used fot flattening the input or change the dimension*

**from** tensorflow.keras.layers **import** Conv2D,MaxPooling2D *#Convolutional layer*

*#MaxPooling2D-for downsampling the image*

**from** keras.preprocessing.image **import** ImageDataGenerator

​

### Image Data Agumentation

In [3]:

*#setting parameter for Image Data agumentation to the training data*

train\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255,shear\_range**=**0.2,zoom\_range**=**0.2,horizontal\_flip**=True**)

*#Image Data agumentation to the testing data*

test\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255)

In [ ]:

​

### Loading our data and performing data agumentation

In [4]:

*#performing data agumentation to train data*

x\_train **=** train\_datagen.flow\_from\_directory(

r'C:\Users\botla\Desktop\Rock\_Classification\_Dataset\Rock\_Classification\_Dataset\train\_set',

target\_size**=**(64, 64),batch\_size**=**5,color\_mode**=**'rgb',class\_mode**=**'categorical')

*#performing data agumentation to test data*

x\_test **=** test\_datagen.flow\_from\_directory(

r'C:\Users\botla\Desktop\Rock\_Classification\_Dataset\Rock\_Classification\_Dataset\test\_set',

target\_size**=**(64, 64),batch\_size**=**5,color\_mode**=**'rgb',class\_mode**=**'categorical')

Found 700 images belonging to 5 classes.

Found 300 images belonging to 5 classes.

In [5]:

print(x\_train.class\_indices)*#checking the number of classes*

{'blue calcite': 0, 'limestone': 1, 'marble': 2, 'olivine': 3, 'red crystal': 4}

In [6]:

print(x\_test.class\_indices)*#checking the number of classes*

{'blue calcite': 0, 'limestone': 1, 'marble': 2, 'olivine': 3, 'red crystal': 4}

In [7]:

**from** collections **import** Counter **as** c

c(x\_train .labels)

Out[7]:

Counter({0: 140, 1: 140, 2: 140, 3: 140, 4: 140})

### Creating the model

In [8]:

*# Initializing the CNN*

classifier **=** Sequential()

​

*# First convolution layer and pooling*

classifier.add(Conv2D(32, (3, 3), input\_shape**=**(64, 64, 3), activation**=**'relu'))

classifier.add(MaxPooling2D(pool\_size**=**(2, 2)))

*# Second convolution layer and pooling*

classifier.add(Conv2D(32, (3, 3), activation**=**'relu'))

*# input\_shape is going to be the pooled feature maps from the previous convolution layer*

classifier.add(MaxPooling2D(pool\_size**=**(2, 2)))

​

*# Flattening the layers*

classifier.add(Flatten())

​

*# Adding a fully connected layer*

classifier.add(Dense(units**=**128, activation**=**'relu'))

classifier.add(Dense(units**=**5, activation**=**'softmax')) *# softmax for more than 2*

​

​

In [9]:

classifier.summary()*#summary of our model*

### Compiling the model

In [10]:

*# Compiling the CNN*

*# categorical\_crossentropy for more than 2*

classifier.compile(optimizer**=**'adam', loss**=**'categorical\_crossentropy', metrics**=**['accuracy'])

## Fitting the model

In [11]:

classifier.fit\_generator(

generator**=**x\_train,steps\_per\_epoch **=** len(x\_train),

epochs**=**50, validation\_data**=**x\_test,validation\_steps **=** len(x\_test))*# No of images in test set*

**Saving our model**

In [13]:

*# Save the model*

classifier.save('rock.h5')

In [14]:

model\_json **=** classifier.to\_json()

**with** open("model-bw.json", "w") **as** json\_file:

json\_file.write(model\_json)

In [ ]:

**Predicting our results**

In [15]:

**from** tensorflow.keras.models **import** load\_model

**from** keras.preprocessing **import** image

model **=** load\_model("rock.h5") *#loading the model for testing*

In [16]:

img **=** image.load\_img(r"C:\Users\botla\Desktop\Rock\_Classification\_Dataset\Rock\_Classification\_Dataset\train\_set\marble\marble15.jpg",

grayscale**=False**,target\_size**=** (64,64))*#loading of the image*

x **=** image.img\_to\_array(img)*#image to array*

x **=** np.expand\_dims(x,axis **=** 0)*#changing the shape*

pred **=** model.predict\_classes(x)*#predicting the classes*

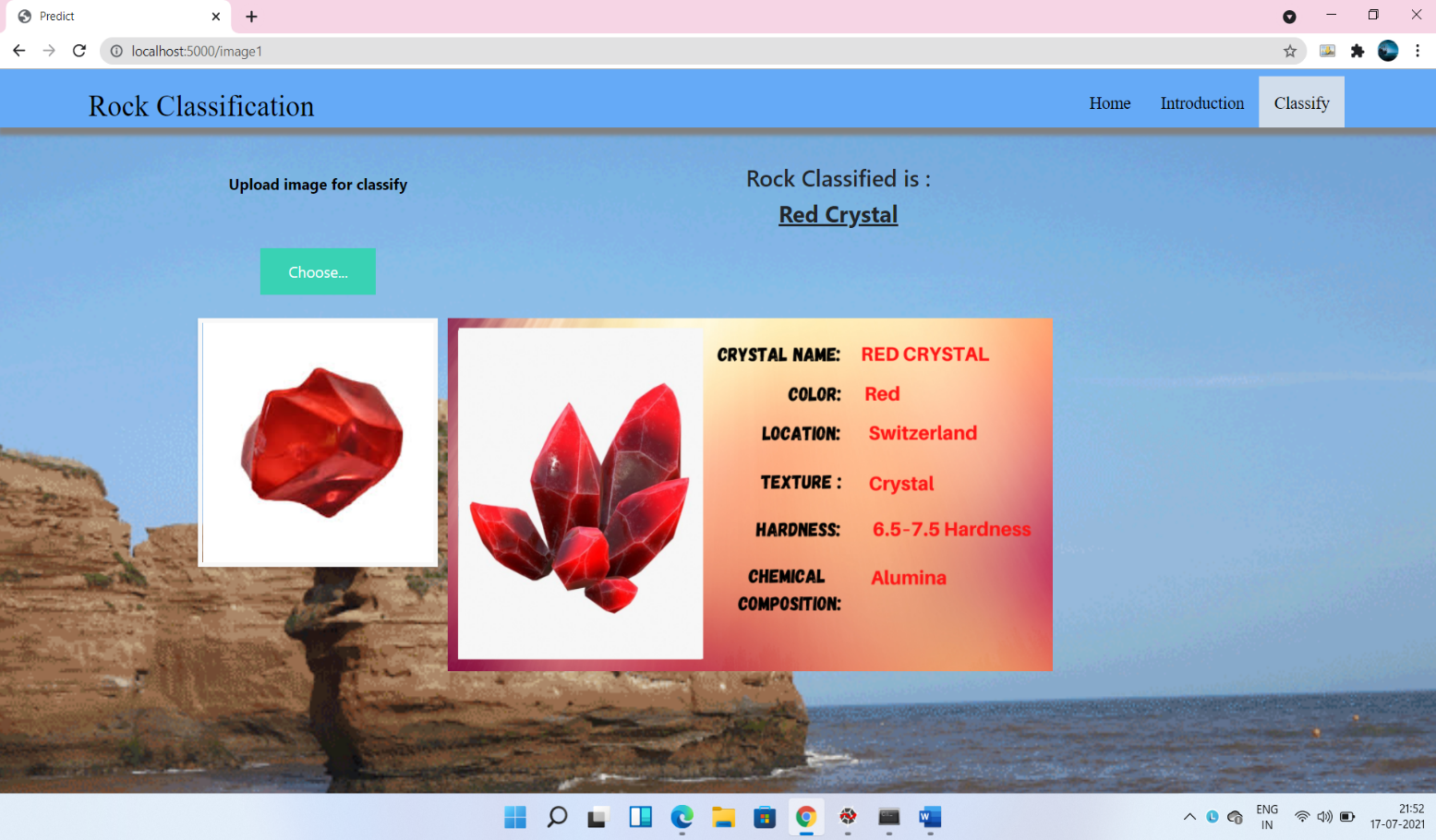
pred

index**=**['blue calcite', 'limestone', 'marble', 'olivine', 'red crystal']

result**=**str(index[pred[0]])

result

**UI output Screenshot:**

****

