Rock Identification Using Deep Learning CNN Project Report

Submitted By:
Abinaya S
Anusha S
Elamathi J
Indhumathi C

Abstract:

Granularity analysis is one of the most essential issues in authenticate under microscope. To improve the efficiency and accuracy of traditional manual work, an convolutional neural network based method is proposed for granularity analysis from thin section image, which chooses and extracts features from image samples while build classifier to recognize granularity of input image samples. 4800 samples from Ordos basin are used for experiments under color spaces of HSV, YCbCr and RGB respectively. On the test dataset, the correct rate in RGB color space is 98.5%, and it is believable in HSV and YCbCr color space. The results show that the convolution neural network can classify the rock images with high reliability.

1.Introduction:

The traditional method for rock classification is a manual work with many problems such as time-consuming and low accuracy. With the development of science and technology, Artificial intelligence is successfully applied in all walks of life. Many domestic and foreign scholars have done researches in the automatic classification of rock images. Hossein Izadi et al.[5] established a neural network to identify the rock mineral, whose accuracy was 93.81%. The above methods show that the application of machine learning in rock classification can improve its efficiency and accuracy. Convolution neural network (CNN) is an important deep learning architecture. It can extract the image features automatically and has a high classify accuracy. In this paper, we construct a new convolution neural network for rock classification, rock images.

1.1Overview:

Rock identification using traditional method is time taken process to identify the rocks. Using deep convolutional neural networks we can accurately identify the rock types based on image analysis. In this paper, we construct a new convolution neural network for rock classification, rock images.

1.2 Purpose:

This project is capable of classifying images based on the type of rock. A convolutional neural network (CNN) convolves an input image which identifies the type of rock. There will be a web application through which user can give their input image then they can check the predicted output and this application is integrated with trained CNN model.

2. Literature survey:

2.1 Existing problem:

The main aim of the project is to classify the rocks. So my task is to create an automated analysis system capable of identifying the rocks.

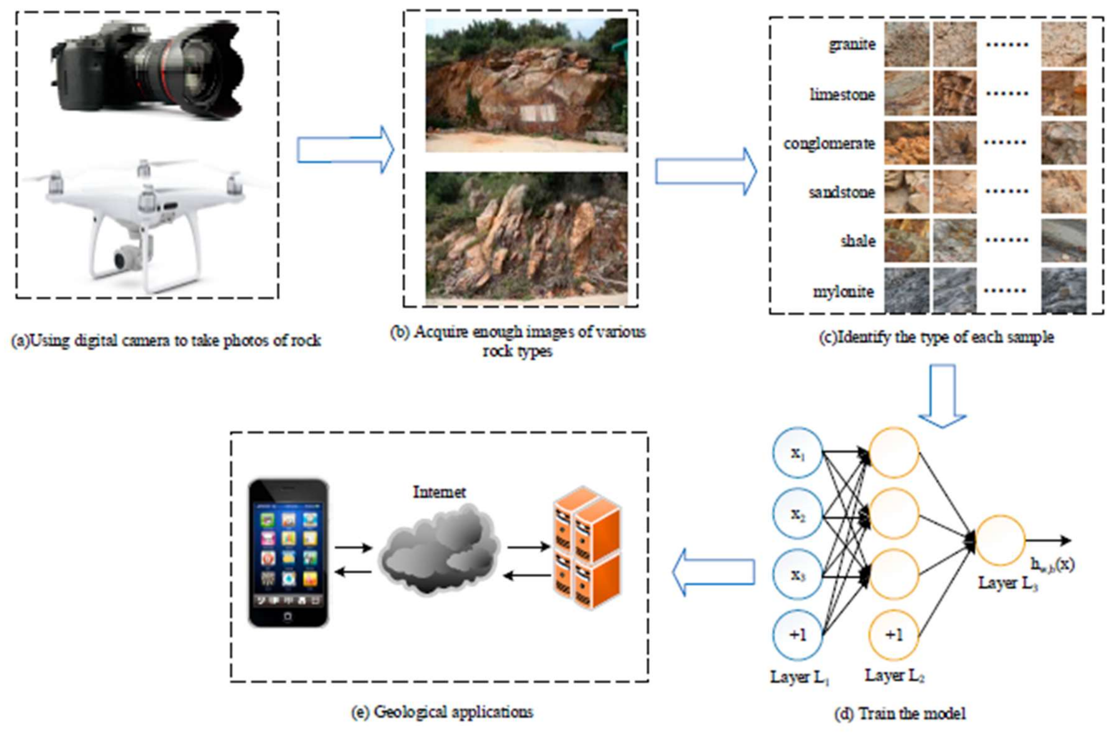
2.2 Proposed solution:

First module is optimization model from which the maxima and minima pixel values are selected and is gives to next module. Second module is similarity measures from where the pixel similarity calculated based on location or color. Third module is computational complexity from which data and time complexity measured. Fourth one is the application to collection of images in which we not only taking single image segmentation, our method implements to multiple images. Fifth module is deep learning neural network (DLNN) which provides complete tuning of image and predicting the type of rock. In this module training the image with labeled data and un labeled data because of Deep learning could satisfies to both supervised segmentation and un-supervised segmentation. Last module is the performance evaluation from this module CPU performance time for running each image and accuracy calculated based on PSNR.

Solution proposed by D. Thompson, S. Niekum, T.Smith, and D.Wettergreen.

3.Theoretical Analysis:

3.1 Block Diagram:


(a)Using digital camera to take photos of rock
(b) Acquire enough images of various rock types
granite
limestone
conglomerate
sandstone
shale
mylonite
(c)Identify the type of each sample
x_1
x_2
x_3
+1
+1
Layer L_1
Layer L_2
Layer L_3
$h_{w,b}(x)$
(d) Train the model
Internet
(e) Geological applications

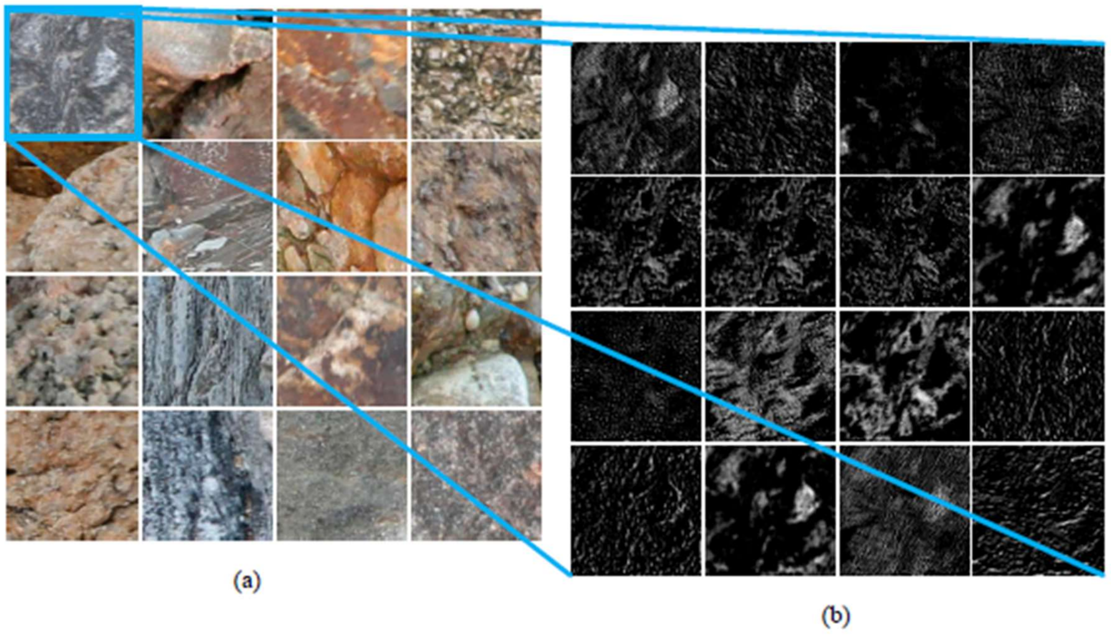
3.2 Hardware / Software designing:

For hardware we would need ICT tools such as digital camera**.** Here we use Digital Camera for capturing the rock image which is given as input to the web page.
For software we would need a compatible operating system for python, java script and HTML.

Software needed are:
1) Tensorflow
2) OpenCV
3) Keras
4) Flask

4. Experimental Investigation:

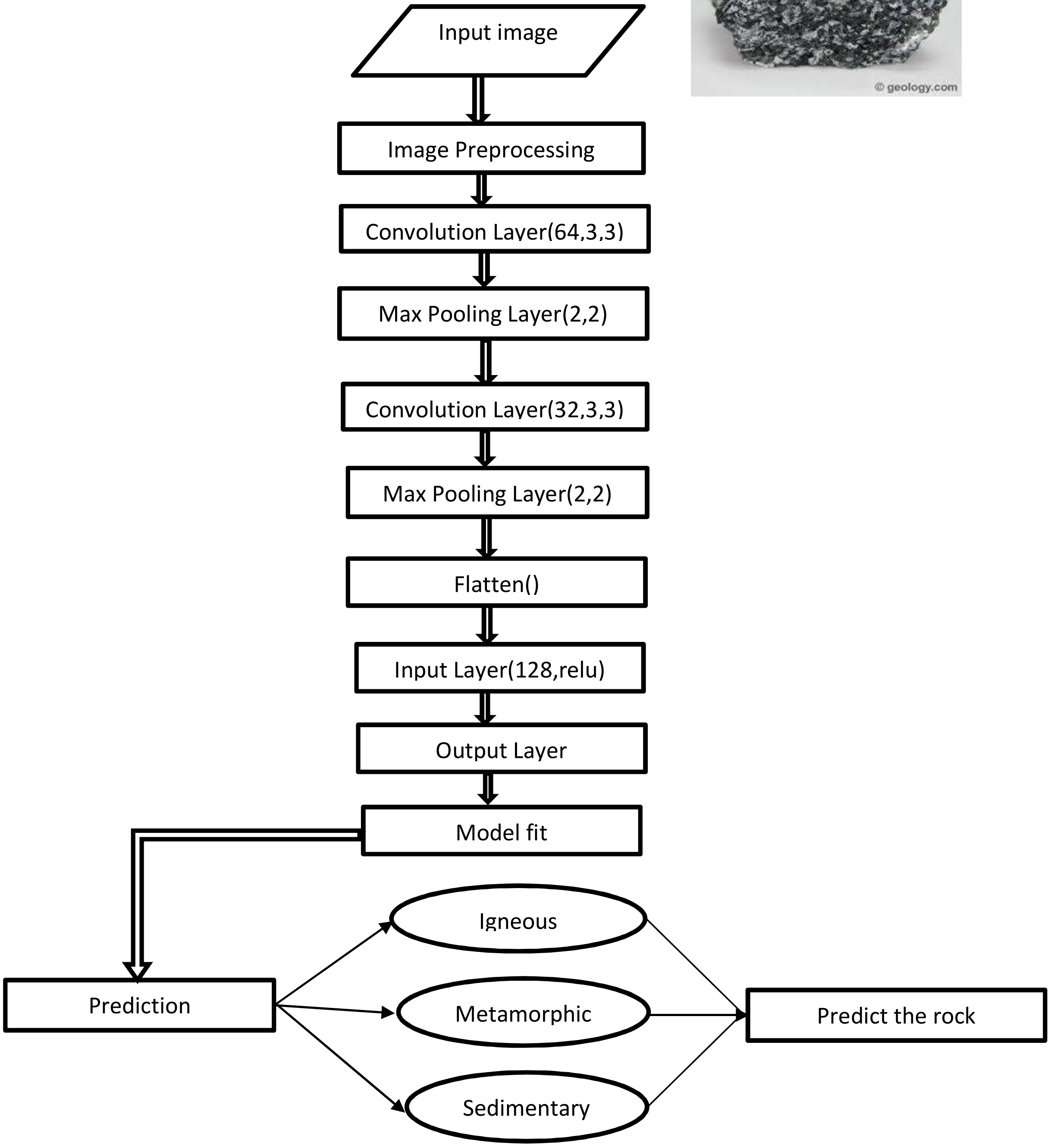
A convolution layer extracts the features of the input images by convolution and outputs the feature maps. It is composed of a series of fixed size filters, known as convolution kernels, which are used to perform convolution operations on image data to produce the feature maps.



(**a**) Input patched field rock sample images. (**b**) Outputted feature maps partly after the first convolution of the input image

5. Flow Chart:

© geology.com


Input image
Image Preprocessing
Convolution Layer(64,3,3)
Max Pooling Layer(2,2)
Convolution Layer(32,3,3)
Max Pooling Layer(2,2)
Flatten()
Input Layer(128,relu)
Output Layer
Model fit
Prediction
Igneous
Metamorphic
Sedimentary
Predict the rock

6. Result:

We got an accuracy of 0.96 which is good measure for a Convolution Neural Network. The Model predicts the type of rock with good efficiency.

7.Advantages & Disadvantages:

Advantages:

- Effective and predicts accurately.
- Predicts the type of rock using color, shape and size.
- Efficiency is maintained by updating the dataset frequently.

Disadvantages:

- The input image must have clarity to predict the correct output.
- Cost of the digital camera is high.
- High computational cost.

8.Application:

The traditional method for rock classification is a manual work with many problems such as time-consuming and low accuracy. Hence we use this project to predict the type of rock quickly and accurately.

9.Conclusion:

In this paper, CNN is used to identify the rock granularity. The experiments show that it has high reliability whether in HSV, YCbCr or RGB color space. In RGB color space, the classification accuracy achieves 96 with high efficiency. In view of the high reliability of the application of CNN in rock image classification based on rock granularity, it can be considered to apply to the classification of rock components.

10.Future scope:

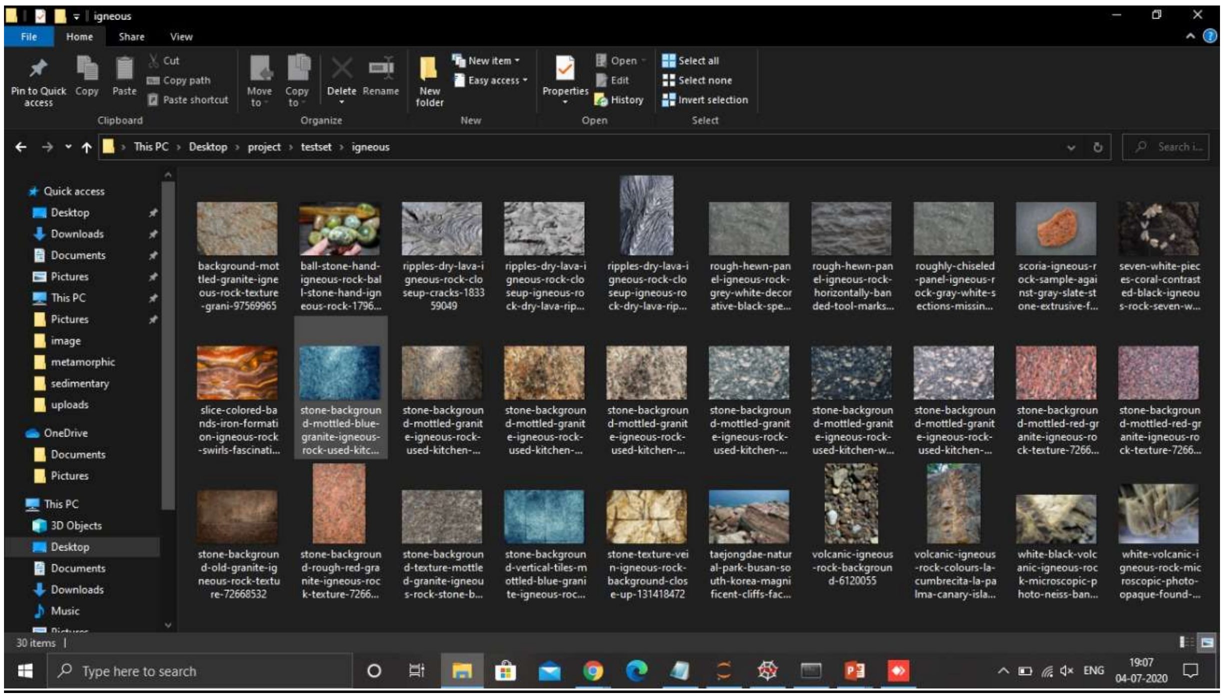
Rock identification using convolutional neural network minimizes the time of predicting the type of rock.

11.Bibliography:

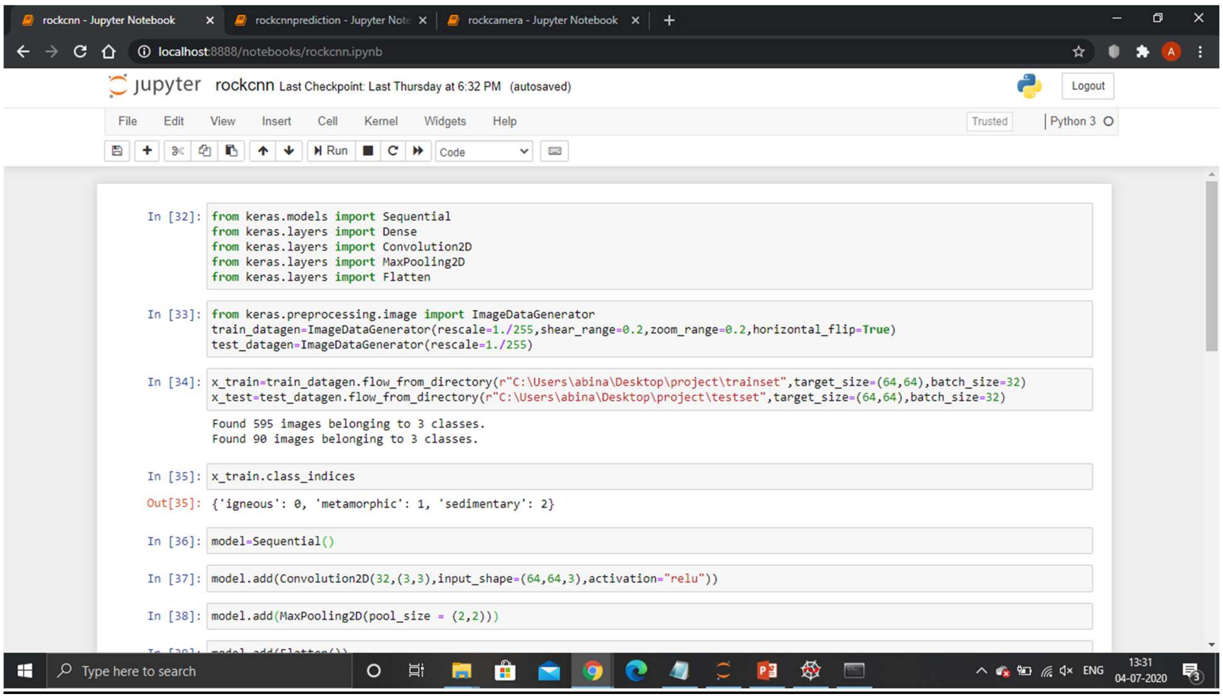
Proposed Idea:
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8964384

12.Appendix:

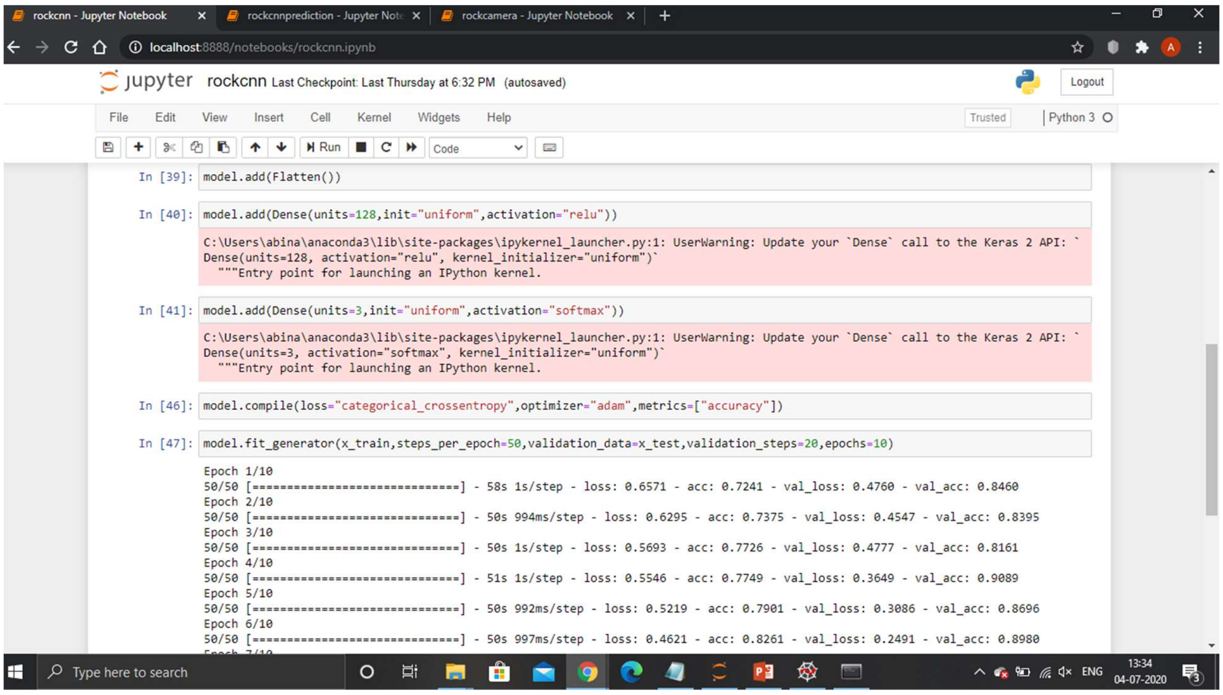
12.1 Dataset:


igneous
This PC > Desktop > project > testset > igneous
background-mottled-granite-igneous-rock-texture-grani-97569965
ball-stone-hand-igneous-rock-ball-stone-hand-igneous-rock-1796...
ripples-dry-lava-igneous-rock-closeup-cracks-183359049
ripples-dry-lava-igneous-rock-closeup-igneous-rock-dry-lava-rip...
ripples-dry-lava-igneous-rock-closeup-igneous-rock-dry-lava-rip...
rough-hewn-panel-igneous-rock-grey-white-decorative-black-spe...
rough-hewn-panel-igneous-rock-horizontally-banded-tool-marks...
roughly-chiseled-panel-igneous-rock-gray-white-sections-missin...
scoria-igneous-rock-sample-against-gray-slate-stone-extrusive-f...
seven-white-pieces-coral-contrasted-black-igneous-rock-seven-w...
slice-colored-bands-iron-formation-igneous-rock-swirls-fascinati...
stone-background-mottled-blue-granite-igneous-rock-used-kitc...
stone-background-mottled-granite-igneous-rock-used-kitchen-...
stone-background-mottled-granite-igneous-rock-used-kitchen-...
stone-background-mottled-granite-igneous-rock-used-kitchen-...
stone-background-mottled-granite-igneous-rock-used-kitchen-...
stone-background-mottled-granite-igneous-rock-used-kitchen-w...
stone-background-mottled-granite-igneous-rock-used-kitchen-...
stone-background-mottled-red-granite-igneous-rock-texture-7266...
stone-background-mottled-red-granite-igneous-rock-texture-7266...
stone-background-old-granite-igneous-rock-texture-72668532
stone-background-rough-red-granite-igneous-rock-texture-7266...
stone-background-texture-mottled-granite-igneous-rock-stone-b...
stone-background-vertical-tiles-mottled-blue-granite-igneous-roc...
stone-texture-vein-igneous-rock-background-close-up-131418472
taejongdae-natural-park-busan-south-korea-magnificent-cliffs-fac...
volcanic-igneous-rock-background-6120055
volcanic-igneous-rock-colours-la-cumbrecita-la-palma-canary-isla...
white-black-volcanic-igneous-rock-microscopic-photo-neiss-ban...
white-volcanic-igneous-rock-microscopic-photo-opaque-found-...
30 items
19:07
04-07-2020

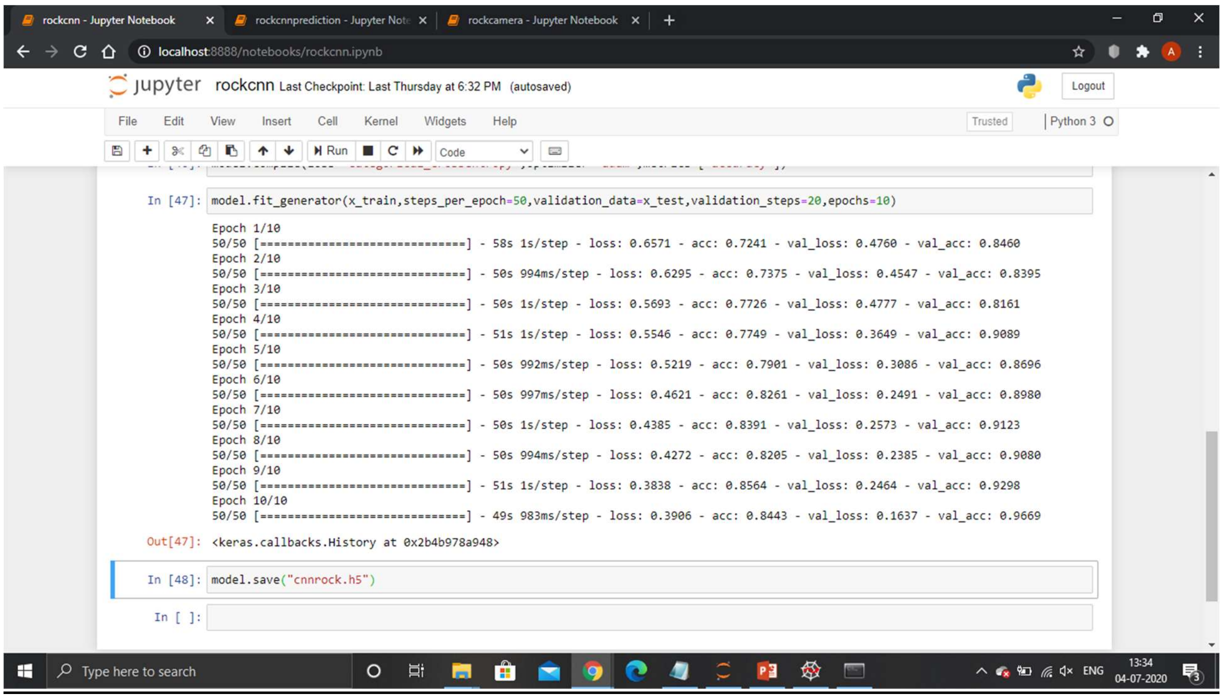
12.2 Model training:


rockcnn - Jupyter Notebook
rockcnnprediction - Jupyter Note
rockcamera - Jupyter Notebook
localhost:8888/notebooks/rockcnn.ipynb
jupyter rockcnn Last Checkpoint: Last Thursday at 6:32 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
13:31
04-07-2020

```
In [32]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Convolution2D
         from keras.layers import MaxPooling2D
         from keras.layers import Flatten

In [33]: from keras.preprocessing.image import ImageDataGenerator
         train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen=ImageDataGenerator(rescale=1./255)

In [34]: x_train=train_datagen.flow_from_directory(r"C:\Users\abina\Desktop\project\trainset",target_size=(64,64),batch_size=32)
         x_test=test_datagen.flow_from_directory(r"C:\Users\abina\Desktop\project\testset",target_size=(64,64),batch_size=32)

         Found 595 images belonging to 3 classes.
         Found 90 images belonging to 3 classes.

In [35]: x_train.class_indices

Out[35]: {'igneous': 0, 'metamorphic': 1, 'sedimentary': 2}

In [36]: model=Sequential()

In [37]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))

In [38]: model.add(MaxPooling2D(pool_size = (2,2)))
```

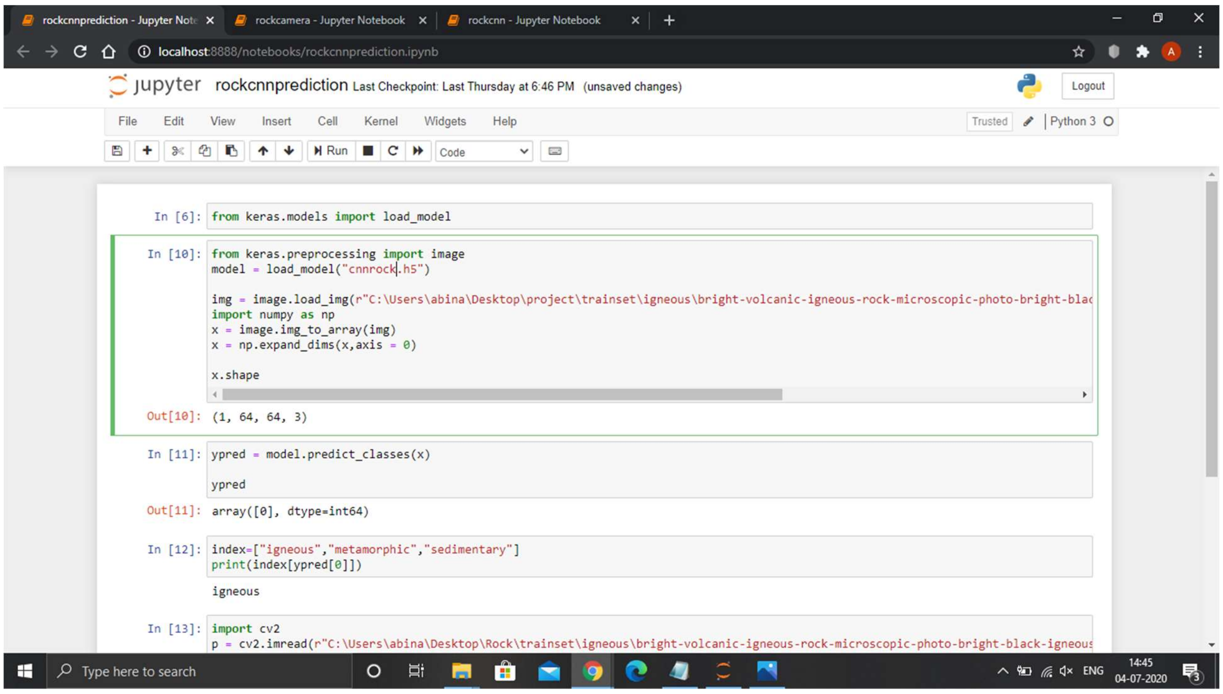

rockcnn - Jupyter Notebook
rockcnnprediction - Jupyter Note
rockcamera - Jupyter Notebook
localhost:8888/notebooks/rockcnn.ipynb
jupyter rockcnn Last Checkpoint: Last Thursday at 6:32 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
13:34
04-07-2020

```
In [39]: model.add(Flatten())

In [40]: model.add(Dense(units=128,init="uniform",activation="relu"))

         C:\Users\abina\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API: `
         Dense(units=128, activation="relu", kernel_initializer="uniform")`
           """Entry point for launching an IPython kernel.

In [41]: model.add(Dense(units=3,init="uniform",activation="softmax"))

         C:\Users\abina\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API: `
         Dense(units=3, activation="softmax", kernel_initializer="uniform")`
           """Entry point for launching an IPython kernel.

In [46]: model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accuracy"])

In [47]: model.fit_generator(x_train,steps_per_epoch=50,validation_data=x_test,validation_steps=20,epochs=10)

         Epoch 1/10
         50/50 [==============================] - 58s 1s/step - loss: 0.6571 - acc: 0.7241 - val_loss: 0.4760 - val_acc: 0.8460
         Epoch 2/10
         50/50 [==============================] - 50s 994ms/step - loss: 0.6295 - acc: 0.7375 - val_loss: 0.4547 - val_acc: 0.8395
         Epoch 3/10
         50/50 [==============================] - 50s 1s/step - loss: 0.5693 - acc: 0.7726 - val_loss: 0.4777 - val_acc: 0.8161
         Epoch 4/10
         50/50 [==============================] - 51s 1s/step - loss: 0.5546 - acc: 0.7749 - val_loss: 0.3649 - val_acc: 0.9089
         Epoch 5/10
         50/50 [==============================] - 50s 992ms/step - loss: 0.5219 - acc: 0.7901 - val_loss: 0.3086 - val_acc: 0.8696
         Epoch 6/10
         50/50 [==============================] - 50s 997ms/step - loss: 0.4621 - acc: 0.8261 - val_loss: 0.2491 - val_acc: 0.8980
```

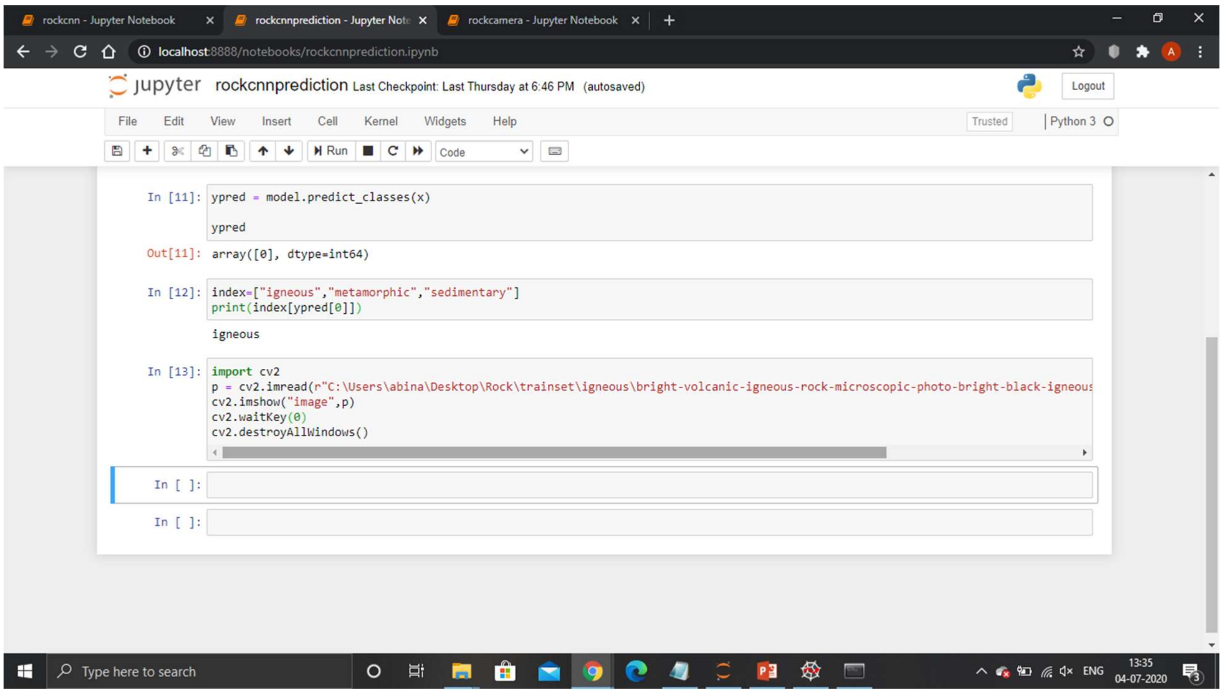

rockcnn - Jupyter Notebook
rockcnnprediction - Jupyter Note
rockcamera - Jupyter Notebook
localhost:8888/notebooks/rockcnn.ipynb
jupyter rockcnn Last Checkpoint: Last Thursday at 6:32 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
13:34
04-07-2020

```
In [47]: model.fit_generator(x_train,steps_per_epoch=50,validation_data=x_test,validation_steps=20,epochs=10)

         Epoch 1/10
         50/50 [==============================] - 58s 1s/step - loss: 0.6571 - acc: 0.7241 - val_loss: 0.4760 - val_acc: 0.8460
         Epoch 2/10
         50/50 [==============================] - 50s 994ms/step - loss: 0.6295 - acc: 0.7375 - val_loss: 0.4547 - val_acc: 0.8395
         Epoch 3/10
         50/50 [==============================] - 50s 1s/step - loss: 0.5693 - acc: 0.7726 - val_loss: 0.4777 - val_acc: 0.8161
         Epoch 4/10
         50/50 [==============================] - 51s 1s/step - loss: 0.5546 - acc: 0.7749 - val_loss: 0.3649 - val_acc: 0.9089
         Epoch 5/10
         50/50 [==============================] - 50s 992ms/step - loss: 0.5219 - acc: 0.7901 - val_loss: 0.3086 - val_acc: 0.8696
         Epoch 6/10
         50/50 [==============================] - 50s 997ms/step - loss: 0.4621 - acc: 0.8261 - val_loss: 0.2491 - val_acc: 0.8980
         Epoch 7/10
         50/50 [==============================] - 50s 1s/step - loss: 0.4385 - acc: 0.8391 - val_loss: 0.2573 - val_acc: 0.9123
         Epoch 8/10
         50/50 [==============================] - 50s 994ms/step - loss: 0.4272 - acc: 0.8205 - val_loss: 0.2385 - val_acc: 0.9080
         Epoch 9/10
         50/50 [==============================] - 51s 1s/step - loss: 0.3838 - acc: 0.8564 - val_loss: 0.2464 - val_acc: 0.9298
         Epoch 10/10
         50/50 [==============================] - 49s 983ms/step - loss: 0.3906 - acc: 0.8443 - val_loss: 0.1637 - val_acc: 0.9669

Out[47]: <keras.callbacks.History at 0x2b4b978a948>

In [48]: model.save("cnnrock.h5")

In [ ]:
```

12.3 Rock Prediction:

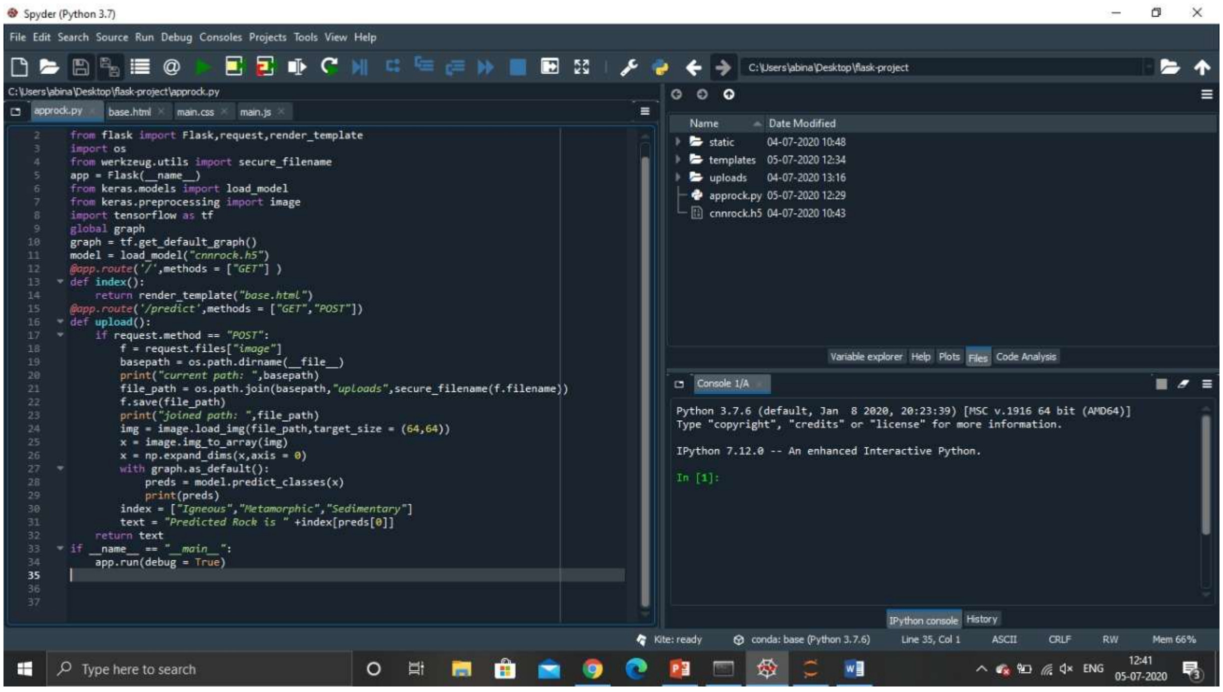

rockcnnprediction - Jupyter Note
rockcamera - Jupyter Notebook
rockcnn - Jupyter Notebook
localhost:8888/notebooks/rockcnnprediction.ipynb
jupyter rockcnnprediction Last Checkpoint: Last Thursday at 6:46 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
ENG
14:45
04-07-2020

```
In [6]: from keras.models import load_model

In [10]: from keras.preprocessing import image
         model = load_model("cnnrock.h5")

         img = image.load_img(r"C:\Users\abina\Desktop\project\trainset\igneous\bright-volcanic-igneous-rock-microscopic-photo-bright-blac
         import numpy as np
         x = image.img_to_array(img)
         x = np.expand_dims(x,axis = 0)

         x.shape

Out[10]: (1, 64, 64, 3)

In [11]: ypred = model.predict_classes(x)

         ypred

Out[11]: array([0], dtype=int64)

In [12]: index=["igneous","metamorphic","sedimentary"]
         print(index[ypred[0]])

         igneous

In [13]: import cv2
         p = cv2.imread(r"C:\Users\abina\Desktop\Rock\trainset\igneous\bright-volcanic-igneous-rock-microscopic-photo-bright-black-igneous
```



```
In [11]: ypred = model.predict_classes(x)

         ypred
Out[11]: array([0], dtype=int64)

In [12]: index=["igneous","metamorphic","sedimentary"]
         print(index[ypred[0]])

         igneous

In [13]: import cv2
         p = cv2.imread(r"C:\Users\abina\Desktop\Rock\trainset\igneous\bright-volcanic-igneous-rock-microscopic-photo-bright-black-igneous
         cv2.imshow("image",p)
         cv2.waitKey(0)
         cv2.destroyAllWindows()

In [ ]:

In [ ]:
```

12.4 Flask:



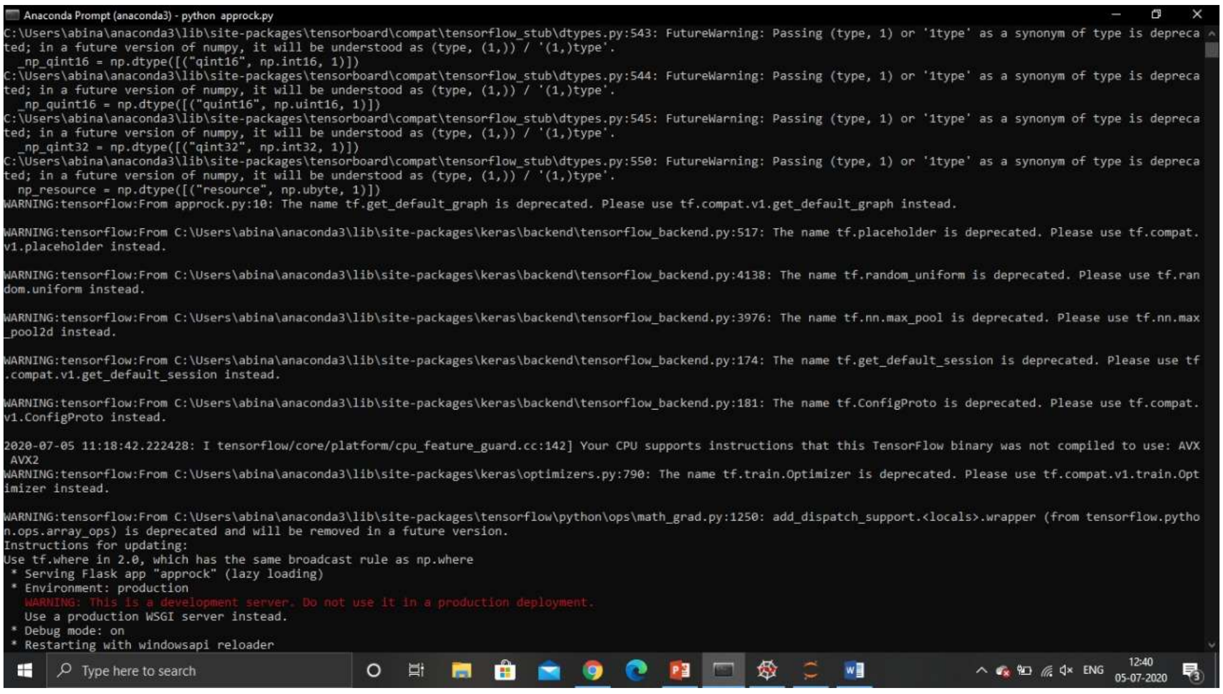
C:\Users\abina\Desktop\flask-project\approck.py

```python
from flask import Flask,request,render_template
import os
from werkzeug.utils import secure_filename
app = Flask(__name__)
from keras.models import load_model
from keras.preprocessing import image
import tensorflow as tf
global graph
graph = tf.get_default_graph()
model = load_model("cnnrock.h5")
@app.route('/',methods = ["GET"] )
def index():
    return render_template("base.html")
@app.route('/predict',methods = ["GET","POST"])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        basepath = os.path.dirname(__file__)
        print("current path: ",basepath)
        file_path = os.path.join(basepath,"uploads",secure_filename(f.filename))
        f.save(file_path)
        print("joined path: ",file_path)
        img = image.load_img(file_path,target_size = (64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis = 0)
        with graph.as_default():
            preds = model.predict_classes(x)
            print(preds)
        index = ["Igneous","Metamorphic","Sedimentary"]
        text = "Predicted Rock is " +index[preds[0]]
    return text
if __name__ == "__main__":
    app.run(debug = True)
```

C:\Users\abina\Desktop\flask-project

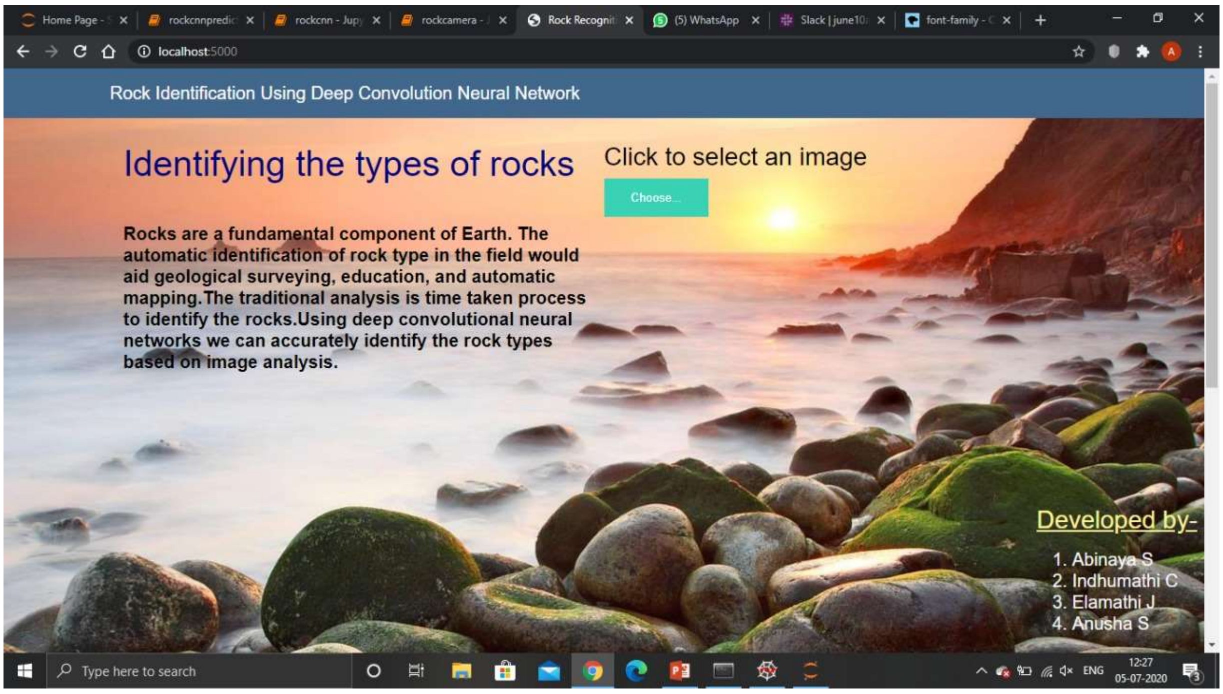
Name	Date Modified
static	04-07-2020 10:48
templates	05-07-2020 12:34
uploads	04-07-2020 13:16
approck.py	05-07-2020 12:29
cnnrock.h5	04-07-2020 10:43

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.12.0 -- An enhanced Interactive Python.

In [1]:
```



```
C:\Users\abina\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is depreca
ted; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\abina\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is depreca
ted; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\abina\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is depreca
ted; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\abina\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is depreca
ted; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
WARNING:tensorflow:From approck.py:10: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.
v1.placeholder instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.ran
dom.uniform instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max
_pool2d instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:174: The name tf.get_default_session is deprecated. Please use tf
.compat.v1.get_default_session instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:181: The name tf.ConfigProto is deprecated. Please use tf.compat.
v1.ConfigProto instead.

2020-07-05 11:18:42.222428: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX
 AVX2
WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Opt
imizer instead.

WARNING:tensorflow:From C:\Users\abina\anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.pytho
n.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
 * Serving Flask app "approck" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with windowsapi reloader
```

12.5 Output:



Rock Identification Using Deep Convolution Neural Network

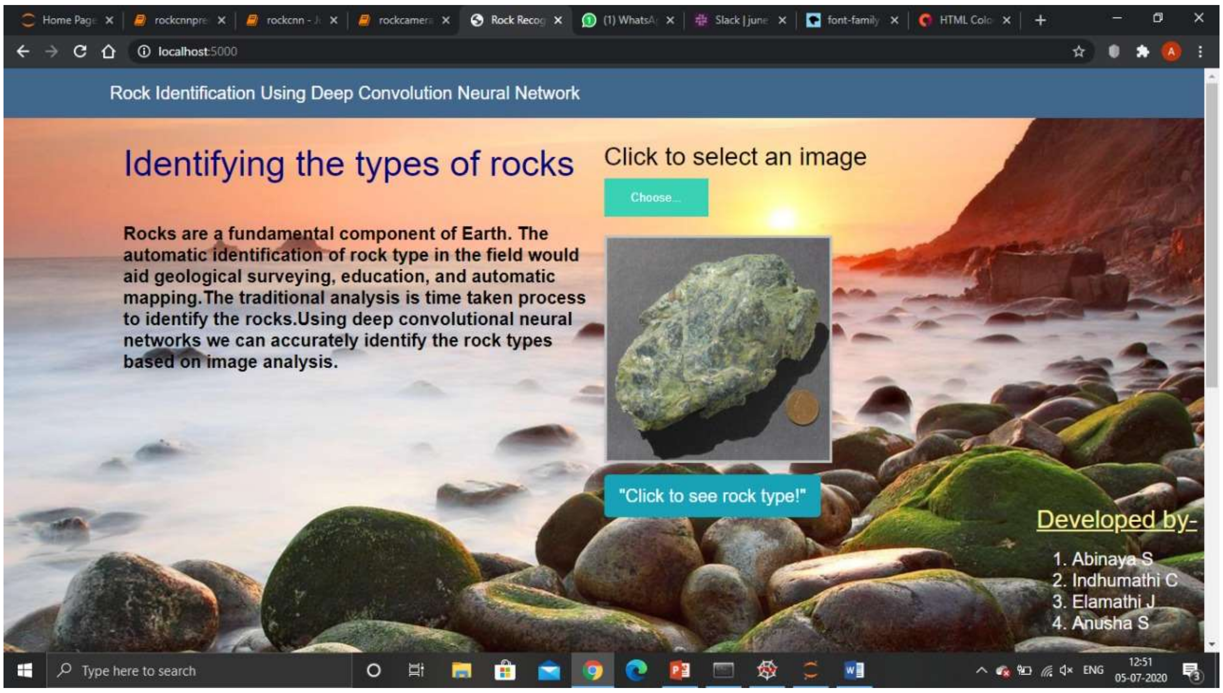
Identifying the types of rocks

Click to select an image

Rocks are a fundamental component of Earth. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping.The traditional analysis is time taken process to identify the rocks.Using deep convolutional neural networks we can accurately identify the rock types based on image analysis.

Developed by-

1. Abinaya S
2. Indhumathi C
3. Elamathi J
4. Anusha S



localhost:5000

Rock Identification Using Deep Convolution Neural Network

Identifying the types of rocks

Rocks are a fundamental component of Earth. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping.The traditional analysis is time taken process to identify the rocks.Using deep convolutional neural networks we can accurately identify the rock types based on image analysis.

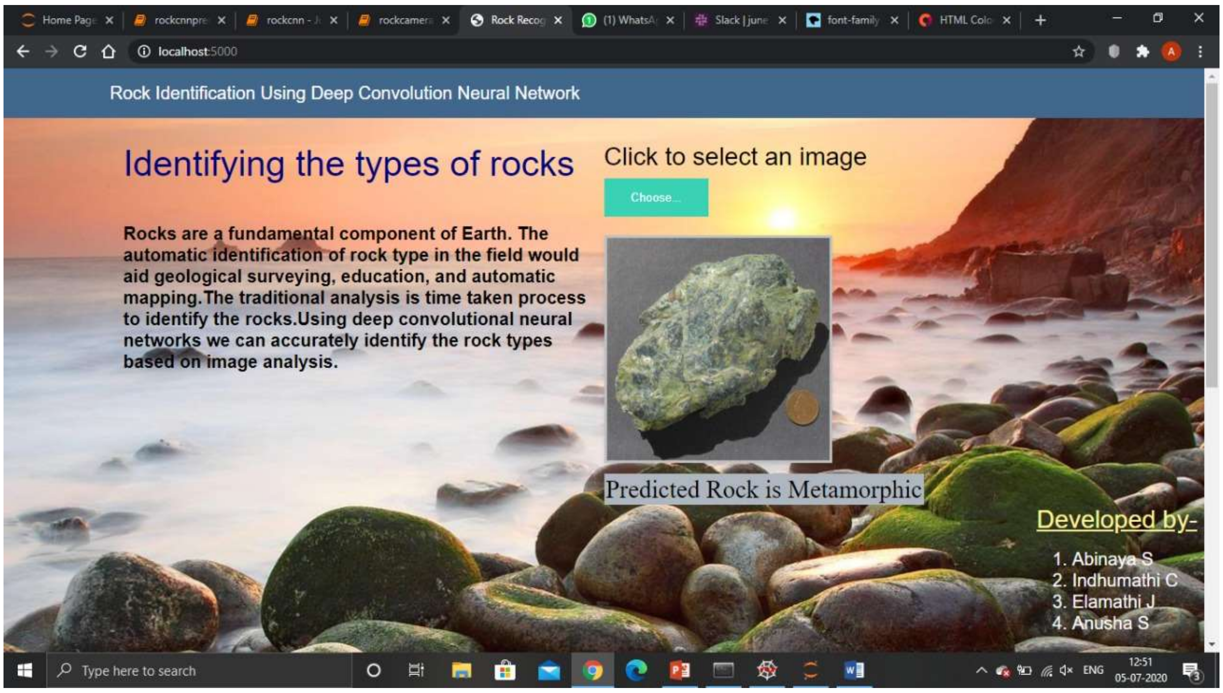
Click to select an image

Choose...

"Click to see rock type!"

Developed by-

1. Abinaya S
2. Indhumathi C
3. Elamathi J
4. Anusha S



localhost:5000

Rock Identification Using Deep Convolution Neural Network

Identifying the types of rocks

Rocks are a fundamental component of Earth. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping.The traditional analysis is time taken process to identify the rocks.Using deep convolutional neural networks we can accurately identify the rock types based on image analysis.

Click to select an image

Choose...

Predicted Rock is Metamorphic

Developed by-

1. Abinaya S
2. Indhumathi C
3. Elamathi J
4. Anusha S