RSIP Career Basic AI 116

Project Report

on

Rock Identification using Deep Convolution Neural Network

PREPARED BY,
P.S.S PRUDHVI RAJ
MEHER DHAVALA

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. Working conditions in the field generally limit identification to visual methods, including using a magnifying glass for fine-grained rocks. Visual inspection assesses properties such as colour, composition, grain size, and structure. The attributes of rocks reflect their mineral and chemical composition, formation environment, and genesis. The colour of rock reflects its chemical composition. But these analysis is time taken process to identify the rocks.Its application here has effectively identified rock types from images captured in the field. This paper proposes an accurate approach for identifying rock types in the field based on image analysis using deep convolutional neural networks.

Purpose:

To get the type of rock within seconds or minutes.

Detecting the type of rock plays a major role in geological surveying. Type of the rock reflects their chemical composition and in turn would be very useful if the time to predicting the type would be less. This model is also helpful for those who don't have that much geological knowledge but want to know the type of rock.

2. Literature survey:

This section summarises some of the scholarly and research works in the field of deep convolution neural network for classifying the type of rock.

Existing problem:

The main aim of this project is to classify the rocks. So, I have to create a system that can identify rocks with high accuracy than the traditional method.

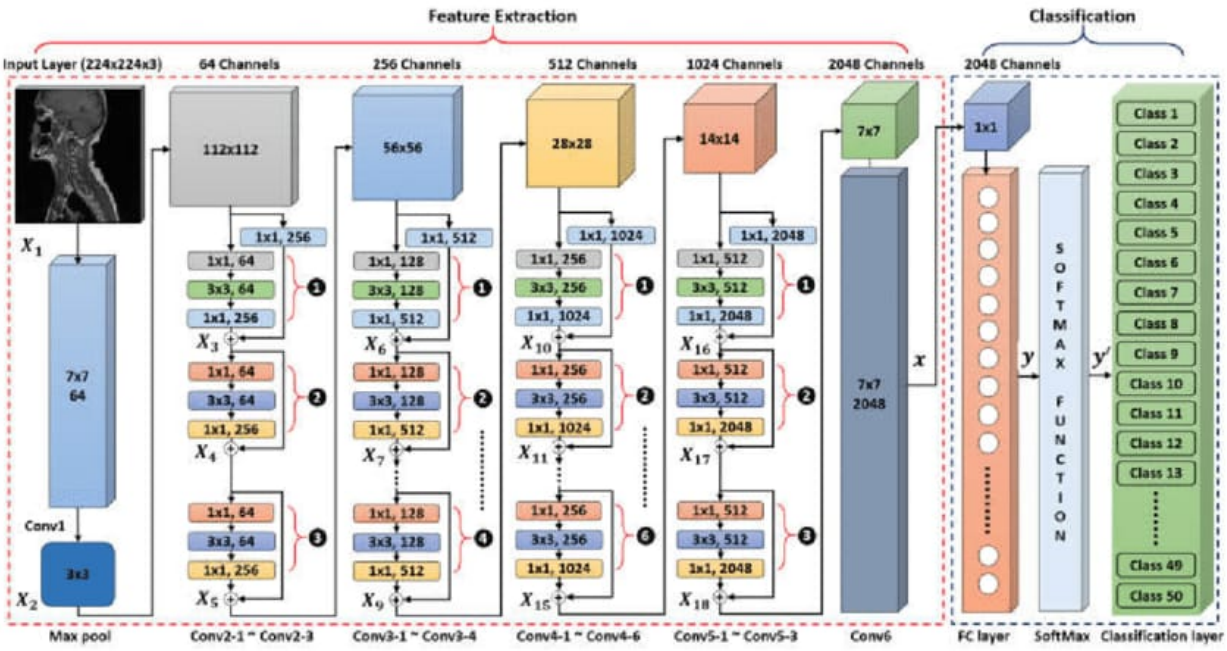
Proposed solution:

First model is optimization model for minima and maxima values. Second module is for colour similarity measure. Third is for time complexity measure. Fourth is for collection of images. Fifth module is Deep learning CNN. Lastly, they have performance evaluation.

Referred pdf: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9042306

3. THEORITICAL ANALYSIS

3.1 Block Diagram



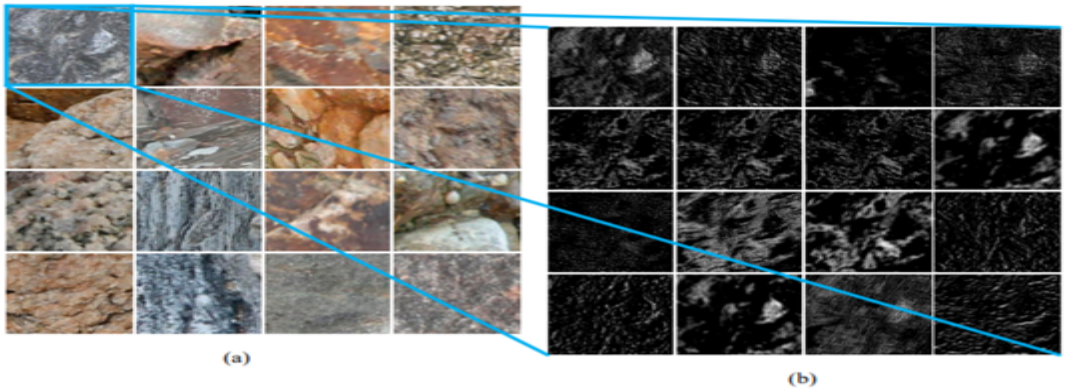
3.2 Hardware / Software designing:

For hardware we would need ICT tools such as digital camera. Here we used both Digital Camera for capturing the rock image and collected the images from internet which is given as input to the web page.

Software tools needed are:
1. Jupyter Notebook
2. Tensorflow
3. Keras
4. Flask
5. Operating system for python, html, css, js

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.



Learned rock features after convolution by the RTCNNs model. **(a)** Input patched field rock sample images. **(b)** Outputted feature maps partly after the first convolution of the input image, from the upper left corner in (a).

5. Flowchart:


Data Preprocessing
Spliting data & Traing data
Model Building
Train data
Machine Learning Models Training
Test Data
70%
30%
Working on regression models for predicting the accurate model
Evaluation
Application Bulilding
Creating HTML File
Python code

6. Result:

I got an accuracy of 0.93 which is a good measure for convolution neural networks.

When we give a rock image as input the model, it classifies them as the categorical output-

- Igneous Rock
- Metamorphic Rock
- Sedimentary Rock

7. Advantages andDisadvantages:

Advantages:

● Effective and predictsaccurately.
● Predicts the type of rock using color, shape andsize.
● Efficiency is maintained by updating the datasetfrequently.

Disadvantages:

● The input image must have clarity to predict the correctoutput.
● Cost of the digital camera ishigh.
● High computationalcost.

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 andaccurately.
● Can be used to identify type of rocks in mines and its implementable on the website.

9. Conclusion:

This project helps in identifying the rock type effectively using CNN. This experiment shows that this has high reliability whether in HSV or RGB color space. In RGB color space the efficiency acquired is 93 percent which is good. In view of using CNN for rock classification this can be considered a good way for classifying rocks.

10. Future scope:

Although CNNs have helped to identify and classify rock types in the field, some challenges remain. First, the recognition accuracy still needs to be improved. The accuracy of 96.96% achieved using the proposed model meant that 89 images were misidentified in the testing dataset. The model attained relatively low identification accuracy for Igneous rock type, which is attributed to the small grain size and similar colors of these rocks. Furthermore, only a narrow range of sample types (Three rock types overall) was considered in this study. The three main rock groups (igneous, sedimentary, and metamorphic) can be divided into hundreds of types (and subtypes) according to mineral composition. Therefore, our future work will combine the deep learning model with a knowledge library, containing more rock knowledge and relationships among different rock types, to classify more rock types and improve both the accuracy and the range of rock-type identification in the field. In addition, each field photograph often contains more than one rock type, but the proposed model can classify each image into only one category, stressing the importance of the quality of the original image capture.

11. Biblography:

Idea:

https://www.researchgate.net/publication/319590805_Rock_images_classification_by_using_deep_convolution_neural_networkhttps://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9042306

12. Appendix:

Model training:

In [1]:
```
#Import the required libraries.

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
```

```
Using TensorFlow backend.
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [2]:
```
#Initialise the model.

model = Sequential()
```

In [3]:
```
#Adding Convolution layer.

model.add(Convolution2D(32,(3,3), input_shape=(64,64,3), activation='relu'))
```

```
WARNING:tensorflow:From C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

In [4]:
```
#Add Max Pooling layer.

model.add(MaxPooling2D(pool_size=(2,2)))
```

In [5]:
```
#Add Flattening layer.

model.add(Flatten())
```

In [6]:
```
#Add hidden layer.

model.add(Dense(output_dim=128, init='uniform', activation='relu'))
```

```
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(activation="relu", units=128, kernel_initializer="uniform")`
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [7]:
```
#Add output layer.

model.add(Dense(units=3, init="uniform", activation="softmax"))
```

```
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=3, activation="softmax", kernel_initializer="uniform")`
```

In [7]:
```
#Add output layer.

model.add(Dense(units=3, init="uniform", activation="softmax"))
```

```
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarning: Update your `Dense` call to the Keras 2
API: `Dense(units=3, activation="softmax", kernel_initializer="uniform")`
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [8]:
```
#Apply Image Preprocessing.

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 [9]:
```
x_train=train_datagen.flow_from_directory(r'F:\Rock\Data_set\train_set',target_size=(64,64),batch_size=32,class_mode='categorica
x_test=test_datagen.flow_from_directory(r'F:\Rock\Data_set\test_set',target_size=(64,64),batch_size=32,class_mode='categorical')
```

```
Found 240 images belonging to 3 classes.
Found 60 images belonging to 3 classes.
```

In [10]:
```
x_train.class_indices
```

Out[10]:
```
{'IGNEOUS ROCKS': 0, 'METAMORPHIC ROCKS': 1, 'SEDIMENTARY ROCKS': 2}
```

In [11]:
```
#Compile the model.

model.compile(optimizer="adam", loss='categorical_crossentropy', metrics=["accuracy"])
```

In [12]:
```
#Training and testing the model.

model.fit_generator(x_train, steps_per_epoch=300, epochs=10, validation_data=x_test, validation_steps=50)
```

```
WARNING:tensorflow:From C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/10
300/300 [==============================] - 107s 355ms/step - loss: 0.8722 - acc: 0.5763 - val_loss: 0.5894 - val_acc: 0.7500
Epoch 2/10
300/300 [==============================] - 103s 343ms/step - loss: 0.4454 - acc: 0.8291 - val_loss: 0.4901 - val_acc: 0.8167
Epoch 3/10
300/300 [==============================] - 102s 341ms/step - loss: 0.2029 - acc: 0.9354 - val_loss: 0.8934 - val_acc: 0.8000
Epoch 4/10
300/300 [==============================] - 104s 348ms/step - loss: 0.1055 - acc: 0.9696 - val_loss: 0.4984 - val_acc: 0.8833
Epoch 5/10
300/300 [==============================] - 104s 346ms/step - loss: 0.0672 - acc: 0.9828 - val_loss: 0.6233 - val_acc: 0.8333
Epoch 6/10
300/300 [==============================] - 103s 344ms/step - loss: 0.0452 - acc: 0.9883 - val_loss: 1.0222 - val_acc: 0.7833
Epoch 7/10
300/300 [==============================] - 104s 345ms/step - loss: 0.0458 - acc: 0.9882 - val_loss: 0.6961 - val_acc: 0.8333
Epoch 8/10
300/300 [==============================] - 104s 345ms/step - loss: 0.0256 - acc: 0.9936 - val_loss: 0.9140 - val_acc: 0.7667
Epoch 9/10
300/300 [==============================] - 103s 344ms/step - loss: 0.0194 - acc: 0.9949 - val_loss: 0.7381 - val_acc: 0.8833
Epoch 10/10
300/300 [==============================] - 103s 345ms/step - loss: 0.0234 - acc: 0.9943 - val_loss: 0.6890 - val_acc: 0.8000
```

Out[12]:
```
<keras.callbacks.History at 0x2d8bddc7808>
```

In [13]:
```
#Saving the model.

model.save("cnn_model.h5")
```

Running the Code:

In [1]:
```
from __future__ import division , print_function
import sys
import os
import glob
import numpy as np
from keras.preprocessing import image
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
```

```
Using TensorFlow backend.
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; 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\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [2]:
```
from keras.models import load_model
from keras import backend
from tensorflow.keras import backend
```

In [3]:
```
import tensorflow as tf
global graph
graph = tf.get_default_graph()
from skimage.transform import resize
from flask import Flask,request,render_template,redirect,url_for
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
```

In [4]:
```
app = Flask(__name__)
MODEL_PATH = 'cnn_model.h5'
model = load_model(MODEL_PATH)
```

```
WARNING:tensorflow:From C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: col
ocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\PRUDHVI RAJ\anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
```

In [5]:
```
@app.route('/',methods = ["GET"] )
def index():
    return render_template("base.html")
```

In [5]:
```
@app.route('/',methods = ["GET"] )
def index():
    return render_template("base.html")
```

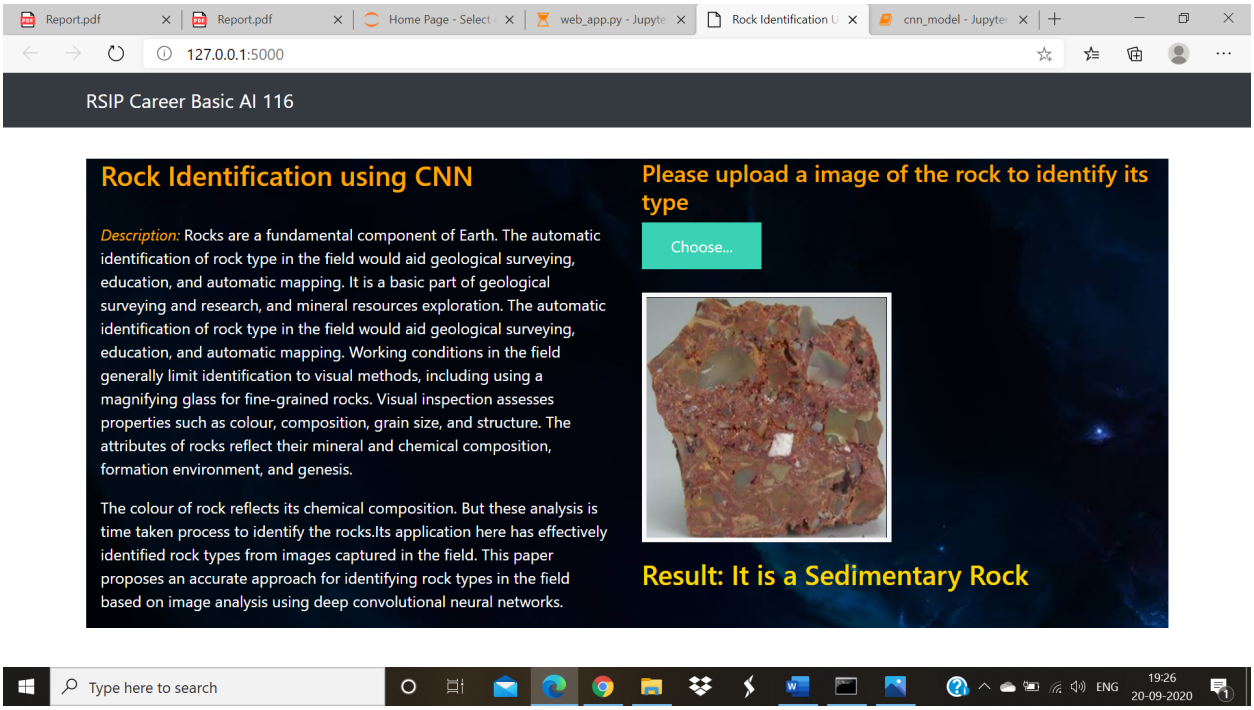
In [6]:
```
@app.route('/predict',methods = ["GET","POST"])
def upload():
    if request.method == "POST":
        f = request.files['image']
        basepath = os.path.dirname('uploads')
        print("current path: ",basepath)
        file_path = os.path.join(basepath,"uploads",secure_filename(f.filename))
        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():
            prediction_class = model.predict_classes(x)
            print("prediction",prediction_class)

        i=prediction_class.flatten()
        index = ['Igneous','Metamorphic','Sedimentary']
        if(str(index[i[0]])=="Igneous"):
            text="It is an " + str(index[i[0]] +" Rock")
        elif (str(index[i[0]])=="Metamorphic"):
            text ="It is a " + str(index[i[0]] +" Rock")
        elif(str(index[i[0]])=="Sedimentary"):
            text="It is a " + str(index[i[0]] +" Rock")
    return text
```

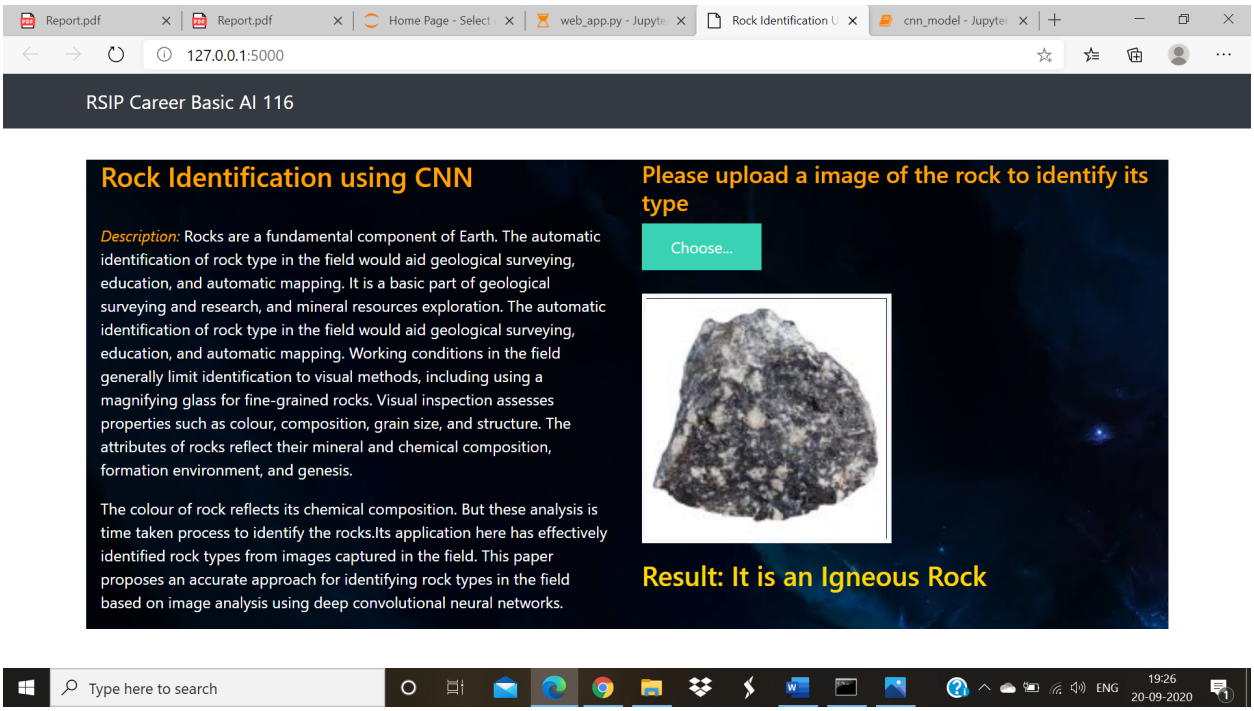
In [*]:
```
if __name__ == "__main__":
    app.run(debug = False)
```

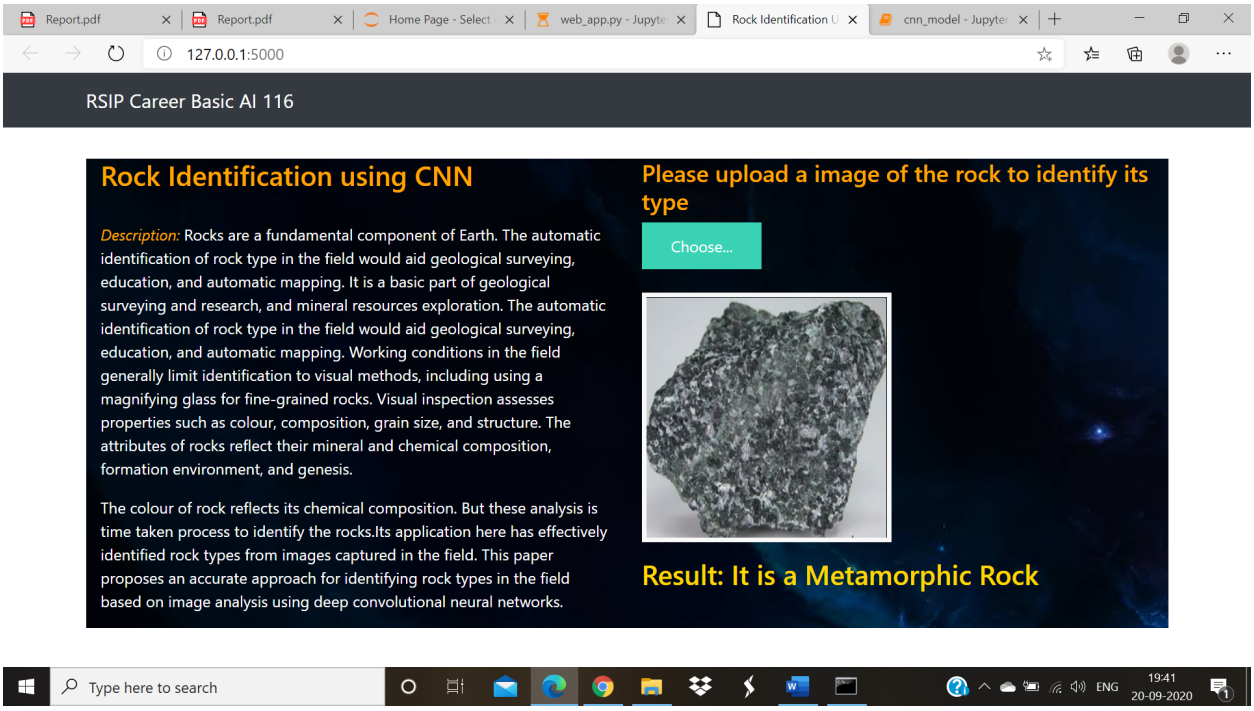
```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

In [ ]:

Output:




Report.pdf
Report.pdf
Home Page - Select
web_app.py - Jupyte
Rock Identification U
cnn_model - Jupyter
127.0.0.1:5000
RSIP Career Basic AI 116
Rock Identification using CNN
Description: 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. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping. Working conditions in the field generally limit identification to visual methods, including using a magnifying glass for fine-grained rocks. Visual inspection assesses properties such as colour, composition, grain size, and structure. The attributes of rocks reflect their mineral and chemical composition, formation environment, and genesis.
The colour of rock reflects its chemical composition. But these analysis is time taken process to identify the rocks.Its application here has effectively identified rock types from images captured in the field. This paper proposes an accurate approach for identifying rock types in the field based on image analysis using deep convolutional neural networks.
Please upload a image of the rock to identify its type
Choose...
Result: It is an Igneous Rock
Type here to search
ENG
19:26
20-09-2020


Report.pdf
Report.pdf
Home Page - Select
web_app.py - Jupyte
Rock Identification U
cnn_model - Jupyter
127.0.0.1:5000
RSIP Career Basic AI 116
Rock Identification using CNN
Description: 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. The automatic identification of rock type in the field would aid geological surveying, education, and automatic mapping. Working conditions in the field generally limit identification to visual methods, including using a magnifying glass for fine-grained rocks. Visual inspection assesses properties such as colour, composition, grain size, and structure. The attributes of rocks reflect their mineral and chemical composition, formation environment, and genesis.
The colour of rock reflects its chemical composition. But these analysis is time taken process to identify the rocks.Its application here has effectively identified rock types from images captured in the field. This paper proposes an accurate approach for identifying rock types in the field based on image analysis using deep convolutional neural networks.
Please upload a image of the rock to identify its type
Choose...
Result: It is a Metamorphic Rock
Type here to search
ENG
19:41
20-09-2020