JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Classifying Space rocks using AI

Mr. Tanuj Dharmesh Patel
Department of Computer Engineering
Universal College of Engineering
Vasai, India
tanujpatel46@gmail.com

Mr. Umang Dilip Patel
Department of Computer Engineering
Universal College of Engineering
Vasai, India
umangpatel2682@gmail.com

Mr. Raj Rahul Parekh
Department of Computer Engineering
Universal College of Engineering
Vasai, India
parekhraj575@gmail.com

Mr. John Kenny
Department of Computer Engineering
Universal College of Engineering
Vasai, India
john.kenny@universal.edu.in

Abstract-A lot of resources are spent by space agencies for the gathering and transportation of rocks that are brought back to earth for research and studies. Geologists have to conduct various tests and experiments on the rocks to classify them. Machine learning based unsupervised hierarchical approaches for space object detection was illustrated in literature. Unfortunately these methods are limited to detection of rocks but not classifying them into categories. This research aims at implementing deep learning network model VGG19 with Convolutional neural network, for the classification to learn the associations between the space rock features e.g. Curves, edges and textures and thus classifying into rock type based on this. This would save a lot of precious time and resources spent by the agencies on space rock research.

Keywords-Space rocks, Neural networks, rock classification, deep learning.

I. INTRODUCTION

As technology advances, many disciplines outside the field of computer science have found ways to introduce advanced technologies into their work. Artificial intelligence (AI) is a relatively new technology that's being applied to many tasks. Artificial intelligence helps teach a computer to detect which rock an astronaut should collect based on the rock type. Artificial intelligence can be a useful tool for the brilliant research minds that are behind some of Earth's most inspiring discoveries. Classification of rocks is one of the most essential issues for research. In this project, a new method is proposed to achieve space rock classification work automatically with the help of deep learning networks models. Rocks types such as Basalt, Highland etc can be classified.

II. LITERATURE SURVEY

Saad ALBAWI , Tareq Abed MOHAMMED and Saad AL-ZAWI defined the term Deep Learning or Deep Neural Network refers to Artificial Neural Networks (ANN) with multi layers. It has become very popular as it is one of the most powerful tools. The interest in finding deeper hidden layers is increasing and has surpassed traditional methods in different fields, especially in pattern recognition. Convolutional Neural Network(CNN) is one of the most popular deep learning networks. It is called convolution from the mathematical operation between matrixes with the same name. The convolutional and fully- connected layers have parameters but pooling and non-linearity layers don't have parameters. The applications that deal with image data, such as the largest image classification data set (Image Net), computer vision, and in natural language processing (NLP) and the results achieved were quite excellent. In this paper they defined all the elements and the important issues related to CNN and how the network actually works[1]

This study by Seong-Hyeon and Han Kwang-Yeob Lee implemented an image classification CNN using a multi-thread GPU. For the CNN, the CIFAR10 dataset was used, and the multi-thread GPU had 256 threads. Using the 256 threads limited to each layer, allocation and parallel processing were conducted. The image classification CNN took 807 ms for computation.[2]

Keiron O'Shea and Ryan Nash stated Artificial Neural Networks (ANNs) are computational processing systems of which are heavily inspired by biological nervous systems (such as the way the human brain operates). ANNs are mainly comprised of a high number of interconnected computational nodes (referred to as neurons), of which work entwine in a distributed fashion to collectively learn from the input in order to optimize its final output.[3]

V. Gor, R. Castano, R. Manduchi, R. C. Anderson, and E. Mjolsness, have proposed a general framework for an image based autonomous rock detection process for Martian terrain. A rock detection algorithm is described and demonstrated on examples of real Mars Rover data. An attempt is made to produce a system that is independent of parameters to ease on-board implementation for real time in situ operation. The process makes use of unsupervised hierarchical approaches for object detection and is easily expandable to more complex data sets. It uses intensity information to detect small rocks and range information (derived from a pair of intensity images) to detect large rocks in the image. The range-based and intensity-based algorithms tend to be complementary, with one working when the other fails, together they detect most of the rocks in Mars images.The module closes the loop between data acquisition, data analysis and decision-making in situ. It can be used to prioritize what information will be sent back to Earth, where to take more scientific measurements using more time-consuming instrumentation, and which surface regions to explore further. In this way the system contributes to reducing data downlink and maximizing science return per bit of data [4].

LIU Ye, GUO Chao, CHENG Guojian have proposed a rock classification technology based on features extracted from rocks images to classify the rock type automatically with the images of core thin sections. The elements of feature space are from color and morphology features of rock images, and constructed with the statistical analysis result of standard arithmetic value into different color spaces [5].

Shuet.al. have used a Support Vector Machine (SVM) classification algorithm to classify uniform rock images into 9 different classes with the image features extracted autonomously. Through this method, they had achieved an accuracy of classification of 96.71%. In this rock image classification with various combinations of manual features as well as unsupervised feature learning. The results of these experiments showed that different combinations of manual features affected classification substantially; whereas unsupervised feature learning based on K-means performed pretty well. This technique can also be applied to geological image archives (e.g. autonomous labeling) or image retrieval etc. Future work will be to improve and test this technique on a larger and more general rock image dataset [6].

Alexis David P. Pascual, Lei Shu, Justin Szoke-Sieswerda, Kenneth McIsaac, Gordon Osinski have proposed in this paper that demonstrates the use of CNNs to classify the same set of rock images. With the addition of dataset augmentation, a 3-layer CNN is shown to have a significant improvement over Shu et. al.'s results, which achieved an average accuracy of 99.60% across 10 trials on the test set. Here the images are taken during field exploration without a standardized method and specialized equipment. The task had been simplified into a binary classification problem where the images are classified into breccia and non-breccia. This research shows that a 5-layer CNN achieves 89.43% classification accuracy for this task [7]

III. EXISTING SYSTEM VS PROPOSED SYSTEM

Machine learning based unsupervised hierarchical approaches for space object detection was completed in the existing system. Unfortunately these methods are limited to detection of rocks but not classifying them into categories. This was only meant to detect the rocks in the path of the mars rover and steer it away from it.

The detailed differences and improvements are shown in the table below:

Parameter	Existing System	Proposed system
Time complexity	Took a lot of time to process	Will take very less time
Newer technologies	Used technologies which were old	Uses technologies such as AI
No human intervention	Some level of input from the Geologist is needed	Completely automated just needs a input of photo
Only detection of rocks	The earlier systems could only detect the rocks presence in the image	This would detect as well as classify the rock
Performance	Because of manual work and dependency on manpower system has less performance	As the system is computer based so performance is better than manual work

Table 1:

PROPOSED SYSTEM:

'Classifying Space rocks using Python and AI' is a classifying deep learning model based on CNN. This system involves a basic mechanism used for detecting and classifying the space rocks. The algorithm is left to train on a dataset of hundreds of images to improve its accuracy and then the algorithm is used to classify rocks. This requires the user to provide the image of the rock for which he/she needs to know the space rock name. This would save the manual work done by the transporters as well the geologists done throughout the years. This tries to automate 90% of this system and makes the process simpler and convenient to use. In Fig . 1. , we show the detailed system architecture.

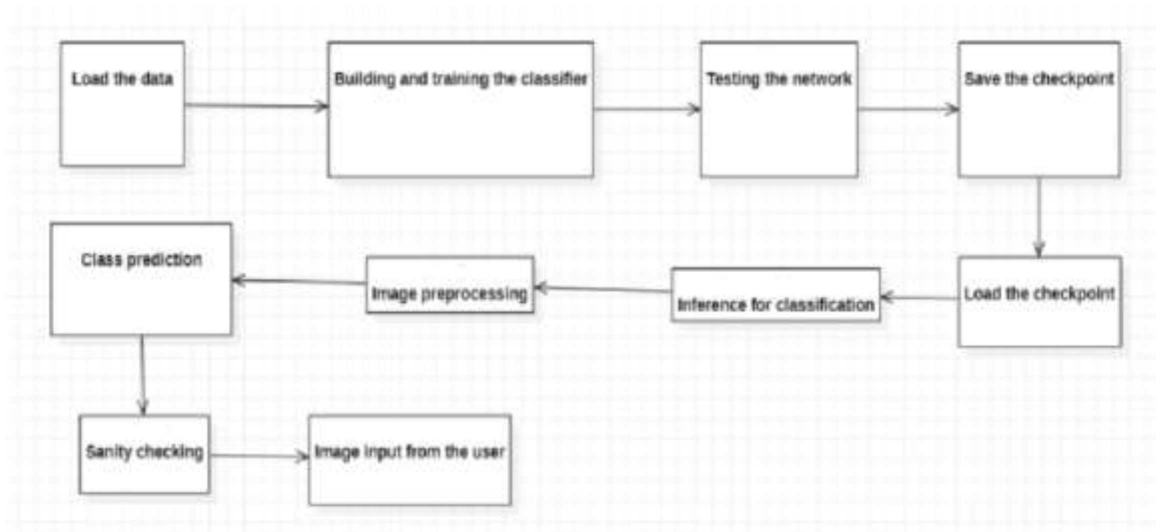

Figure 1 System Architecture

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

VGG19 is the first to detect objects within an image coming from 200 classes, which is called object localization. The second is to classify images, each labeled with one of 1000 categories, which is called image classification.

Torchvision :- Torchvision is a library for Computer Vision that goes hand in hand with PyTorch. It has utilities for efficient Image and Video transformations, some commonly used pre-trained models, and some datasets.

1.1 The modules are as follows:

A. Load the data

Preprocessing on the dataset is done through the application of transformations such as random scaling, cropping, and flipping before model training. This will help the network generalize leading to better performance. It will also need to make sure the input data is resized to 224x224 pixels as required by the pre-trained networks.

B. Label Mapping

We load in a mapping from category label to category name. It's a JSON object file which gives the dictionary mapping the integer encoded categories to the actual names of the rocks in figure 2

```
3
{'1': 'Basalt', '2': 'Highland', '3': 'Breccia'}
```

Figure 2. Class Mapping

C. Building and training the classifier

Now that the data is ready, it's time to build and train the classifier. It will use one of the pretrained models from torchvision models to get the image features. Build and train a new feed-forward classifier using those features.

i. First we load a pre-trained network
ii.Then we define a new, untrained feed-forward network as a classifier, using ReLU activations and dropout.
iii.Then we train the classifier layers using backpropagation using the pre-trained network to get the features
iv.We track the loss and accuracy on the validation set to determine the best hyperparameters.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): ReLU(inplace=True)
    (18): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (19): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (24): ReLU(inplace=True)
    (25): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (26): ReLU(inplace=True)
    (27): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (31): ReLU(inplace=True)
    (32): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (33): ReLU(inplace=True)
    (34): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (35): ReLU(inplace=True)
    (36): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
```

Figure 3.VGG 19

C. Testing the network

It runs the test images through the network and measures the accuracy, the same way it did validation. It tests if the model has been trained well.

E. Save the checkpoint

Now that our network is trained, save the model so it can load it later for making predictions. It saves other things such as the mapping of classes to indices which it gets from one of the image datasets.

F. Load the checkpoint

Function that can load a checkpoint and rebuild the model. That way it can come back to this project and keep working on it without having to retrain the network.

G. Inference for classification

A function to use a trained network for inference. That is, it passes an image into the network and predicts the class of the rock in the image. Write a function that takes an image and a model, then returns the top most likely classes along with the probabilities.

H. Image Preprocessing

First it resizes the images where the shortest side is 256 pixels, keeping the aspect ratio.Then it need to crop out the center 224x224 portion of the image. Color channels of images are typically encoded as integers 0-255, but the model expected floats 0-1. It needs to convert the values. As before, the network expects the images to be normalized in a specific way. For the means, it's [0.485, 0.456, 0.406] and for the standard deviations [0.229, 0.224, 0.225].

I. Class prediction

Once images are in the correct format, a function is written for making predictions with this model. A common practice is to predict the top 5 or so most probable classes. Finally class probabilities are calculated to find the largest values. To get the top largest values in a tensor.This method returns both the highest probabilities and the indices of those probabilities corresponding to the classes. A need to convert from these indices to the actual class which is added to the model or the image folder we used to load the data.

J. Sanity Checking

Now that trained model for predictions is made, a check to make sure it makes sense. It uses matplotlib to plot the probabilities for the 3 classes as a bar graph, along with the input image.

1.2 Dataset used

We used a custom database which was gathered by us by searching the internet and through the official NASA website for public domain.

IV. RESULT AND DISCUSSION

Now that the system is ready. We provide the test images to classify them to their respective names with the help of the trained deep learning network.The rocks are classified into their respective names.

We have achieved close to 90 percent accuracy with this system.

Pseudocode to predict the class from an image file

```
Def predict
Image from image path
Image unsqueeze
Output = torch.exp(out)

Predict top probabilities
Convert to lists
Index to class
Print probabilities
Print classes
```

Pseudocode to display the classes

Plot figure
Image process(image path)
Rock title = rock_to_name[3]
Rock_name = [rock_to_name for i in classes]
#plot the
Plt.subplot
#print the bar graph
Sb.barplot
plt.show()

The classification is shown below:

1.Basalt rock

Basalt is an aphanitic extrusive igneous rock formed from the rapid cooling of low-viscosity lava rich in magnesium and iron exposed at or very near the surface of a rocky planet or moon.

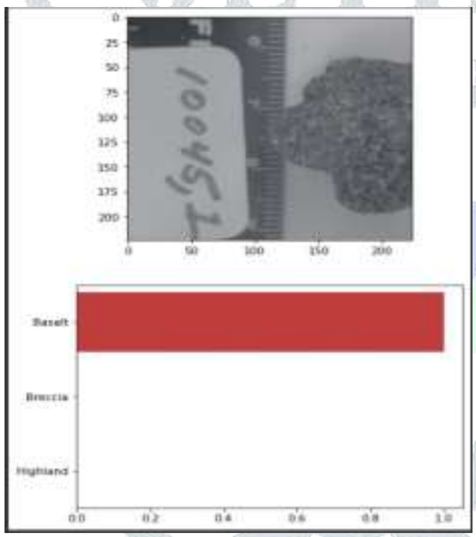

Figure 4. Basalt rock classification

2.Highland Rock

The highlands consist of the ancient lunar surface rock, anorthosite, and materials thrown out during the creation of the impact basins. Relatively young basins are shown in light colors; the oldest basins are in dark color.

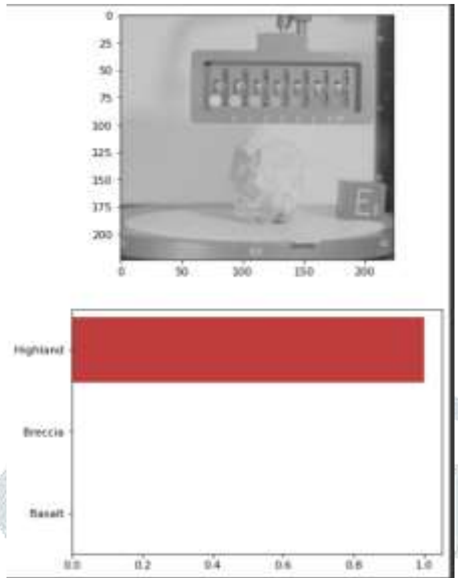

Figure 5. Highland rock classification

Breccia is a sedimentary rock composed of broken fragments of minerals or rocks cemented together by a fine-grained matrix that can be similar to or different from the composition of the fragments

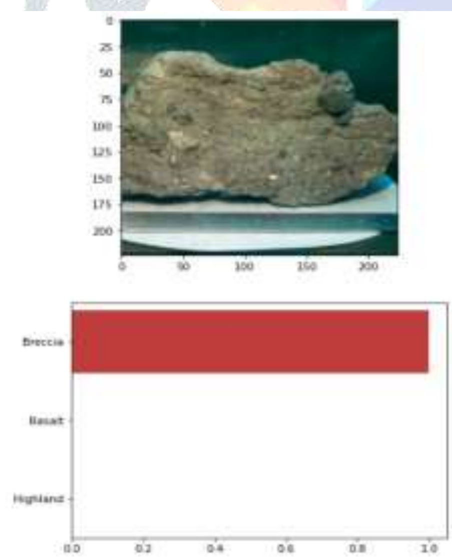

Figure 6. Breccia rock classification

V. CONCLUSION

Tremendous resources are spent by space agencies for the gathering and transportation of rocks that are brought back to earth for research and studies. Geologists have to conduct various tests and experiments on the rocks to classify them. With the help of neural networks /deep learning network to learn the associations between the features eg. Curves, edges and textures and the rock type. The model would be trained to identify the rocks from the images and display their types.

VI. REFERENCES

[1] Saad ALBAWI , Tareq Abed MOHAMMED and Saad AL-ZAWI , “Understanding of a Convolutional Neural Network” , IEEE(2017)

[2] Seong-Hyeon and Han Kwang-Yeob Lee , “Implemetation of Image Classification CNN using Multi thread GPU ” , IEEE(2017)

[3] Keiron O’Shea and Ryan Nash , “An Introduction to Convolutional Neural Networks”, Lancaster University(2015)

[4] V. Gor, R. Castano, R. Manduchi, R. C. Anderson, and E. Mjolsness, “Autonomous rock detection for mars terrain” , AIAA American Institute of Aeronautics and Astronautics (2001)

[5] LIU Ye, GUO Chao, CHENG Guojian , “Rock Classification Based on Images Color Spaces and Artificial Neural Network” , IEEE (2014)

[6] Lei Shu, Kenneth McIsaac, Gordon R. Osinski, Raymond Francis, ”Unsupervised feature learning for autonomous rock image classification” , Computers and Geosciences, vol. 106, no. February, pp. 10–17, (2017).

[7] Alexis David P. Pascual, Lei Shu, Justin Szoke-Sieswerda, Kenneth McIsaac and Gordon Osinski, “Towards Natural Scene Rock Image Classification with Convolutional Neural Networks” , IEEE (2019)