Igneous rock classification using Convolutional neural networks (CNN)

Surendra Patro, Dr. D.C. Jhariya, Dr. Mridu Sahu, Dr Pankaj Dewangan, Dr P. Y. Dhekne

1B. Tech Mining Engineering Pre-Final Year, National Institute of Technology Raipur, G.E. Road, Raipur - 492010, Chhattisgarh, India
2Assistant Professor, Department of Applied Geology, National Institute of Technology Raipur, G.E. Road, Raipur - 492 010, Chhattisgarh, India
3Assistant Professor, Department of Information Technology, National Institute of Technology Raipur, G.E. Road, Raipur-492010, Chhattisgarh, India
4Associate Professor, Department of Mining Engineering, National Institute of Technology Raipur, G.E. Road, Raipur-492010, Chhattisgarh, India
5Associate Professor, Department of Mining Engineering, National Institute of Technology Raipur, G.E. Road, Raipur-492010, Chhattisgarh, India

Abstract –
This paper describes how convolutional neural networks are used to identify and classify igneous rocks (CNN). Igneous rocks are formed while still hot, hot magma crystallises and solidifies. Melt originates deep beneath the Earth's surface, amid active plate borders or hot zones, and then rises to the surface. There are also various kinds of igneous rocks, which are addressed throughout this work, distinguishing each one is a difficult feat in and of itself. Machine learning, which is fundamentally a three-layer neural network, is a subset of deep learning. These neural networks attempt to mimic conscious brain function of humans by letting it "learn" from massive volumes of data, but they continue to fail. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning approach for assigning priority (learnable weights and biases) to various aspects/objects in a picture while also identifying them. Classification of images, audio and video segmentation, decision support systems, speech recognition, and image analysis are just a few of the applications for CNNs.

Keywords – Igneous Rocks, CNN (convolutional neural networks), layers, Python and Jupyter Notebook, training and testing.

1. Introduction –
Igneous rocks are formed by the cooling and crystallisation of molten rock, often known as magma. Magma is molten rock that exists deep within the ground. It is the source of all igneous rock. Because the planet was mainly melted when it originated, magma may be regarded as the beginning of the rock cycle. Igneous rocks' origins are documented in their composition. We can derive processes that occur within the earth and comprehend volcanic activities that occur on the planet's surface by carefully examining igneous rocks and interpreting the information they carry. We can learn about the igneous portion of geologic history by studying igneous rocks [1]. Igneous petrology is the study of the identification, classification, origin, development, and formation and crystallisation processes of igneous rocks [2].
Igneous rock categorization and identification are critical because they can assist us understand the relative mineral richness of the rock. Based on their chemical/mineral composition, texture, and grain size, igneous rocks can be classified as felsic, intermediate, mafic, or ultramafic: Extrusive igneous rocks are fine-grained i.e., fine textured (infinitesimally small crystals) or amorphous, whereas intrusive igneous rocks are coarse-grained (tiny crystals) (no minerals; no crystalline structure). The porphyritic texture of volcanic rocks, notably felsic and intermediate, is characterised by visible crystals floating in a fine-grained groundmass [3].

Physically identifying each igneous rock by its texture, grain size, colours, faults, and patterns is a difficult process that necessitates the use of a skilled geologist with rock and mineral understanding. The act of identifying and classifying rocks may be arduous and time consuming for persons from other fields. This paper discusses a tool that can aid in the task's completion. We employed some specific deep learning methods to create this tool. Based on the image that the user uploads, our system offers an accurate forecast of rock [4].

The method of deep learning is a machine learning and artificial intelligence (AI) methodology that mimics human learning and biological neural networks. In data science statistical insights and predictive modelling, has embraced deep learning technology as an essential component. Data scientists in charge of acquiring, evaluating, and perceiving large amounts of data gain tremendously from deep learning because it improves and simplifies the task [7]. Deep Learning advancements in computer vision and image identification have been established and improved throughout time, mainly due to the use of a single algorithm: the Convolutional Neural Network.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system which can ingest a picture and assign relevance (learnable weights and biases) to various aspects/objects while
also discerning between them. ConvNet/CNN requires substantially more pre-processing than other classification techniques. With enough training, ConvNets can learn these filters/characteristics [17], whereas simple approaches need filter hand-engineering. Other classification approaches need substantially higher pre-processing than a ConvNet. While foundational approaches need the hand-engineering of filters, CNN can acquire these filters/characteristics with enough training [17]. The extent of pre-processing necessary by a ConvNet is much less than that required by other classification algorithms. While basic approaches need hand-engineering of filters, CNN can learn these filters/characteristics provided with sufficient training [17]. The amount of pre-processing required by a ConvNet is significantly lower than that required by other classification techniques. While simple techniques need filter designing by hand, CNN can learn these filters/characteristics with enough training. The arrangement of the Visual Cortex influenced the construction of a CNN, which is similar to the connection pattern of Neurons in the Human Brain. Individual neurons can only respond to stimuli in the Receptive Field, a tiny portion of the visual field. If a set of comparable fields overlap, the entire visual region is covered. The CNN's role is to compress the images into a format that is easier to handle while keeping crucial properties for accurate prediction [17].

**Literature Review**

Sharma N., Jain V., Mishra A., “An analysis of convolutional networks for image classification”. This study looks at the performance of common convolutional neural networks (CNNs) for recognising objects in real-time image processing. [23].

Xin M. and Wang Y., “Research on image classification model based on deep convolution neural network”. This paper explains architecture of image classification models based deep convolutional neural networks [24].

H. C. Shin; Holger R. Roth; Gao M.; Le Lu; Xu Z.; Nogues I.; Yao J., “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”. Three crucial, but formerly unexplored aspects of using deep convolutional neural networks to solve computer-aided detection challenges are described in this research. To begin, the research and assessment of several CNN architectures are described. The models analysed ranging in size from 5,000 to 160 million variables, with varying amounts of layers. The effect of dataset scale and spatial image context on performance is then determined. [25].

JP Iddings, “The origin of igneous rocks”. This paper discusses in depth analysis of igneous rocks – formation, features and types [26].

MacKenzie W.S., Donaldson C. H. and C. Guilford, “Atlas of igneous rocks and their textures”. The textures of various volcanic and plutonic igneous rocks are discussed [27].

**2. Methodology**

To predict and identify rocks the model first needs to be trained. For training, a good number of images with decent resolution and quality are required. The following flow chart explains the sequence of tasks we are ought to perform to create an accurate model.

| Data collection | Splitting data into training and testing folders | Writing the code |
Data collection and data splitting –

Data collection is the process of obtaining, measuring, and interpreting precise information from a multitude of sources in order to explore solutions to examine situations, answer questions, and assess outcomes [12]. To train our model images of various igneous rocks are required. It is recommended to collect at least 150-200 distinct images of each class of igneous rocks. This paper aims to perform a multiclass classification of 4 types of igneous rocks; however, any number of igneous rocks can be considered if enough image data is available for each class. The four classes of Igneous rocks used in the scope of this paper are andesite, basalt, diorite and obsidian.

Training and testing folders

First create two folders in any directory and name them as “train” and “test”. Create 4 folders of “andesite”, “basalt”, “diorite” and “obsidian” inside each of these training and testing folders. Split the total dataset of each class into a ratio of 80:20 in training and testing folders (For example – If 200 images of Andesite are available, move 160 images to andesite folder present inside training folder and remaining 40 images to andesite folder of testing folder). Once images are moved to their respective folders, the process of training and testing the model begins.

Tools and techniques -

Python is the programming language utilised in this project. Jupyter Notebook, a cloud-based IDE, is used to develop the python code. Because it saves information on a cloud
server and does all processing on the cloud, Jupyter notebooks allow you to interact with applications and APIs that may demand a high system specification (RAM, GPU).

Keras Sequential API is used to generate and train a model. This API simplifies the process of building and training with only a few lines of code. TensorFlow is accompanied by the Keras API. As a result, simply installing TensorFlow on your command prompt will enough [15].

TensorFlow is a machine learning framework that streamlines the entire process. It attributes a diverse ecosystem of tools, libraries, and community resources that enable academics to continue to progress the trailblazing techniques in machine learning and developers to instantaneously compose and implement machine learning optimization algorithms [7][13]. Google designed and published TensorFlow, a Python library for swift computationally intensive tasks [13].

Keras is an API that is based on Python-programming language. It is a deep learning API that operates atop TensorFlow machine learning framework. It was founded with a specific aim of enabling for speedy experimentation and research [8]. Keras uses many optimization methods to make high-level neural network API simpler and more performant. Keras is built on TensorFlow, an open-source machine learning framework. Keras is based on a straightforward framework for creating deep learning models with TensorFlow [7]. It's a simple framework that may be used on both the CPU and GPU. Established neural network deployment approaches either adds substantial latencies, cost a lot of money and time to incorporate into embedded code bases, or specifically supports a few numbers of model types.

---

**Igneous rock classification using CNN**

- By Surenda Patro

```python
In [6]: import numpy as np
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K

In [7]: train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
training_set = train_datagen.flow_from_directory(
    'data/train/',
    target_size=(64, 64),
    batch_size=32,
    class_mode='categorical')
Found 760 images belonging to 4 classes.
```

**Importing the libraries**

Import the TensorFlow Platform and Keras API first. Data augmentation is a crucial aspect of the training process. For data augmentation, we use the "ImageDataGenerator" technique that comes with Keras. Then add the Sequential class to your project. The basic concept of Sequential API is to arrange the Keras layers in a sequential manner, hence the name. Most ANNs have layers that are arranged in a sequential sequence, with data flowing from one layer to the next in the specified order until it reaches the output layer. Then add layers for activation, dropout, flatten, and dense CNN. An activation function governs how the sum
of weighted inputs is generated into an outcome from a network as well as nodes in a layer in a neural network. Dropout is a method of preventing a model beyond being overfit. Dropout converts the information the outgoing points of hidden nodes (neurons that comprises the hidden layers) to 0 at random throughout every cycle of the training procedure [8][15].

Training the model –

Once arranging the images in all folders is completed, we define two variables “train_datagen” and “training_set”. The data gen variable calls out the ImageDataGenerator function which helps in the process of image augmentation. Second variable training set stores the path of the directory in which the training images are stored. These images can be stored locally as well in cloud storage platforms like Google drive and dropbox [11].

ImageDataGenerator( ) class has several arguments, rescale – This argument rescales the RGB pixel value into numbers between 0-1 (usually they’re somewhere near 0 – 255), shear range and zoom range- The flow from directory method has following arguments – The directory path, target size – Pixel size of the image, batch size – Number of images to be taken in a single batch, class mode- Here categorical class mode as an argument is used since we have multiple classes.

Now the process of creating the model starts. To create a CNN model, we are going to add several layers to it using “keras.layers” and add() function. The CNN is made up of three types of standard layers they are-

Convolutional layers, Pooling layers, and Fully-connected (FC) layers. When these layers are combined, a standard CNN architecture is formed. Remaining two significant components are the activation layer and the dropout function [10]. A standard diagram of CNN architecture is shown below.

i. Convolutional Layer-

To detect picture characteristics, a convolution filter is applied to the image. The following is how it works:
A convolution is a function that multiplies a collection of weights by the neural network's inputs. Kernels or screens kernel (for 2D arrays of parameters) or a filter (for 3D structures) passes throughout an image several times during the multiplication process. The filtering is employed from right to left and top to bottom to cover the entire image. During the convolution, a mathematical operation known as the dot or scalar product is used. The weights are multiplied by each filter with various input values. The entire inputs are combined together to provide each filter position a unique value [18][19][21].

ii. Pooling Layer-

The pooling layers shrink the image over time, preserving just the most crucial details. For each set of pixels, the pixel with the peak score is kept (this is known as max pooling), or simply the average is kept (average pooling). By lowering the number of calculations and parameters in the network, pooling layers aid in the management of overfitting. [18].

iii. Fully Connected Layer-

A fully connected neural network is made up of layers that are all linked. A fully connected layer is expressed as a function ranging from R m to R n. Each input dimension determines the output dimension. "Neurons" is a term used to describe the nodes in fully connected networks. As a result, fully connected networks will be referred to as "neural networks" elsewhere in the literature [21].

iv. Dropout-

The Dropout layer, which helps minimise overfitting, changes input unit values to null at random with a frequency of specific rate at each step during training time. Inputs that aren't adjusted to 0 are escalated up by 1/(1 - r), where r is that specific rate that was mentioned, so that the overall sum does not change its value [19].

```python
In [7]: train_dataset = ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True)
training_set = train_dataset.flow_from_directory(
    'data/train/',
    target_size = (64, 64),
    batch_size = 32,
    class_mode = 'categorical')
Found 748 images belonging to 4 classes.

In [8]: test_dataset = ImageDataGenerator(rescale = 1./255)
test_set = test_dataset.flow_from_directory(
    'data/test/',
    target_size = (64, 64),
    batch_size = 32,
    class_mode = 'categorical')
Found 186 images belonging to 4 classes.

In [9]: cm = tf.keras.models.Sequential()
cm.add(tf.keras.layers.Conv2D(filters = 32, kernel_size = 3, activation='relu', input_shape = (64, 64, 3)))
cm.add(tf.keras.layers.MaxPooling2D(pool_size = 2, strides = 2))
cm.add(tf.keras.layers.Conv2D(filters = 64, kernel_size = 3, activation='relu'))
cm.add(tf.keras.layers.MaxPooling2D(pool_size = 2, strides = 2))
cm.add(tf.keras.layers.Dropout(0.5))
cm.add(tf.keras.layers.Flatten())
cm.add(tf.keras.layers.Dense(units = 128, activation='relu'))
cm.add(tf.keras.layers.Dense(units = 4, activation='softmax'))
cm.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

Training the model and adding layers to it
v. Activation Functions-

Furthermore, the activation function can be considered as one of the most critical parts of the CNN model. These functions are employed to train themselves and estimate specific sort of network mode to node correlation, including continuous and complicated. It defines which model characteristics should be transmitted forward and which will be returned at the network’s end, in simple words [12][20].

Note: The activation function employed is RELU, or rectified linear activation function, is a piecewise function that returns 0 if the input is negative and returns 1 if it is positive. Because it is easier to make it learn and often produces better quality results, it has transformed itself into the preferred activation function for several modelling techniques [13].

A nonlinear activation function is applied to the outcomes of a linear operations, such as convolution. Regardless of the fact that relatively smoother nonlinear functions like the sigmoid or hyperbolic tangent (tanh) function were usually prominent as mathematical representations of actual neuron behaviour, the rectified linear unit (ReLU) is now the most prominent nonlinear activation function [22].

\[ f(x) = \max(0, x) \]

**Testing and validating the data**-

Testing data can help us understand whether our model is performing correctly on newer data. In this line of code, we fit the trained CNN model with test data set. The `epoch` argument shows the number of cycles the data passes in and out of the neural network.

---

Testing data
The testing of our model is completed. The performance parameter shows an accuracy of 95%, which is great. *(Note – This accuracy might vary depending upon the type and quality of data).*

**Model summary –**

Here we show our model summary. We need model summary to confirm that everything (layers, arguments and other parameters) is as expected. [14][15]

| In [6]: | cnn.summary() |
|-------|----------------|
| Model: | "sequential" |
| Layer (type) | Output Shape | Param # |
| conv2d (Conv2D) | (None, 62, 62, 64) | 1792 |
| max_pooling2d (MaxPooling2D) | (None, 31, 31, 64) | 0 |
| conv2d_1 (Conv2D) | (None, 29, 29, 64) | 36928 |
| max_pooling2d_1 (MaxPooling2D) | (None, 14, 14, 64) | 0 |
| dropout (Dropout) | (None, 14, 14, 64) | 0 |
| flatten (Flatten) | (None, 12544) | 0 |
| dense (Dense) | (None, 128) | 1605760 |
| dense_1 (Dense) | (None, 4) | 516 |

**CNN model summary**

| Total params: | 1,644,996 |
| Trainable params: | 1,644,996 |
| Non-trainable params: | 0 |

**Performance parameters –**

For performance parameter, two Line graphs are plotted. One is of *accuracy vs epochs* and other is of *loss vs epochs*. The graph shows that the validation and training accuracies were almost identical, with a combined accuracy of over 90%. The CNN model's loss is a negative trailing graph, indicating that the model is functioning as predicted, with the loss decreasing with each epoch [16][19].
Results and Outcome –

After training, validating and testing, a CNN model is created which can identify igneous rocks. To test the model on new data, following lines of python code can be used. In this cell, an input box is provided where path location of any locally saved image is added. (Note – make sure the new image data is locally saved in your device)
After entering the image path -

Once the image path is entered into the input box shown above. The image data is passed through the neural network, which predicts the rock type.

The uploaded test image is showcased -

And finally, the model predicts an outcome. The out is usually in a form of an array. A simple if-else python statement can solve this. Finally, the desired result is obtained.

In this case the Igneous rock is **diorite**!

**Conclusion –**

Classification of igneous rocks using CNN algorithm can help us automate many aspects of geological surveys. First organise the image dataset into two folders “training and testing”. After this the model is provided with several layers like convolutional layer, pooling layer and activation function (In this case RELU). This CNN algorithm can classify images with considerably good accuracy. The four igneous rock classes we’ve taken for this paper are
Andesite, basalt, diorite and obsidian. Once the user inputs a rock image to the algorithm, the result is predicted. In our case we provided an image of Diorite rock, which our model predicted very accurately. There is no restriction in the number of classes. However, it is important to organise the rock images in proper folders and images should be of good quality.

This tool makes geologist’s life easy. Geological surveying can be automated by using these machine learning tools. This rock classification algorithm can also be used for surveying rocks and minerals on a larger scale and also for understanding underlying geological structures. The CNN image classification algorithm can also be used for several other geological/mining problems like predicting the rock fragmentation, predicting the river basin sediments etc...,

References

[1] MacKenzie, William Scott, Colin H. Donaldson, and Corin Guilford. Atlas of igneous rocks and their textures. Vol. 148. Harlow: Longman, 1982.
[2] Gill, Robin. Igneous rocks and processes: a practical guide. John Wiley & Sons, 2011.
[3] Mc Birney, Alexander R. Igneous petrology. Jones & Bartlett Learning, 1993.
[4] Marsh, Bruce D. "On some fundamentals of igneous petrology." Contributions to Mineralogy and Petrology 166.3 (2013): 665-690.
[5] Maaloe, Sven. Principles of igneous petrology. Springer Science & Business Media, 2012.
[6] Cox, Keith Gordon, ed. The interpretation of igneous rocks. Springer Science & Business Media, 2013.
[7] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016.
[8] Arnold, Taylor B. "kerasR: R Interface to the Keras Deep Learning Library." J. Open Source Softw. 2.14 (2017): 296.
[9] Learning, Deep. "Deep learning." High-Dimensional Fuzzy Clustering (2020).
[10] Lu, Dengsheng, and Qihao Weng. "A survey of image classification methods and techniques for improving classification performance." International journal of Remote sensing 28.5 (2007): 823-870.
[11] Kattenborn, Teja, et al. "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing." ISPRS Journal of Photogrammetry and Remote Sensing 173 (2021): 24-49.
[12] Sapsford, Roger, and Victor Jupp, eds. Data collection and analysis. Sage, 1996.
[13] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016.
[14] Gulli, Antonio, and Sujit Pal. Deep learning with Keras. Packt Publishing Ltd, 2017.
[15] Moolayil, Jojo. "An introduction to deep learning and keras." Learn Keras for Deep Neural Networks. Apress, Berkeley, CA, 2019. 1-16.
[16] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 international conference on engineering and technology (ICET). Ieee, 2017.
[17] Hussain, Mahbub, Jordan J. Bird, and Diego R. Faria. "A study on cnn transfer learning for image classification." UK Workshop on computational Intelligence. Springer, Cham, 2018.
[18] Bakhshi, Ali, Stephan Chalup, and Nasimul Noman. "Fast evolution of CNN architecture for image classification." Deep Neural Evolution. Springer, Singapore, 2020. 209-229.
[19] Jmour, Nadia, Sehla Zayen, and Afef Abdelkrim. "Convolutional neural networks for image classification." 2018 international conference on advanced systems and electric technologies (IC_ASET). IEEE, 2018.
[20] Sharma, Sagar, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." Towards data science 6.12 (2017): 310-316.
[21] Ma, Wei, and Jun Lu. "An equivalence of fully connected layer and convolutional layer." arXiv preprint arXiv:1712.01252 (2017).
[22] Agarap, Abien Fred. "Deep learning using rectified linear units (relu)." arXiv preprint arXiv:1803.08375 (2018).
[23] Sharma, Neha, Vibhor Jain, and Anju Mishra. "An analysis of convolutional neural networks for image classification." Procedia computer science 132 (2018): 377-384.
[24] Xin, Mingyuan, and Yong Wang. "Research on image classification model based on deep convolution neural network." EURASIP Journal on Image and Video Processing 2019.1 (2019): 1-11.
[25] Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE transactions on medical imaging 35.5 (2016): 1285-1298.
[26] Iddings, Joseph Paxson. The origin of igneous rocks. Vol. 12. Philosophical Society, 1892.
[27] MacKenzie, William Scott, Colin H. Donaldson, and Corin Guilford. Atlas of igneous rocks and their textures. Vol. 148. Harlow: Longman, 1982.