Classifying Minerals using Deep Learning Algorithms

Tajendra Singh1, a, Dr. D.C. Jhariya2, b, *, Dr. Mridu Sahu3, c, Dr Pankaj Dewangan4, d, Dr. P.Y. Dhekne5, e.

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

Email: *singhtajendra01@gmail.com; bdcjhariya.geo@nitrr.ac.in; cmrisahu.it@nitrr.ac.in; dpankajnitrr@yahoo.co.in, epdhekne@nitrr.ac.in

Abstract: A mineral is an inorganic substance that occurs in nature with specific chemical content and an ordered atomic positioning. Minerals are identified by their physical properties. Minerals’ physical properties are related to their chemical composition and bonding. Quartz is extremely valuable economically. Valuable minerals and some examples of gemstones are citrine, amethyst, quartz with smoky texture and quartz of rose color can be said as rose quartz are some examples of gemstones. Sandstone, primarily composing quartz, is the most used building stone. Biotite has limited number of applications for commercial use. Deep learning is the subset of machine learning. It is based on self-learning and improvement through the examination of computer algorithms. TensorFlow library of machine learning combines a number of different algorithms and models which allows users to build deep neural networks for projects/model such as image recognition/classification and many more. Image Classification is the assignment of one label from a fixed set of categories to an input image. In this paper Convolutional neural networks (CNNs) are used primarily for image processing, classification, segmentation, and other auto-correlated data. This paper will explain the techniques and explanation for classifying minerals images using a deep learning algorithm called a convolutional neural network. Identifying minerals on a field is a tedious activity and requires a lot of information and conformation here with the help of deep learning algorithms we made a deep learning model which has all its feature already embedded in it that can be used to classify minerals with a reasonable accuracy furthermore in future it can be made more accurate and fit accordingly to the conditions.

Keywords: Minerals, Quartz, Biotite, Deep learning, TensorFlow, Keras, Convolutional neural network.
1. Introduction

Minerals either alone or in a combination called compounds forms the Earth. Mineral can be of a single element or can be compound. It is an inorganic substance that occurs naturally having specified chemical composition and ordered atomic structure.

Mineral Economics (ME) Division of the Indian Bureau of Mines was established in 1948. It was elevated to the rank of division of the department in 1956. The Mineral Economics division provides information support and advisory services to the mineral industry and the government, mainly being focused on issues of marketing, mineral specifications, uses of minerals, mineral legislation, mineral resource inventory, mining leases and taxation, and many more.

In India, Metallic minerals are valued at Rs 54,000 crore, while for non-metallic minerals including minor minerals, are valued at Rs 61,000 crore, according to the report. According to the reports of 2016-17 the value of metallic minerals such as ores of: iron, manganese, zinc, bauxite, copper, gold(native), chromite, and lead was around Rupees 40,000 crore.

Andhra Pradesh (AP), Chhattisgarh (CG), Goa (GA), Gujarat (GJ), Jharkhand, Karnataka, Madhya Pradesh (MP), Maharashtra (MH), Orissa (OR), Rajasthan (RJ), and Tamil Nadu(TN) are among the states with the most mineral reserves and contribute significantly to the country's mineral production.

India's major mineral resources constitute Iron Ore, Coal, Mica, Manganese ore, Bauxite Chromite, Diamonds, Limestone, and Thorium.

Identification of Minerals on Field: Minerals can be distinguished by a variety of characteristics. Colour, streak, lustre, hardness, crystal form, cleavage, specific gravity, and habit are the most frequent characteristics used to identify minerals. Even when a geologist is out in the field, most of these can be easily analysed. Mineral identification is required to identify rocks and can be used to understand the landscape also the geologic history of the area.

Deep learning: Is one of the types of machine learning in which computers tends to learn by doing, much like people do. It is also an important component of self-driven cars because it allows them to identify traffic signs and distinguish between pedestrians and obstacles on road. In this, a computer model learns to perform classification tasks by taking input as pictures, text, or sound directly. These models may achieve the greatest accuracy of time also sometimes they can even outperform humans. Multilayer neural network topologies and a large amount of labelled data are used to train models. Depending on the type of the earth science exploration, a variety of algorithms may be used For specific goals, number of algorithms may perform much more accurate than others.

CNN: Artificial neural networks (ANN) are good at solid classification and convolutional neural networks (CNN) are good at throwing light on photographs, but they are more computationally costly to train than support-vector machine (SVM) learning. In recent decades, machine learning has grown in popularity as other technologies such as unmanned aerial vehicles (UAVs), ultra-high resolution remote sensing technology, and high-performance computer units have made vast, high-quality datasets and more powerful algorithms available.
The CNN is an artificial neural network that is being used for image identification and processing, because of its capacity to detect patterns in photos. A CNN is a powerful tool, but training it takes millions of labelled data points\(^\text{[12]}\). If CNNs are to generate results quickly enough to be effective, they must be trained with high-power processors such as a GPU or an NPU.

**CNN for Mineral Classification:** Extracting characteristics from a picture (in this example of biotite and quartz) in order to discover trends in a dataset is known as image classification\(^\text{[13]}\). ANN for image classification would be highly computationally costly due to the large number of trainable parameters\(^\text{[14]}\). Let’s say, if there are 50 X 50 image of a minerals and want to train our standard ANN to classify it as biotite or quartz, the trainable parameters are \( (50*50) \times 2,50,302 \) image pixels multiplied by hidden layer + 100 bias + 2 = \( 2,50,302 \) image pixels multiplied by hidden layer + 100 bias + 2.

**Problem Statement:** Mineral exploration is costly and risky due to limited geologic knowledge and an incomplete understanding of the ore-forming processes that control mineral distribution over geologic space and over hundreds of millions of years of geologic time. Classifying them by the means of deep learning model will come out to be the great change in the mineral identification problem and have potential to overcome the difficulties during the process.

2. **Literature Review**

Folorunso, I. O., et al. "A rule-based expert system for mineral identification." Journal of Emerging Trends in Computing and Information. In this project, an expert system is used to identify minerals. The expert system's design and execution are based on the physical characteristics of the forty minerals studied as the knowledge domain. The inference engine is rule-based, as past research has shown to be more effective.

Mlynarczuk, Mariusz, and Marta Skiba in "The application of artificial intelligence for the identification of the maceral groups and mineral components of coal." Computers & Geosciences 103 (2017). It was demonstrated that artificial intelligence approaches may be used to successfully identify specific petrographic properties of coal.

Zhang, Ye, in. "Intelligent identification for rock-mineral microscopic images using ensemble machine learning algorithms." Sensors 19.18 (2019): Deep learning and convolutional neural networks (CNNs) provide a mechanism for rapidly and intelligently analysing mineral microscopic pictures. The transfer learning model of mineral microscopic pictures is developed in this study using the Inception-v3 architecture.

Maitre, Julien, Kévin Bouchard, and L. Paul Bédard. "Mineral grains recognition using computer vision and machine learning." Computers & Geosciences 130 (2019): Computer vision combined with machine learning can categorise minerals in sand grains, with mineral recognition results reaching 90% accuracy.
3. Methodology

For a good and accurate working model, it first needs to be trained on a dataset containing separate folder for each class as in this case “Biotite” & “Quartz”, of good quality images that are free of background noises, so the first step is to collect the data that will be fed into the model.

3.1 Data Collection

Data collection is the systematic way/technique of collection and gathering of data in the systematic form with features of interest clubbed together so that it can be used for testing hypotheses and for evaluating the outcome of problems \[^{[15]}\]. For gathering the image data of minerals, using “bingimagedownloader” api, for each class there are 199 images (Biotite and Quartz) so total is 199*2 which is 398. Make folder for each class and name them accordingly \[^{[16]}\].

Now, compress this folder and upload it to your cloud account so that it can be imported directly for use. For this “dropbox” is used as it creates direct shareable link.
3.2 Tools used

For writing script in python google colab is used as development environment. It's a cloud-based integrated development environment [17]. Colab is thought to be more portable and simpler to use than Jupyter since it is easier to set up. Colab also facilitates team collaboration, which is not available with Jupyter [18].

For this project, Python syntax is used. As a framework, Keras is implemented, It is a Python-based high-level neural network API. But for lower-level tasks it cannot be utilized or function without a backend [19].

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 [20].

Step 1: Import the libraries you’ll need.

For this project, Keras library is used for building and training of the deep learning model. Also used Matplotlib to visualise our dataset so that we can better grasp the photos we'll be dealing with. Opencv is another useful library for dealing with picture data.

```python
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
```

TensorFlow accepts tensor data forms, tensors are high-dimensional and multi-dimensional arrays. These multi-dimensional arrays are useful for taking in account with large volumes of data [21].
TensorFlow is based on graphs that depict data flows and have nodes and edges. Because the execution method is in the form of graphs, TensorFlow code is significantly easier to spread over a cluster of computers when using GPUs [22].

Keras is an API of high-level in deep learning has a use for creating neural networks developed by Google. Its script is written in Python and is made for simplification to the development of neural networks [23]. Backend calculation of several neural networks is also enabled in keras.

2nd Step of Methodology: Next, is to define the data's path, and will be downloading the data for the url that we had created before. Name the folder same as it was in compressed files(“Minerals”).

Code is shown below:

```python
import pathlib
dataset_url = "https://dl.dropboxusercontent.com/s/w0jyd2f23mk6wu0/minerals2.tgz?dl=0"
data_dir = tf.keras.utils.get_file(‘minerals’, origin=dataset_url, untar=True)
data_dir = pathlib.Path(data_dir)
```

Figure 4: Importing the dataset

Next is checking if the data is imported correctly or not.

Code along with the output is shown below:

```python
[ ] biotite = list(data_dir.glob(‘biotite/*’))
print(biotite[0])
PIL.Image.open(str(biotite[0]))
```

Figure 5: Dataset Output
The third step is to split the dataset. Setting up the training, validation and test sets has a huge impact on productivity. It is important to choose the validation and test sets from the same distribution and it must be taken randomly from all the data [24].

```python
[ ] img_height, img_width=180,180
batch_size=32
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
data_dir,
validation_split=0.2,
subset="training",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

Found 371 files belonging to 2 classes.
Using 297 files for training.

[ ] val_ds = tf.keras.preprocessing.image_dataset_from_directory(
data_dir,
validation_split=0.2,
subset="validation",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

Found 371 files belonging to 2 classes.
Using 74 files for validation.
```

Figure 6: Splitting the dataset into training and validation classes

**Step 3: Create a visual representation of the data.**

Let's take a look at the data and see what we're dealing with. Here, utilises “mat plot” library for plotting the total quantity of photos in both classes, and result can be seen below.

```python
[ ] import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(9):
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.title(class_names[labels[i]])
    plt.axis("off")

Figure 7: Checking the Split Dataset.
The next step is to Define the model:

Starting with the simple CNN model having three Convolutional layers and then three max-pooling layers [25]. Succeeding the third maxpool operation a dropout layer was inserted so that overfitting is avoided.

```
num_classes = 2
model = Sequential([
    layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])
```

Figure 8: Importing Functions

The input is half padded when padding == "SAME." Because the size of the output is same as of input (with stride=1), the padding type is termed SAME. Using the keyword 'SAME' ensures that the filter is applied to all of the input's items. Padding is usually set to "SAME" when training the model [26]. For subsequent computation, the output size is mathematically convenient [27].

Activation function is originally a basic function for turning the inputs into the outputs within a certain range of values, and is set to "relu." In contrast to other forms of activation functions, the sigmoid activation function accepts input and converts the output values between 0 and 1 in a unique fashion [28].

For compilation of the model optimization type is chosen as Adam optimizer, and for the loss function, the SparseCategoricalCrossEntropy is chosen. For obtaining the smoother curve learning rate was set to “1x10^-6” [29].

Here is the code:

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])#compiling the model
```

Figure 9: Compiling Model
This graph is showing the learning rate with the loss of pictures approximately=0.4% (which is a pretty less loss value and indicates an accurate result). Learning rate and loss graph was obtained to check if the training of the model was not up to the mark or not, it tells the rate at which model is being trained and also the loss of features due to the learning rate, a good learning rate is much needed so that loss of the features is less and also the training of the model is efficient.

![Learning Rate and Loss Graph](image)

**Figure 10: Learning Rate and Loss**

The next step is to train our model. Now, because our learning rate is so low, let's train our model for 10 epochs.

```python
[ ] epochs=10
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)
```

**Figure 11: Setting Epochs for model fitting**

On the next, the output is shown along with the time, loss, accuracy, valid loss and valid accuracy.
Further Step: Assessing the outcome:

Training and Validation accuracy and as well as training and validation loss was plotted.

Now for the model summary:

Keras makes it possible to summarise a model.

The textual summary contains information about:

- The layers of the model, as well as their arrangement.
- The ultimate output form of each layer.
- About parameters of each tier (weights)
Also, all the parameters in the model (weights). The summary is constructed by invoking the model’s summary() method, which returns a string that may then be printed[30].

```
model.summary()
```

| Layer (type)        | Output Shape       | Param # |
|---------------------|--------------------|---------|
| rescaling (Rescaling) | (None, 180, 180, 3) | 0       |
| conv2d (Conv2D)     | (None, 180, 180, 16) | 448     |
| max_pooling2d (MaxPooling2D) | (None, 90, 90, 16) | 0       |
| conv2d_1 (Conv2D)   | (None, 90, 90, 32)  | 4640    |
| max_pooling2d_1 (MaxPooling2D) | (None, 45, 45, 32) | 0       |
| conv2d_2 (Conv2D)   | (None, 45, 45, 64)  | 18496   |
| max_pooling2d_2 (MaxPooling2D) | (None, 22, 22, 64) | 0       |
| flatten (Flatten)   | (None, 30976)      | 0       |
| dense (Dense)       | (None, 128)        | 3965056 |
| dense_1 (Dense)     | (None, 2)          | 258     |

Total params: 3,988,898
Trainable params: 3,988,898
Non-trainable params: 0

Figure 14: Model Summary Output

Next step is to:

A function is defined which can accept images, and reshape those images to (180, 180):

```
def predict_image(img):
    img_4d=img.reshape(-1,180,180,3)
    prediction=model.predict(img_4d)[0]
    return {class_names[i]: float(prediction[i]) for i in range(2)} #function to reshape image so that it can be fed into the input
```

Next step is to deploy the model on web:
For deployment of my deep learning model, gradio is used because, Gradio is a free, open-source Python toolkit that lets you quickly develop UI components for your machine learning model that are easy to use and customise[31].
Gradio makes it easier to use by integrating the GUI straight into your Python notebook.

Install gradio:

```
!pip install gradio # for deployment of model on web as demo.

import gradio as gr
```

Figure 15: Importing gradio

A simple pre-made web template for taking input from local host and to show the images along with prediction made by model.

Gradio can be used as a webpage or embedded in Python notebooks. A Gradio interface can build a public connection automatically. Furthermore, model result is displayed by taking an input picture, into the UI, it can be easily fed to the model by drag and drop function or by directly clicking the upload button.

4. Model Result

Figure 16: Simple UI for accepting images and giving output.

Images can be uploaded by drag and drop or can be uploaded directly:

The uploaded image can be seen the frame and can be edited in that frame also.

Figure 17: Prediction output for biotite image.
5. Conclusion

A deep learning model which can classify minerals using CNN can be deployed with reasonable accuracy. In this paper, employed Convolutional Neural Networks (CNN) for image categorization in this paper, and we used photos that got using the bingimagedownloader api for “Biotite” and “Quartz”. This data collection was used for both training and testing using CNN. It has a 98 percent accuracy rating. The processing time for these images is significantly long when compared to regular JPEG shots. Image classification results will be more accurate if clusters of GPUs are used to stack the model with extra layers and train the network with more picture data. The next update will concentrate on identifying large-scale colored images, which will be highly valuable in the image segmentation process. Although it has a great accuracy but it can be better if a large number of noise free images being used for better precision and accuracy can be used classifying minerals.

References:

[1] Magnus, Albertus. The Book of Minerals. Dalcassian Publishing Company, 1967.

[2] Brown, J. Coggin. India's Mineral wealth: A guide to the occurrences and economics of the useful minerals of the Indian Empire. Oxford University Press, London, 1936.

[3] Kalra, G. D. "The Indian minerals industry—an overview." Minerals and Energy 7.2 (1990): 35-37.

[4] MINES, INDIAN BUREAU OF. "Indian Minerals Yearbook 2016." Gov. India Minist. Mines Nagpur 13 (2018): 1-17.

[5] Randive, Kirtikumar, and Sanjeevani Jawadand. "Strategic minerals in India: present status and future challenges." Mineral Economics 32.3 (2019): 337-352.

[6] Bowie, S. HU, and K. Taylor. A system of ore mineral identification. No. A/CONF. 15/P/42. Geological Survey, London, 1959.

[7] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[8] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[9] Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. Deep learning. Vol. 1. Cambridge, MA, USA: MIT press, 2017.

[10] 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.

[11] Roska, Tamas, and Leon O. Chua. "The CNN universal machine: an analogic array computer." IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 40.3 (1993): 163-173.
[12] Wang, Jiang, et al. "Cnn-rnn: A unified framework for multi-label image classification." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[13] Sang, Xiancheng, et al. "One-Dimensional Deep Convolutional Neural Network for Mineral Classification from Raman Spectroscopy." *Neural Processing Letters* 54.1 (2022): 677-690.

[14] Lou, Wei, Dexian Zhang, and Richard C. Bayless. "Review of mineral recognition and its future." *Applied Geochemistry* 122 (2020): 104727.

[15] Weller, Susan C., and A. Kimball Romney. *Systematic data collection*. Vol. 10. Sage publications, 1988.

[16] Couper, Mick P. "The future of modes of data collection." *Public Opinion Quarterly* 75.5 (2011): 889-908.

[17] Carneiro, Tiago, et al. "Performance analysis of google colaboratory as a tool for accelerating deep learning applications." *IEEE Access* 6 (2018): 61677-61685.

[18] Kluyver, Thomas, et al. *Jupyter Notebooks-a publishing format for reproducible computational workflows*. Vol. 2016. 2016.

[19] Raschka, Sebastian. *Python machine learning*. Packt publishing ltd, 2015.

[20] Gulli, Antonio, and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[21] Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., ... & Saurous, R. A. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.

[22] Pang, Bo, Erik Nijkamp, and Ying Nian Wu. "Deep learning with tensorflow: A review." *Journal of Educational and Behavioral Statistics* 45.2 (2020): 227-248.

[23] Ketkar, Nikhil. "Introduction to keras." *Deep learning with Python*. Apress, Berkeley, CA, 2017. 97-111.

[24] Wei, Qiong, and Roland L. Dunbrack Jr. "The role of balanced training and testing data sets for binary classifiers in bioinformatics." *PloS one* 8.7 (2013): e67863.

[25] Jung, Wonkyung, et al. "Restructuring batch normalization to accelerate CNN training." *Proceedings of Machine Learning and Systems* 1 (2019): 14-26.

[26] Liu, Guilin, et al. "Partial convolution based padding." *arXiv preprint arXiv:1811.11718* (2018).

[27] Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." *arXiv preprint arXiv:1603.07285* (2016).

[28] Sharma, Sagar, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." *Towards data science* 6.12 (2017): 310-316.
[29] Jais, Imran Khan Mohd, Amelia Ritahani Ismail, and Syed Qamrun Nisa. "Adam optimization algorithm for wide and deep neural network." *Knowledge Engineering and Data Science* 2.1 (2019): 41-46.

[30] Shlezinger, Nir, et al. "Model-based deep learning." *arXiv preprint arXiv:2012.08405* (2020).