Abhay Singh

Email: abhay.s-26@scds.saiuniversity.edu.in

School of Computing and Data Science

Hutton Rock Dataset

Total Number of images: 505

- The number of images in granite class are: 187
- The number of images in basalt class are: 130
- The number of images in coal class are: 85
- The number of images in andesite class are: 103

378 images are used for trining and 127 images are used for testing.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import tensorflow as tf
print(tf.__version__)

from tensorflow import keras
tf.random.set_seed(42)

import numpy as np
np.random.seed(42)

import matplotlib.pyplot as plt
%matplotlib inline

import glob
import PIL
from PIL import Image
```

```
2.17.1
```

```
from numpy import load

X_train_std = load('/content/drive/MyDrive/DLPROJECT/X_train_std.npy')
X_test_std = load('/content/drive/MyDrive/DLPROJECT/X_test_std.npy')

y_train = load('/content/drive/MyDrive/DLPROJECT/y_train.npy')
y_test = load('/content/drive/MyDrive/DLPROJECT/y_test.npy')
```

```
ConvNeXtTiny=keras.applications.ConvNeXtTiny(
    include_top=False,
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/convnext/convnext_tiny_notop.h5
111650432/111650432 ━━━━━━━━━━━━━━━━━━━━ 6s 0us/step
```

```
base_model3_TL = keras.applications.DenseNet201(weights='imagenet',
                                                 include_top=False)
for layer in base_model3_TL.layers:
  layer.trainabe = False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/densenet/densenet201_weights_tf_dim_ordering_tf_k
74836368/74836368 ━━━━━━━━━━━━━━━━━━━━ 4s 0us/step
```

```
# Adding classifier

global_pool = keras.layers.GlobalAveragePooling2D()(base_model3_TL.output)

Drop_out = keras.layers.Dropout(rate=0.15)(global_pool)

output_ = keras.layers.Dense(units=4, activation='softmax')(Drop_out)


model3_TL = keras.models.Model(inputs=[base_model3_TL.input], outputs=[output_])
```

```
model3_TL.summary()
```

Model: "functional_4"

Layer (type)	Output Shape	Param #	Connected to
input_layer_5 (InputLayer)	(None, None, None, 3)	0	-
zero_padding2d (ZeroPadding2D)	(None, None, None, 3)	0	input_layer_5[0][0]
conv1_conv (Conv2D)	(None, None, None, 64)	9,408	zero_padding2d[0][0]
conv1_bn (BatchNormalization)	(None, None, None, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, None, None, 64)	0	conv1_bn[0][0]
zero_padding2d_1 (ZeroPadding2D)	(None, None, None, 64)	0	conv1_relu[0][0]
pool1 (MaxPooling2D)	(None, None, None, 64)	0	zero_padding2d_1[0][0]
conv2_block1_0_bn (BatchNormalization)	(None, None, None, 64)	256	pool1[0][0]
conv2_block1_0_relu (Activation)	(None, None, None, 64)	0	conv2_block1_0_bn[0][…
conv2_block1_1_conv (Conv2D)	(None, None, None, 128)	8,192	conv2_block1_0_relu[0…
conv2_block1_1_bn (BatchNormalization)	(None, None, None, 128)	512	conv2_block1_1_conv[0…
conv2_block1_1_relu (Activation)	(None, None, None, 128)	0	conv2_block1_1_bn[0][…
conv2_block1_2_conv (Conv2D)	(None, None, None, 32)	36,864	conv2_block1_1_relu[0…
conv2_block1_concat (Concatenate)	(None, None, None, 96)	0	pool1[0][0], conv2_block1_2_conv[0…
conv2_block2_0_bn (BatchNormalization)	(None, None, None, 96)	384	conv2_block1_concat[0…
conv2_block2_0_relu (Activation)	(None, None, None, 96)	0	conv2_block2_0_bn[0][…
conv2_block2_1_conv (Conv2D)	(None, None, None, 128)	12,288	conv2_block2_0_relu[0…
conv2_block2_1_bn (BatchNormalization)	(None, None, None, 128)	512	conv2_block2_1_conv[0…
conv2_block2_1_relu (Activation)	(None, None, None, 128)	0	conv2_block2_1_bn[0][…
conv2_block2_2_conv	(None, None, None, 32)	36,864	conv2_block2_1_relu[0…

```
model3_TL.compile(loss='sparse_categorical_crossentropy',
                optimizer='adam',
                metrics=['accuracy'])

callbacks_TL = [
            keras.callbacks.ModelCheckpoint("bestTL3.weights.h5",
                                            monitor='val_accuracy',
                                            save_weights_only=True,
                                            save_best_only=True)]

history_TL3 = model3_TL.fit(x = X_train_std, y = y_train, epochs=10,
                        validation_split=0.1, batch_size=32, callbacks=callbacks_TL)
```

```
Epoch 1/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 830s 40s/step - accuracy: 0.5363 - loss: 1.3094 - val_accuracy: 0.4359 - val_loss: 80.5229
Epoch 2/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 8s 756ms/step - accuracy: 0.6388 - loss: 0.9866 - val_accuracy: 0.3333 - val_loss: 1506.2859
Epoch 3/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 8s 749ms/step - accuracy: 0.7656 - loss: 0.6408 - val_accuracy: 0.3590 - val_loss: 276.5112
Epoch 4/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 10s 754ms/step - accuracy: 0.8198 - loss: 0.4295 - val_accuracy: 0.2308 - val_loss: 17.3383
Epoch 5/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 10s 756ms/step - accuracy: 0.8702 - loss: 0.3419 - val_accuracy: 0.3590 - val_loss: 45.9896
Epoch 6/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 10s 753ms/step - accuracy: 0.8723 - loss: 0.3539 - val_accuracy: 0.3590 - val_loss: 146.7544
Epoch 7/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 13s 1s/step - accuracy: 0.8637 - loss: 0.3265 - val_accuracy: 0.4615 - val_loss: 271.4401
```

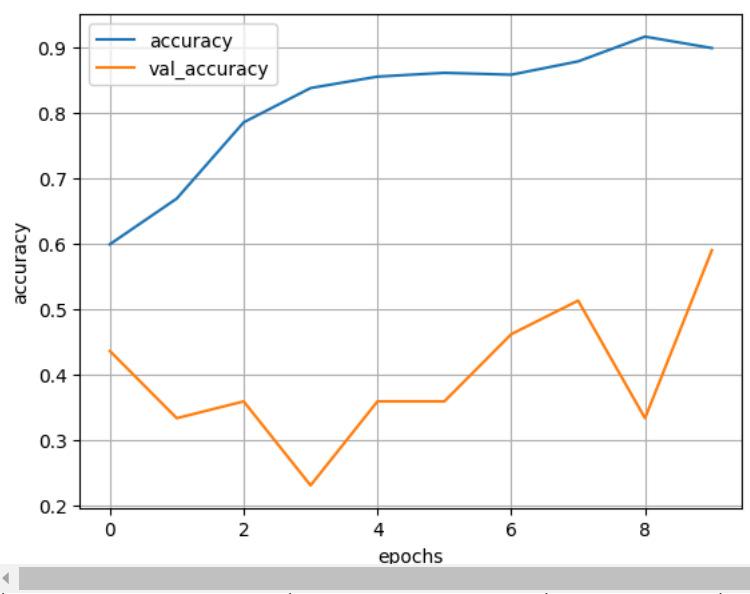
```
Epoch 8/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 11s 984ms/step - accuracy: 0.8867 - loss: 0.3292 - val_accuracy: 0.5128 - val_loss: 114.3833
Epoch 9/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 18s 750ms/step - accuracy: 0.8996 - loss: 0.2730 - val_accuracy: 0.3333 - val_loss: 30.6528
Epoch 10/10
11/11 ━━━━━━━━━━━━━━━━━━━━ 12s 970ms/step - accuracy: 0.9018 - loss: 0.2668 - val_accuracy: 0.5897 - val_loss: 10.5563
```

```
keys = ['accuracy', 'val_accuracy']
progress = {k:v for k,v in history_TL3.history.items() if k in keys}

import pandas as pd
pd.DataFrame(progress).plot()

plt.xlabel("epochs")
plt.ylabel("accuracy")

plt.grid(True)
plt.show()
```




```
model3_TL.load_weights("bestTL3.weights.h5")
```

```
model3_TL.save('/content/drive/MyDrive/DLPROJECT/01_ConvNeXtTiny_TransferLearning_Best_Model.keras')
```

conv2_block6_2_conv (Conv2D)	(None, None, None, 32)	36,864	conv2_block6_1_relu[0…
conv2_block6_concat (Concatenate)	(None, None, None, 256)	0	conv2_block5_concat[0… conv2_block6_2_conv[0…
pool2_bn (BatchNormalization)	(None, None, None, 256)	1,024	conv2_block6_concat[0…
pool2_relu (Activation)	(None, None, None, 256)	0	pool2_bn[0][0]
pool2_conv (Conv2D)	(None, None, None, 128)	32,768	pool2_relu[0][0]
pool2_pool (AveragePooling2D)	(None, None, None, 128)	0	pool2_conv[0][0]
conv3_block1_0_bn (BatchNormalization)	(None, None, None, 128)	512	pool2_pool[0][0]
conv3_block1_0_relu (Activation)	(None, None, None, 128)	0	conv3_block1_0_bn[0][…