DATA AUGMENTATION

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

```
     Mounted at /content/gdrive
```

```
!pip install augmentor
```

```
     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Collecting augmentor
       Downloading Augmentor-0.2.12-py2.py3-none-any.whl (38 kB)
     Requirement already satisfied: Pillow>=5.2.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (
     Requirement already satisfied: tqdm>=4.9.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (4.
     Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (
     Installing collected packages: augmentor
     Successfully installed augmentor-0.2.12
```

```
!pip install augmentor
# Importing necessary library
import Augmentor
# Passing the path of the image directory
p = Augmentor.Pipeline('/content/gdrive/MyDrive/Hidden hunger/original dataset')

# Defining augmentation parameters and generating 5 samples
p.zoom(probability = 0.5, min_factor = 0.8, max_factor = 1.5)
```

Saving...

```
     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Requirement already satisfied: augmentor in /usr/local/lib/python3.10/dist-packages (0.2.12)
     Requirement already satisfied: Pillow>=5.2.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (
     Requirement already satisfied: tqdm>=4.9.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (4.
     Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from augmentor) (
     Initialised with 5827 image(s) found.
     Output directory set to /content/gdrive/MyDrive/Hidden hunger/original dataset/output.Processing <PIL.Jpeg
```

```
pip install split-folders
```

```
     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Collecting split-folders
       Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
     Installing collected packages: split-folders
     Successfully installed split-folders-0.5.1
```

```
import splitfolders

input_folder = r'/content/gdrive/MyDrive/Hidden hunger/original dataset/output'
splitfolders.ratio(input_folder, output= r'/content/gdrive/MyDrive/augmentation 8000/split dataset_8k',
                   seed=42, ratio=(.8, .2),
                   group_prefix=None)
```

```
     Copying files: 1439 files [01:00, 23.69 files/s]
```

```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.inception_v3 import preprocess_input
from tensorflow.keras.applications.inception_v3 import preprocess_input
from tensorflow.keras import layers
from tensorflow.python.keras.layers import Dense, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
```

```
train_path = '/content/gdrive/MyDrive/hidden hunger/split dataset_orig/train'
test_path = '/content/gdrive/MyDrive/hidden hunger/split dataset_orig/val'
```

```
from tensorflow.keras.layers import Input,Lambda,Dense,Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
```

```
IMAGE_SIZE = [299,299]
```

```
# Use the Image Data Generator to import the images from the dataset
from tensorflow.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,
                                  )
```

Saving...

```
                                   rget size as initialied for the image size
train_set=train_datagen.flow_from_directory('/content/gdrive/MyDrive/hidden hunger/split dataset_orig/train',tar
                                                 batch_size=32,
                                                 class_mode='categorical')

     Found 4608 images belonging to 6 classes.
```

```
test_set = test_datagen.flow_from_directory('/content/gdrive/MyDrive/hidden hunger/split dataset_orig/val',
                                            target_size = (299, 299),
                                            batch_size = 32,
                                            class_mode = 'categorical')

     Found 1155 images belonging to 6 classes.
```

```
class_name = train_set.class_indices
print(class_name)

     {'Iodine Deficiency': 0, 'Vitamin - B12  Deficiency': 1, 'Vitamin D deficiency': 2, 'Zinc Deficiency': 3,
```

```
inceptionv3_model = Sequential()
pretrained_model = InceptionV3(input_shape=(299,299,3), weights='imagenet', include_top=False,
                   pooling='avg',classes=6)
for layer in pretrained_model.layers:
        layer.trainable=False

inceptionv3_model.add(pretrained_model)
inceptionv3_model.add(Flatten())
inceptionv3_model.add(Dense(512, activation='relu'))
inceptionv3_model.add(Dense(6, activation='softmax'))

     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_
     87910968/87910968 [==============================] - 3s 0us/step
```

```
inceptionv3_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 inception_v3 (Functional)   (None, 2048)              21802784  
                                                                 
 flatten (Flatten)           (None, 2048)              0         
                                                                 
 dense (Dense)               (None, 512)               1049088   
                                                                 
 dense_1 (Dense)             (None, 6)                 3078      
                                                                 
=================================================================
Total params: 22,854,950
Trainable params: 1,052,166
Non-trainable params: 21,802,784
_________________________________________________________________
```

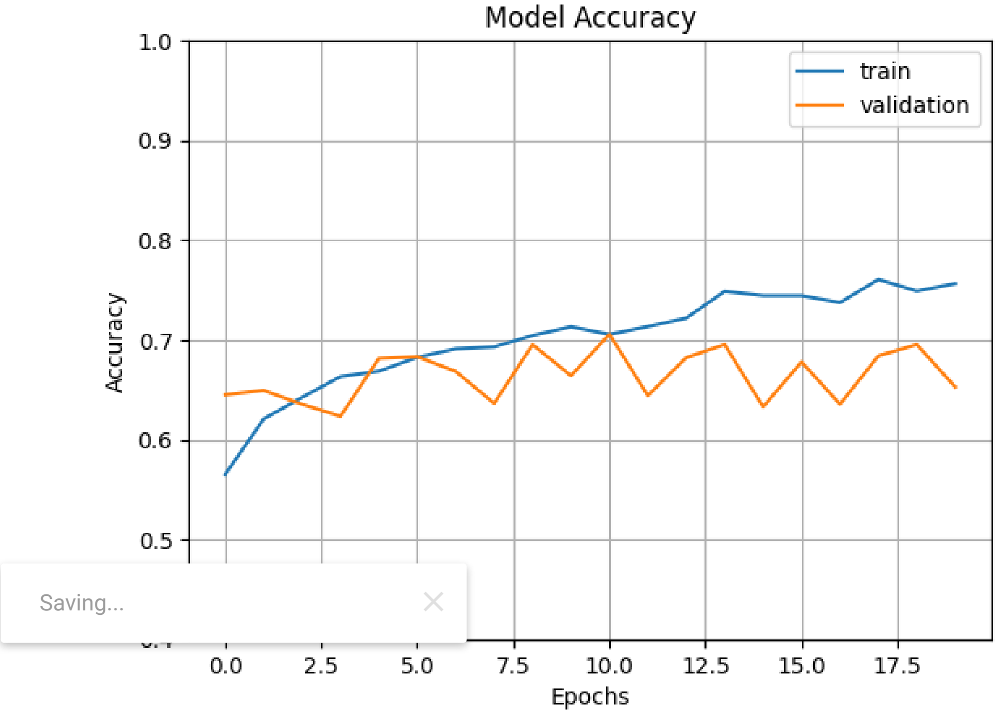
```
inceptionv3_model.compile(optimizer=Adam(learning_rate=0.01),loss='categorical_crossentropy',metrics=['accuracy'
```

```
epochs=20
history = inceptionv3_model.fit(train_set,validation_data=test_set,epochs=epochs)
```

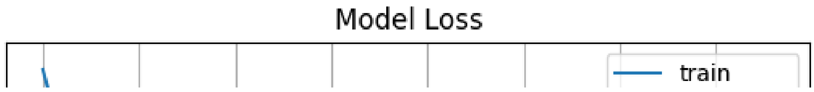
```
Epoch 1/20
144/144 [==============================] - 1221s 9s/step - loss: 1.1623 - accuracy: 0.5651 - val_loss: 1.0
Epoch 2/20
144/144 [==============================] - 127s 883ms/step - loss: 1.0275 - accuracy: 0.6207 - val_loss: 0
Saving...
         =========] - 129s 896ms/step - loss: 0.9584 - accuracy: 0.6424 - val_loss: 1
144/144 [==============================] - 127s 884ms/step - loss: 0.9387 - accuracy: 0.6632 - val_loss: 1
Epoch 5/20
144/144 [==============================] - 128s 888ms/step - loss: 0.9003 - accuracy: 0.6686 - val_loss: 0
Epoch 6/20
144/144 [==============================] - 129s 896ms/step - loss: 0.8587 - accuracy: 0.6825 - val_loss: 0
Epoch 7/20
144/144 [==============================] - 129s 894ms/step - loss: 0.8403 - accuracy: 0.6910 - val_loss: 0
Epoch 8/20
144/144 [==============================] - 128s 891ms/step - loss: 0.8238 - accuracy: 0.6929 - val_loss: 0
Epoch 9/20
144/144 [==============================] - 127s 880ms/step - loss: 0.7866 - accuracy: 0.7044 - val_loss: 0
Epoch 10/20
144/144 [==============================] - 130s 899ms/step - loss: 0.7678 - accuracy: 0.7131 - val_loss: 0
Epoch 11/20
144/144 [==============================] - 128s 888ms/step - loss: 0.7842 - accuracy: 0.7055 - val_loss: 0
Epoch 12/20
144/144 [==============================] - 128s 891ms/step - loss: 0.7539 - accuracy: 0.7133 - val_loss: 0
Epoch 13/20
144/144 [==============================] - 126s 879ms/step - loss: 0.7496 - accuracy: 0.7218 - val_loss: 0
Epoch 14/20
144/144 [==============================] - 127s 882ms/step - loss: 0.6910 - accuracy: 0.7487 - val_loss: 0
Epoch 15/20
144/144 [==============================] - 126s 873ms/step - loss: 0.7073 - accuracy: 0.7444 - val_loss: 1
Epoch 16/20
144/144 [==============================] - 127s 885ms/step - loss: 0.6959 - accuracy: 0.7444 - val_loss: 0
Epoch 17/20
144/144 [==============================] - 126s 874ms/step - loss: 0.6991 - accuracy: 0.7374 - val_loss: 0
Epoch 18/20
144/144 [==============================] - 127s 883ms/step - loss: 0.6523 - accuracy: 0.7604 - val_loss: 0
Epoch 19/20
144/144 [==============================] - 127s 884ms/step - loss: 0.6881 - accuracy: 0.7489 - val_loss: 0
Epoch 20/20
144/144 [==============================] - 133s 922ms/step - loss: 0.6571 - accuracy: 0.7563 - val_loss: 0
```

```
inceptionv3_model.save("/content/gdrive/MyDrive/hidden hunger/split dataset_orig/inceptionv3_orig.h5")
```

```
fig1 = plt.gcf()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.axis(ymin=0.4,ymax=1)
plt.grid()
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'validation'])
plt.show()
```



```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.grid()
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'validation'])
plt.show()
```



```
from keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model("/content/gdrive/MyDrive/hidden hunger/split dataset_orig/inceptionv3_orig.h5")
```

```
import numpy as np
def predictImage(filename,model):
  img1=image.load_img(filename,target_size=(299,299))
  plt.imshow(img1)
  Y=image.img_to_array(img1)
  X=np.expand_dims(Y,axis=0)
  pred=model.predict(X/255)
  print(pred)
  pred = np.array(pred)
  val = np.argmax(pred)
  print(val)
  if (val==0).all():
    plt.xlabel("Iodine Deficiency",fontsize=50)
  elif (val==1).all():
    plt.xlabel("Iron Deficiency",fontsize=50)
  elif (val==2).all():
    plt.xlabel("Vitamin - B12  Deficiency",fontsize=50)
  elif (val==3).all():
    plt.xlabel("Vitamin D - Deficiency",fontsize=25)
  elif (val==4).all():
    plt.xlabel("Zinc Deficiency",fontsize=50)
```

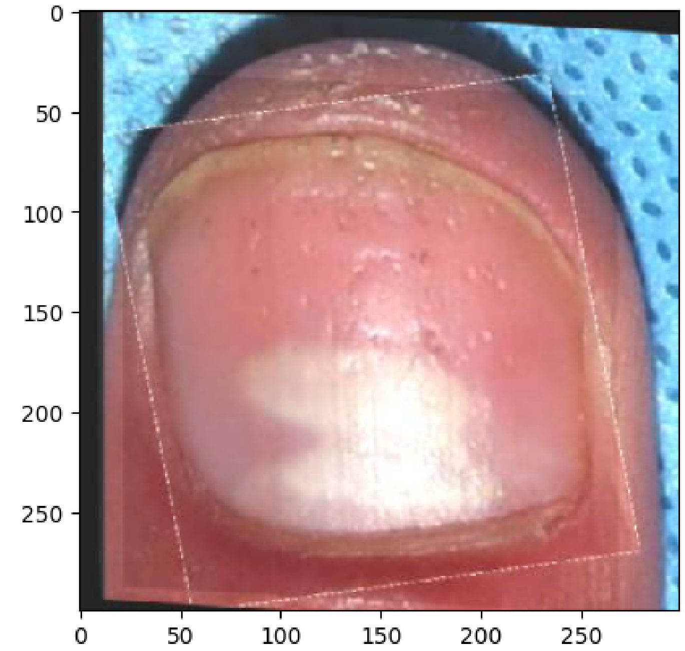
Saving...

```
                                             5)
```

```
predictImage("/content/gdrive/MyDrive/Hidden hunger/val/Iodine Deficiency/Iodine Deficiency_original_Screen-Shot
    1/1 [==============================] - 3s 3s/step
```

```
predictImage("/content/gdrive/MyDrive/Hidden hunger/val/Iron Deficiency/Iron Deficiency_original_112_JPG.rf.4a61
```

```
    1/1 [==============================] - 0s 47ms/step
    [[1.2578721e-01 7.1040863e-01 2.5478872e-02 8.6294105e-03 1.2921669e-01
      4.7909268e-04]]
    1
```

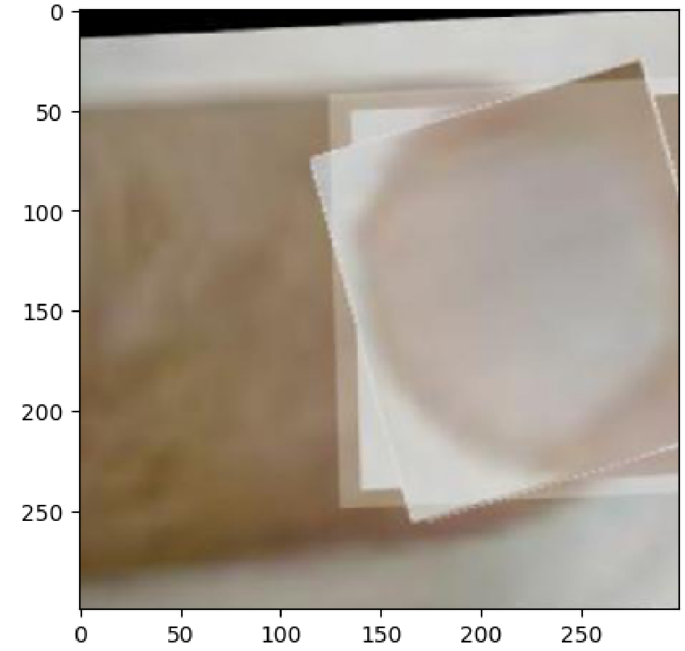


ron Deficienc

Saving...

```
predictImage("/content/gdrive/MyDrive/Hidden hunger/val/Vitamin - B12  Deficiency/Vitamin - B12  Deficiency_orig
```

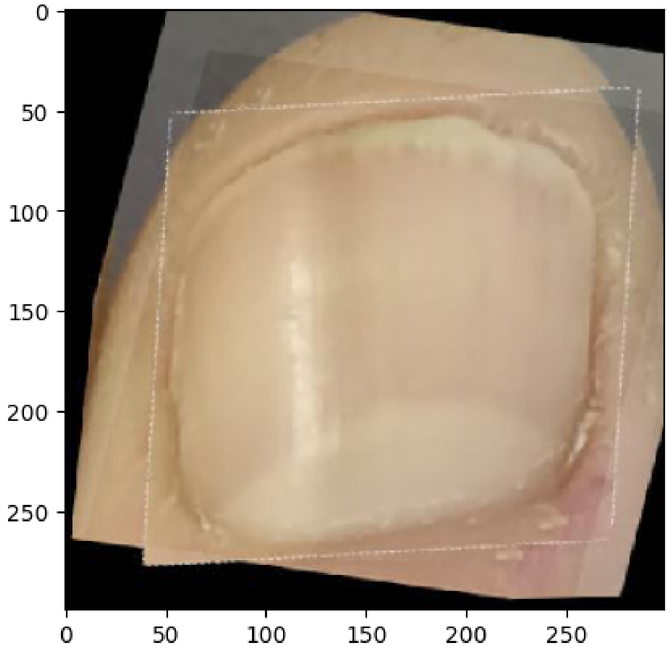
```
    1/1 [==============================] - 0s 45ms/step
    [[3.1044530e-02 3.8101595e-02 9.2412591e-01 1.9836647e-04 8.3671941e-04
      5.6927935e-03]]
    2
```



in - B12 Def

```
predictImage("/content/gdrive/MyDrive/Hidden hunger/val/Vitamin D - Deficiency/Vitamin D - Deficiency_original_S
```

```
1/1 [==============================] - 0s 28ms/step
[[0.00152076 0.29042125 0.03032159 0.5030235  0.1166429  0.05807005]]
3
```

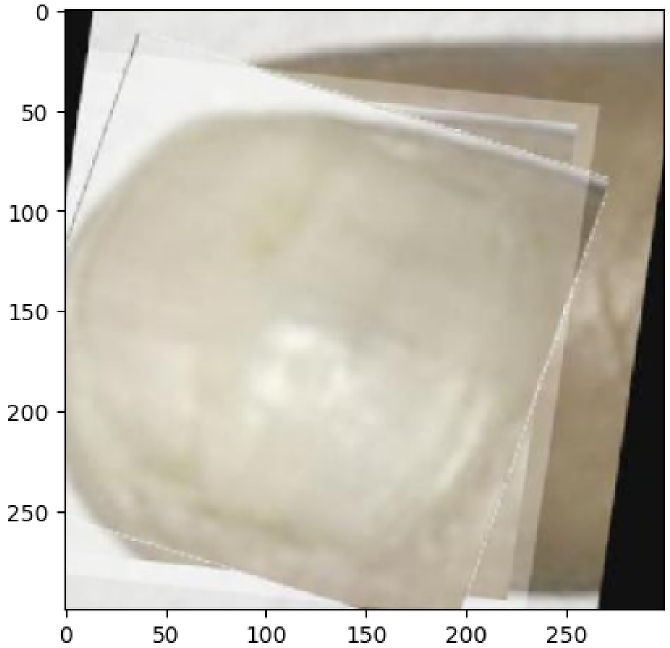


Vitamin D - Deficiency

Saving... ✕

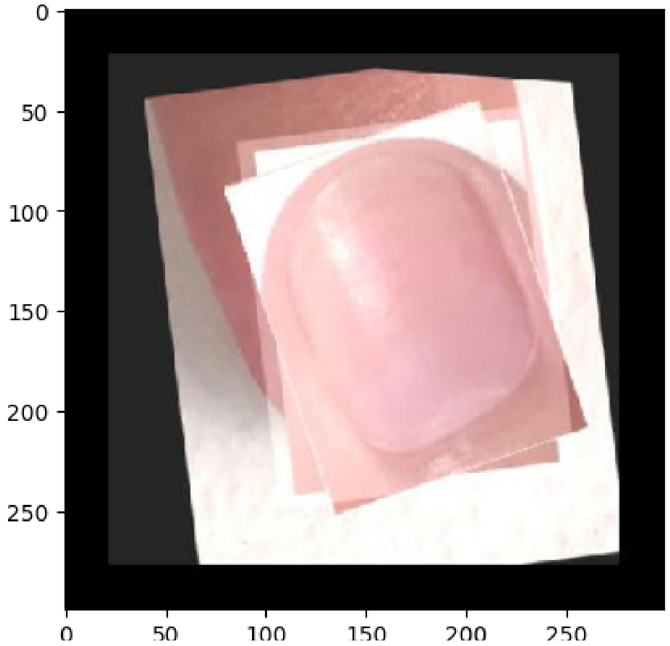
```
ve/Hidden hunger/val/Zinc Deficiency/Zinc Deficiency_original_Screen-Shot-202
```

```
1/1 [==============================] - 0s 27ms/step
[[0.4563299  0.04907864 0.33956146 0.0005035  0.15108983 0.00343669]]
0
```



dine Deficier

```
predictImage("/content/gdrive/MyDrive/Hidden hunger/val/healthy/healthy_original_Screen-Shot-2021-11-15-at-12-52
1/1 [==============================] - 0s 97ms/step
[[0.00148466 0.02746078 0.01187978 0.00133189 0.2960549  0.6617879 ]]
5
```



```
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import numpy as np
```

Saving...

```
gdrive/MyDrive/hidden hunger/split dataset_orig/inceptionv3_orig.h5")
all_y_pred = []
all_y_true = []

for i in range(len(test_set)):
    x, y = test_set[i]
    y_pred = saved_model.predict(x)

    all_y_pred.append(y_pred)
    all_y_true.append(y)

all_y_pred = np.concatenate(all_y_pred, axis=0)
all_y_true = np.concatenate(all_y_true, axis=0)
```

```
1/1 [==============================] - 2s 2s/step
1/1 [==============================] - 0s 66ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 65ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 55ms/step
```

```
    1/1 [==============================] - 0s 52ms/step
    1/1 [==============================] - 0s 60ms/step
    1/1 [==============================] - 0s 54ms/step
    1/1 [==============================] - 0s 55ms/step
    1/1 [==============================] - 0s 43ms/step
    1/1 [==============================] - 0s 72ms/step
    1/1 [==============================] - 0s 75ms/step
    1/1 [==============================] - 0s 72ms/step
    1/1 [==============================] - 0s 71ms/step
    1/1 [==============================] - 0s 76ms/step
    1/1 [==============================] - 0s 81ms/step
    1/1 [==============================] - 2s 2s/step
```

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# compute confusion matrix
cm = confusion_matrix(all_y_true.argmax(axis=1), all_y_pred.argmax(axis=1))

# create heatmap from confusion matrix
fig, ax = plt.subplots(figsize=(6,6))
sns.heatmap(cm, annot=True, cmap="Greens", fmt="d", xticklabels=train_set.class_indices.keys(),
            yticklabels=train_set.class_indices.keys(), ax=ax)

# set axis labels and title
ax.set_xlabel('Predicted labels')
ax.set_ylabel('True labels')
ax.set_title('Confusion Matrix')
```

Saving... ×

```
Text(0.5, 1.0, 'Confusion Matrix')
```

```
from sklearn.metrics import classification_report

# Get the predicted class labels
y_pred = np.argmax(all_y_pred, axis=1)

# Get the true class labels
y_true = np.argmax(all_y_true, axis=1)

# Compute classification report
report = classification_report(y_true, y_pred, target_names=train_set.class_indices.keys())

# Print classification report
print(report)
```

```
                            precision    recall  f1-score   support

           Iodine Deficiency       0.68      0.59      0.63       157
     Vitamin - B12  Deficiency       0.43      0.80      0.56       192
          Vitamin D deficiency       0.66      0.73      0.70       162
              Zinc Deficiency       0.66      0.52      0.58        91
                      healthy       0.76      0.57      0.66       134
              iron deficiency       0.87      0.63      0.73       419

                     accuracy                           0.65      1155
                    macro avg       0.68      0.64      0.64      1155
                 weighted avg       0.71      0.65      0.66      1155
```

Saving...

```
# Load the saved model
model = load_model('/content/gdrive/MyDrive/hidden hunger/split dataset_orig/inceptionv3_orig.h5')

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.01),loss='categorical_crossentropy',metrics=['accuracy'])

# Load the training data
# Assuming X_train and y_train are already defined

# Evaluate the model on the training data
train_loss, train_accuracy = model.evaluate(train_set, verbose=0)

# Print the training accuracy
print("Training accuracy:", train_accuracy)
```

```
Training accuracy: 0.755859375
```

```
# Import necessary libraries
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator

# Load the saved model
model = load_model('/content/gdrive/MyDrive/hidden hunger/split dataset_orig/inceptionv3_orig.h5')
scores = model.evaluate(test_set, steps=len(test_set), verbose=1)
scores2 = model.evaluate(train_set, steps=len(test_set), verbose=1)


# Print the accuracy score
print("Test Accuracy: %.2f%%" % (scores[1]*100))
print("Train Accuracy: %.2f%%" % (scores2[1]*100))
```

```
37/37 [==============================] - 12s 288ms/step - loss: 0.9021 - accuracy: 0.6528
37/37 [==============================] - 30s 818ms/step - loss: 0.6270 - accuracy: 0.7559
```

```
Test Accuracy: 65.28%
Train Accuracy: 75.59%
```

✓ 47s completed at 12:15 PM ● ✕

Saving... ✕