

In [1]:

```
1 import numpy as np # linear algebra
2 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
3
4 # Input data files are available in the read-only "../input/" directory
5 # For example, running this (by clicking run or pressing Shift+Enter) will list all files under
6
7 import os
8 for dirname, _, filenames in os.walk('talh'):
9     for filename in filenames:
10         print(os.path.join(dirname, filename))
11 os.environ['CUDA_VISIBLE_DEVICES'] = '-1'
```

In [2]:

```

1 import tensorflow as tf
2 from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
3 from tensorflow.keras.models import Model
4 from tensorflow.keras.applications.resnet50 import ResNet50
5 from tensorflow.keras.applications.resnet50 import preprocess_input
6 from tensorflow.keras.preprocessing import image
7 from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
8 from tensorflow.keras.models import Sequential
9 import numpy as np
10 from glob import glob
11 import matplotlib.pyplot as plt

```

```

/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
516: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype [("qint8", np.int8, 1)]
/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
517: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint8 = np.dtype [("quint8", np.uint8, 1)]
/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
518: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
519: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint16 = np.dtype [("quint16", np.uint16, 1)]
/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
520: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype [("qint32", np.int32, 1)]
/home/onstak/miniconda3/envs/tf1/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:
525: FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future ve
rsion of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    np_resource = np.dtype [("resource", np.ubyte, 1)]

```

In [3]:

```
1 image_set = "talha/lung_colon_image_set/lung_image_sets"
```

```
In [4]: 1 SIZE_X = SIZE_Y = 224
2
3 datagen = tf.keras.preprocessing.image.ImageDataGenerator(validation_split = 0.2)
4
5 train_set = datagen.flow_from_directory(image_set,
6                                         class_mode = "categorical",
7                                         target_size = (SIZE_X, SIZE_Y),
8                                         color_mode="rgb",
9                                         batch_size = 128,
10                                        shuffle = False,
11                                        subset='training',
12                                        seed = 42)
13
14 validate_set = datagen.flow_from_directory(image_set,
15                                             class_mode = "categorical",
16                                             target_size = (SIZE_X, SIZE_Y),
17                                             color_mode="rgb",
18                                             batch_size = 128,
19                                             shuffle = False,
20                                             subset='validation',
21                                             seed = 42)
```

Found 20000 images belonging to 5 classes.
Found 5000 images belonging to 5 classes.

```
In [5]: 1 IMAGE_SIZE = [224, 224]
```

```
In [6]: 1 resnet = ResNet50(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

/home/onstak/.local/lib/python3.6/site-packages/keras_applications/resnet50.py:265: UserWarning: The output shape of `ResNet50(include_top=False)` has been changed since Keras 2.2.0.
warnings.warn('The output shape of `ResNet50(include_top=False)`')

```
In [7]: 1 for layer in resnet.layers:
2       layer.trainable = False
```

```
In [8]: 1 flatten = Flatten()(resnet.output)
2 dense = Dense(256, activation = 'relu')(flatten)
3 dense = Dense(128, activation = 'relu')(dense)
4 prediction = Dense(5, activation = 'softmax')(dense)
5 #creating a model
6 model = Model(inputs = resnet.input, outputs = prediction )
7 model.summary()
```

WARNING:tensorflow:From /home/onstak/miniconda3/envs/tfl/lib/python3.6/site-packages/tensorflow/python/ops/init_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.

Instructions for updating:

Call initializer instance with the dtype argument instead of passing it to the constructor
Model: "model"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	
=====			
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	input_1[0][0]
=====			
conv1 (Conv2D)	(None, 112, 112, 64)	9472	conv1_pad[0][0]
=====			

```
In [10]: 1 model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

In [11]:

```
1 #executing the model
2 history = model.fit_generator(train_set, validation_data = (validate_set), epochs = 5, verbose =
```

Epoch 1/5

157/157 [=====] - 579s 4s/step - loss: 11.8224 - acc: 0.3395 - val_loss: 7.9277 - val_acc: 0.2000

Epoch 2/5

157/157 [=====] - 575s 4s/step - loss: 1.4030 - acc: 0.6400 - val_loss: 1.9102 - val_acc: 0.3242

Epoch 3/5

157/157 [=====] - 578s 4s/step - loss: 0.5175 - acc: 0.8181 - val_loss: 2.1538 - val_acc: 0.3432

Epoch 4/5

157/157 [=====] - 579s 4s/step - loss: 0.6293 - acc: 0.8063 - val_loss: 4.0252 - val_acc: 0.2552

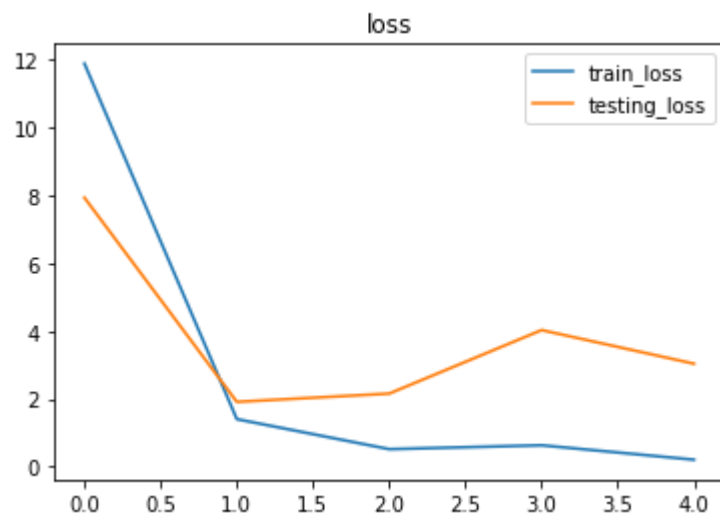
Epoch 5/5

157/157 [=====] - 578s 4s/step - loss: 0.2055 - acc: 0.9212 - val_loss: 3.0351 - val_acc: 0.2700

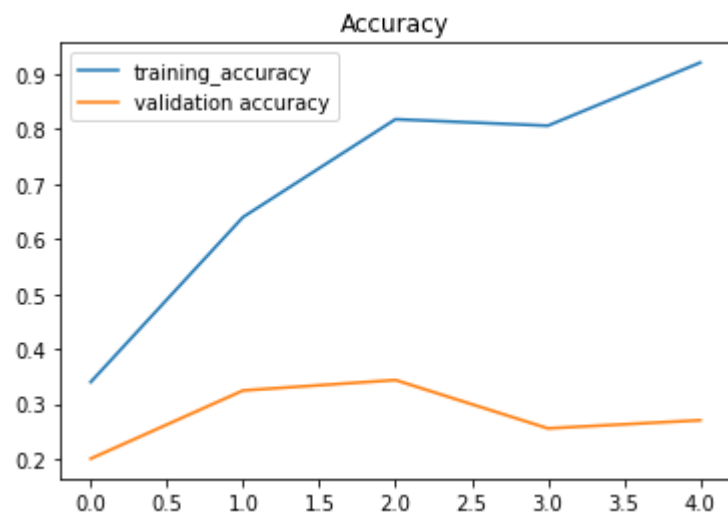
In [12]:

```
1 model.save_weights('it1_pretrained_model.h5')
```

```
In [13]: 1 # plotting the loss
2 plt.plot(history.history['loss'],label = 'train_loss')
3 plt.plot(history.history['val_loss'], label = 'testing_loss')
4 plt.title('loss')
5 plt.legend()
6 plt.show()
```



```
In [23]: 1 plt.plot(history.history['acc'], label='training_accuracy')
2         plt.plot(history.history['val_acc'], label='validation accuracy')
3         plt.title('Accuracy')
4         plt.legend()
5         plt.show()
```



```
In [21]: 1 from sklearn.metrics import classification_report
2 from sklearn.metrics import confusion_matrix
3 from sklearn.metrics import f1_score
4 Y_pred = model.predict_generator(validate_set)
5 y_pred = np.argmax(Y_pred ,axis =1)
6 print('Confusion Matrix')
7 confusion_matrix = confusion_matrix(validate_set.classes, y_pred)
8 print(confusion_matrix)
9 print('Classification Report')
10 print(classification_report(validate_set.classes, y_pred))
```

Confusion Matrix

```
[[769  3  0  0 228]
 [681 240  0  0  79]
 [798 10  2  5 185]
 [882 106  0 12  0]
 [670  2  1  0 327]]
```

Classification Report

	precision	recall	f1-score	support
0	0.20	0.77	0.32	1000
1	0.66	0.24	0.35	1000
2	0.67	0.00	0.00	1000
3	0.71	0.01	0.02	1000
4	0.40	0.33	0.36	1000
accuracy			0.27	5000
macro avg	0.53	0.27	0.21	5000
weighted avg	0.53	0.27	0.21	5000

```
In [22]: 1 result = model.evaluate(validate_set,batch_size=128)
2 print("test_loss, test accuracy",result)
```

40/40 [=====] - 86s 2s/step - loss: 3.0351 - acc: 0.2700
test_loss, test accuracy [3.035135233402252, 0.27]

In []:

1

