In [ ]:

```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/gdrive
```

In [ ]:

```
cd /content/gdrive/My\ Drive/ttv
```

```
/content/gdrive/My Drive/ttv
```

In [ ]:

```
train_path = 'train'
valid_path = 'val'
test_path  = 'test'
```

In [ ]:

```
import os

num_train_samples = len(os.listdir(train_path+'/'+"measles")) + len(os.listdir(train_path+'/'+"melanoma")) + len(os.listdir(train_path+'/'+"Psoriasis")) + len(os.listdir(train_path+'/'+"ringworm"))
num_val_samples = len(os.listdir(test_path+'/'+"measles")) + len(os.listdir(test_path+'/'+"melanoma")) + len(os.listdir(test_path+'/'+"Psoriasis")) + len(os.listdir(test_path+'/'+"ringworm"))

print(num_train_samples,num_val_samples)
```

```
1012 128
```

In [ ]:

```
import numpy as np

# train_batch_size = int((np.ceil(num_train_samples / train_batch_size))*30)
# val_batch_size = int((np.ceil(num_val_samples / val_batch_size))*30)
image_size = 224

# train_steps = np.ceil(num_train_samples / train_batch_size)
# val_steps = np.ceil(num_val_samples / val_batch_size)
```

In [ ]:

```
import pandas as pd
import numpy as np


import tensorflow
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
```

```
import os

from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
import itertools
import shutil
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:

```
datagen = ImageDataGenerator(
    rotation_range          = 180,
    width_shift_range       = 0.1,
    height_shift_range      = 0.1,
    zoom_range              = 0.1,
    horizontal_flip         = True,
    vertical_flip           = True,
    fill_mode               = 'nearest',
    preprocessing_function  = tensorflow.keras.applications.mobilenet.preprocess_input
    )
```

In [ ]:

```
train_batches = datagen.flow_from_directory(train_path,
                                            target_size=(image_size,image_size),
                                            batch_size=80)

valid_batches = datagen.flow_from_directory(valid_path,
                                            target_size=(image_size,image_size),
                                            batch_size=10)

test_batches = datagen.flow_from_directory(test_path,
                                            target_size=(image_size,image_size),
                                            batch_size=10,
                                            shuffle=False)
```

```
Found 918 images belonging to 4 classes.
Found 111 images belonging to 4 classes.
Found 111 images belonging to 4 classes.
```

In [ ]:

```
mobile = tensorflow.keras.applications.mobilenet.MobileNet()
```

In [ ]:

```
x = mobile.layers[-6].output

x = Dropout(0.25)(x)
predictions = Dense(4, activation='softmax')(x)

model = Model(inputs=mobile.input, outputs=predictions)
```

In [ ]:

```
for layer in model.layers[:-23]:
    layer.trainable = False
```

In [ ]:

```
from tensorflow.keras.metrics import categorical_accuracy, top_k_categorical_accuracy

def top_3_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=3)

def top_2_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=2)
```

In [ ]:

```
model.compile(Adam(lr=0.01), loss='categorical_crossentropy', metrics=[categorical_accuracy, top_2_accuracy, top_3_accuracy])
```

In [ ]:

```
valid_batches.class_indices
```

Out[ ]:

```
{'Psoriasis': 0, 'measles': 1, 'melanoma': 2, 'ringworm': 3}
```

In [ ]:

```
train_batches.class_indices
```

Out[ ]:

```
{'Psoriasis': 0, 'measles': 1, 'melanoma': 2, 'ringworm': 3}
```

In [ ]:

```
class_weights={
    0: 3.0,
    1: 1.5,
    2: 2.0,
    3: 2.4,
}
```

In [ ]:

```
filepath = "model.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_top_3_accuracy', verbose=1,
                             save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='val_top_3_accuracy', factor=0.5, patience=2,
                                    verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]

history = model.fit_generator(train_batches,
                              # steps_per_epoch=int(918/45),
                              class_weight=class_weights,
                    validation_data=valid_batches,
                    # validation_steps=int(111/45),
                    epochs=30, verbose=1,
                   callbacks=callbacks_list)
```

```
Epoch 1/30
/usr/local/lib/python3.6/dist-packages/PIL/TiffImagePlugin.py:788: UserWarning: Corrupt EXIF data.  Expecting to read 4 bytes but only got 0.
  warnings.warn(str(msg))
12/12 [==============================] - ETA: 0s - loss: 1.4276 - categorical_accuracy: 0.7484 - top_2_accuracy: 0.9216 - top_3_accuracy: 0.9804
Epoch 00001: val_top_3_accuracy improved from -inf to 0.91892, saving model to model.h5
12/12 [==============================] - 27s 2s/step - loss: 1.4276 - categorical_accuracy: 0.7484 - top_2_accuracy: 0.9216 - top_3_accuracy: 0.9804 - val_loss: 1.6695 - val_categorical_accuracy: 0.6486 - val_top_2_accuracy: 0.8378 - val_top_3_accuracy: 0.9189 - lr: 0.0100
Epoch 2/30
12/12 [==============================] - ETA: 0s - loss: 1.2314 - categorical_accuracy: 0.7843 - top_2_accuracy: 0.9423 - top_3_accuracy: 0.9858
Epoch 00002: val_top_3_accuracy did not improve from 0.91892
12/12 [==============================] - 27s 2s/step - loss: 1.2314 - categorical_accuracy: 0.7843 - top_2_accuracy: 0.9423 - top_3_accuracy: 0.9858 - val_loss: 1.4726 - val_categorical_accuracy: 0.7117 - val_top_2_accuracy: 0.8378 - val_top_3_accuracy: 0.9189 - lr: 0.0100
```

```
Epoch 3/30
12/12 [==============================] - ETA: 0s - loss: 1.1780 - categorical_accuracy: 0
.7898 - top_2_accuracy: 0.9444 - top_3_accuracy: 0.9869
Epoch 00003: val_top_3_accuracy did not improve from 0.91892

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.004999999888241291.
12/12 [==============================] - 27s 2s/step - loss: 1.1780 - categorical_accurac
y: 0.7898 - top_2_accuracy: 0.9444 - top_3_accuracy: 0.9869 - val_loss: 1.1681 - val_cate
gorical_accuracy: 0.6937 - val_top_2_accuracy: 0.8378 - val_top_3_accuracy: 0.9009 - lr:
0.0100
Epoch 4/30
12/12 [==============================] - ETA: 0s - loss: 1.0816 - categorical_accuracy: 0
.8170 - top_2_accuracy: 0.9477 - top_3_accuracy: 0.9902
Epoch 00004: val_top_3_accuracy did not improve from 0.91892
12/12 [==============================] - 27s 2s/step - loss: 1.0816 - categorical_accurac
y: 0.8170 - top_2_accuracy: 0.9477 - top_3_accuracy: 0.9902 - val_loss: 1.0890 - val_cate
gorical_accuracy: 0.6757 - val_top_2_accuracy: 0.8018 - val_top_3_accuracy: 0.8739 - lr:
0.0050
Epoch 5/30
12/12 [==============================] - ETA: 0s - loss: 1.0424 - categorical_accuracy: 0
.8214 - top_2_accuracy: 0.9466 - top_3_accuracy: 0.9880
Epoch 00005: val_top_3_accuracy did not improve from 0.91892

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.0024999999441206455.
12/12 [==============================] - 27s 2s/step - loss: 1.0424 - categorical_accurac
y: 0.8214 - top_2_accuracy: 0.9466 - top_3_accuracy: 0.9880 - val_loss: 1.0454 - val_cate
gorical_accuracy: 0.6847 - val_top_2_accuracy: 0.8198 - val_top_3_accuracy: 0.8829 - lr:
0.0050
Epoch 6/30
12/12 [==============================] - ETA: 0s - loss: 1.0122 - categorical_accuracy: 0
.8333 - top_2_accuracy: 0.9510 - top_3_accuracy: 0.9902
Epoch 00006: val_top_3_accuracy did not improve from 0.91892
12/12 [==============================] - 28s 2s/step - loss: 1.0122 - categorical_accurac
y: 0.8333 - top_2_accuracy: 0.9510 - top_3_accuracy: 0.9902 - val_loss: 1.0204 - val_cate
gorical_accuracy: 0.6757 - val_top_2_accuracy: 0.7838 - val_top_3_accuracy: 0.8829 - lr:
0.0025
Epoch 7/30
12/12 [==============================] - ETA: 0s - loss: 1.0116 - categorical_accuracy: 0
.8235 - top_2_accuracy: 0.9532 - top_3_accuracy: 0.9902
Epoch 00007: val_top_3_accuracy did not improve from 0.91892

Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.0012499999720603228.
12/12 [==============================] - 27s 2s/step - loss: 1.0116 - categorical_accurac
y: 0.8235 - top_2_accuracy: 0.9532 - top_3_accuracy: 0.9902 - val_loss: 0.9415 - val_cate
gorical_accuracy: 0.6937 - val_top_2_accuracy: 0.8468 - val_top_3_accuracy: 0.8919 - lr:
0.0025
Epoch 8/30
12/12 [==============================] - ETA: 0s - loss: 0.9882 - categorical_accuracy: 0
.8388 - top_2_accuracy: 0.9575 - top_3_accuracy: 0.9902
Epoch 00008: val_top_3_accuracy did not improve from 0.91892
12/12 [==============================] - 27s 2s/step - loss: 0.9882 - categorical_accurac
y: 0.8388 - top_2_accuracy: 0.9575 - top_3_accuracy: 0.9902 - val_loss: 0.8350 - val_cate
gorical_accuracy: 0.7027 - val_top_2_accuracy: 0.8468 - val_top_3_accuracy: 0.8739 - lr:
0.0012
Epoch 9/30
12/12 [==============================] - ETA: 0s - loss: 0.9321 - categorical_accuracy: 0
.8279 - top_2_accuracy: 0.9510 - top_3_accuracy: 0.9891
Epoch 00009: val_top_3_accuracy did not improve from 0.91892

Epoch 00009: ReduceLROnPlateau reducing learning rate to 0.0006249999930150807.
12/12 [==============================] - 28s 2s/step - loss: 0.9321 - categorical_accurac
y: 0.8279 - top_2_accuracy: 0.9510 - top_3_accuracy: 0.9891 - val_loss: 0.7792 - val_cate
gorical_accuracy: 0.7297 - val_top_2_accuracy: 0.8559 - val_top_3_accuracy: 0.9189 - lr:
0.0012
Epoch 10/30
12/12 [==============================] - ETA: 0s - loss: 0.9770 - categorical_accuracy: 0
.8268 - top_2_accuracy: 0.9542 - top_3_accuracy: 0.9924
Epoch 00010: val_top_3_accuracy did not improve from 0.91892
12/12 [==============================] - 27s 2s/step - loss: 0.9770 - categorical_accurac
y: 0.8268 - top_2_accuracy: 0.9542 - top_3_accuracy: 0.9924 - val_loss: 0.7428 - val_cate
```

```
gorical_accuracy: 0.7658 - val_top_2_accuracy: 0.8649 - val_top_3_accuracy: 0.9099 - lr:
6.2500e-04
Epoch 11/30
12/12 [==============================] - ETA: 0s - loss: 0.8858 - categorical_accuracy: 0
.8486 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924
Epoch 00011: val_top_3_accuracy improved from 0.91892 to 0.92793, saving model to model.h
5
12/12 [==============================] - 28s 2s/step - loss: 0.8858 - categorical_accurac
y: 0.8486 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924 - val_loss: 0.6962 - val_cate
gorical_accuracy: 0.7477 - val_top_2_accuracy: 0.8649 - val_top_3_accuracy: 0.9279 - lr:
6.2500e-04
Epoch 12/30
12/12 [==============================] - ETA: 0s - loss: 0.9350 - categorical_accuracy: 0
.8508 - top_2_accuracy: 0.9597 - top_3_accuracy: 0.9902
Epoch 00012: val_top_3_accuracy improved from 0.92793 to 0.95495, saving model to model.h
5
12/12 [==============================] - 28s 2s/step - loss: 0.9350 - categorical_accurac
y: 0.8508 - top_2_accuracy: 0.9597 - top_3_accuracy: 0.9902 - val_loss: 0.7240 - val_cate
gorical_accuracy: 0.7748 - val_top_2_accuracy: 0.8739 - val_top_3_accuracy: 0.9550 - lr:
6.2500e-04
Epoch 13/30
12/12 [==============================] - ETA: 0s - loss: 0.9153 - categorical_accuracy: 0
.8475 - top_2_accuracy: 0.9619 - top_3_accuracy: 0.9935
Epoch 00013: val_top_3_accuracy did not improve from 0.95495
12/12 [==============================] - 28s 2s/step - loss: 0.9153 - categorical_accurac
y: 0.8475 - top_2_accuracy: 0.9619 - top_3_accuracy: 0.9935 - val_loss: 0.6839 - val_cate
gorical_accuracy: 0.7568 - val_top_2_accuracy: 0.8739 - val_top_3_accuracy: 0.9369 - lr:
6.2500e-04
Epoch 14/30
12/12 [==============================] - ETA: 0s - loss: 0.8912 - categorical_accuracy: 0
.8464 - top_2_accuracy: 0.9662 - top_3_accuracy: 0.9946
Epoch 00014: val_top_3_accuracy did not improve from 0.95495

Epoch 00014: ReduceLROnPlateau reducing learning rate to 0.0003124999930150807.
12/12 [==============================] - 27s 2s/step - loss: 0.8912 - categorical_accurac
y: 0.8464 - top_2_accuracy: 0.9662 - top_3_accuracy: 0.9946 - val_loss: 0.6365 - val_cate
gorical_accuracy: 0.7838 - val_top_2_accuracy: 0.9009 - val_top_3_accuracy: 0.9459 - lr:
6.2500e-04
Epoch 15/30
12/12 [==============================] - ETA: 0s - loss: 0.9267 - categorical_accuracy: 0
.8453 - top_2_accuracy: 0.9673 - top_3_accuracy: 0.9902
Epoch 00015: val_top_3_accuracy improved from 0.95495 to 0.96396, saving model to model.h
5
12/12 [==============================] - 28s 2s/step - loss: 0.9267 - categorical_accurac
y: 0.8453 - top_2_accuracy: 0.9673 - top_3_accuracy: 0.9902 - val_loss: 0.6075 - val_cate
gorical_accuracy: 0.7838 - val_top_2_accuracy: 0.8919 - val_top_3_accuracy: 0.9640 - lr:
3.1250e-04
Epoch 16/30
12/12 [==============================] - ETA: 0s - loss: 0.8493 - categorical_accuracy: 0
.8584 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9935
Epoch 00016: val_top_3_accuracy did not improve from 0.96396
12/12 [==============================] - 27s 2s/step - loss: 0.8493 - categorical_accurac
y: 0.8584 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9935 - val_loss: 0.6148 - val_cate
gorical_accuracy: 0.7568 - val_top_2_accuracy: 0.8829 - val_top_3_accuracy: 0.9550 - lr:
3.1250e-04
Epoch 17/30
12/12 [==============================] - ETA: 0s - loss: 0.9086 - categorical_accuracy: 0
.8497 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9946
Epoch 00017: val_top_3_accuracy did not improve from 0.96396

Epoch 00017: ReduceLROnPlateau reducing learning rate to 0.00015624999650754035.
12/12 [==============================] - 27s 2s/step - loss: 0.9086 - categorical_accurac
y: 0.8497 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9946 - val_loss: 0.6298 - val_cate
gorical_accuracy: 0.7838 - val_top_2_accuracy: 0.8829 - val_top_3_accuracy: 0.9459 - lr:
3.1250e-04
Epoch 18/30
12/12 [==============================] - ETA: 0s - loss: 0.8863 - categorical_accuracy: 0
.8497 - top_2_accuracy: 0.9662 - top_3_accuracy: 0.9946
Epoch 00018: val_top_3_accuracy improved from 0.96396 to 0.97297, saving model to model.h
5
12/12 [==============================] - 28s 2s/step - loss: 0.8863 - categorical_accurac
```

```
y: 0.8497 - top_2_accuracy: 0.9662 - top_3_accuracy: 0.9946 - val_loss: 0.5543 - val_cate
gorical_accuracy: 0.7928 - val_top_2_accuracy: 0.9099 - val_top_3_accuracy: 0.9730 - lr:
1.5625e-04
Epoch 19/30
12/12 [==============================] - ETA: 0s - loss: 0.9053 - categorical_accuracy: 0
.8497 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9913
Epoch 00019: val_top_3_accuracy did not improve from 0.97297
12/12 [==============================] - 27s 2s/step - loss: 0.9053 - categorical_accurac
y: 0.8497 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9913 - val_loss: 0.5774 - val_cate
gorical_accuracy: 0.8018 - val_top_2_accuracy: 0.9099 - val_top_3_accuracy: 0.9640 - lr:
1.5625e-04
Epoch 20/30
12/12 [==============================] - ETA: 0s - loss: 0.8348 - categorical_accuracy: 0
.8671 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9924
Epoch 00020: val_top_3_accuracy did not improve from 0.97297

Epoch 00020: ReduceLROnPlateau reducing learning rate to 7.812499825377017e-05.
12/12 [==============================] - 27s 2s/step - loss: 0.8348 - categorical_accurac
y: 0.8671 - top_2_accuracy: 0.9608 - top_3_accuracy: 0.9924 - val_loss: 0.5945 - val_cate
gorical_accuracy: 0.7928 - val_top_2_accuracy: 0.8739 - val_top_3_accuracy: 0.9640 - lr:
1.5625e-04
Epoch 21/30
12/12 [==============================] - ETA: 0s - loss: 0.8570 - categorical_accuracy: 0
.8627 - top_2_accuracy: 0.9651 - top_3_accuracy: 0.9924
Epoch 00021: val_top_3_accuracy did not improve from 0.97297
12/12 [==============================] - 27s 2s/step - loss: 0.8570 - categorical_accurac
y: 0.8627 - top_2_accuracy: 0.9651 - top_3_accuracy: 0.9924 - val_loss: 0.5294 - val_cate
gorical_accuracy: 0.7928 - val_top_2_accuracy: 0.9099 - val_top_3_accuracy: 0.9730 - lr:
7.8125e-05
Epoch 22/30
12/12 [==============================] - ETA: 0s - loss: 0.8542 - categorical_accuracy: 0
.8638 - top_2_accuracy: 0.9597 - top_3_accuracy: 0.9924
Epoch 00022: val_top_3_accuracy did not improve from 0.97297

Epoch 00022: ReduceLROnPlateau reducing learning rate to 3.9062499126885086e-05.
12/12 [==============================] - 27s 2s/step - loss: 0.8542 - categorical_accurac
y: 0.8638 - top_2_accuracy: 0.9597 - top_3_accuracy: 0.9924 - val_loss: 0.5414 - val_cate
gorical_accuracy: 0.7838 - val_top_2_accuracy: 0.9189 - val_top_3_accuracy: 0.9730 - lr:
7.8125e-05
Epoch 23/30
12/12 [==============================] - ETA: 0s - loss: 0.9530 - categorical_accuracy: 0
.8388 - top_2_accuracy: 0.9651 - top_3_accuracy: 0.9935
Epoch 00023: val_top_3_accuracy improved from 0.97297 to 0.98198, saving model to model.h
5
12/12 [==============================] - 28s 2s/step - loss: 0.9530 - categorical_accurac
y: 0.8388 - top_2_accuracy: 0.9651 - top_3_accuracy: 0.9935 - val_loss: 0.5084 - val_cate
gorical_accuracy: 0.7928 - val_top_2_accuracy: 0.9369 - val_top_3_accuracy: 0.9820 - lr:
3.9062e-05
Epoch 24/30
12/12 [==============================] - ETA: 0s - loss: 0.8377 - categorical_accuracy: 0
.8584 - top_2_accuracy: 0.9619 - top_3_accuracy: 0.9902
Epoch 00024: val_top_3_accuracy did not improve from 0.98198
12/12 [==============================] - 27s 2s/step - loss: 0.8377 - categorical_accurac
y: 0.8584 - top_2_accuracy: 0.9619 - top_3_accuracy: 0.9902 - val_loss: 0.5631 - val_cate
gorical_accuracy: 0.8108 - val_top_2_accuracy: 0.9099 - val_top_3_accuracy: 0.9820 - lr:
3.9062e-05
Epoch 25/30
12/12 [==============================] - ETA: 0s - loss: 0.8697 - categorical_accuracy: 0
.8442 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924
Epoch 00025: val_top_3_accuracy did not improve from 0.98198

Epoch 00025: ReduceLROnPlateau reducing learning rate to 1.9531249563442543e-05.
12/12 [==============================] - 28s 2s/step - loss: 0.8697 - categorical_accurac
y: 0.8442 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924 - val_loss: 0.4724 - val_cate
gorical_accuracy: 0.8468 - val_top_2_accuracy: 0.9279 - val_top_3_accuracy: 0.9820 - lr:
3.9062e-05
Epoch 26/30
12/12 [==============================] - ETA: 0s - loss: 0.9214 - categorical_accuracy: 0
.8377 - top_2_accuracy: 0.9553 - top_3_accuracy: 0.9935
Epoch 00026: val_top_3_accuracy did not improve from 0.98198
```

```
12/12 [==============================] - 27s 2s/step - loss: 0.9214 - categorical_accurac
y: 0.8377 - top_2_accuracy: 0.9553 - top_3_accuracy: 0.9935 - val_loss: 0.5487 - val_cate
gorical_accuracy: 0.7928 - val_top_2_accuracy: 0.9009 - val_top_3_accuracy: 0.9820 - lr:
1.9531e-05
Epoch 27/30
12/12 [==============================] - ETA: 0s - loss: 0.8536 - categorical_accuracy: 0
.8715 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9902
Epoch 00027: val_top_3_accuracy did not improve from 0.98198

Epoch 00027: ReduceLROnPlateau reducing learning rate to 1e-05.
12/12 [==============================] - 27s 2s/step - loss: 0.8536 - categorical_accurac
y: 0.8715 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9902 - val_loss: 0.5364 - val_cate
gorical_accuracy: 0.8198 - val_top_2_accuracy: 0.9099 - val_top_3_accuracy: 0.9820 - lr:
1.9531e-05
Epoch 28/30
12/12 [==============================] - ETA: 0s - loss: 0.8833 - categorical_accuracy: 0
.8769 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924
Epoch 00028: val_top_3_accuracy did not improve from 0.98198
12/12 [==============================] - 27s 2s/step - loss: 0.8833 - categorical_accurac
y: 0.8769 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9924 - val_loss: 0.5609 - val_cate
gorical_accuracy: 0.8108 - val_top_2_accuracy: 0.9279 - val_top_3_accuracy: 0.9730 - lr:
1.0000e-05
Epoch 29/30
12/12 [==============================] - ETA: 0s - loss: 0.8822 - categorical_accuracy: 0
.8540 - top_2_accuracy: 0.9553 - top_3_accuracy: 0.9956
Epoch 00029: val_top_3_accuracy did not improve from 0.98198
12/12 [==============================] - 26s 2s/step - loss: 0.8822 - categorical_accurac
y: 0.8540 - top_2_accuracy: 0.9553 - top_3_accuracy: 0.9956 - val_loss: 0.5112 - val_cate
gorical_accuracy: 0.8198 - val_top_2_accuracy: 0.9189 - val_top_3_accuracy: 0.9820 - lr:
1.0000e-05
Epoch 30/30
12/12 [==============================] - ETA: 0s - loss: 0.8708 - categorical_accuracy: 0
.8671 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9956
Epoch 00030: val_top_3_accuracy did not improve from 0.98198
12/12 [==============================] - 26s 2s/step - loss: 0.8708 - categorical_accurac
y: 0.8671 - top_2_accuracy: 0.9630 - top_3_accuracy: 0.9956 - val_loss: 0.5005 - val_cate
gorical_accuracy: 0.8018 - val_top_2_accuracy: 0.9189 - val_top_3_accuracy: 0.9820 - lr:
1.0000e-05
```

In [ ]:

```
val_loss, val_cat_acc, val_top_2_acc, val_top_3_acc = model.evaluate_generator(test_batch
es)
```

In [ ]:

```
print('val_loss:', val_loss)
print('val_cat_acc:', val_cat_acc)
print('val_top_2_acc:', val_top_2_acc)
print('val_top_3_acc:', val_top_3_acc)
```

```
val_loss: 0.4090295732021332
val_cat_acc: 0.8648648858070374
val_top_2_acc: 0.9639639854431152
val_top_3_acc: 0.9819819927215576
```

In [ ]:

```
model.load_weights('model.h5')

val_loss, val_cat_acc, val_top_2_acc, val_top_3_acc = \
model.evaluate_generator(test_batches,
                         )

print('val_loss:', val_loss)
print('val_cat_acc:', val_cat_acc)
print('val_top_2_acc:', val_top_2_acc)
print('val_top_3_acc:', val_top_3_acc)
```

```
val_loss: 0.48321300745010376
val_cat_acc: 0.792792797088623
```

```
val_top_2_acc: 0.9459459185600281
val_top_3_acc: 0.9819819927215576
```

In [ ]:

```
predictions = model.predict_generator(test_batches, verbose=1)
```

```
WARNING:tensorflow:From <ipython-input-73-f2d0f45ba84e>:1: Model.predict_generator (from
tensorflow.python.keras.engine.training) is deprecated and will be removed in a future ve
rsion.
Instructions for updating:
Please use Model.predict, which supports generators.
12/12 [==============================] - 3s 279ms/step
```

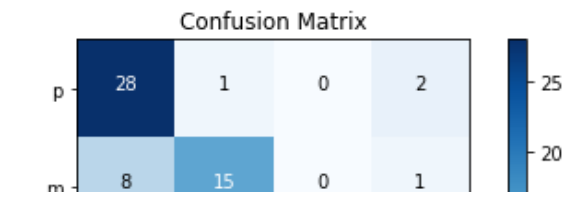
In [ ]:

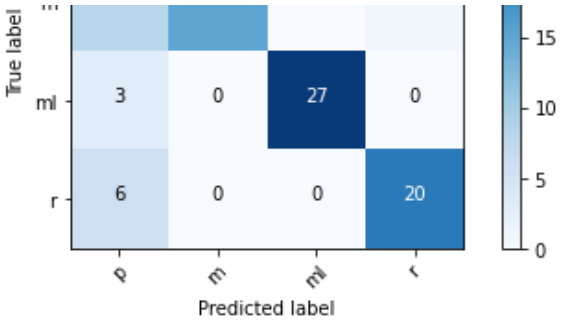
```
test_labels = test_batches.classes
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

In [ ]:

```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.tight_layout()


plot_confusion_matrix(cm,["p","m","ml","r"], title='Confusion Matrix')
```

```
Confusion matrix, without normalization
[[28  1  0  2]
 [ 8 15  0  1]
 [ 3  0 27  0]
 [ 6  0  0 20]]
```





In [ ]:

```
from keras.preprocessing import image

def loadImages(path):
  img = image.load_img(path,target_size=(224, 224))
  img_data = image.img_to_array(img)
  img_data = np.expand_dims(img_data, axis=0)
  img_data = tensorflow.keras.applications.mobilenet.preprocess_input(img_data)
  features = np.array(model.predict(img_data))
  return features
```

```
Using TensorFlow backend.
```

In [ ]:

```
ls
```

```
model.h5  test/  train/  val/
```

In [ ]:

```
loadImages("index.jpeg")
```

Out[ ]:

```
array([[7.0998053e-06, 8.3098044e-08, 9.9998176e-01, 1.1061923e-05]],
      dtype=float32)
```

In [ ]:

```
loadImages("measles.jpg")
```

Out[ ]:

```
array([[2.2336345e-02, 9.7485709e-01, 4.3617096e-04, 2.3704956e-03]],
      dtype=float32)
```

In [ ]:

```

```