Information on Dataset used:

This model is trained on HAM 10000 (Human Againest Machine with 10000 training images) data set, it can be found in the following link: https://arxiv.org/abs/1803.10417 (https://arxiv.org/abs/1803.10417)

It is a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions.

Labels are as following:

1- nv Melanocytic nevi are benign neoplasms of melanocytes and appear in a myriad of variants, which all are included in our series. The variants may differ significantly from a dermatoscopic point of view. [6705 images]

2- mel Melanoma is a malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage it can be cured by simple surgical excision. Melanomas can be invasive or non-invasive (in situ). We included all variants of melanoma including melanoma in situ, but did exclude non-pigmented, subungual, ocular or mucosal melanoma. [1113 images]

3- bkl "Benign keratosis" is a generic class that includes seborrheic ker- atoses ("senile wart"), solar lentigo - which can be regarded a flat variant of seborrheic keratosis - and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression. The three subgroups may look different dermatoscop- ically, but we grouped them together because they are similar biologically and often reported under the same generic term histopathologically. From a dermatoscopic view, lichen planus-like keratoses are especially challeng- ing because they can show morphologic features mimicking melanoma and are often biopsied or excised for diagnostic reasons. [1099 images]

4- bcc Basal cell carcinoma is a common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated. It appears in different morphologic variants (flat, nodular, pigmented, cystic, etc) [21], which are all included in this set. [514 images]

5- akiec Actinic Keratoses (Solar Keratoses) and intraepithelial Carcinoma (Bowen's disease) are common non-invasive, variants of squamous cell car- cinoma that can be treated locally without surgery. Some authors regard them as precursors of squamous cell carcinomas and not as actual carci- nomas. There is, however, agreement that these lesions may progress to invasive squamous cell carcinoma - which is usually not pigmented. Both neoplasms commonly show surface scaling and commonly are devoid of pigment. Actinic keratoses are more common on the face and Bowen's disease is more common on other body sites. Because both types are in- duced by UV-light the surrounding skin is usually typified by severe sun damaged except in cases of Bowen's disease that are caused by human papilloma virus infection and not by UV. Pigmented variants exists for Bowen's disease and for actinic keratoses. Both are included in this set. [327 images]

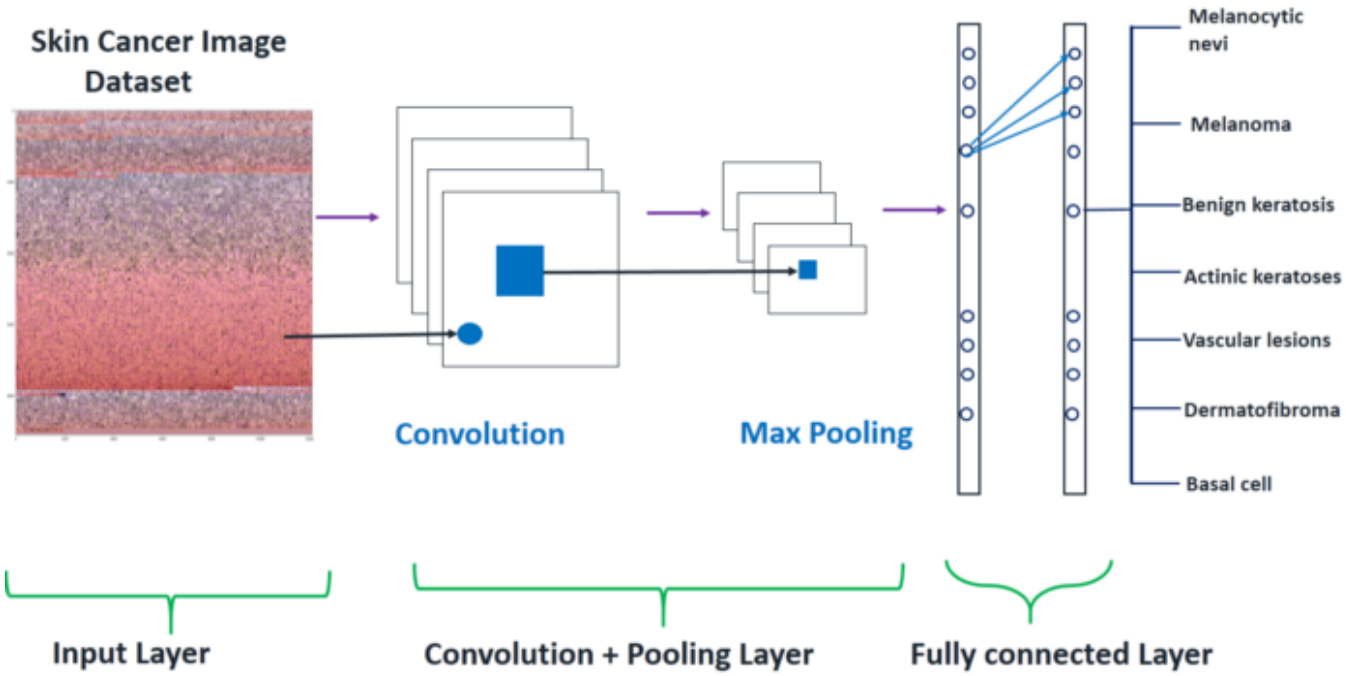
6- vasc Vascular skin lesions in the dataset range from cherry angiomas to angiokeratomas and pyogenic granulomas. Hemorrhage is also included in this category. [142 images]

7- df Dermatofibroma is a benign skin lesion regarded as either a benign proliferation or an inflammatory reaction to minimal trauma. It is brown often showing a central zone of fibrosis dermatoscopically. [115 images]

[Total images = 10015]

Table of Contents:

01- 02- 03- 04- 05- 06- 07- 08- 09- 10-



Start of the code

In [1]:

```
from numpy.random import seed
seed(101)
# from tensorflow import set_random_seed
# set_random_seed(101)

import pandas as pd
import numpy as np
import keras
from keras import backend as K

import tensorflow
tensorflow.random.set_seed(101)

from keras.layers.core import Dense, Dropout
from keras.optimizers import Adam
from keras.metrics import categorical_crossentropy
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model
from 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
```

```
Using TensorFlow backend.
```

The following line will read the image library file names and list them, they must be extracted into a separate file inside the project directory.

For this case, they are extracted into a file named as *input* inside Test 2 folder directory.

In [2]:

```
os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000')
```

Out[2]:

```
['HAM10000_images_part_1',
 'HAM10000_images_part_2',
 'HAM10000_metadata.csv',
 'hmnist_28_28_L.csv',
 'hmnist_28_28_RGB.csv',
 'hmnist_8_8_L.csv',
 'hmnist_8_8_RGB.csv']
```

Directory Structure These folders will store the images that will be fed to the Keras generators.

In [3]:

```
# Create a new directory
base_dir = 'base_dir'
os.mkdir(base_dir)

#[CREATE FOLDERS INSIDE THE BASE DIRECTORY]

# now we create 7 folders inside 'base_dir':
# train_dir
    # nv
    # mel
    # bkl
    # bcc
    # akiec
    # vasc
    # df

# val_dir
    # nv
    # mel
    # bkl
    # bcc
    # akiec
    # vasc
    # df

# create a path to 'base_dir' to which we will join the names of the new folders
# train_dir
train_dir = os.path.join(base_dir, 'train_dir')
os.mkdir(train_dir)

# val_dir
val_dir = os.path.join(base_dir, 'val_dir')
os.mkdir(val_dir)


# [CREATE FOLDERS INSIDE THE TRAIN, VALIDATION AND TEST FOLDERS]
# Inside each folder we create seperate folders for each class

# create new folders inside train_dir
nv = os.path.join(train_dir, 'nv')
os.mkdir(nv)
mel = os.path.join(train_dir, 'mel')
os.mkdir(mel)
bkl = os.path.join(train_dir, 'bkl')
os.mkdir(bkl)
bcc = os.path.join(train_dir, 'bcc')
os.mkdir(bcc)
akiec = os.path.join(train_dir, 'akiec')
os.mkdir(akiec)
vasc = os.path.join(train_dir, 'vasc')
os.mkdir(vasc)
df = os.path.join(train_dir, 'df')
os.mkdir(df)

# create new folders inside val_dir
nv = os.path.join(val_dir, 'nv')
os.mkdir(nv)
mel = os.path.join(val_dir, 'mel')
os.mkdir(mel)
```

```
bkl = os.path.join(val_dir, 'bkl')
os.mkdir(bkl)
bcc = os.path.join(val_dir, 'bcc')
os.mkdir(bcc)
akiec = os.path.join(val_dir, 'akiec')
os.mkdir(akiec)
vasc = os.path.join(val_dir, 'vasc')
os.mkdir(vasc)
df = os.path.join(val_dir, 'df')
os.mkdir(df)
```

Creating Train and Validation Sets

In [4]:

```
df_data = pd.read_csv('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/HAM10000_metadata.csv')

df_data.head()
```

Out[4]:

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

In [5]:

```
# this will tell us how many images are associated with each lesion_id
df = df_data.groupby('lesion_id').count()

# now we filter out lesion_id's that have only one image associated with it
df = df[df['image_id'] == 1]

df.reset_index(inplace=True)

df.head()
```

Out[5]:

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000001	1	1	1	1	1	1
1	HAM_0000003	1	1	1	1	1	1
2	HAM_0000004	1	1	1	1	1	1
3	HAM_0000007	1	1	1	1	1	1
4	HAM_0000008	1	1	1	1	1	1

In [6]:

```
# here we identify lesion_id's that have duplicate images and those that have only
# one image.

def identify_duplicates(x):

    unique_list = list(df['lesion_id'])

    if x in unique_list:
        return 'no_duplicates'
    else:
        return 'has_duplicates'

# create a new colum that is a copy of the lesion_id column
df_data['duplicates'] = df_data['lesion_id']
# apply the function to this new column
df_data['duplicates'] = df_data['duplicates'].apply(identify_duplicates)

df_data.head()
```

Out[6]:

	lesion_id	image_id	dx	dx_type	age	sex	localization	duplicates
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	has_duplicates
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	has_duplicates
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	has_duplicates
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	has_duplicates
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	has_duplicates

In [7]:

```
df_data['duplicates'].value_counts()
```

Out[7]:

```
no_duplicates     5514
has_duplicates    4501
Name: duplicates, dtype: int64
```

In [8]:

```
# now we filter out images that don't have duplicates
df = df_data[df_data['duplicates'] == 'no_duplicates']

df.shape
```

Out[8]:

```
(5514, 8)
```

In [9]:

```
# now we create a val set using df because we are sure that none of these images
# have augmented duplicates in the train set
y = df['dx']

_, df_val = train_test_split(df, test_size=0.17, random_state=101, stratify=y)

df_val.shape
```

Out[9]:

```
(938, 8)
```

In [10]:

```
df_val['dx'].value_counts()
```

Out[10]:

```
nv       751
bkl       75
mel       39
bcc       30
akiec     26
vasc      11
df         6
Name: dx, dtype: int64
```

Creating a training set with excluding images in the validation (val) set This is to separate the images to be fed to the model from others.

In [11]:

```
# This set will be df_data excluding all rows that are in the val set

# This function identifies if an image is part of the train
# or val set.
def identify_val_rows(x):
    # create a list of all the lesion_id's in the val set
    val_list = list(df_val['image_id'])

    if str(x) in val_list:
        return 'val'
    else:
        return 'train'

# identify train and val rows

# create a new colum that is a copy of the image_id column
df_data['train_or_val'] = df_data['image_id']
# apply the function to this new column
df_data['train_or_val'] = df_data['train_or_val'].apply(identify_val_rows)

# filter out train rows
df_train = df_data[df_data['train_or_val'] == 'train']


print(len(df_train))
print(len(df_val))
```

```
9077
938
```

In [12]:

```
df_train['dx'].value_counts()
```

Out[12]:

```
nv       5954
mel      1074
bkl      1024
bcc       484
akiec     301
vasc      131
df        109
Name: dx, dtype: int64
```

In [13]:

```
df_val['dx'].value_counts()
```

Out[13]:

```
nv       751
bkl       75
mel       39
bcc       30
akiec     26
vasc      11
df         6
Name: dx, dtype: int64
```

Transfering the images into folders

In [14]:

```
# Set the image_id as the index in df_data
df_data.set_index('image_id', inplace=True)
```

In [15]:

```
# Get a list of images in each of the two folders
folder_1 = os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_image
s_part_1')
folder_2 = os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_image
s_part_2')

# Get a list of train and val images
train_list = list(df_train['image_id'])
val_list = list(df_val['image_id'])


# Transfer the train images
for image in train_list:

    fname = image + '.jpg'
    label = df_data.loc[image,'dx']

    if fname in folder_1:
        # source path to image
        src = os.path.join('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_
images_part_1', fname)
        # destination path to image
        dst = os.path.join(train_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)

    if fname in folder_2:
        # source path to image
        src = os.path.join('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_
images_part_2', fname)
        # destination path to image
        dst = os.path.join(train_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)


# Transfer the val images
for image in val_list:

    fname = image + '.jpg'
    label = df_data.loc[image,'dx']

    if fname in folder_1:
        # source path to image
        src = os.path.join('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_
images_part_1', fname)
        # destination path to image
        dst = os.path.join(val_dir, label, fname)
        # copy the image from the source to the destination
        shutil.copyfile(src, dst)

    if fname in folder_2:
        # source path to image
        src = os.path.join('C:/Users/admi/Desktop/ML Model Tech Team/HAM10000/ham10000_
images_part_2', fname)
        # destination path to image
        dst = os.path.join(val_dir, label, fname)
        # copy the image from the source to the destination
```

```
        shutil.copyfile(src, dst)
```

In [16]:

```
# check how many train images we have in each folder

print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/nv'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/mel'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/bkl'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/bcc'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/akie
c')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/vasc'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/df'
)))
```

```
5954
1074
1024
484
301
131
109
```

In [17]:

```
# check how many val images we have in each folder

print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/nv')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/mel')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/bkl')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/bcc')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/akiec'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/vasc'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir/df')))
```

```
751
39
75
30
26
11
6
```

Image Data Augmentation Data augmentation is a technique that artificially creates a new training data for the purpose of creating new training data for the model, through some transofrmation processes to the original images in the training dataset directory, for each class selected.

Transformation includes shifting, flipping, zooming, rotation and brightness adjustment.

The main purpose is to expand the training dataset with new examples so that the model can be trained on more variation of images.

It is applied to the training dataset only, excluding validation and testing datasets. It also helps when the dataset is baised to a certain class of image.

Image data augmentation is done using ImageDataGenerator() class.

Copying the training images to augmented directory "aug_dir"

In [18]:

```
# note that we are not augmenting class 'nv'
class_list = ['mel','bkl','bcc','akiec','vasc','df']

for item in class_list:

    # We are creating temporary directories here because we delete these directories la
ter
    # create a base dir
    aug_dir = 'aug_dir'
    os.mkdir(aug_dir)
    # create a dir within the base dir to store images of the same class
    img_dir = os.path.join(aug_dir, 'img_dir')
    os.mkdir(img_dir)

    # Choose a class
    img_class = item

    # list all images in that directory
    img_list = os.listdir('base_dir/train_dir/' + img_class)

    # Copy images from the class train dir to the img_dir e.g. class 'mel'
    for fname in img_list:
            # source path to image
            src = os.path.join('base_dir/train_dir/' + img_class, fname)
            # destination path to image
            dst = os.path.join(img_dir, fname)
            # copy the image from the source to the destination
            shutil.copyfile(src, dst)


    # point to a dir containing the images and not to the images themselves
    path = aug_dir
    save_path = 'base_dir/train_dir/' + img_class

    # Create a data generator
    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,
        #brightness_range=(0.9,1.1),
        fill_mode='nearest')

    batch_size = 50

    aug_datagen = datagen.flow_from_directory(path,
                                          save_to_dir=save_path,
                                          save_format='jpg',
                                                  target_size=(224,224),
                                                  batch_size=batch_size)



    # Generate the augmented images and add them to the training folders

    ############
```

```
    num_aug_images_wanted = 6000 # total number of images we want to have in each class

    ############

    num_files = len(os.listdir(img_dir))
    num_batches = int(np.ceil((num_aug_images_wanted-num_files)/batch_size))

    # run the generator and create about 6000 augmented images
    for i in range(0,num_batches):

        imgs, labels = next(aug_datagen)

    # delete temporary directory with the raw image files
    shutil.rmtree('aug_dir')
```

```
Found 1074 images belonging to 1 classes.
Found 1024 images belonging to 1 classes.
Found 484 images belonging to 1 classes.
Found 301 images belonging to 1 classes.
Found 131 images belonging to 1 classes.
Found 109 images belonging to 1 classes.
```

In [19]:

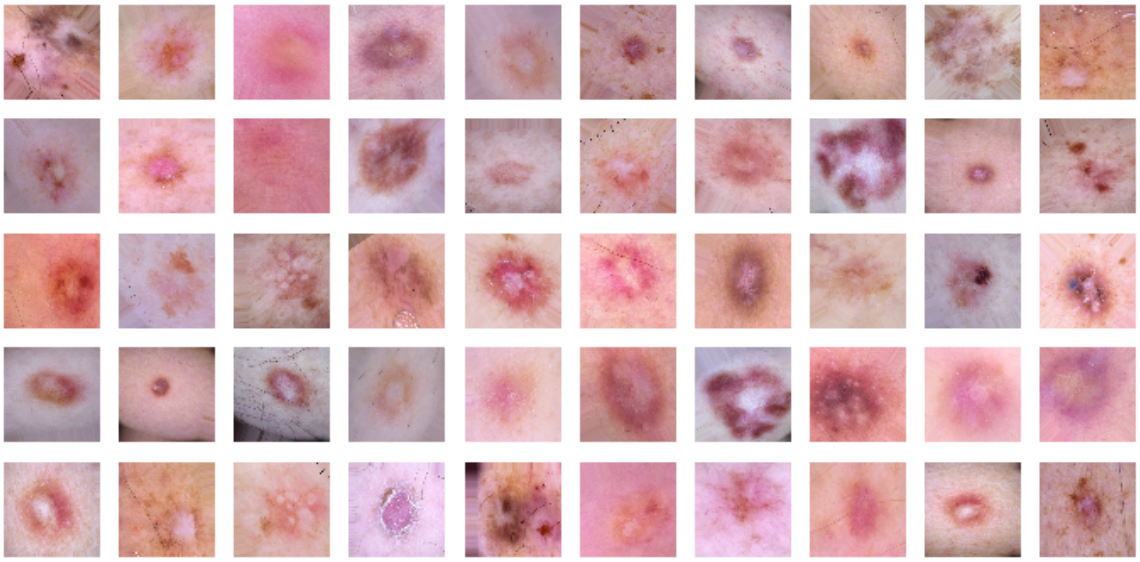
```
# Check how many train images we now have in each folder.
# This is the original images plus the augmented images.


print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/nv'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/mel'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/bkl'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/bcc'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/akie
c')))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/vasc'
)))
print(len(os.listdir('C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir/df'
)))
```

```
5954
5920
5920
5858
5217
5290
4410
```

Visualize 50 Augmented Images

In [20]:

```
# plots images with labels within jupyter notebook
# source: https://github.com/smileservices/keras_utils/blob/master/utils.py

def plots(ims, figsize=(12,6), rows=5, interp=False, titles=None): # 12,6
    if type(ims[0]) is np.ndarray:
        ims = np.array(ims).astype(np.uint8)
        if (ims.shape[-1] != 3):
            ims = ims.transpose((0,2,3,1))
    f = plt.figure(figsize=figsize)
    cols = len(ims)//rows if len(ims) % 2 == 0 else len(ims)//rows + 1
    for i in range(len(ims)):
        sp = f.add_subplot(rows, cols, i+1)
        sp.axis('Off')
        if titles is not None:
            sp.set_title(titles[i], fontsize=16)
        plt.imshow(ims[i], interpolation=None if interp else 'none')

plots(imgs, titles=None) # titles=labels will display the image labels, None won't
```



End of Data Preparation

==========

Start of Model Building

Set Up the Generators

In [21]:

```
train_path = 'C:/Users/admi/Desktop/ML Model Tech Team/base_dir/train_dir'
valid_path = 'C:/Users/admi/Desktop/ML Model Tech Team/base_dir/val_dir'

num_train_samples = len(df_train)
num_val_samples = len(df_val)
train_batch_size = 10
val_batch_size = 10
image_size = 224

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

In [22]:

```
datagen = ImageDataGenerator(
    preprocessing_function= \
    keras.applications.mobilenet.preprocess_input)

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

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

# Note: shuffle=False causes the test dataset to not be shuffled
test_batches = datagen.flow_from_directory(valid_path,
                                            target_size=(image_size,image_size),
                                            batch_size=1,
                                            shuffle=False)
```

```
Found 38569 images belonging to 7 classes.
Found 938 images belonging to 7 classes.
Found 938 images belonging to 7 classes.
```

Modify MobileNEt Model MobileNet is a CNN architecture model for Image Classification and Mobile Vision. It uses very less computation power to run or apply transfer learning to. This makes it a perfect fit for Mobile devices,embedded systems and computers without GPU or low computational efficiency with compromising significantly with the accuracy of the results.It is also best suited for web browsers as browsers have limitation over computation,graphic processing and storage. The core layer of MobileNet is depthwise separable filters, named as Depthwise Separable Convolution. The network structure is another factor to boost the performance. Finally, the width and resolution can be tuned to trade off between latency and accuracy.

Modify Mobilenet Model

In [23]:

```
# create a copy of a mobilenet model

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

```
Downloading data from https://github.com/fchollet/deep-learning-models/rel
eases/download/v0.6/mobilenet_1_0_224_tf.h5
17227776/17225924 [==============================] - 53s 3us/step
```

In [24]:

```
mobile.summary()
```

```
Model: "mobilenet_1.00_224"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 225, 225, 3)       0
_________________________________________________________________
conv1 (Conv2D)               (None, 112, 112, 32)      864
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 32)      128
_________________________________________________________________
conv1_relu (ReLU)            (None, 112, 112, 32)      0
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512
_________________________________________________________________
conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768
```

conv_pw_4_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2304
conv_dw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65536
conv_pw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2304
conv_dw_6_bn (BatchNormaliza	(None, 14, 14, 256)	1024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_pw_6 (Conv2D)	(None, 14, 14, 512)	131072
conv_pw_6_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_6_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_7_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_7 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_7_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_8_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_8 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_8_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_9_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0

```
conv_pw_9 (Conv2D)           (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_9_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_9_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_10 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_10 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_11 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_11 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pad_12 (ZeroPadding2D)  (None, 15, 15, 512)       0
_________________________________________________________________
conv_dw_12 (DepthwiseConv2D) (None, 7, 7, 512)         4608
_________________________________________________________________
conv_dw_12_bn (BatchNormaliz (None, 7, 7, 512)         2048
_________________________________________________________________
conv_dw_12_relu (ReLU)       (None, 7, 7, 512)         0
_________________________________________________________________
conv_pw_12 (Conv2D)          (None, 7, 7, 1024)        524288
_________________________________________________________________
conv_pw_12_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_12_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_dw_13 (DepthwiseConv2D) (None, 7, 7, 1024)        9216
_________________________________________________________________
conv_dw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
reshape_1 (Reshape)          (None, 1, 1, 1024)        0
_________________________________________________________________
dropout (Dropout)            (None, 1, 1, 1024)        0
```

```
_________________________________________________________________
conv_preds (Conv2D)          (None, 1, 1, 1000)        1025000
_________________________________________________________________
reshape_2 (Reshape)          (None, 1000)              0
_________________________________________________________________
act_softmax (Activation)     (None, 1000)              0
=================================================================
Total params: 4,253,864
Trainable params: 4,231,976
Non-trainable params: 21,888
_________________________________________________________________
```

In [25]:

```
# MobileNet has 28 layers.
# Source: https://arxiv.org/pdf/1704.04861.pdf
type(mobile.layers)
```

Out[25]:

```
list
```

In [26]:

```
# CREATE THE MODEL ARCHITECTURE

# Exclude the last 5 layers of the above model.
# This will include all layers up to and including global_average_pooling2d_1
x = mobile.layers[-6].output

# Create a new dense layer for predictions
# 7 corresponds to the number of classes
x = Dropout(0.25)(x)
predictions = Dense(7, activation='softmax')(x)

# inputs=mobile.input selects the input layer, outputs=predictions refers to the
# dense layer we created above.

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

In [27]:

```
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 225, 225, 3)       0
_________________________________________________________________
conv1 (Conv2D)               (None, 112, 112, 32)      864
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 32)      128
_________________________________________________________________
conv1_relu (ReLU)            (None, 112, 112, 32)      0
_________________________________________________________________
conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)      288
_________________________________________________________________
conv_dw_1_bn (BatchNormaliza (None, 112, 112, 32)      128
_________________________________________________________________
conv_dw_1_relu (ReLU)        (None, 112, 112, 32)      0
_________________________________________________________________
conv_pw_1 (Conv2D)           (None, 112, 112, 64)      2048
_________________________________________________________________
conv_pw_1_bn (BatchNormaliza (None, 112, 112, 64)      256
_________________________________________________________________
conv_pw_1_relu (ReLU)        (None, 112, 112, 64)      0
_________________________________________________________________
conv_pad_2 (ZeroPadding2D)   (None, 113, 113, 64)      0
_________________________________________________________________
conv_dw_2 (DepthwiseConv2D)  (None, 56, 56, 64)        576
_________________________________________________________________
conv_dw_2_bn (BatchNormaliza (None, 56, 56, 64)        256
_________________________________________________________________
conv_dw_2_relu (ReLU)        (None, 56, 56, 64)        0
_________________________________________________________________
conv_pw_2 (Conv2D)           (None, 56, 56, 128)       8192
_________________________________________________________________
conv_pw_2_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_2_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_dw_3 (DepthwiseConv2D)  (None, 56, 56, 128)       1152
_________________________________________________________________
conv_dw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_dw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pw_3 (Conv2D)           (None, 56, 56, 128)       16384
_________________________________________________________________
conv_pw_3_bn (BatchNormaliza (None, 56, 56, 128)       512
_________________________________________________________________
conv_pw_3_relu (ReLU)        (None, 56, 56, 128)       0
_________________________________________________________________
conv_pad_4 (ZeroPadding2D)   (None, 57, 57, 128)       0
_________________________________________________________________
conv_dw_4 (DepthwiseConv2D)  (None, 28, 28, 128)       1152
_________________________________________________________________
conv_dw_4_bn (BatchNormaliza (None, 28, 28, 128)       512
_________________________________________________________________
conv_dw_4_relu (ReLU)        (None, 28, 28, 128)       0
_________________________________________________________________
conv_pw_4 (Conv2D)           (None, 28, 28, 256)       32768
```

conv_pw_4_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2304
conv_dw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65536
conv_pw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2304
conv_dw_6_bn (BatchNormaliza	(None, 14, 14, 256)	1024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_pw_6 (Conv2D)	(None, 14, 14, 512)	131072
conv_pw_6_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_6_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_7_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_7 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_7_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_8_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_8 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_8_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_pw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_9_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0

```
conv_pw_9 (Conv2D)           (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_9_bn (BatchNormaliza (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_9_relu (ReLU)        (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_10 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_10 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_10_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_10_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_dw_11 (DepthwiseConv2D) (None, 14, 14, 512)       4608
_________________________________________________________________
conv_dw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_dw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pw_11 (Conv2D)          (None, 14, 14, 512)       262144
_________________________________________________________________
conv_pw_11_bn (BatchNormaliz (None, 14, 14, 512)       2048
_________________________________________________________________
conv_pw_11_relu (ReLU)       (None, 14, 14, 512)       0
_________________________________________________________________
conv_pad_12 (ZeroPadding2D)  (None, 15, 15, 512)       0
_________________________________________________________________
conv_dw_12 (DepthwiseConv2D) (None, 7, 7, 512)         4608
_________________________________________________________________
conv_dw_12_bn (BatchNormaliz (None, 7, 7, 512)         2048
_________________________________________________________________
conv_dw_12_relu (ReLU)       (None, 7, 7, 512)         0
_________________________________________________________________
conv_pw_12 (Conv2D)          (None, 7, 7, 1024)        524288
_________________________________________________________________
conv_pw_12_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_12_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_dw_13 (DepthwiseConv2D) (None, 7, 7, 1024)        9216
_________________________________________________________________
conv_dw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_dw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
conv_pw_13 (Conv2D)          (None, 7, 7, 1024)        1048576
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, 7, 7, 1024)        4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, 7, 7, 1024)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 7175
```

```
=================================================================
Total params: 3,236,039
Trainable params: 3,214,151
Non-trainable params: 21,888
_________________________________________________________________
```

In [28]:

```
# We need to choose how many layers we actually want to be trained.

# Here we are freezing the weights of all layers except the
# last 23 layers in the new model.
# The last 23 layers of the model will be trained.

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

Train the Model

In [29]:

```
# Define Top2 and Top3 Accuracy

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 [30]:

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

In [31]:

```
# Get the labels that are associated with each index
print(valid_batches.class_indices)
```

```
{'akiec': 0, 'bcc': 1, 'bkl': 2, 'df': 3, 'mel': 4, 'nv': 5, 'vasc': 6}
```

In [32]:

```
# Add weights to try to make the model more sensitive to melanoma

class_weights={
    0: 1.0, # akiec
    1: 1.0, # bcc
    2: 1.0, # bkl
    3: 1.0, # df
    4: 3.0, # mel # Try to make the model more sensitive to Melanoma.
    5: 1.0, # nv
    6: 1.0, # vasc
}
```

In [33]:

```
# load model, after first training in case training was interrupted,
# uncomment the next line then train back, will have higher accuracy
# will restore previous weights values (the "state of the model")

# model.load_weights("model.h5")
```

In [34]:

```
# You can use a trained model without having to retrain it,
# or pick-up training where you left off—in case the training
# process was interrupted. The tf.keras.callbacks.ModelCheckpoint
# callback allows to continually save the model both during and at
# the end of training.

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=train_steps,
                              class_weight=class_weights,
                    validation_data=valid_batches,
                    validation_steps=val_steps,
                    epochs=30, verbose=1,
                   callbacks=callbacks_list) # Pass callback to training

# This may generate warnings related to saving the state of the optimizer.
# These warnings (and similar warnings throughout this notebook)
# are in place to discourage outdated usage, and can be ignored.
```

```
Epoch 1/30
908/908 [==============================] - 929s 1s/step - loss: 1.7708 - c
ategorical_accuracy: 0.4968 - top_2_accuracy: 0.7144 - top_3_accuracy: 0.8
446 - val_loss: 0.4756 - val_categorical_accuracy: 0.7783 - val_top_2_accu
racy: 0.8891 - val_top_3_accuracy: 0.9435

Epoch 00001: val_top_3_accuracy improved from -inf to 0.94350, saving mode
l to model.h5
Epoch 2/30
908/908 [==============================] - 940s 1s/step - loss: 1.3182 - c
ategorical_accuracy: 0.6021 - top_2_accuracy: 0.7985 - top_3_accuracy: 0.9
073 - val_loss: 5.5279 - val_categorical_accuracy: 0.3156 - val_top_2_accu
racy: 0.4478 - val_top_3_accuracy: 0.5309

Epoch 00002: val_top_3_accuracy did not improve from 0.94350
Epoch 3/30
908/908 [==============================] - 929s 1s/step - loss: 1.2046 - c
ategorical_accuracy: 0.6327 - top_2_accuracy: 0.8213 - top_3_accuracy: 0.9
217 - val_loss: 1.3003 - val_categorical_accuracy: 0.6066 - val_top_2_accu
racy: 0.8465 - val_top_3_accuracy: 0.8763

Epoch 00003: val_top_3_accuracy did not improve from 0.94350

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.004999999888241
291.
Epoch 4/30
908/908 [==============================] - 932s 1s/step - loss: 0.9881 - c
ategorical_accuracy: 0.6883 - top_2_accuracy: 0.8729 - top_3_accuracy: 0.9
501 - val_loss: 1.2890 - val_categorical_accuracy: 0.5522 - val_top_2_accu
racy: 0.8380 - val_top_3_accuracy: 0.9360

Epoch 00004: val_top_3_accuracy did not improve from 0.94350
Epoch 5/30
908/908 [==============================] - 922s 1s/step - loss: 0.8798 - c
ategorical_accuracy: 0.7241 - top_2_accuracy: 0.8891 - top_3_accuracy: 0.9
612 - val_loss: 1.0476 - val_categorical_accuracy: 0.8102 - val_top_2_accu
racy: 0.8774 - val_top_3_accuracy: 0.9328

Epoch 00005: val_top_3_accuracy did not improve from 0.94350

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.002499999944120
6455.
Epoch 6/30
908/908 [==============================] - 927s 1s/step - loss: 0.7830 - c
ategorical_accuracy: 0.7534 - top_2_accuracy: 0.9126 - top_3_accuracy: 0.9
704 - val_loss: 0.8508 - val_categorical_accuracy: 0.7655 - val_top_2_accu
racy: 0.8998 - val_top_3_accuracy: 0.9446

Epoch 00006: val_top_3_accuracy improved from 0.94350 to 0.94456, saving m
odel to model.h5
Epoch 7/30
908/908 [==============================] - 922s 1s/step - loss: 0.7475 - c
ategorical_accuracy: 0.7745 - top_2_accuracy: 0.9219 - top_3_accuracy: 0.9
749 - val_loss: 0.6132 - val_categorical_accuracy: 0.6780 - val_top_2_accu
racy: 0.8859 - val_top_3_accuracy: 0.9467

Epoch 00007: val_top_3_accuracy improved from 0.94456 to 0.94670, saving m
odel to model.h5
Epoch 8/30
908/908 [==============================] - 918s 1s/step - loss: 0.7073 - c
ategorical_accuracy: 0.7834 - top_2_accuracy: 0.9271 - top_3_accuracy: 0.9
```

```
792 - val_loss: 0.5366 - val_categorical_accuracy: 0.7186 - val_top_2_accu
racy: 0.9051 - val_top_3_accuracy: 0.9574

Epoch 00008: val_top_3_accuracy improved from 0.94670 to 0.95736, saving m
odel to model.h5
Epoch 9/30
908/908 [==============================] - 920s 1s/step - loss: 0.6831 - c
ategorical_accuracy: 0.7839 - top_2_accuracy: 0.9280 - top_3_accuracy: 0.9
812 - val_loss: 1.1652 - val_categorical_accuracy: 0.6269 - val_top_2_accu
racy: 0.8657 - val_top_3_accuracy: 0.9318

Epoch 00009: val_top_3_accuracy did not improve from 0.95736
Epoch 10/30
908/908 [==============================] - 922s 1s/step - loss: 0.6165 - c
ategorical_accuracy: 0.8137 - top_2_accuracy: 0.9434 - top_3_accuracy: 0.9
835 - val_loss: 1.6530 - val_categorical_accuracy: 0.5981 - val_top_2_accu
racy: 0.8731 - val_top_3_accuracy: 0.9275

Epoch 00010: val_top_3_accuracy did not improve from 0.95736

Epoch 00010: ReduceLROnPlateau reducing learning rate to 0.001249999972060
3228.
Epoch 11/30
908/908 [==============================] - 921s 1s/step - loss: 0.5696 - c
ategorical_accuracy: 0.8193 - top_2_accuracy: 0.9479 - top_3_accuracy: 0.9
881 - val_loss: 0.3955 - val_categorical_accuracy: 0.7676 - val_top_2_accu
racy: 0.9051 - val_top_3_accuracy: 0.9478

Epoch 00011: val_top_3_accuracy did not improve from 0.95736
Epoch 12/30
908/908 [==============================] - 922s 1s/step - loss: 0.5390 - c
ategorical_accuracy: 0.8318 - top_2_accuracy: 0.9530 - top_3_accuracy: 0.9
873 - val_loss: 0.5136 - val_categorical_accuracy: 0.6823 - val_top_2_accu
racy: 0.8785 - val_top_3_accuracy: 0.9403

Epoch 00012: val_top_3_accuracy did not improve from 0.95736

Epoch 00012: ReduceLROnPlateau reducing learning rate to 0.000624999986030
1614.
Epoch 13/30
908/908 [==============================] - 919s 1s/step - loss: 0.4928 - c
ategorical_accuracy: 0.8433 - top_2_accuracy: 0.9593 - top_3_accuracy: 0.9
915 - val_loss: 0.1569 - val_categorical_accuracy: 0.7186 - val_top_2_accu
racy: 0.8902 - val_top_3_accuracy: 0.9488

Epoch 00013: val_top_3_accuracy did not improve from 0.95736
Epoch 14/30
908/908 [==============================] - 912s 1s/step - loss: 0.4445 - c
ategorical_accuracy: 0.8612 - top_2_accuracy: 0.9627 - top_3_accuracy: 0.9
916 - val_loss: 0.8228 - val_categorical_accuracy: 0.7527 - val_top_2_accu
racy: 0.8987 - val_top_3_accuracy: 0.9446

Epoch 00014: val_top_3_accuracy did not improve from 0.95736

Epoch 00014: ReduceLROnPlateau reducing learning rate to 0.000312499993015
0807.
Epoch 15/30
908/908 [==============================] - 914s 1s/step - loss: 0.4179 - c
ategorical_accuracy: 0.8698 - top_2_accuracy: 0.9710 - top_3_accuracy: 0.9
931 - val_loss: 1.0050 - val_categorical_accuracy: 0.7719 - val_top_2_accu
racy: 0.8955 - val_top_3_accuracy: 0.9499
```

```
Epoch 00015: val_top_3_accuracy did not improve from 0.95736
Epoch 16/30
908/908 [==============================] - 919s 1s/step - loss: 0.4108 - c
ategorical_accuracy: 0.8725 - top_2_accuracy: 0.9706 - top_3_accuracy: 0.9
924 - val_loss: 1.1721 - val_categorical_accuracy: 0.7708 - val_top_2_accu
racy: 0.9019 - val_top_3_accuracy: 0.9467

Epoch 00016: val_top_3_accuracy did not improve from 0.95736

Epoch 00016: ReduceLROnPlateau reducing learning rate to 0.000156249996507
54035.
Epoch 17/30
908/908 [==============================] - 902s 994ms/step - loss: 0.4062
- categorical_accuracy: 0.8726 - top_2_accuracy: 0.9694 - top_3_accuracy:
0.9938 - val_loss: 0.7211 - val_categorical_accuracy: 0.7559 - val_top_2_a
ccuracy: 0.8934 - val_top_3_accuracy: 0.9478

Epoch 00017: val_top_3_accuracy did not improve from 0.95736
Epoch 18/30
908/908 [==============================] - 890s 981ms/step - loss: 0.3678
- categorical_accuracy: 0.8878 - top_2_accuracy: 0.9747 - top_3_accuracy:
0.9954 - val_loss: 0.6871 - val_categorical_accuracy: 0.7569 - val_top_2_a
ccuracy: 0.8966 - val_top_3_accuracy: 0.9488

Epoch 00018: val_top_3_accuracy did not improve from 0.95736

Epoch 00018: ReduceLROnPlateau reducing learning rate to 7.812499825377017
e-05.
Epoch 19/30
908/908 [==============================] - 892s 982ms/step - loss: 0.3843
- categorical_accuracy: 0.8816 - top_2_accuracy: 0.9732 - top_3_accuracy:
0.9941 - val_loss: 0.0677 - val_categorical_accuracy: 0.7569 - val_top_2_a
ccuracy: 0.8966 - val_top_3_accuracy: 0.9499

Epoch 00019: val_top_3_accuracy did not improve from 0.95736
Epoch 20/30
908/908 [==============================] - 892s 982ms/step - loss: 0.3540
- categorical_accuracy: 0.8881 - top_2_accuracy: 0.9739 - top_3_accuracy:
0.9955 - val_loss: 0.6636 - val_categorical_accuracy: 0.7633 - val_top_2_a
ccuracy: 0.9019 - val_top_3_accuracy: 0.9478

Epoch 00020: val_top_3_accuracy did not improve from 0.95736

Epoch 00020: ReduceLROnPlateau reducing learning rate to 3.906249912688508
6e-05.
Epoch 21/30
908/908 [==============================] - 891s 981ms/step - loss: 0.3716
- categorical_accuracy: 0.8820 - top_2_accuracy: 0.9738 - top_3_accuracy:
0.9948 - val_loss: 0.9225 - val_categorical_accuracy: 0.7633 - val_top_2_a
ccuracy: 0.9009 - val_top_3_accuracy: 0.9478

Epoch 00021: val_top_3_accuracy did not improve from 0.95736
Epoch 22/30
908/908 [==============================] - 891s 981ms/step - loss: 0.3628
- categorical_accuracy: 0.8902 - top_2_accuracy: 0.9732 - top_3_accuracy:
0.9960 - val_loss: 1.5689 - val_categorical_accuracy: 0.7601 - val_top_2_a
ccuracy: 0.9009 - val_top_3_accuracy: 0.9478

Epoch 00022: val_top_3_accuracy did not improve from 0.95736
```

```
Epoch 00022: ReduceLROnPlateau reducing learning rate to 1.953124956344254
3e-05.
Epoch 23/30
908/908 [==============================] - 890s 980ms/step - loss: 0.3600
- categorical_accuracy: 0.8889 - top_2_accuracy: 0.9756 - top_3_accuracy:
0.9950 - val_loss: 0.8504 - val_categorical_accuracy: 0.7772 - val_top_2_a
ccuracy: 0.9062 - val_top_3_accuracy: 0.9510

Epoch 00023: val_top_3_accuracy did not improve from 0.95736
Epoch 24/30
908/908 [==============================] - 890s 980ms/step - loss: 0.3571
- categorical_accuracy: 0.8861 - top_2_accuracy: 0.9744 - top_3_accuracy:
0.9957 - val_loss: 0.5233 - val_categorical_accuracy: 0.7814 - val_top_2_a
ccuracy: 0.9051 - val_top_3_accuracy: 0.9478

Epoch 00024: val_top_3_accuracy did not improve from 0.95736

Epoch 00024: ReduceLROnPlateau reducing learning rate to 1e-05.
Epoch 25/30
908/908 [==============================] - 890s 980ms/step - loss: 0.3639
- categorical_accuracy: 0.8907 - top_2_accuracy: 0.9743 - top_3_accuracy:
0.9947 - val_loss: 0.6755 - val_categorical_accuracy: 0.7804 - val_top_2_a
ccuracy: 0.9051 - val_top_3_accuracy: 0.9510

Epoch 00025: val_top_3_accuracy did not improve from 0.95736
Epoch 26/30
908/908 [==============================] - 890s 980ms/step - loss: 0.3607
- categorical_accuracy: 0.8935 - top_2_accuracy: 0.9772 - top_3_accuracy:
0.9961 - val_loss: 0.1559 - val_categorical_accuracy: 0.7729 - val_top_2_a
ccuracy: 0.9051 - val_top_3_accuracy: 0.9478

Epoch 00026: val_top_3_accuracy did not improve from 0.95736
Epoch 27/30
908/908 [==============================] - 891s 981ms/step - loss: 0.3496
- categorical_accuracy: 0.8874 - top_2_accuracy: 0.9752 - top_3_accuracy:
0.9959 - val_loss: 0.3550 - val_categorical_accuracy: 0.7729 - val_top_2_a
ccuracy: 0.9041 - val_top_3_accuracy: 0.9478

Epoch 00027: val_top_3_accuracy did not improve from 0.95736
Epoch 28/30
908/908 [==============================] - 890s 980ms/step - loss: 0.3743
- categorical_accuracy: 0.8846 - top_2_accuracy: 0.9758 - top_3_accuracy:
0.9949 - val_loss: 0.1785 - val_categorical_accuracy: 0.7687 - val_top_2_a
ccuracy: 0.9030 - val_top_3_accuracy: 0.9478

Epoch 00028: val_top_3_accuracy did not improve from 0.95736
Epoch 29/30
908/908 [==============================] - 890s 981ms/step - loss: 0.3539
- categorical_accuracy: 0.8906 - top_2_accuracy: 0.9729 - top_3_accuracy:
0.9955 - val_loss: 0.9402 - val_categorical_accuracy: 0.7751 - val_top_2_a
ccuracy: 0.9051 - val_top_3_accuracy: 0.9499

Epoch 00029: val_top_3_accuracy did not improve from 0.95736
Epoch 30/30
908/908 [==============================] - 891s 981ms/step - loss: 0.3647
- categorical_accuracy: 0.8871 - top_2_accuracy: 0.9760 - top_3_accuracy:
0.9954 - val_loss: 0.6519 - val_categorical_accuracy: 0.7719 - val_top_2_a
ccuracy: 0.9041 - val_top_3_accuracy: 0.9499

Epoch 00030: val_top_3_accuracy did not improve from 0.95736
```

Evaluate the model using the val set

In [35]:

```
# get the metric names so we can use evaulate_generator
model.metrics_names
```

Out[35]:

```
['loss', 'categorical_accuracy', 'top_2_accuracy', 'top_3_accuracy']
```

In [36]:

```
# Here the the last epoch will be used.

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

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: 0.014978794381022453
val_cat_acc: 0.7718549966812134
val_top_2_acc: 0.9040511846542358
val_top_3_acc: 0.9498934149742126
```

In [37]:

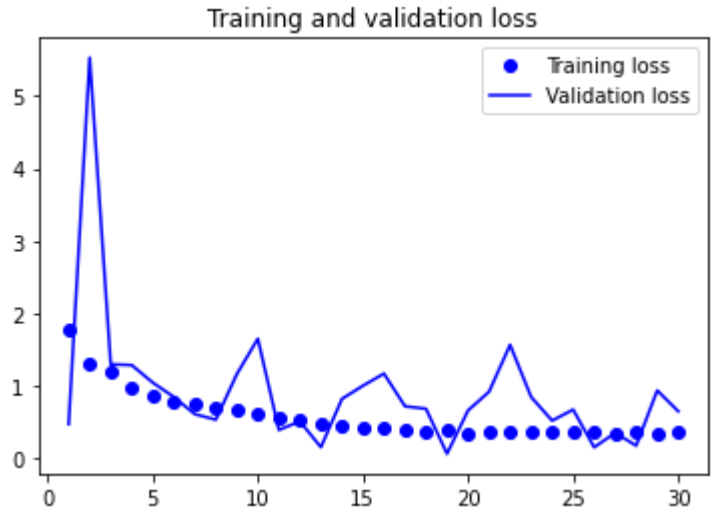
```
# Here the best epoch will be used.

model.load_weights('model.h5')

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

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)
```

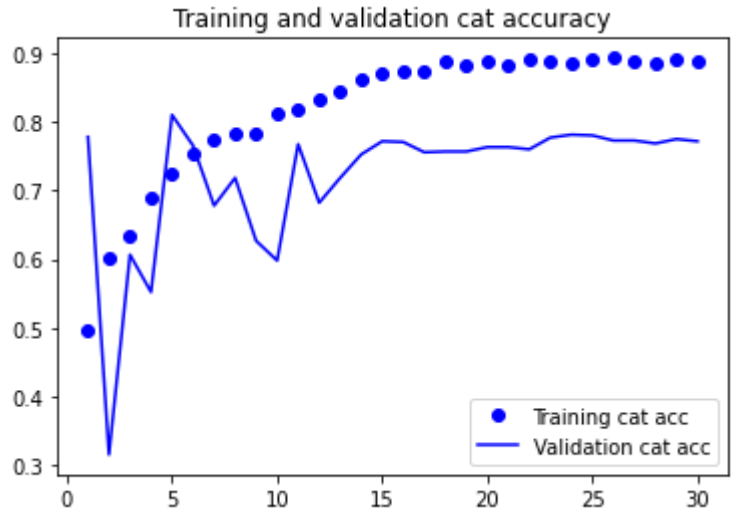
```
val_loss: 0.06077069044113159
val_cat_acc: 0.7185500860214233
val_top_2_acc: 0.9051172733306885
val_top_3_acc: 0.9573560953140259
```

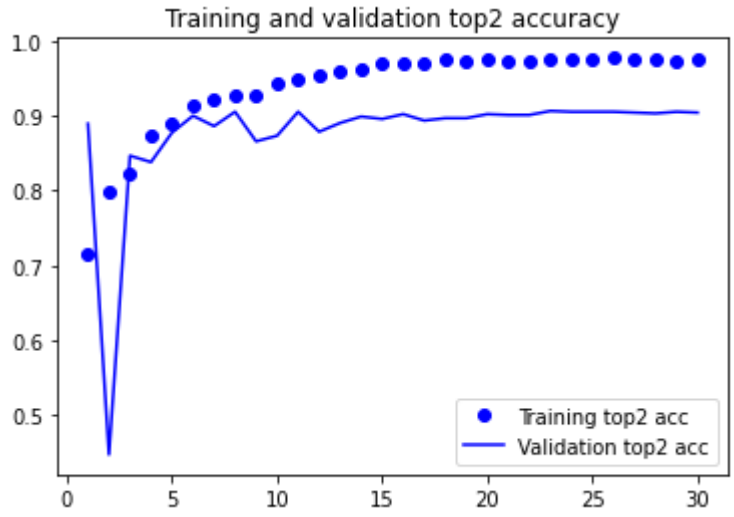
Plot the Training Curves

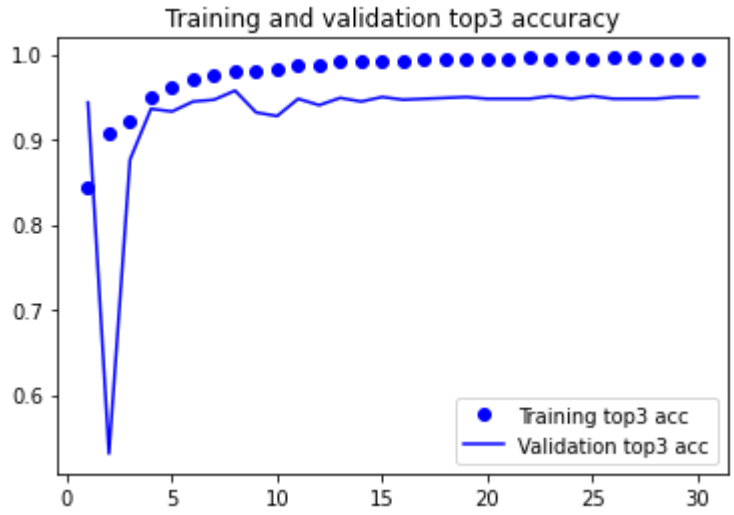
In [38]:

```
# display the loss and accuracy curves

import matplotlib.pyplot as plt

acc = history.history['categorical_accuracy']
val_acc = history.history['val_categorical_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
train_top2_acc = history.history['top_2_accuracy']
val_top2_acc = history.history['val_top_2_accuracy']
train_top3_acc = history.history['top_3_accuracy']
val_top3_acc = history.history['val_top_3_accuracy']
epochs = range(1, len(acc) + 1)

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.figure()

plt.plot(epochs, acc, 'bo', label='Training cat acc')
plt.plot(epochs, val_acc, 'b', label='Validation cat acc')
plt.title('Training and validation cat accuracy')
plt.legend()
plt.figure()


plt.plot(epochs, train_top2_acc, 'bo', label='Training top2 acc')
plt.plot(epochs, val_top2_acc, 'b', label='Validation top2 acc')
plt.title('Training and validation top2 accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, train_top3_acc, 'bo', label='Training top3 acc')
plt.plot(epochs, val_top3_acc, 'b', label='Validation top3 acc')
plt.title('Training and validation top3 accuracy')
plt.legend()


plt.show()
```


Training and validation loss
Training loss
Validation loss


Training and validation cat accuracy
Training cat acc
Validation cat acc


Training and validation top2 accuracy
Training top2 acc
Validation top2 acc



Create a Confusion Matrix

In [39]:

```
# Get the labels of the test images.

test_labels = test_batches.classes
```

In [40]:

```
# We need these to plot the confusion matrix.
test_labels
```

Out[40]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3,
       3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6])
```

In [41]:

```
# Print the label associated with each class
test_batches.class_indices
```

Out[41]:

```
{'akiec': 0, 'bcc': 1, 'bkl': 2, 'df': 3, 'mel': 4, 'nv': 5, 'vasc': 6}
```

In [42]:

```
# make a prediction
predictions = model.predict_generator(test_batches, steps=len(df_val), verbose=1)
```

```
938/938 [==============================] - 58s 62ms/step
```

In [43]:

```
predictions.shape
```

Out[43]:

```
(938, 7)
```

In [44]:

```
# Source: Scikit Learn website
# http://scikit-learn.org/stable/auto_examples/
# model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-
# selection-plot-confusion-matrix-py


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()
```

In [45]:

```
test_labels.shape
```

Out[45]:

```
(938,)
```

In [46]:

```
# argmax returns the index of the max value in a row
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

In [47]:

```
test_batches.class_indices
```

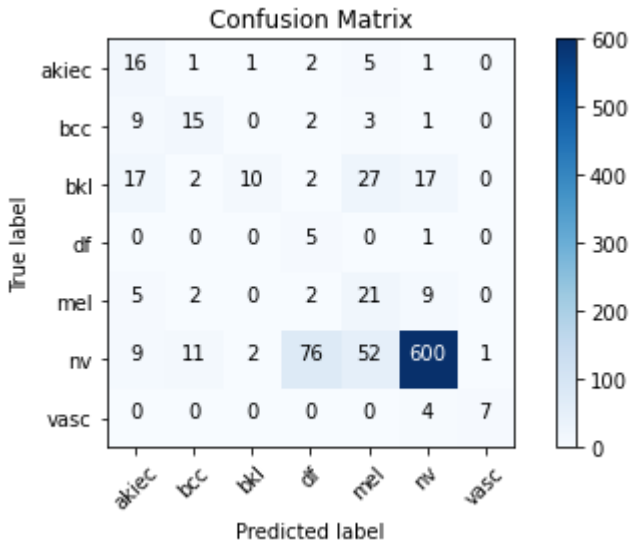
Out[47]:

```
{'akiec': 0, 'bcc': 1, 'bkl': 2, 'df': 3, 'mel': 4, 'nv': 5, 'vasc': 6}
```

In [48]:

```
# Define the labels of the class indices. These need to match the
# order shown above.
cm_plot_labels = ['akiec', 'bcc', 'bkl', 'df', 'mel','nv', 'vasc']

plot_confusion_matrix(cm, cm_plot_labels, title='Confusion Matrix')
```

```
Confusion matrix, without normalization
[[ 16   1   1   2   5   1   0]
 [  9  15   0   2   3   1   0]
 [ 17   2  10   2  27  17   0]
 [  0   0   0   5   0   1   0]
 [  5   2   0   2  21   9   0]
 [  9  11   2  76  52 600   1]
 [  0   0   0   0   0   4   7]]
```



Generate the Classification Report

In [49]:

```
# Get the index of the class with the highest probability score
y_pred = np.argmax(predictions, axis=1)

# Get the labels of the test images.
y_true = test_batches.classes
```

In [50]:

```
from sklearn.metrics import classification_report

# Generate a classification report
report = classification_report(y_true, y_pred, target_names=cm_plot_labels)

print(report)
```

```
              precision    recall  f1-score   support

       akiec       0.29      0.62      0.39        26
         bcc       0.48      0.50      0.49        30
         bkl       0.77      0.13      0.23        75
          df       0.06      0.83      0.11         6
         mel       0.19      0.54      0.29        39
          nv       0.95      0.80      0.87       751
        vasc       0.88      0.64      0.74        11

    accuracy                           0.72       938
   macro avg       0.52      0.58      0.44       938
weighted avg       0.86      0.72      0.76       938
```

Recall = Given a class, will the classifier be able to detect it? **Precision** = Given a class prediction from a classifier, how likely is it to be correct? **F1 Score** = The harmonic mean of the recall and precision. Essentially, it punishes extreme values.

In [51]:

```
# End of Model Building

# Convert the Model from Keras to Tensorflow.js
```

Install Tensorflow.js

In [52]:

```
%pip install tensorflowjs
```

```
ERROR: tensorflow 2.0.0 has requirement gast==0.2.2, but you'll have gast
0.3.3 which is incompatible.
```

```
Collecting tensorflowjs
  Downloading tensorflowjs-2.1.0-py3-none-any.whl (60 kB)
Collecting tensorflow-hub==0.7.0
  Downloading tensorflow_hub-0.7.0-py2.py3-none-any.whl (89 kB)
Collecting PyInquirer==1.0.3
  Downloading PyInquirer-1.0.3.tar.gz (27 kB)
Requirement already satisfied: h5py>=2.8.0 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflowjs) (2.10.0)
Collecting numpy<1.19.0,>=1.16.4
  Downloading numpy-1.18.5-cp37-cp37m-win_amd64.whl (12.7 MB)
Collecting tensorflow-cpu<3,>=2.1.0
  Downloading tensorflow_cpu-2.3.0-cp37-cp37m-win_amd64.whl (195.8 MB)
Requirement already satisfied: six>=1.12.0 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflowjs) (1.15.0)
Requirement already satisfied: protobuf>=3.4.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorflow-hub==0.7.0->tensorflow
js) (3.12.3)
Collecting prompt_toolkit==1.0.14
  Downloading prompt_toolkit-1.0.14-py3-none-any.whl (248 kB)
Requirement already satisfied: Pygments>=2.2.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from PyInquirer==1.0.3->tensorflowjs)
(2.6.1)
Collecting regex>=2016.11.21
  Downloading regex-2020.7.14-cp37-cp37m-win_amd64.whl (268 kB)
Collecting tensorflow-estimator<2.4.0,>=2.3.0
  Downloading tensorflow_estimator-2.3.0-py2.py3-none-any.whl (459 kB)
Requirement already satisfied: wheel>=0.26 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflowj
s) (0.34.2)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorf
lowjs) (1.1.0)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensor
flowjs) (3.1.0)
Collecting gast==0.3.3
  Downloading gast-0.3.3-py2.py3-none-any.whl (9.7 kB)
Collecting tensorboard<3,>=2.3.0
  Downloading tensorboard-2.3.0-py3-none-any.whl (6.8 MB)
Collecting scipy==1.4.1
  Downloading scipy-1.4.1-cp37-cp37m-win_amd64.whl (30.9 MB)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\admi\anaconda3\en
vs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflow
js) (1.27.2)
Collecting keras-preprocessing<1.2,>=1.1.1
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Collecting astunparse==1.6.3
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: wrapt>=1.11.1 in c:\users\admi\anaconda3\en
vs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflow
js) (1.12.1)
Requirement already satisfied: google-pasta>=0.1.8 in c:\users\admi\anacon
da3\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tens
orflowjs) (0.2.0)
Requirement already satisfied: absl-py>=0.7.0 in c:\users\admi\anaconda3\e
nvs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflo
wjs) (0.9.0)
Requirement already satisfied: setuptools in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from protobuf>=3.4.0->tensorflow-hub==0.7.0
->tensorflowjs) (49.2.0.post20200714)
Requirement already satisfied: wcwidth in c:\users\admi\anaconda3\envs\ten
```

```
sorflow\lib\site-packages (from prompt_toolkit==1.0.14->PyInquirer==1.0.3-
>tensorflowjs) (0.2.5)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\a
dmi\anaconda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.
0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (1.6.0)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorflo
w-cpu<3,>=2.1.0->tensorflowjs) (0.16.1)
Requirement already satisfied: markdown>=2.6.8 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorflow
-cpu<3,>=2.1.0->tensorflowjs) (3.1.1)
Requirement already satisfied: google-auth<2,>=1.6.3 in c:\users\admi\anac
onda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tenso
rflow-cpu<3,>=2.1.0->tensorflowjs) (1.17.2)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\user
s\admi\anaconda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=
2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (0.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\admi\anacon
da3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorf
low-cpu<3,>=2.1.0->tensorflowjs) (2.24.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\admi\anac
onda3\envs\tensorflow\lib\site-packages (from google-auth<2,>=1.6.3->tenso
rboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (0.2.7)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\admi\ana
conda3\envs\tensorflow\lib\site-packages (from google-auth<2,>=1.6.3->tens
orboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (4.1.0)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3" in
c:\users\admi\anaconda3\envs\tensorflow\lib\site-packages (from google-aut
h<2,>=1.6.3->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowj
s) (4.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\admi\a
naconda3\envs\tensorflow\lib\site-packages (from google-auth-oauthlib<0.5,
>=0.4.1->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs)
(1.3.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\admi\anaconda3\env
s\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=
2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboard
<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\admi\anaconda3\envs\tensorflow\lib\site-packages (from requests<
3,>=2.21.0->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs)
(1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admi\anacond
a3\envs\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboar
d<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (2020.6.20)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\admi\anaco
nda3\envs\tensorflow\lib\site-packages (from pyasn1-modules>=0.2.1->google
-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorfl
owjs) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0
->tensorflowjs) (3.1.0)
Building wheels for collected packages: PyInquirer
  Building wheel for PyInquirer (setup.py): started
  Building wheel for PyInquirer (setup.py): finished with status 'done'
  Created wheel for PyInquirer: filename=PyInquirer-1.0.3-py3-none-any.whl
size=32853 sha256=0207a5c58853051276a03cb9bba25e5fbe93e81fc76f29688068676a
3f50ae1e
```

```
  Stored in directory: c:\users\admi\appdata\local\pip\cache\wheels\89\3b
\7b\8b3cc8ac47137eabaeb6937a3ff0d33e78a12e2ba1e3ad4ba1
Successfully built PyInquirer
Installing collected packages: numpy, tensorflow-hub, prompt-toolkit, rege
x, PyInquirer, tensorflow-estimator, gast, tensorboard, scipy, keras-prepr
ocessing, astunparse, tensorflow-cpu, tensorflowjs
  Attempting uninstall: numpy
    Found existing installation: numpy 1.19.1
    Uninstalling numpy-1.19.1:


ERROR: tensorflow 2.0.0 has requirement tensorboard<2.1.0,>=2.0.0, but yo
u'll have tensorboard 2.3.0 which is incompatible.
ERROR: tensorflow 2.0.0 has requirement tensorflow-estimator<2.1.0,>=2.0.
0, but you'll have tensorflow-estimator 2.3.0 which is incompatible.
ERROR: jupyter-console 6.1.0 has requirement prompt-toolkit!=3.0.0,!=3.0.
1,<3.1.0,>=2.0.0, but you'll have prompt-toolkit 1.0.14 which is incompati
ble.
ERROR: ipython 7.16.1 has requirement prompt-toolkit!=3.0.0,!=3.0.1,<3.1.
0,>=2.0.0, but you'll have prompt-toolkit 1.0.14 which is incompatible.
ERROR: Could not install packages due to an EnvironmentError: [WinError 5]
Access is denied: 'c:\\users\\admi\\anaconda3\\envs\\tensorflow\\lib\\site
-packages\\~umpy\\core\\_multiarray_tests.cp37-win_amd64.pyd'
Consider using the `--user` option or check the permissions.
```

Convert the Model Note: Do not load a saved model and try to convert it. It will not work.

In [53]:

```
# create a directory to store the model files
os.mkdir('tfjs_dir')

# convert to Tensorflow.js
import tensorflowjs as tfjs

tfjs.converters.save_keras_model(model, 'tfjs_dir')
```

```
      Successfully uninstalled numpy-1.19.1

---------------------------------------------------------------------------
-
ModuleNotFoundError                       Traceback (most recent call las
t)
<ipython-input-53-1a12f1c8dcf6> in <module>
      3
      4 # convert to Tensorflow.js
----> 5 import tensorflowjs as tfjs
      6
      7 tfjs.converters.save_keras_model(model, 'tfjs_dir')

ModuleNotFoundError: No module named 'tensorflowjs'
```

In [ ]:

```
# check the the directory containing the model is available
%ls
```

In [ ]:

```
# view the files that make up the tensorflow.js model
os.listdir('tfjs_dir')
```

In [54]:

```
# Delete the image data directory we created if needed (uncomment the line below)
# shutil.rmtree('base_dir')
```

In [57]:

```

```

Out[57]:

```
Available line magics:
%alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %bookm
ark  %cd  %clear  %cls  %colors  %conda  %config  %connect_info  %copy  %d
dir  %debug  %dhist  %dirs  %doctest_mode  %echo  %ed  %edit  %env  %gui
%hist  %history  %killbgscripts  %ldir  %less  %load  %load_ext  %loadpy
%logoff  %logon  %logstart  %logstate  %logstop  %ls  %lsmagic  %macro  %m
agic  %matplotlib  %mkdir  %more  %notebook  %page  %pastebin  %pdb  %pdef
%pdoc  %pfile  %pinfo  %pinfo2  %pip  %popd  %pprint  %precision  %prun  %
psearch  %psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickref  %r
ecall  %rehashx  %reload_ext  %ren  %rep  %rerun  %reset  %reset_selective
%rmdir  %run  %save  %sc  %set_env  %store  %sx  %system  %tb  %time  %tim
eit  %unalias  %unload_ext  %who  %who_ls  %whos  %xdel  %xmode

Available cell magics:
%%!  %%HTML  %%SVG  %%bash  %%capture  %%cmd  %%debug  %%file  %%html  %%j
avascript  %%js  %%latex  %%markdown  %%perl  %%prun  %%pypy  %%python  %%
python2  %%python3  %%ruby  %%script  %%sh  %%svg  %%sx  %%system  %%time
%%timeit  %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

In [58]:

```
%%HTML
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js"></scri
pt>
```

In [64]:

```
# create a directory to store the model files
os.mkdir('tfjs_dir')

# convert to Tensorflow.js
from tensorflow import tensorflowjs as tfjs
# import tensorflowjs as tfjs

tfjs.converters.save_keras_model(model, 'tfjs_dir')
```

```
---------------------------------------------------------------------------
-
ImportError                               Traceback (most recent call las
t)
<ipython-input-64-34e41fc2b37f> in <module>
      3
      4 # convert to Tensorflow.js
----> 5 from tensorflow import tensorflowjs as tfjs
      6 # import tensorflowjs as tfjs
      7

ImportError: cannot import name 'tensorflowjs' from 'tensorflow' (C:\Users
\admi\anaconda3\envs\tensorflow\lib\site-packages\tensorflow\__init__.py)
```

In [66]:

```
!pip install tensorflowjs
```

```
ERROR: tensorflow 2.0.0 has requirement gast==0.2.2, but you'll have gast
0.3.3 which is incompatible.

Collecting tensorflowjs
  Using cached tensorflowjs-2.1.0-py3-none-any.whl (60 kB)
Processing c:\users\admi\appdata\local\pip\cache\wheels\89\3b\7b\8b3cc8ac4
7137eabaeb6937a3ff0d33e78a12e2ba1e3ad4ba1\pyinquirer-1.0.3-py3-none-any.wh
l
Requirement already satisfied: numpy<1.19.0,>=1.16.4 in c:\users\admi\anac
onda3\envs\tensorflow\lib\site-packages (from tensorflowjs) (1.18.5)
Requirement already satisfied: six>=1.12.0 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflowjs) (1.15.0)
Collecting tensorflow-hub==0.7.0
  Using cached tensorflow_hub-0.7.0-py2.py3-none-any.whl (89 kB)

ERROR: tensorflow 2.0.0 has requirement tensorboard<2.1.0,>=2.0.0, but yo
u'll have tensorboard 2.3.0 which is incompatible.
ERROR: tensorflow 2.0.0 has requirement tensorflow-estimator<2.1.0,>=2.0.
0, but you'll have tensorflow-estimator 2.3.0 which is incompatible.
ERROR: jupyter-console 6.1.0 has requirement prompt-toolkit!=3.0.0,!=3.0.
1,<3.1.0,>=2.0.0, but you'll have prompt-toolkit 1.0.14 which is incompati
ble.
ERROR: ipython 7.16.1 has requirement prompt-toolkit!=3.0.0,!=3.0.1,<3.1.
0,>=2.0.0, but you'll have prompt-toolkit 1.0.14 which is incompatible.
ERROR: Could not install packages due to an EnvironmentError: [WinError 5]
Access is denied: 'c:\\users\\admi\\anaconda3\\envs\\tensorflow\\lib\\site
-packages\\~cipy\\fft\\_pocketfft\\pypocketfft.cp37-win_amd64.pyd'
Consider using the `--user` option or check the permissions.
```

```
Requirement already satisfied: h5py>=2.8.0 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflowjs) (2.10.0)
Collecting tensorflow-cpu<3,>=2.1.0
  Using cached tensorflow_cpu-2.3.0-cp37-cp37m-win_amd64.whl (195.8 MB)
Collecting regex>=2016.11.21
  Using cached regex-2020.7.14-cp37-cp37m-win_amd64.whl (268 kB)
Requirement already satisfied: Pygments>=2.2.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from PyInquirer==1.0.3->tensorflowjs)
(2.6.1)
Collecting prompt-toolkit==1.0.14
  Using cached prompt_toolkit-1.0.14-py3-none-any.whl (248 kB)
Requirement already satisfied: protobuf>=3.4.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorflow-hub==0.7.0->tensorflow
js) (3.12.3)
Requirement already satisfied: absl-py>=0.7.0 in c:\users\admi\anaconda3\e
nvs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflo
wjs) (0.9.0)
Requirement already satisfied: google-pasta>=0.1.8 in c:\users\admi\anacon
da3\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tens
orflowjs) (0.2.0)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\admi\anaconda3\en
vs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflow
js) (1.27.2)
Collecting astunparse==1.6.3
  Using cached astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensor
flowjs) (3.1.0)
Collecting tensorboard<3,>=2.3.0
  Using cached tensorboard-2.3.0-py3-none-any.whl (6.8 MB)
Requirement already satisfied: wrapt>=1.11.1 in c:\users\admi\anaconda3\en
vs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflow
js) (1.12.1)
Requirement already satisfied: wheel>=0.26 in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorflowj
s) (0.34.2)
Collecting tensorflow-estimator<2.4.0,>=2.3.0
  Using cached tensorflow_estimator-2.3.0-py2.py3-none-any.whl (459 kB)
Collecting scipy==1.4.1
  Using cached scipy-1.4.1-cp37-cp37m-win_amd64.whl (30.9 MB)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorflow-cpu<3,>=2.1.0->tensorf
lowjs) (1.1.0)
Collecting gast==0.3.3
  Using cached gast-0.3.3-py2.py3-none-any.whl (9.7 kB)
Collecting keras-preprocessing<1.2,>=1.1.1
  Using cached Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Requirement already satisfied: wcwidth in c:\users\admi\anaconda3\envs\ten
sorflow\lib\site-packages (from prompt-toolkit==1.0.14->PyInquirer==1.0.3-
>tensorflowjs) (0.2.5)
Requirement already satisfied: setuptools in c:\users\admi\anaconda3\envs
\tensorflow\lib\site-packages (from protobuf>=3.4.0->tensorflow-hub==0.7.0
->tensorflowjs) (49.2.0.post20200714)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\admi\anacon
da3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorf
low-cpu<3,>=2.1.0->tensorflowjs) (2.24.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\user
s\admi\anaconda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=
2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (0.4.1)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorflo
```

```
w-cpu<3,>=2.1.0->tensorflowjs) (0.16.1)
Requirement already satisfied: google-auth<2,>=1.6.3 in c:\users\admi\anac
onda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tenso
rflow-cpu<3,>=2.1.0->tensorflowjs) (1.17.2)
Requirement already satisfied: markdown>=2.6.8 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.0->tensorflow
-cpu<3,>=2.1.0->tensorflowjs) (3.1.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\a
dmi\anaconda3\envs\tensorflow\lib\site-packages (from tensorboard<3,>=2.3.
0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (1.6.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\admi\anaconda3\envs\tensorflow\lib\site-packages (from requests<
3,>=2.21.0->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs)
(1.24.3)
Requirement already satisfied: idna<3,>=2.5 in c:\users\admi\anaconda3\env
s\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=
2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admi\anacond
a3\envs\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboar
d<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\admi\anaconda
3\envs\tensorflow\lib\site-packages (from requests<3,>=2.21.0->tensorboard
<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (3.0.4)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\admi\a
naconda3\envs\tensorflow\lib\site-packages (from google-auth-oauthlib<0.5,
>=0.4.1->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs)
(1.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\admi\anac
onda3\envs\tensorflow\lib\site-packages (from google-auth<2,>=1.6.3->tenso
rboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (0.2.7)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\admi\ana
conda3\envs\tensorflow\lib\site-packages (from google-auth<2,>=1.6.3->tens
orboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowjs) (4.1.0)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3" in
c:\users\admi\anaconda3\envs\tensorflow\lib\site-packages (from google-aut
h<2,>=1.6.3->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorflowj
s) (4.0)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\admi\anaconda3
\envs\tensorflow\lib\site-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0
->tensorflowjs) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\admi\anaco
nda3\envs\tensorflow\lib\site-packages (from pyasn1-modules>=0.2.1->google
-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->tensorflow-cpu<3,>=2.1.0->tensorfl
owjs) (0.4.8)
Installing collected packages: regex, prompt-toolkit, PyInquirer, tensorfl
ow-hub, astunparse, tensorboard, tensorflow-estimator, scipy, gast, keras-
preprocessing, tensorflow-cpu, tensorflowjs
  Attempting uninstall: prompt-toolkit
    Found existing installation: prompt-toolkit 3.0.5
    Uninstalling prompt-toolkit-3.0.5:
      Successfully uninstalled prompt-toolkit-3.0.5
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.2.1
    Uninstalling tensorboard-2.2.1:
      Successfully uninstalled tensorboard-2.2.1
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.0.0
    Uninstalling tensorflow-estimator-2.0.0:
      Successfully uninstalled tensorflow-estimator-2.0.0
  Attempting uninstall: scipy
```

```
Found existing installation: scipy 1.5.0
Uninstalling scipy-1.5.0:
  Successfully uninstalled scipy-1.5.0
```