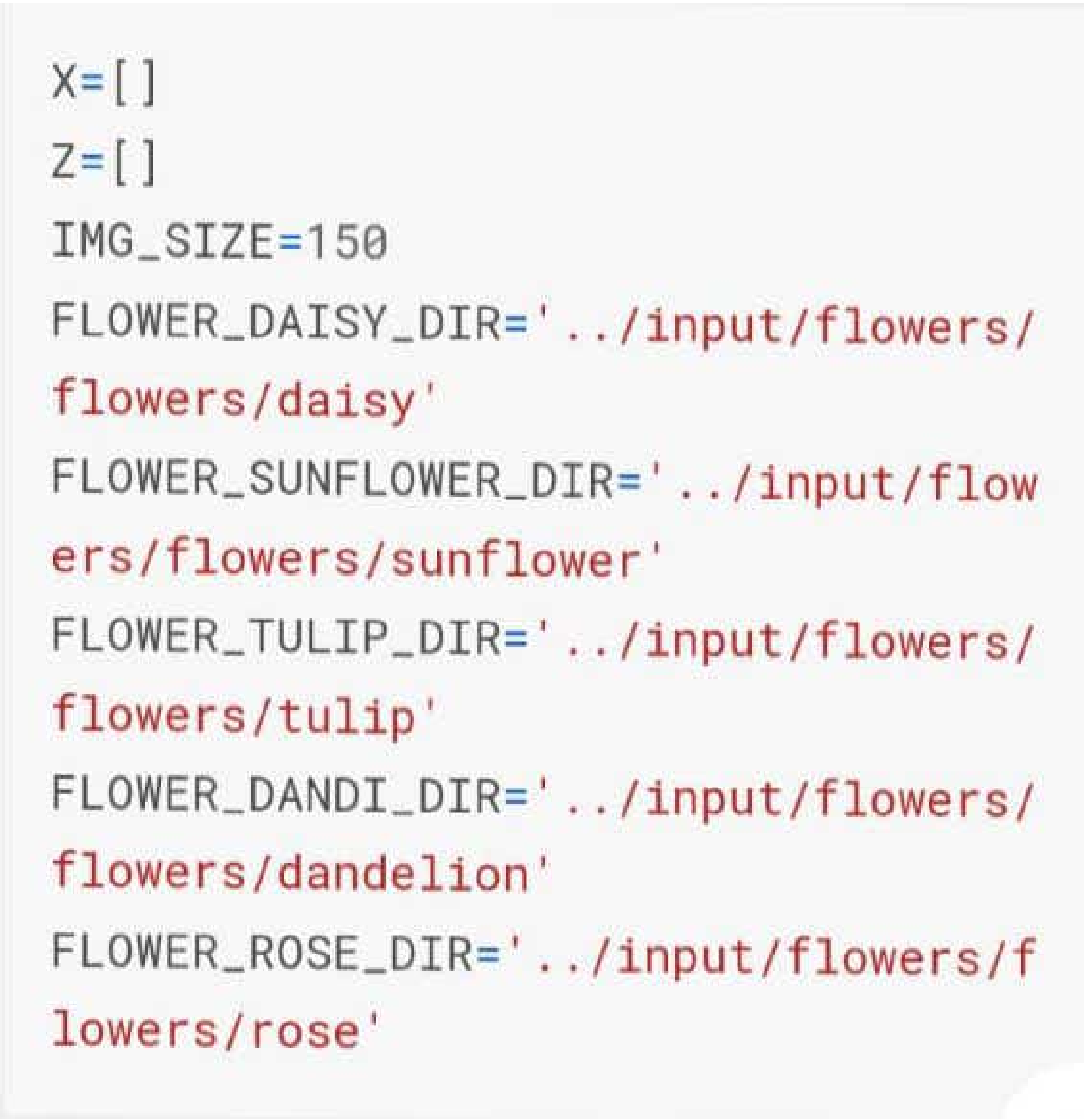
```
# Ignore  the warnings
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')

# data visualisation and manipulati
on
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

#configure
# sets matplotlib to inline and dis
plays graphs below the corresspondi
ng cell.
%matplotlib inline
style.use('fivethirtyeight')
sns.set(style='whitegrid',color_cod
es=True)
```

```
#model selection
from sklearn.model_selection import
train_test_split
from sklearn.model_selection import
KFold
from sklearn.metrics import accurac
y_score,precision_score,recall_scor
e,confusion_matrix,roc_curve,roc_au
c_score
from sklearn.model_selection import
GridSearchCV
from sklearn.preprocessing import L
abelEncoder

#preprocess.
from keras.preprocessing.image impo
rt ImageDataGenerator

#dl libraraies
from keras import backend as K
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam,S
from keras.optimizers import Adam,S
GD,Adagrad,Adadelta,RMSprop
from keras.utils import to_categori
cal

# specifically for cnn
from keras.layers import Dropout, F
latten,Activation
from keras.layers import Conv2D, Ma
xPooling2D, BatchNormalization

import tensorflow as tf
import random as rn

# specifically for manipulating zip
ped images and getting numpy arrays
of pixel values of images.
import cv2
import numpy as np
from tqdm import tqdm
import os
from random import shuffle
from zipfile import ZipFile
```



```
X=[]
Z=[]
IMG_SIZE=150
FLOWER_DAISY_DIR='../input/flowers/
flowers/daisy'
FLOWER_SUNFLOWER_DIR='../input/flow
ers/flowers/sunflower'
FLOWER_TULIP_DIR='../input/flowers/
flowers/tulip'
FLOWER_DANDI_DIR='../input/flowers/
flowers/dandelion'
FLOWER_ROSE_DIR='../input/flowers/f
lowers/rose'
```

```
def assign_label(img,flower_type):
    return flower_type
```

In [5]:

```
def make_train_data(flower_type,DIR):
    for img in tqdm(os.listdir(DIR)):
        label=assign_label(img,flower_type)
        path = os.path.join(DIR,img)
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))

        X.append(np.array(img))
        Z.append(str(label))
```

```
make_train_data('Tulip',FLOWER_TULI
P_DIR)
print(len(X))
```

```
100%|██████████| 984/984 [0
0:04<00:00, 224.01it/s]


2487
```

In [9]:

```
make_train_data('Dandelion',FLOWER_
DANDI_DIR)
print(len(X))
```

```
  9%|█          | 97/1055 [00:00<0
0:04, 235.89it/s]
```

```python
make_train_data('Daisy',FLOWER_DAIS
Y_DIR)
print(len(X))
```

```
100%|██████████| 769/769 [0
0:03<00:00, 215.70it/s]
```

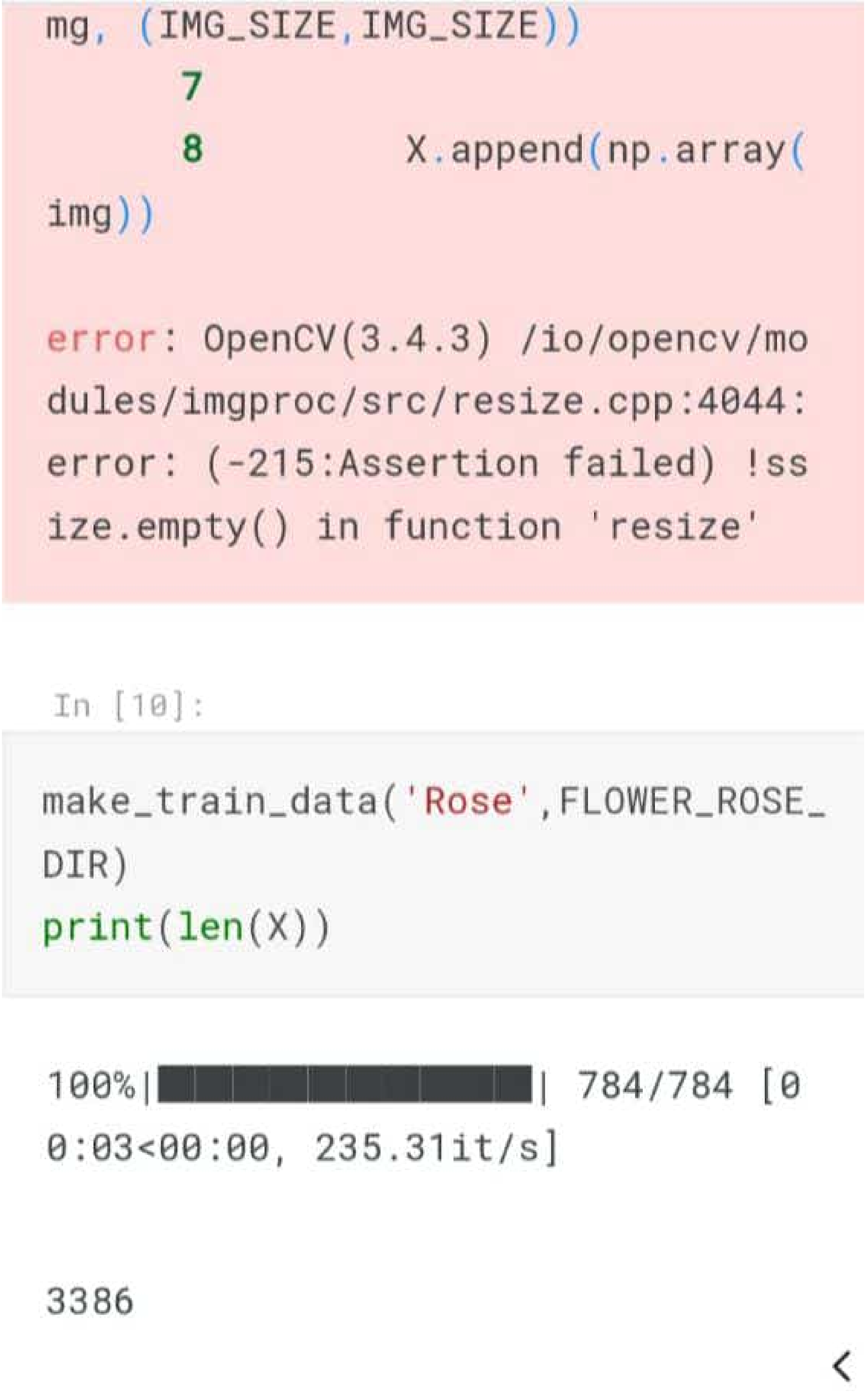
```
769
```

In [7]:

```python
make_train_data('Sunflower',FLOWER_
SUNFLOWER_DIR)
print(len(X))
```

```
100%|██████████| 734/734 [0
0:03<00:00, 206.81it/s]
```

```
------------------------------------------
------------------------------------------
--------
error
Traceback (most recent call last)
<ipython-input-9-95c78ead0c98> in
<module>
----> 1 make_train_data('Dandelio
n',FLOWER_DANDI_DIR)
      2 print(len(X))

<ipython-input-5-001b1f747236> in
make_train_data(flower_type, DIR)
      4           path = os.path.joi
n(DIR,img)
      5           img = cv2.imread(p
ath,cv2.IMREAD_COLOR)
----> 6           img = cv2.resize(i
mg, (IMG_SIZE,IMG_SIZE))
      7
      8           X.append(np.array(
img))
```



```
mg, (IMG_SIZE,IMG_SIZE))
      7
      8             X.append(np.array(
img))

error: OpenCV(3.4.3) /io/opencv/mo
dules/imgproc/src/resize.cpp:4044:
error: (-215:Assertion failed) !ss
ize.empty() in function 'resize'
```

In [10]:

```
make_train_data('Rose',FLOWER_ROSE_
DIR)
print(len(X))
```

```
100%|██████████| 784/784 [0
0:03<00:00, 235.31it/s]


3386
```