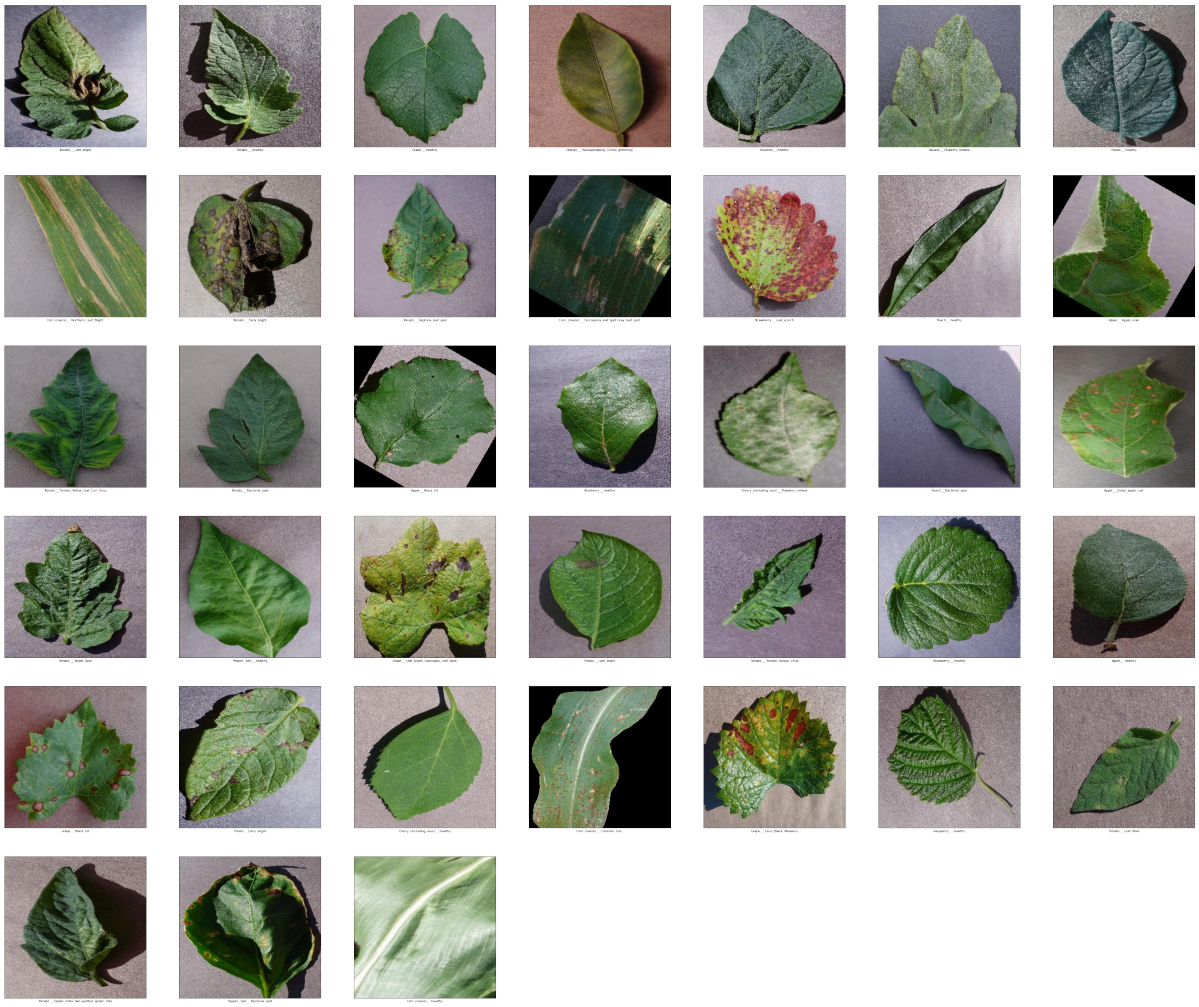
In [1]:
```
#IMPORTING LIBRARIES
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
```

In [2]:
```
path='/kaggle/input/new-plant-diseases-dataset/New Plant Diseases Dataset(Augmented
plt.figure(figsize=(70,70))
count=0
plant_names=[]
total_images=0
for i in os.listdir(path):
  count+=1
  plant_names.append(i)
  plt.subplot(7,7,count)

  images_path=os.listdir(path+"/"+i)
  print("Number of images of "+i+":",len(images_path),"||",end=" ")
  total_images+=len(images_path)

  image_show=plt.imread(path+"/"+i+"/"+images_path[0])

  plt.imshow(image_show)
  plt.xlabel(i)

  plt.xticks([])
  plt.yticks([])


print("Total number of images we have",total_images)
```

```
Number of images of Tomato___Late_blight: 1851 || Number of images of Tomato___hea
lthy: 1926 || Number of images of Grape___healthy: 1692 || Number of images of Ora
nge___Haunglongbing_(Citrus_greening): 2010 || Number of images of Soybean___healt
hy: 2022 || Number of images of Squash___Powdery_mildew: 1736 || Number of images
of Potato___healthy: 1824 || Number of images of Corn_(maize)___Northern_Leaf_Blig
ht: 1908 || Number of images of Tomato___Early_blight: 1920 || Number of images of
Tomato___Septoria_leaf_spot: 1745 || Number of images of Corn_(maize)___Cercospora
_leaf_spot Gray_leaf_spot: 1642 || Number of images of Strawberry___Leaf_scorch: 1
774 || Number of images of Peach___healthy: 1728 || Number of images of Apple___Ap
ple_scab: 2016 || Number of images of Tomato___Tomato_Yellow_Leaf_Curl_Virus: 1961
|| Number of images of Tomato___Bacterial_spot: 1702 || Number of images of Apple_
__Black_rot: 1987 || Number of images of Blueberry___healthy: 1816 || Number of im
ages of Cherry_(including_sour)___Powdery_mildew: 1683 || Number of images of Peac
h___Bacterial_spot: 1838 || Number of images of Apple___Cedar_apple_rust: 1760 ||
Number of images of Tomato___Target_Spot: 1827 || Number of images of Pepper,_bell
___healthy: 1988 || Number of images of Grape___Leaf_blight_(Isariopsis_Leaf_Spo
t): 1722 || Number of images of Potato___Late_blight: 1939 || Number of images of
Tomato___Tomato_mosaic_virus: 1790 || Number of images of Strawberry___healthy: 18
24 || Number of images of Apple___healthy: 2008 || Number of images of Grape___Bla
ck_rot: 1888 || Number of images of Potato___Early_blight: 1939 || Number of image
s of Cherry_(including_sour)___healthy: 1826 || Number of images of Corn_(maize)__
_Common_rust_: 1907 || Number of images of Grape___Esca_(Black_Measles): 1920 || N
umber of images of Raspberry___healthy: 1781 || Number of images of Tomato___Leaf_
Mold: 1882 || Number of images of Tomato___Spider_mites Two-spotted_spider_mite: 1
741 || Number of images of Pepper,_bell___Bacterial_spot: 1913 || Number of images
of Corn_(maize)___healthy: 1859 || Total number of images we have 70295
```



In [3]:
```
print(plant_names)
print(len(plant_names))
```

```
['Tomato___Late_blight', 'Tomato___healthy', 'Grape___healthy', 'Orange___Haunglon
gbing_(Citrus_greening)', 'Soybean___healthy', 'Squash___Powdery_mildew', 'Potato_
__healthy', 'Corn_(maize)___Northern_Leaf_Blight', 'Tomato___Early_blight', 'Tomat
o___Septoria_leaf_spot', 'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot', 'St
rawberry___Leaf_scorch', 'Peach___healthy', 'Apple___Apple_scab', 'Tomato___Tomato
_Yellow_Leaf_Curl_Virus', 'Tomato___Bacterial_spot', 'Apple___Black_rot', 'Blueber
ry___healthy', 'Cherry_(including_sour)___Powdery_mildew', 'Peach___Bacterial_spo
t', 'Apple___Cedar_apple_rust', 'Tomato___Target_Spot', 'Pepper,_bell___healthy',
'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)', 'Potato___Late_blight', 'Tomato___To
mato_mosaic_virus', 'Strawberry___healthy', 'Apple___healthy', 'Grape___Black_ro
t', 'Potato___Early_blight', 'Cherry_(including_sour)___healthy', 'Corn_(maize)___
Common_rust_', 'Grape___Esca_(Black_Measles)', 'Raspberry___healthy', 'Tomato___Le
af_Mold', 'Tomato___Spider_mites Two-spotted_spider_mite', 'Pepper,_bell___Bacteri
al_spot', 'Corn_(maize)___healthy']
38
```

IMPORTING NECESSARY LIBRARIES FOR TRAINING OF MODEL

In [4]:
```
import tensorflow
from tensorflow import keras
from keras.models import Sequential,load_model,Model
from keras.layers import Conv2D,MaxPool2D,AveragePooling2D,Dense,Flatten,ZeroPaddir
from keras.optimizers import SGD
from keras.initializers import glorot_uniform
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint,EarlyStopping,ReduceLROnPlateau
```

In [5]:
```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input
```

In [6]:
```
base_model_tf=ResNet50(include_top=False,weights='imagenet',input_shape=(224,224,3)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-application
s/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94773248/94765736 [==============================] - 3s 0us/step
```

In [7]:
```
#Model building
base_model_tf.trainable=False

pt=Input(shape=(224,224,3))
func=tensorflow.cast(pt,tensorflow.float32)
x=preprocess_input(func) #This function used to zero-center each color channel wrt
model_resnet=base_model_tf(x,training=False)
model_resnet=GlobalAveragePooling2D()(model_resnet)
model_resnet=Dense(128,activation='relu')(model_resnet)
model_resnet=Dense(64,activation='relu')(model_resnet)
model_resnet=Dense(38,activation='softmax')(model_resnet)


model_main=Model(inputs=pt,outputs=model_resnet)
model_main.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
tf.cast (TFOpLambda)         (None, 224, 224, 3)       0
_________________________________________________________________
tf.__operators__.getitem (Sl (None, 224, 224, 3)       0
_________________________________________________________________
tf.nn.bias_add (TFOpLambda)  (None, 224, 224, 3)       0
_________________________________________________________________
resnet50 (Functional)        (None, 7, 7, 2048)        23587712
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 128)               262272
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 38)                2470
=================================================================
Total params: 23,860,710
Trainable params: 272,998
Non-trainable params: 23,587,712
_________________________________________________________________
```

In [8]:
```
#Image augmentation
train_datagen= ImageDataGenerator(shear_range=0.2,zoom_range=0.2,horizontal_flip=Fa
                                  ,fill_mode='nearest',width_shift_range=0.2,height

val_datagen=ImageDataGenerator()

path_train='/kaggle/input/new-plant-diseases-dataset/New Plant Diseases Dataset(Aug

path_valid='/kaggle/input/new-plant-diseases-dataset/New Plant Diseases Dataset(Aug

train= train_datagen.flow_from_directory(directory=path_train,batch_size=32,target_
                                         color_mode='rgb',class_mode='categorical',
```

```
valid=val_datagen.flow_from_directory(directory=path_valid,batch_size=32,target_siz
```

```
Found 70295 images belonging to 38 classes.
Found 17572 images belonging to 38 classes.
```

In [9]:
```
#CallBacks
es=EarlyStopping(monitor='val_accuracy',verbose=1,patience=7,mode='auto')
mc=ModelCheckpoint(filepath='/content',monitor='val_accuracy',verbose=1,save_best_c
lr=ReduceLROnPlateau(monitor='val_accuracy',verbose=1,patience=5,min_lr=0.001)
```

In [10]:
```
model_main.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accur
```

In [11]:
```
#Training
model_main.fit(train,validation_data=valid,epochs=30,steps_per_epoch=200,verbose=1,
```

```
Epoch 1/30
200/200 [==============================] - 220s 1s/step - loss: 2.3978 - accuracy:
0.3754 - val_loss: 0.5253 - val_accuracy: 0.8482

Epoch 00001: val_accuracy improved from -inf to 0.84822, saving model to /content
Epoch 2/30
200/200 [==============================] - 142s 709ms/step - loss: 0.5075 - accura
cy: 0.8560 - val_loss: 0.3299 - val_accuracy: 0.8987

Epoch 00002: val_accuracy improved from 0.84822 to 0.89870, saving model to /conte
nt
Epoch 3/30
200/200 [==============================] - 136s 682ms/step - loss: 0.3505 - accura
cy: 0.8951 - val_loss: 0.3256 - val_accuracy: 0.8984

Epoch 00003: val_accuracy did not improve from 0.89870
Epoch 4/30
200/200 [==============================] - 135s 674ms/step - loss: 0.2756 - accura
cy: 0.9120 - val_loss: 0.2372 - val_accuracy: 0.9228

Epoch 00004: val_accuracy improved from 0.89870 to 0.92283, saving model to /conte
nt
Epoch 5/30
200/200 [==============================] - 135s 677ms/step - loss: 0.2562 - accura
cy: 0.9225 - val_loss: 0.2561 - val_accuracy: 0.9165

Epoch 00005: val_accuracy did not improve from 0.92283
Epoch 6/30
200/200 [==============================] - 133s 668ms/step - loss: 0.2301 - accura
cy: 0.9233 - val_loss: 0.2279 - val_accuracy: 0.9254

Epoch 00006: val_accuracy improved from 0.92283 to 0.92539, saving model to /conte
nt
Epoch 7/30
200/200 [==============================] - 137s 688ms/step - loss: 0.1726 - accura
cy: 0.9434 - val_loss: 0.2289 - val_accuracy: 0.9231

Epoch 00007: val_accuracy did not improve from 0.92539
Epoch 8/30
200/200 [==============================] - 134s 669ms/step - loss: 0.1967 - accura
cy: 0.9345 - val_loss: 0.1884 - val_accuracy: 0.9354

Epoch 00008: val_accuracy improved from 0.92539 to 0.93541, saving model to /conte
nt
Epoch 9/30
200/200 [==============================] - 132s 660ms/step - loss: 0.2030 - accura
cy: 0.9324 - val_loss: 0.1711 - val_accuracy: 0.9433

Epoch 00009: val_accuracy improved from 0.93541 to 0.94332, saving model to /conte
nt
Epoch 10/30
200/200 [==============================] - 126s 629ms/step - loss: 0.1908 - accura
cy: 0.9399 - val_loss: 0.2269 - val_accuracy: 0.9265

Epoch 00010: val_accuracy did not improve from 0.94332
Epoch 11/30
200/200 [==============================] - 126s 632ms/step - loss: 0.1575 - accura
cy: 0.9496 - val_loss: 0.1583 - val_accuracy: 0.9468

Epoch 00011: val_accuracy improved from 0.94332 to 0.94679, saving model to /conte
nt
Epoch 12/30
200/200 [==============================] - 127s 634ms/step - loss: 0.1747 - accura
cy: 0.9420 - val_loss: 0.1582 - val_accuracy: 0.9491
```

```
Epoch 00012: val_accuracy improved from 0.94679 to 0.94912, saving model to /conte
nt
Epoch 13/30
200/200 [==============================] - 129s 644ms/step - loss: 0.1450 - accura
cy: 0.9511 - val_loss: 0.2017 - val_accuracy: 0.9339

Epoch 00013: val_accuracy did not improve from 0.94912
Epoch 14/30
200/200 [==============================] - 126s 629ms/step - loss: 0.1696 - accura
cy: 0.9428 - val_loss: 0.1917 - val_accuracy: 0.9391

Epoch 00014: val_accuracy did not improve from 0.94912
Epoch 15/30
200/200 [==============================] - 127s 638ms/step - loss: 0.1541 - accura
cy: 0.9508 - val_loss: 0.1157 - val_accuracy: 0.9610

Epoch 00015: val_accuracy improved from 0.94912 to 0.96102, saving model to /conte
nt
Epoch 16/30
200/200 [==============================] - 125s 625ms/step - loss: 0.1268 - accura
cy: 0.9563 - val_loss: 0.1425 - val_accuracy: 0.9532

Epoch 00016: val_accuracy did not improve from 0.96102
Epoch 17/30
200/200 [==============================] - 126s 631ms/step - loss: 0.1400 - accura
cy: 0.9519 - val_loss: 0.1441 - val_accuracy: 0.9535

Epoch 00017: val_accuracy did not improve from 0.96102
Epoch 18/30
200/200 [==============================] - 126s 630ms/step - loss: 0.1333 - accura
cy: 0.9567 - val_loss: 0.1860 - val_accuracy: 0.9412

Epoch 00018: val_accuracy did not improve from 0.96102
Epoch 19/30
200/200 [==============================] - 127s 635ms/step - loss: 0.1140 - accura
cy: 0.9624 - val_loss: 0.1360 - val_accuracy: 0.9542

Epoch 00019: val_accuracy did not improve from 0.96102
Epoch 20/30
200/200 [==============================] - 126s 629ms/step - loss: 0.1033 - accura
cy: 0.9646 - val_loss: 0.1555 - val_accuracy: 0.9519

Epoch 00020: val_accuracy did not improve from 0.96102

Epoch 00020: ReduceLROnPlateau reducing learning rate to 0.001.
Epoch 21/30
200/200 [==============================] - 125s 628ms/step - loss: 0.1483 - accura
cy: 0.9517 - val_loss: 0.2028 - val_accuracy: 0.9344

Epoch 00021: val_accuracy did not improve from 0.96102
Epoch 22/30
200/200 [==============================] - 126s 633ms/step - loss: 0.1181 - accura
cy: 0.9620 - val_loss: 0.1714 - val_accuracy: 0.9446

Epoch 00022: val_accuracy did not improve from 0.96102
Epoch 00022: early stopping
```

Out[11]:
```
<tensorflow.python.keras.callbacks.History at 0x7bf901737ad0>
```

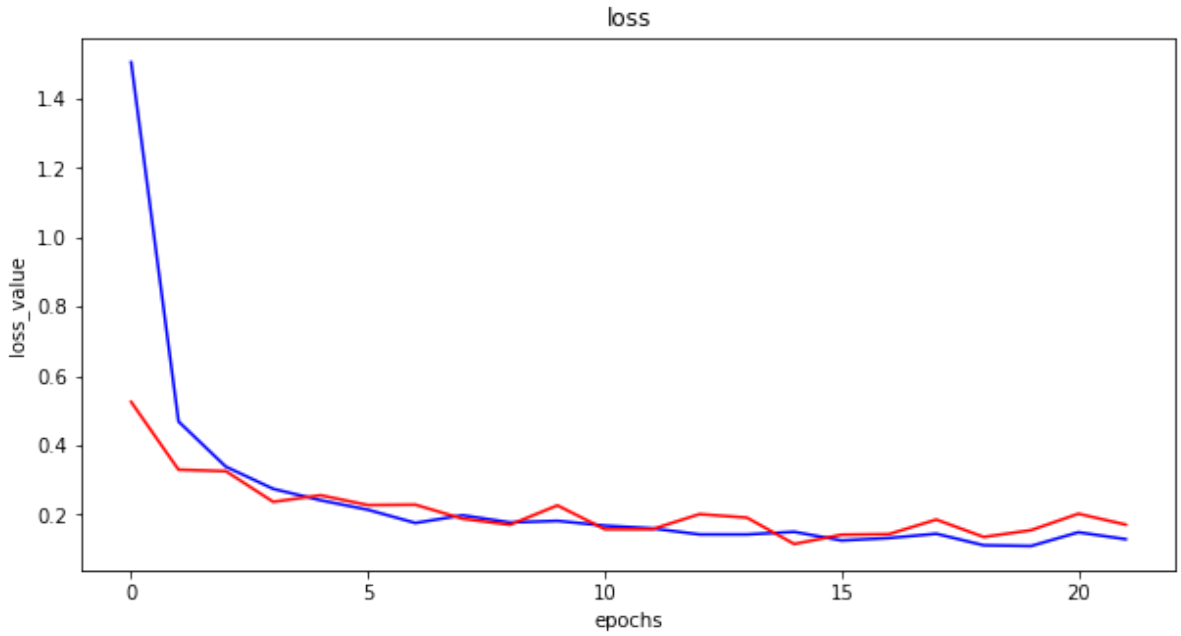
In [12]:
```
model_main.save("RESNET50_PLANT_DISEASE.h5")
```

In [13]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import cv2
from PIL import Image
```

In [14]:
```
plt.figure(figsize=(10,5))
plt.plot(model_main.history.history['loss'],'-bx')
plt.plot(model_main.history.history['val_loss'],'-rx')
plt.xlabel("epochs")
plt.ylabel("loss_value")
plt.title("loss")
```

Out[14]:
```
Text(0.5, 1.0, 'loss')
```



In [15]:
```
plt.figure(figsize=(10,5))
plt.plot(model_main.history.history['accuracy'],color='b',label='Training accuracy'
plt.plot(model_main.history.history['val_accuracy'],color='r',label='Validation acc
plt.xlabel("epochs")
plt.ylabel("accuracy")
plt.title("accuracy graph")
```

Out[15]:
```
Text(0.5, 1.0, 'accuracy graph')
```