查找并且处理异常图片

定义训练集目录

In [11]:

```
import os, shutil, platform
import numpy as np
import pandas as pd
```

In [12]:

```
train_folder = "train"
train_filenames = os.listdir(train_folder)
bad_image_names = []
top = 50

def append_to_bad_file_names_if_need(preds, file_name,top):
    append = True
    for i in range(0, top):
        type_code = preds[i][0]
        if is_car_or_dog_image(type_code):
            append = False
    if append:
        bad_image_names.append(file_name)
```

定义 Imagenet 数据集中的猫狗类型

In [13]:

```
imagenet_dog_types = [
 'n02085620','n02085782','n02085936','n02086079'
,'n02086240','n02086646','n02086910','n02087046'
,'n02087394','n02088094','n02088238','n02088364'
,'n02088466','n02088632','n02089078','n02089867'
,'n02089973','n02090379','n02090622','n02090721'
,'n02091032','n02091134','n02091244','n02091467'
,'n02091635','n02091831','n02092002','n02092339'
,'n02093256','n02093428','n02093647','n02093754'
,'n02093859','n02093991','n02094114','n02094258'
,'n02094433','n02095314','n02095570','n02095889'
,'n02096051','n02096177','n02096294','n02096437'
,'n02096585','n02097047','n02097130','n02097209'
,'n02097298','n02097474','n02097658','n02098105'
,'n02098286','n02098413','n02099267','n02099429'
,'n02099601','n02099712','n02099849','n02100236'
,'n02100583','n02100735','n02100877','n02101006'
,'n02101388','n02101556','n02102040','n02102177'
,'n02102318','n02102480','n02102973','n02104029'
,'n02104365','n02105056','n02105162','n02105251'
,'n02105412','n02105505','n02105641','n02105855'
,'n02106030','n02106166','n02106382','n02106550'
,'n02106662','n02107142','n02107312','n02107574'
,'n02107683','n02107908','n02108000','n02108089'
,'n02108422','n02108551','n02108915','n02109047'
,'n02109525','n02109961','n02110063','n02110185'
,'n02110341','n02110627','n02110806','n02110958'
,'n02111129','n02111277','n02111500','n02111889'
,'n02112018','n02112137','n02112350','n02112706'
,'n02113023','n02113186','n02113624','n02113712'
,'n02113799','n02113978']

imagenet_cat_types =[
'n02123045','n02123159','n02123394','n02123597'
,'n02124075','n02125311','n02127052']



def is_car_or_dog_image(type_code):
    if type_code in imagenet_dog_types or type_code in imagenet_cat_types:
        return True
    else:
        return False
```

ResNet50 预测非猫非狗

建立模型

In [14]:

```
from keras.preprocessing import image
from keras.applications.resnet50 import ResNet50, preprocess_input, decode_predictions

target_size_resnet = (224, 224)
model_resnet550 = ResNet50(weights='imagenet')
```

预测非猫非狗图片

In [15]:

```
def prepare_input(file_name):
    path = "train/" + file_name
    img = image.load_img(path, target_size= target_size_resnet)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return x


for file_name in train_filenames:
    x = prepare_input(file_name)
    preds = model_resnet550.predict(x)
    pred_results = decode_predictions(preds, top)[0]
    append_to_bad_file_names_if_need(pred_results, file_name,top)

print ("done!!!")
```

done!!!

In [16]:

```
print(len(bad_image_names))
```

45

Xception预测非猫非狗

导入类库

In [19]:

```
from keras.preprocessing import image
from keras.applications.xception import Xception, preprocess_input, decode_predictions

target_size_xception = (299, 299)
model_xception = Xception(weights='imagenet')
```

预测非猫非狗图片

In [20]:

```
def prepare_input(file_name):
    path = "train/" + file_name
    img = image.load_img(path, target_size= target_size_xception)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return x


for file_name in train_filenames:
    x = prepare_input(file_name)
    preds = model_xception.predict(x)
    pred_results = decode_predictions(preds, top)[0]
    append_to_bad_file_names_if_need(pred_results, file_name,top)

print ("done!!!")
```

```
done!!!
```

In [21]:

```
print(len(bad_image_names))
```

```
79
```

InceptionResNetV2预测非猫非狗

导入类库

In [22]:

```
from keras.preprocessing import image
from keras.applications.inception_resnet_v2 import InceptionResNetV2, preprocess_input, decode_pred

target_size_irv2 = (299, 299)
model_irv2 = InceptionResNetV2(weights='imagenet')
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/down
load/v0.7/inception_resnet_v2_weights_tf_dim_ordering_tf_kernels.h5 (https://github.
com/fchollet/deep-learning-models/releases/download/v0.7/inception_resnet_v2_weights
_tf_dim_ordering_tf_kernels.h5)
225214464/225209952 [==============================] - 3s 0us/step
```

预测非猫非狗图片

In [23]:

```
def prepare_input(file_name):
    path = "train/" + file_name
    img = image.load_img(path, target_size= target_size_irv2)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return x

for file_name in train_filenames:
    x = prepare_input(file_name)
    preds = model_irv2.predict(x)
    pred_results = decode_predictions(preds, top)[0]
    append_to_bad_file_names_if_need(pred_results, file_name,top)

print ("done!!!")
```

done!!!

In [24]:

```
print("预测完成！！！")
```

预测完成！！！

In [25]:

```
print(len(bad_image_names))
```

115

合并列表中的重复项

In [27]:

```
bad_images_set = sorted(set(bad_image_names),key=bad_image_names.index)
```

In [28]:

```
print("合并重复项目完成！！！")
```

合并重复项目完成！！！

In [29]:

```
print(len(bad_images_set))
```

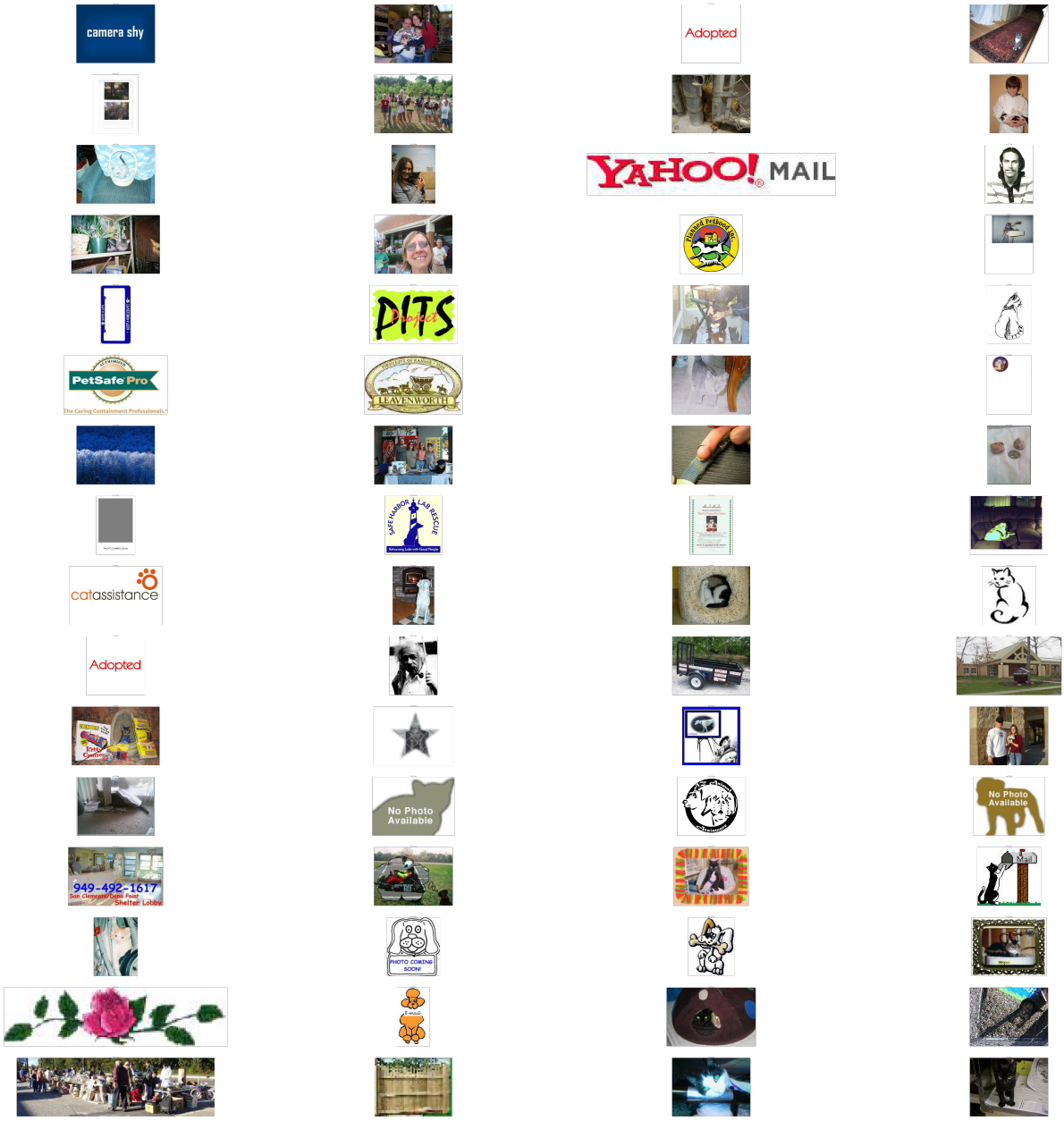
64

可视化

In [47]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

plt.figure(figsize=(256, 256))
for i, img_file_name in enumerate(bad_images_set):
    plt.subplot(16, 4, i+1)
    plt.title(img_file_name)
    plt.axis('off')
    img_path = "./train/" + img_file_name
    img_obj = mpimg.imread(img_path)
    plt.imshow(img_obj, interpolation="nearest")
```



In [ ]: