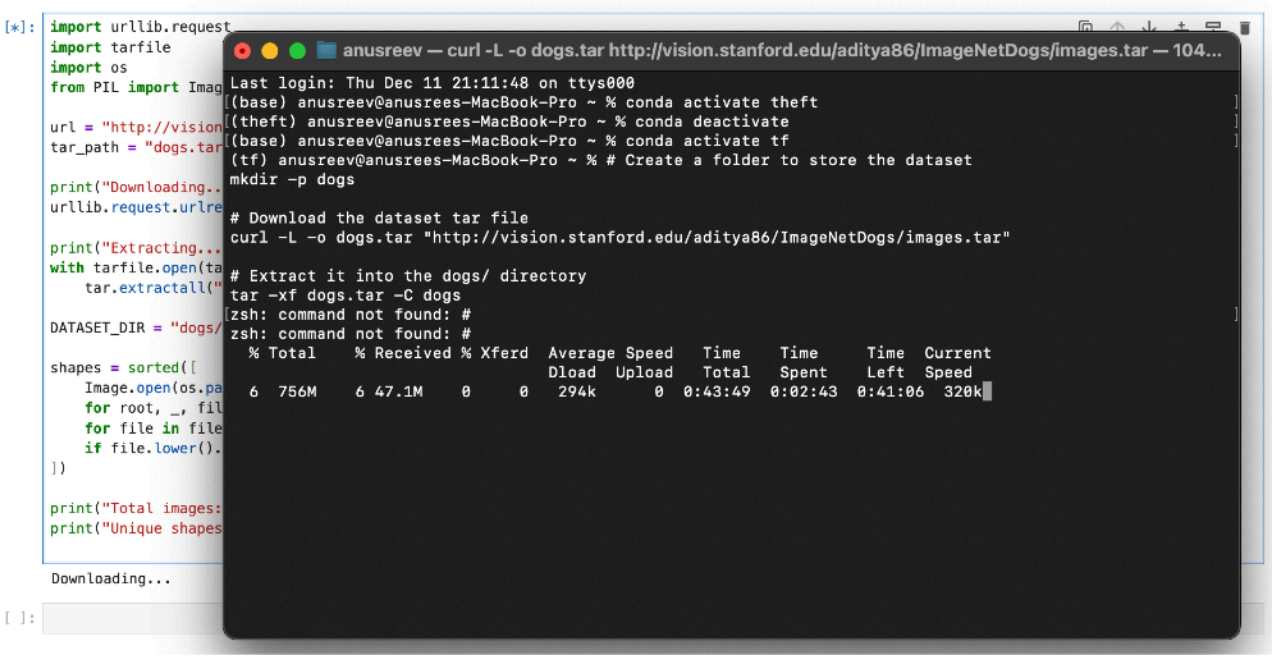
25MML0014
ANUSREE V

EDGE INTELLIGENCE

The main problem faced is loading dataset because it contains large MBs. So I just downloaded in terminal .it took 28 minutes for loading as it is huge data.



os.walk is used to automatically scan a folder.
It finds all files and subfolders without manual checking.
This makes loading large image datasets simple and efficient.

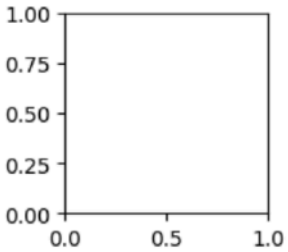
```
4]: all_shapes = []


    count = 0

    for root, _, files in os.walk(DATASET_DIR):
        for file in files:
            if file.lower().endswith((".jpg", ".png", ".jpeg")):
                img_path = os.path.join(root, file)
                img = Image.open(img_path)
                shape = img.size
                all_shapes.append(shape)
                count += 1
                print(f"Image {count}: {shape}")

    print("\nTotal images found:", count)
    print("Unique image shapes:", set(all_shapes))
```

```
Image 1: (448, 400)
Image 2: (375, 500)
Image 3: (500, 375)
Image 4: (333, 500)
Image 5: (500, 333)
Image 6: (375, 500)
Image 7: (375, 500)
Image 8: (500, 375)
Image 9: (375, 500)
Image 10: (500, 375)
Image 11: (280, 252)
Image 12: (375, 500)
Image 13: (500, 375)
Image 14: (500, 375)
Image 15: (500, 375)
Image 16: (500, 400)
Image 17: (375, 500)
Image 18: (500, 375)
```

At first I give .png only its showing index out of range

```
for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".png")):
            img_path = os.path.join(root, file)
            img = Image.open(img_path).convert("RGB")

            image_list.append(np.array(img))
            shape_list.append(img.size)

            count += 1
            if count == limit:
                break
    if count == limit:
        break

plt.figure(figsize=(10, 10))
for i in range(limit):
    plt.subplot(5, 5, i+1)
    plt.imshow(image_list[i])
    plt.title(str(shape_list[i]))
    plt.axis("off")

plt.suptitle("First 25 Images & Their Shapes", fontsize=16)
plt.tight_layout()
plt.show()
```

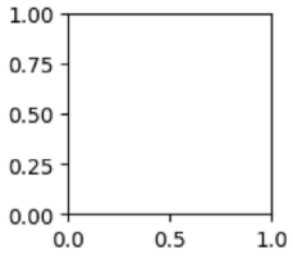
```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[9], line 27
     25 for i in range(limit):
     26     plt.subplot(5, 5, i+1)
---> 27     plt.imshow(image_list[i])
     28     plt.title(str(shape_list[i]))
     29     plt.axis("off")

IndexError: list index out of range
```



Jpeg only is also index out of range

```
for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".jpeg")):
            img_path = os.path.join(root, file)
            img = Image.open(img_path).convert("RGB")

            image_list.append(np.array(img))
            shape_list.append(img.size)

            count += 1
            if count == limit:
                break
    if count == limit:
        break

plt.figure(figsize=(10, 10))
for i in range(limit):
    plt.subplot(5, 5, i+1)
    plt.imshow(image_list[i])
    plt.title(str(shape_list[i]))
    plt.axis("off")

plt.suptitle("First 25 Images & Their Shapes", fontsize=16)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[11], line 27
     25 for i in range(limit):
     26     plt.subplot(5, 5, i+1)
---> 27     plt.imshow(image_list[i])
     28     plt.title(str(shape_list[i]))
     29     plt.axis("off")

IndexError: list index out of range
```



So I given everything jpg, png, jpeg

```
import numpy as np
image_list = []
shape_list = []

limit = 25
count = 0

for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".jpg", ".png", ".jpeg")):
            img_path = os.path.join(root, file)
            img = Image.open(img_path).convert("RGB")

            image_list.append(np.array(img))
            shape_list.append(img.size)

            count += 1
            if count == limit:
                break
    if count == limit:
        break

plt.figure(figsize=(10, 10))
for i in range(limit):
    plt.subplot(5, 5, i+1)
    plt.imshow(image_list[i])
    plt.title(str(shape_list[i]))
    plt.axis("off")

plt.suptitle("First 25 Images & Their Shapes", fontsize=16)
plt.tight_layout()
plt.show()
```

[13]:
```python
import os
from PIL import Image
import numpy as np

TARGET_SIZE = (128, 128)

resized_images = []
shapes = []
count = 0

folder_path = "dogs/Images/"

print("Resizing all images to:", TARGET_SIZE)

for file in os.listdir(folder_path):
    if file.lower().endswith((".jpg", ".jpeg", ".png")):
        img_path = os.path.join(folder_path, file)

        img = Image.open(img_path).convert("RGB")
        img = img.resize(TARGET_SIZE)

        img_np = np.array(img)

        resized_images.append(img_np)
        shapes.append(img_np.shape)

        count += 1
        print(f"Image {count} resized to {img_np.shape}")

resized_images = np.array(resized_images)

print("\nTotal images resized:", resized_images.shape[0])
print("Each image final shape:", resized_images[0].shape)
```

```
Resizing all images to: (128, 128)

Total images resized: 0
```

```
IndexError                                Traceback (most recent call last)
Cell In[13], line 33
     30 resized_images = np.array(resized_images)
     32 print("\nTotal images resized:", resized_images.shape[0])
---> 33 print("Each image final shape:", resized_images[0].shape)

IndexError: index 0 is out of bounds for axis 0 with size 0
```

[14]:
```python
import os
from PIL import Image
import numpy as np

DATASET_DIR = "dogs/Images/"

TARGET_SIZE = (128, 128)

resized_images = []
shapes = []
count = 0

print("Resizing all images to:", TARGET_SIZE)

for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".jpg", ".jpeg", ".png")):
            img_path = os.path.join(root, file)


            img = Image.open(img_path).convert("RGB")


            img = img.resize(TARGET_SIZE)


            img_np = np.array(img)

            resized_images.append(img_np)
            shapes.append(img_np.shape)

            count += 1
            print(f"Image {count} resized to {img_np.shape}")

resized_images = np.array(resized_images)

print("\nTotal images resized:", resized_images.shape[0])
print("Each image final shape:", resized_images[0].shape)
```

```
Resizing all images to: (128, 128)
Image 1 resized to (128, 128, 3)
Image 2 resized to (128, 128, 3)
Image 3 resized to (128, 128, 3)
Image 4 resized to (128, 128, 3)
Image 5 resized to (128, 128, 3)
Image 6 resized to (128, 128, 3)
Image 7 resized to (128, 128, 3)
Image 8 resized to (128, 128, 3)
Image 9 resized to (128, 128, 3)
Image 10 resized to (128, 128, 3)
```

DATASET_DIR helps the code know exactly where to look for images while keeping the script flexible and organized.

[15]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))

for i in range(25):
    plt.subplot(5, 5)
    plt.imshow(resized_images[i])
    plt.axis("off")

plt.suptitle("Resized Images (128x128)", fontsize=16)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 6
      3 plt.figure(figsize=(10, 10))
      5 for i in range(25):
----> 6     plt.subplot(5, 5)
      7     plt.imshow(resized_images[i])
      8     plt.axis("off")

File /opt/anaconda3/envs/tf/lib/python3.10/site-packages/matplotlib/pyplot.py:1551, in subplot(*args, **kwargs)
   1548 fig = gcf()
   1550 # First, search for an existing subplot with a matching spec.
-> 1551 key = SubplotSpec._from_subplot_args(fig, args)
   1553 for ax in fig.axes:
   1554     # If we found an Axes at the position, we can reuse it if the user passed no
   1555     # kwargs or if the Axes class and kwargs are identical.
   1556     if (ax.get_subplotspec() == key
   1557         and (kwargs == {}
   1558              or (ax._projection_init
   1559                  == fig._process_projection_requirements(**kwargs)))):

File /opt/anaconda3/envs/tf/lib/python3.10/site-packages/matplotlib/gridspec.py:576, in SubplotSpec._from_subplot_args(figure, args)
    574     rows, cols, num = args
    575 else:
--> 576     raise _api.nargs_error("subplot", takes="1 or 3", given=len(args))
    578 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    579 if gs is None:

TypeError: subplot() takes 1 or 3 positional arguments but 2 were given
<Figure size 1000x1000 with 0 Axes>
```

[16]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))

for i in range(25):
    plt.subplot(5, 5,i)
    plt.imshow(resized_images[i])
    plt.axis("off")

plt.suptitle("Resized Images (128x128)", fontsize=16)
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[16], line 6
      3 plt.figure(figsize=(10, 10))
      5 for i in range(25):
----> 6     plt.subplot(5, 5,i)
      7     plt.imshow(resized_images[i])
      8     plt.axis("off")

File /opt/anaconda3/envs/tf/lib/python3.10/site-packages/matplotlib/pyplot.py:1551, in subplot(*args, **kwargs)
   1548 fig = gcf()
   1550 # First, search for an existing subplot with a matching spec.
-> 1551 key = SubplotSpec._from_subplot_args(fig, args)
   1553 for ax in fig.axes:
   1554     # If we found an Axes at the position, we can reuse it if the user passed no
   1555     # kwargs or if the Axes class and kwargs are identical.
   1556     if (ax.get_subplotspec() == key
   1557         and (kwargs == {}
   1558              or (ax._projection_init
   1559                  == fig._process_projection_requirements(**kwargs)))):

File /opt/anaconda3/envs/tf/lib/python3.10/site-packages/matplotlib/gridspec.py:589, in SubplotSpec._from_subplot_args(figure, args)
    587 else:
    588     if not isinstance(num, Integral) or num < 1 or num > rows*cols:
--> 589         raise ValueError(
    590             f"num must be an integer with 1 <= num <= {rows*cols}, "
    591             f"not {num!r}"
    592         )
    593     i = j = num
    594 return gs[i-1:j]

ValueError: num must be an integer with 1 <= num <= 25, not 0
<Figure size 1000x1000 with 0 Axes>
```

Added plt.subplot(5,5,i)also, still shown error

subplot index must start from 1, not 0,

plt.subplot(5, 5, 0)

So i+1

[19]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))

for i in range(25):
    plt.subplot(5, 5,i+1)
    plt.imshow(resized_images[i])
    plt.axis("off")

plt.suptitle("Resized Images (128x128)", fontsize=16)
plt.tight_layout()
plt.show()
```

Final results after error correction

dataset loading

[1]:
```
import urllib.request
import tarfile
import os
from PIL import Image

url = "http://vision.stanford.edu/aditya86/ImageNetDogs/images.tar"
tar_path = "dogs.tar"

print("Downloading...")
urllib.request.urlretrieve(url, tar_path)

print("Extracting...")
with tarfile.open(tar_path) as tar:
    tar.extractall("dogs")

DATASET_DIR = "dogs/Images/"

shapes = sorted([
    Image.open(os.path.join(root, file)).size
    for root, _, files in os.walk(DATASET_DIR)
    for file in files
    if file.lower().endswith((".jpg",".png",".jpeg"))
])

print("Total images:", len(shapes))
print("Unique shapes (first 20):", shapes[:20])
```

```
Downloading...
Extracting...
Total images: 20580
Unique shapes (first 20): [(97, 134), (100, 115), (103, 120), (105, 100), (106, 150), (107, 150), (108, 120), (108, 150), (109, 110), (109,
150), (110, 150), (111, 150), (112, 120), (117, 186), (118, 120), (120, 102), (120, 104), (120, 110), (120, 111), (120, 120)]
```

finding shapes of each and every image ¶

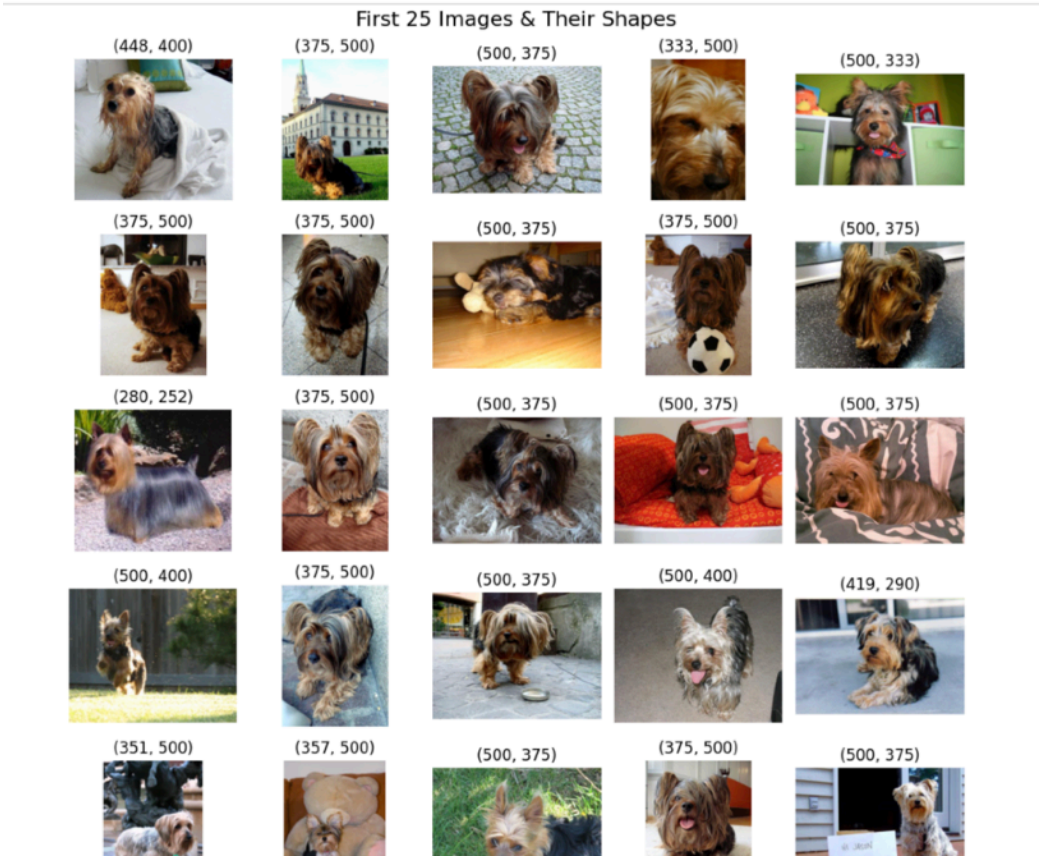
[8]:
```
all_shapes = []


count = 0

for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".jpg", ".png", ".jpeg")):
            img_path = os.path.join(root, file)
            img = Image.open(img_path)
            shape = img.size
            all_shapes.append(shape)
            count += 1
            print(f"Image {count}: {shape}")

print("\nTotal images found:", count)
print("Unique image shapes:", set(all_shapes))
```

```
Image 30: (500, 375)
Image 31: (500, 375)
Image 32: (500, 375)
Image 33: (1024, 768)
Image 34: (500, 399)
Image 35: (500, 375)
Image 36: (500, 375)
Image 37: (500, 361)
Image 38: (500, 384)
Image 39: (500, 375)
Image 40: (351, 500)
Image 41: (375, 500)
Image 42: (500, 375)
Image 43: (500, 375)
Image 44: (500, 375)
Image 45: (500, 359)
Image 46: (500, 350)
Image 47: (375, 500)
Image 48: (500, 375)
```

Reshaping images to same size

[14]:
```
import os
from PIL import Image
import numpy as np

DATASET_DIR = "dogs/Images/"

TARGET_SIZE = (128, 128)

resized_images = []
shapes = []
count = 0

print("Resizing all images to:", TARGET_SIZE)

for root, _, files in os.walk(DATASET_DIR):
    for file in files:
        if file.lower().endswith((".jpg", ".jpeg", ".png")):
            img_path = os.path.join(root, file)


            img = Image.open(img_path).convert("RGB")


            img = img.resize(TARGET_SIZE)


            img_np = np.array(img)

            resized_images.append(img_np)
            shapes.append(img_np.shape)

            count += 1
            print(f"Image {count} resized to {img_np.shape}")

resized_images = np.array(resized_images)

print("\nTotal images resized:", resized_images.shape[0])
print("Each image final shape:", resized_images[0].shape)
```

```
Image 56 resized to (128, 128, 3)
Image 57 resized to (128, 128, 3)
Image 58 resized to (128, 128, 3)
Image 59 resized to (128, 128, 3)
Image 60 resized to (128, 128, 3)
Image 61 resized to (128, 128, 3)
```



visualizing the reshaped images

19]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))

for i in range(25):
    plt.subplot(5, 5,i+1)
    plt.imshow(resized_images[i])
    plt.axis("off")

plt.suptitle("Resized Images (128x128)", fontsize=16)
plt.tight_layout()
plt.show()
```