Edge Intelligence Lab-1

Name: N. Lakshmi Sandeep

Register No: 25MML0006

Tasks:

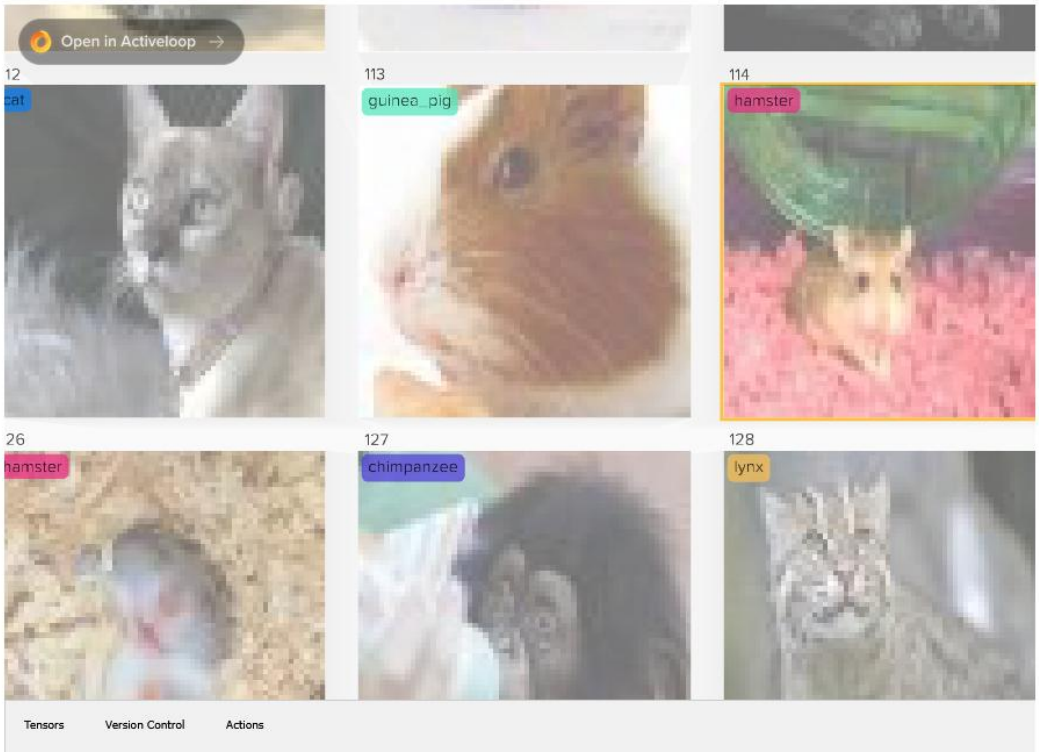
1. Identify Jupyter Notebook
2. Open Jupyter Notebook
3. Choose a Dataformat
4. Choose a Dataset
5. Load the Dataset

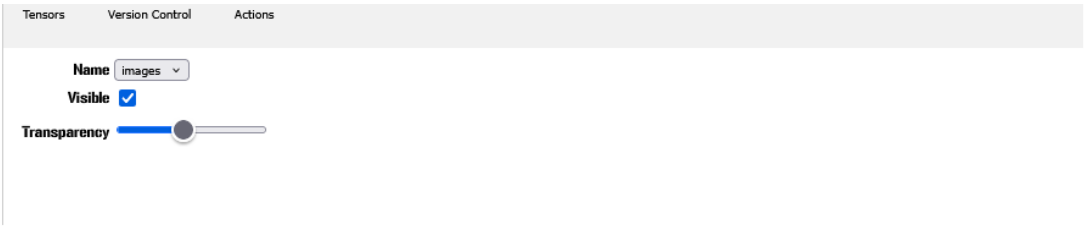
Implementation:

Chosen Image Dataset – animal 10N dataset consists of 5000 images of training data and 5000 images of testing data

By importing **deeplake** library can load the direct dataset from link

```
import deeplake

ds_train=deeplake.load("hub://activeloop/animal10n-train")
ds_test=deeplake.load("hub://activeloop/animal10n-test")
print(ds_train.visualize())
```






```
None
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

Error:

```
In [8]: print(ds_train.shape())

---------------------------------------------------------------------------
TensorDoesNotExistError                   Traceback (most recent call last)
~\AppData\Roaming\Python\Python39\site-packages\deeplake\core\dataset\dataset.py in __getattr__(self, key)
   1387         try:
-> 1388             return self.__getitem__(key)
   1389         except TensorDoesNotExistError as ke:

~\AppData\Roaming\Python\Python39\site-packages\deeplake\core\dataset\dataset.py in __getitem__(self, item, is_iteration)
    589             else:
--> 590                 raise TensorDoesNotExistError(item)
    591         elif isinstance(item, (int, slice, list, tuple, Index, type(Ellipsis))):

TensorDoesNotExistError: "Tensor 'shape' does not exist."

The above exception was the direct cause of the following exception:

AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_23452\2249661798.py in <module>
----> 1 print(ds_train.shape())

~\AppData\Roaming\Python\Python39\site-packages\deeplake\core\dataset\dataset.py in __getattr__(self, key)
   1388             return self.__getitem__(key)
   1389         except TensorDoesNotExistError as ke:
-> 1390             raise AttributeError(
   1391                 f"'{self.__class__}' object has no attribute '{key}'"
   1392             ) from ke

AttributeError: '<class 'deeplake.core.dataset.deeplake_cloud_dataset.DeepLakeCloudDataset'>' object has no attribute 'shape'
```

Error can be rectified by including tensor images as **ds_train.images.shape**

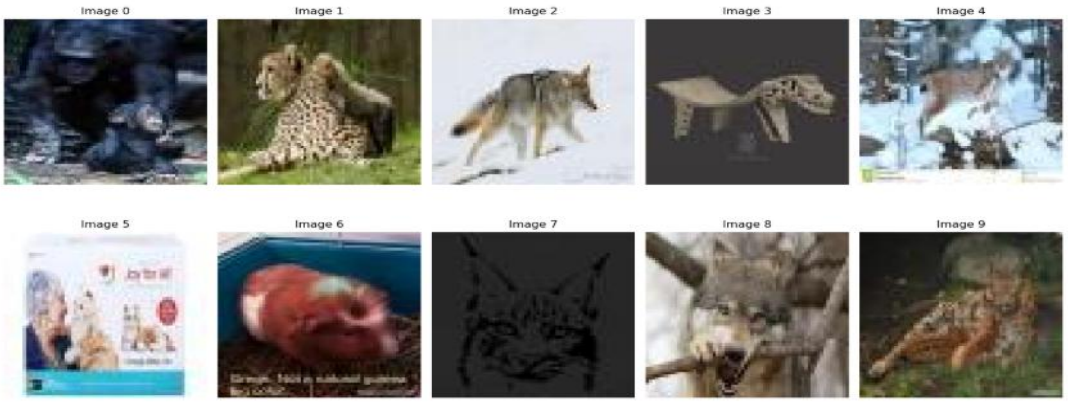
```
In [9]: print(ds_train.images.shape)
        (50000, 64, 64, 3)

In [10]: print(ds_test.images.shape)
         (5000, 64, 64, 3)
```

Actually, here all the images having the same shape as we have taken the preprocessed data, suppose if the dataset contains different shapes that means different resolution we have to reshape each image by using opencv, if we take the testing data as example we can reshape each image into

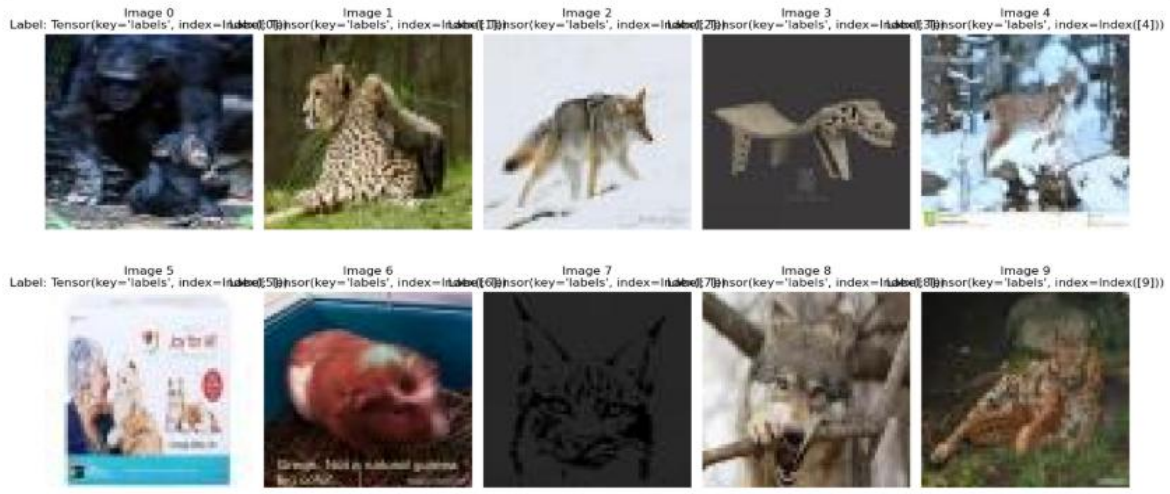
```
import cv2

import numpy as np

TARGET_SIZE = (224, 224)

resized_images = []

for sample in ds_test:

    img = sample['images'].numpy()

    img_resized = cv2.resize(img, TARGET_SIZE)

    resized_images.append(img_resized)

resized_images = np.array(resized_images)

print(resized_images.shape)
```

```
In [16]: unique_shapes = set()
         image_count = 0

         # Iterate through the dataset to check shapes
         for sample in ds_train:
             # Call .numpy() to load the actual image data into memory as a NumPy array
             image = sample['images'].numpy()
             unique_shapes.add(image.shape)
             image_count += 1

         print(f"\nTotal images processed: {image_count}")
         print(f"Found {len(unique_shapes)} unique image shapes:")

         Total images processed: 50000
         Found 1 unique image shapes:

In [17]: # Print all unique shapes found
         for shape in unique_shapes:
             print(f"- {shape}")

         # Summary
         if len(unique_shapes) == 1:
             print("\nResult: All images in the dataset have the same shape.")
         else:
             print("\nResult: Images in the dataset have varying shapes so reshape the images into proper shape.")

         - (64, 64, 3)

         Result: All images in the dataset have the same shape.
```

```
import matplotlib.pyplot as plt

# Show first 10 images
plt.figure(figsize=(15, 8))

for i in range(10):
    sample = ds_train[i]
    img = sample['images']

    plt.subplot(2, 5, i+1)   # 2 rows, 5 columns
    plt.imshow(img)
    plt.title(f"Image {i}")
    plt.axis("off")

plt.tight_layout()
plt.show()
```




```
plt.figure(figsize=(15, 8))
#to print with labels
for i in range(10):
    sample = ds_train[i]
    img = sample['images']
    label = sample['labels']  # extract label

    plt.subplot(2, 5, i+1)
    plt.imshow(img)
    plt.title(f"Image {i}\nLabel: {label}")
    plt.axis("off")

plt.tight_layout()
plt.show()
```



Here we can see an error as all the labels are occurred at once without the specific label so to get that specific label correspond to that image we need to add **.numpy()** to each sample.

After adding **numpy()** to the labels

```
plt.figure(figsize=(15, 8))
#to print with labels
for i in range(10):
    sample = ds_train[i]
    img = sample['images'].numpy()
    label = sample['labels'].numpy()  # extract label

    plt.subplot(2, 5, i+1)
    plt.imshow(img)
    plt.title(f"Image {i}\nLabel: {label}")
    plt.axis("off")

plt.tight_layout()
plt.show()
```