Skin Cancer classification

Overview

Popular dermatoscopic image data set HAM10000 ("Human Against Machine with 10000 training images") is hosted by Harvard Dataverse and includes images from various demographics. The dataset comprises 10015 images that fall into various diagnostic categories, such as: dermatofibroma (df), melanoma (mel), melanocytic nevi (nv), basal cell carcinoma (bcc), actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), melanoma (mel), melanocytic nevi (nv), and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc).

This example will show you how to use Pytorch to link the MONAI framework with Amazon SageMaker. It will also include example code for MONAI pre-processing transforms that can help with picture transformations and imbalanced datasets. Additionally, the code to activate MONAI neural network topologies, like Densenet, will be demonstrated.

SageMaker and MONAI using Pytorch

We will be using MONAI framework into the Amazon SageMaker managed service and designed code of MONAI pre-processing transforms that can assist with imbalanced datasets and image transformations. We will also review the code to invoke MONAI neural network architectures such as Densenet for image classification. Additionally, we will cover the SageMaker API calls to launch and manage the compute infrastructure for both model training and hosting for inference.

Components

This MONAI Skin Cancer model implementation utilizes:

- Amazon S3 bucket for skin cancer images
- Amazon Sagemaker Notebook instance ml.t2.medium with 100 GB EBS and conda_pytorch_p36 kernel
- Amazon Sagemaker Pytorch managed container

Skin Cancer classification

- HAM10000 dataset hosted by Harvard Dataverse
- MONAI 0.3.0

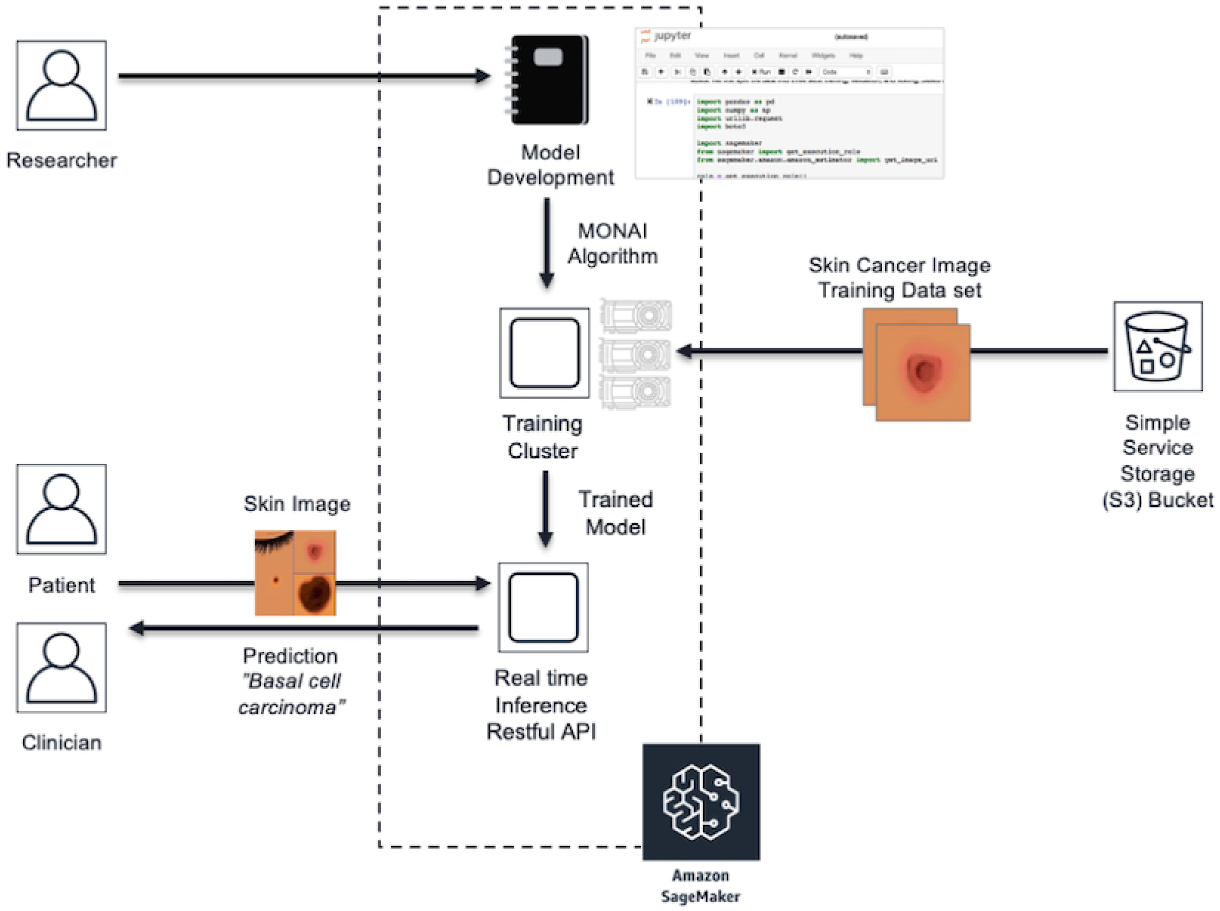
Architecture

The skin cancer predictive model uses Amazon Sagemaker architecture which includes model development within managed Jupyter Notebooks, integration with the MONAI framework by extending a SageMaker managed PyTorch container, and training on ephemeral clusters that use the HAM10000 dataset uploaded to S3. Once the model is trained, it is deployed to a SageMaker real-time endpoint for hosting that is composed of one or more managed Amazon Elastic Compute (EC2) instances which are deployed over different availability zones for high availability. For inferences, we submit a sample of patient skin cancer images from the HAM10000 validation dataset to the endpoint via the SageMaker API which returns the model predictions for class and probability.

SageMaker Model Lifecycle including MONAI framework

The skin cancer predictive model uses Amazon Sagemaker architecture which includes model development within managed Jupyter Notebooks, integration with the MONAI framework by extending a SageMaker managed PyTorch container, and training on ephemeral clusters that use the HAM10000 dataset uploaded to S3. Once the model is trained, it is deployed to a SageMaker real-time endpoint for hosting that is composed of one or more managed Amazon Elastic Compute (EC2) instances which are deployed over different availability zones for high availability. For inferences, we submit a sample of patient skin cancer images from the HAM10000 validation dataset to the endpoint via the SageMaker.

Skin Cancer classification



```
import pandas as pd
import numpy as np
import urllib.request
import boto3

import sagemaker
from sagemaker import get_execution_role
from sagemaker.amazon.amazon_estimator import get_image_uri

role = get_execution_role()
```

Deploy Instructions

To run the skin cancer model using MONAI follow the steps below:

1. Create an S3 bucket in your account and add an empty folder to it.
2. Navigate to the dataset download at HAM10000.
3. Select "Access Dataset" in top right, and review the license Creative Commons Attribution-NonCommercial 4.0 International Public License.
4. If you accept license, then select "Original Format Zip" and upload the zip to the folder in the S3 bucket you created in the previous steps.
5. Navigate to Amazon Sagemaker in the same account to create a Jupyter Notebook instance.
6. Under Notebook > Notebook instances select Create Notebook instance. Fill in the name (ex. skin-cancer-notebook) and instance type ml.t2.medium and select volume size of 100 GB.

7. The Notebook will need permissions to call other services including SageMaker and S3. Choose an existing role or create a role with the AmazonSageMakerFull Access IAM role.
8. Create Notebook instance and once In Service, then Open Jupyter Lab.
9. In the Jupyter Notebook, select File > New > Terminal, cd SageMaker and execute
   git clone https://github.com/vigneshkumar957/skincancerclassification
10. Update the set.env file within Jupyter Notebook with the S3 location (SKIN_CANCER_BUCKET) and prefix/path (SKIN_CANCER_BUCKET_PATH) of the dataverse_files.zip uploaded to S3.
11. Open the Jupyter Notebook monai_skin_cancer.ipynb and Run > Run all cells to observe the skin cancer example using MONAI.
12. The example output shows inference results on five sample skin cancer images from the validation set and includes probablity of the skin cancer classification.

Setup

This notebook was created and tested on an ml.t2.medium notebook instance with 100 GB of EBS and conda_pytorch_p36 kernel.

Let's get started by creating a S3 bucket and uploading the HAM10000 dataset to the bucket.

1. Create an S3 bucket in the same account as the Sagemaker notebook instance.
2. Download the skin cancer dataset at HAM10000.
3. Select "Access Dataset" in top right, and select "Original Format Zip".
4. Upload the dataset to the S3 bucket created in step 1.
5. Update the set.env file located in the current directory with the S3 location of the dataverse_files.zip.

The code below will install MONAI framework and dependent packages and setup environment variables.

```
!python -m pip install -r source/requirements.txt
```

Skin Cancer classification

```
Requirement already satisfied: monai==0.3.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 1)) (0.3.0)
Requirement already satisfied: torchtext==0.6.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 2)) (0.6.0)
Requirement already satisfied: python-dotenv==1.0.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 3)) (1.0.0)
Requirement already satisfied: torch>=1.4 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from monai==0.3.0->-r source/requirements.txt (line 1)) (2.1.0)
Requirement already satisfied: numpy>=1.17 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from monai==0.3.0->-r source/requirements.txt (line 1)) (1.25.1)
Requirement already satisfied: tqdm in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (4.65.0)
Requirement already satisfied: requests in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (2.31.0)
Requirement already satisfied: six in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (1.16.0)
Requirement already satisfied: sentencepiece in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (0.1.99)
Requirement already satisfied: filelock in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (3.12.2)
Requirement already satisfied: typing-extensions in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (4.5.0)
Requirement already satisfied: sympy in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (1.12)
Requirement already satisfied: networkx in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (3.1)
Requirement already satisfied: jinja2 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (3.1.2)
Requirement already satisfied: fsspec in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (2023.6.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.105)
```

Skin Cancer classification

```
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.18.1 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (2.18.1)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: triton==2.1.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (2.1.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.4->monai==0.3.0->-r source/requirements.txt (line 1)) (12.3.52)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from requests->torchtext==0.6.0->-r source/requirements.txt (line 2)) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from requests->torchtext==0.6.0->-r source/requirements.txt (line 2)) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from requests->torchtext==0.6.0->-r source/requirements.txt (line 2)) (1.26.14)
```

Skin Cancer classification

```
Requirement already satisfied: certifi>=2017.4.17 in /home/ec2-user/anaconda3/envs/tensorflow2_p31
0/lib/python3.10/site-packages (from requests->torchtext==0.6.0->-r source/requirements.txt (line 2)) (
2023.5.7)
Requirement already satisfied: MarkupSafe>=2.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310
/lib/python3.10/site-packages (from jinja2->torch>=1.4->monai==0.3.0->-r source/requirements.txt (li
ne 1)) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/li
b/python3.10/site-packages (from sympy->torch>=1.4->monai==0.3.0->-r source/requirements.txt (lin
e 1)) (1.3.0)
```

```
import os
from pathlib import Path
from dotenv import load_dotenv
env_path = Path('.') / 'set.env'
load_dotenv(dotenv_path=env_path)

skin_cancer_bucket=os.environ.get('SKIN_CANCER_BUCKET')
skin_cancer_bucket_path=os.environ.get('SKIN_CANCER_BUCKET_PATH')
skin_cancer_files=os.environ.get('SKIN_CANCER_FILES')
skin_cancer_files_ext=os.environ.get('SKIN_CANCER_FILES_EXT')
base_dir = os.environ.get('BASE_DIR')

print('Skin Cancer Bucket: '+skin_cancer_bucket)
print('Skin Cancer Bucket Prefix: '+skin_cancer_bucket_path)
print('Skin Cancer Files: '+skin_cancer_files)
print('Skin Cancer Files Ext: '+skin_cancer_files_ext)
print('Base Dir: '+base_dir)
```

HAM10000 Data Transformation

The transform_data.ipynb will download the dataverse_files.zip and perform transformations to build directories by class for training and validation sets from the meta-data. It will also augment the data to create a more balanced data set across the classes for training. The script will upload the transformed dataset HAM10000.tar.gz to the same S3 bucket identifed in set.env for model training.

```
%run source/transform_data.ipynb
```

Data

Create Sagemaker session and S3 location for transformed HAM10000 dataset

```
import sagemaker
```

Skin Cancer classification

```
sagemaker_session = sagemaker.Session()
role = sagemaker.get_execution_role()
```

```
inputs =
sagemaker_session.upload_data(path=base_dir+'HAM10000.tar.gz',
bucket=skin_cancer_bucket, key_prefix=skin_cancer_bucket_path)
print('input spec (in this case, just an S3 path):
{}'.format(inputs))
```

Train Model

Training

The monai_skin_cancer.py script provides all the code we need for training and hosting a SageMaker model (model_fn function to load a model). The training script is very similar to a training script you might run outside of SageMaker, but you can access useful properties about the training environment through various environment variables, such as:

- SM_MODEL_DIR: A string representing the path to the directory to write model artifacts to. These artifacts are uploaded to S3 for model hosting.
- SM_NUM_GPUS: The number of gpus available in the current container.
- SM_CURRENT_HOST: The name of the current container on the container network.
- SM_HOSTS: JSON encoded list containing all the hosts .

Supposing one input channel, 'training', was used in the call to the PyTorch estimator's fit() method, the following will be set, following the format SM_CHANNEL_[channel_name]:

- SM_CHANNEL_TRAINING: A string representing the path to the directory containing data in the 'training' channel.

```
!pygmentize source/monai_skin_cancer.py
```

Run training in SageMaker

Skin Cancer classification

The PyTorch class allows us to run our training function as a training job on SageMaker infrastructure. We need to configure it with our training script, an IAM role, the number of training instances, the training instance type, and hyperparameters. In this case we are going to run our training job on ml.p3.8xlarge instance. But this example can be ran on one or multiple, cpu or gpu instances (full list of available instances). The hyperparameters parameter is a dict of values that will be passed to your training script -- you can see how to access these values in the monai_skin_cancer.py script above.

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(entry_point='monai_skin_cancer.py',
                    source_dir='source',
                    role=role,
                    framework_version='1.5.0',
                    py_version='py3',
                    instance_count=1,
                    instance_type='ml.p3.8xlarge',
                    hyperparameters={
                        'backend': 'gloo',
                        'epochs': 30
                    })
```

HOST Model

Create real-time endpoint

After training, we use the PyTorch estimator object to build and deploy a PyTorchPredictor. This creates a Sagemaker Endpoint -- a hosted prediction service that we can use to perform inference.

As mentioned above we have implementation of model_fn in the monai_skin_cancer.py script that is required. We are going to use default implementations of input_fn, predict_fn, output_fn and transform_fm defined in sagemaker-pytorch-containers.

```
predictor = estimator.deploy(initial_instance_count=1,
instance_type='ml.m5.xlarge')
```

Load Validation Images for Inference

Skin Cancer classification

In [ ]:

```
from PIL import Image

print('Load Test Images for Inference')
val_dir = os.path.join(base_dir, 'HAM10000/val_dir')
class_names = sorted([x for x in os.listdir(val_dir) if
os.path.isdir(os.path.join(val_dir, x))])
num_class = len(class_names)
image_files = [[os.path.join(val_dir, class_name, x)
                for x in os.listdir(os.path.join(val_dir,
class_name))[:1]]
               for class_name in class_names]
image_file_list = []
image_label_list = []

for i, class_name in enumerate(class_names):
    image_file_list.extend(image_files[i])
    image_label_list.extend([i] * len(image_files[i]))

num_total = len(image_label_list)
image_width, image_height = Image.open(image_file_list[0]).size
```

MONAI Transform Image using Compose and Skin Cancer Dataset

MONAI has transforms that support both Dictionary and Array format and are specialized for the high-dimensionality of medical images. The transforms include several categories such as Crop & Pad, Intensity, IO, Post-processing, Spatial, and Utilities. In the following excerpt, the Compose class chains a series of image transforms together and returns a single tensor of the image.

```
import torch
from torch.utils.data import DataLoader
from source.skin_cancer_dataset import SkinCancerDataset
from monai.transforms import Compose, LoadPNG, Resize,
AsChannelFirst, ScaleIntensity, ToTensor

val_transforms = Compose([
        LoadPNG(image_only=True),
        AsChannelFirst(channel_dim=2),
        ScaleIntensity(),
        Resize(spatial_size=(64,64)),
        ToTensor()
])

val_ds = SkinCancerDataset(image_file_list, image_label_list,
val_transforms)
val_loader = DataLoader(val_ds, batch_size=1, num_workers=1)
```

Evaluate

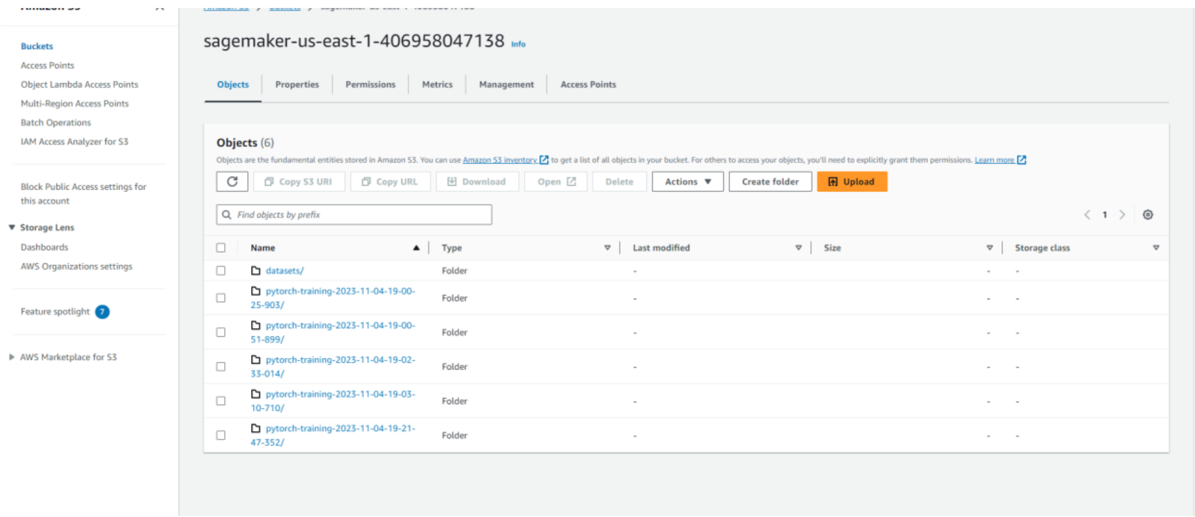
Skin Cancer classification

We can now use the predictor to perform a real-time inference to classify skin cancer images.

In [ ]:

```
print('Sample Inference Results By Class:')
for i, val_data in enumerate(val_loader):
    response = predictor.predict(val_data[0])
    actual_label = val_data[1]
    pred = torch.nn.functional.softmax(torch.tensor(response), dim=1)
    top_p, top_class = torch.topk(pred, 1)
    print('actual class: '+class_names[actual_label.numpy()[0]])
    print('predicted class: '+class_names[top_class])
    print('predicted class probablity: '+str(round(top_p.item(),2)))
    print()
```

Screenshots:

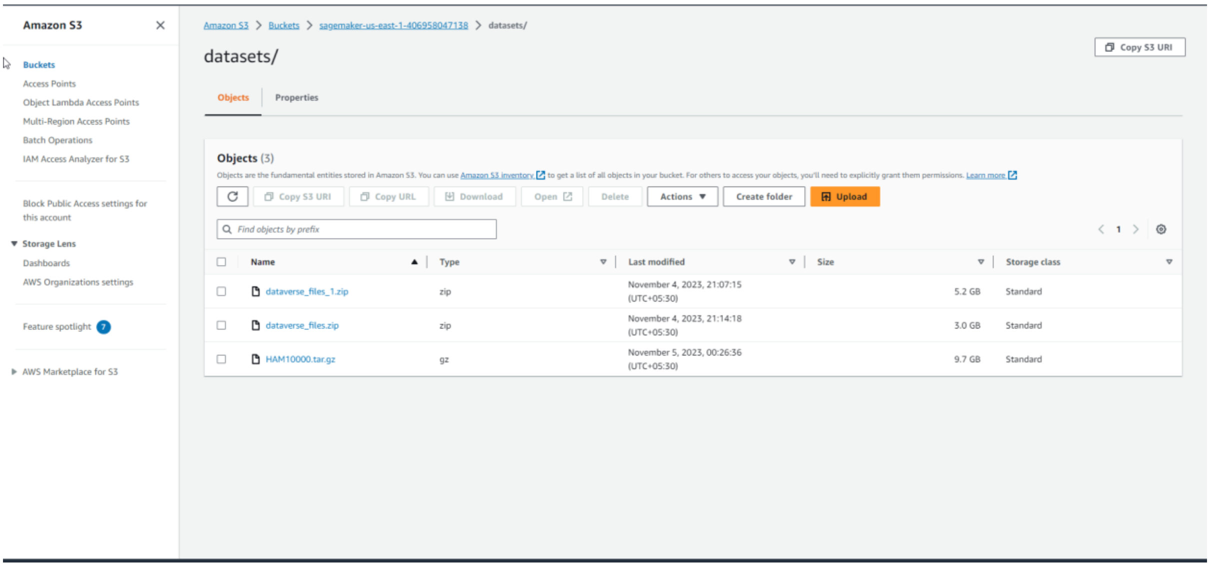


sagemaker-us-east-1-406958047138

Objects (6)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	datasets/	Folder	-	-	-
□	pytorch-training-2023-11-04-19-00-25-903/	Folder	-	-	-
□	pytorch-training-2023-11-04-19-00-51-899/	Folder	-	-	-
□	pytorch-training-2023-11-04-19-02-33-014/	Folder	-	-	-
□	pytorch-training-2023-11-04-19-03-10-710/	Folder	-	-	-
□	pytorch-training-2023-11-04-19-21-47-352/	Folder	-	-	-



Amazon S3 > Buckets > sagemaker-us-east-1-406958047138 > datasets/

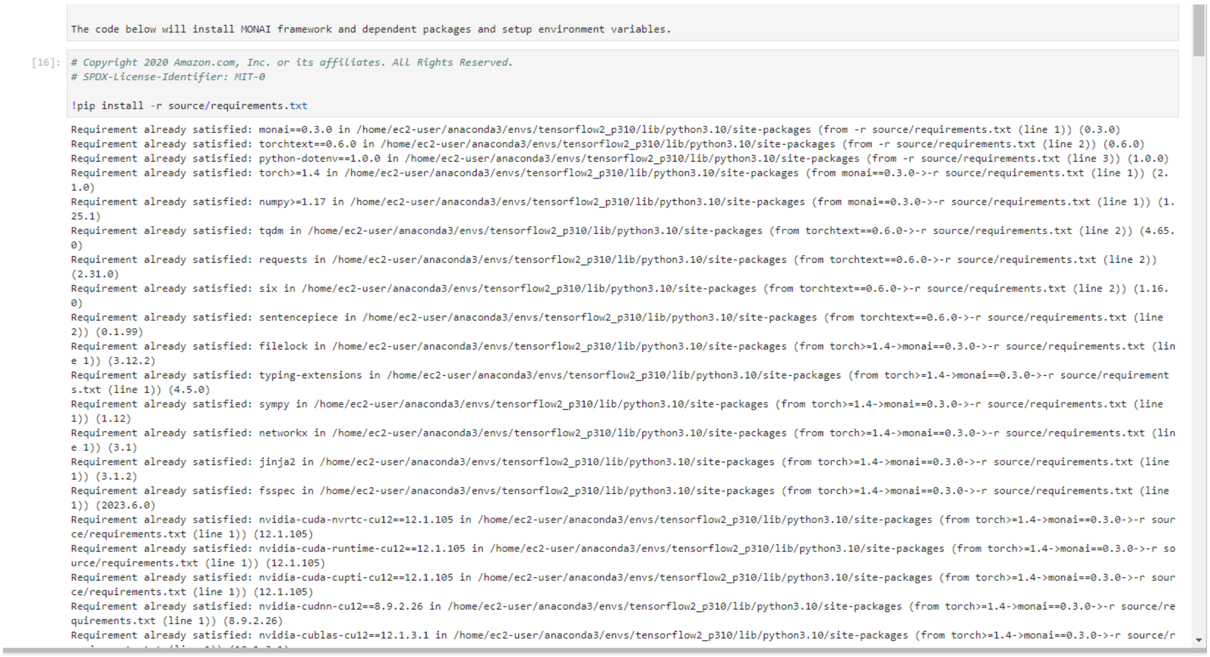
datasets/

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	dataverse_files_1.zip	zip	November 4, 2023, 21:07:15 (UTC+05:30)	5.2 GB	Standard
□	dataverse_files.zip	zip	November 4, 2023, 21:14:18 (UTC+05:30)	3.0 GB	Standard
□	HAM10000.tar.gz	gz	November 5, 2023, 00:26:36 (UTC+05:30)	9.7 GB	Standard

Skin Cancer classification

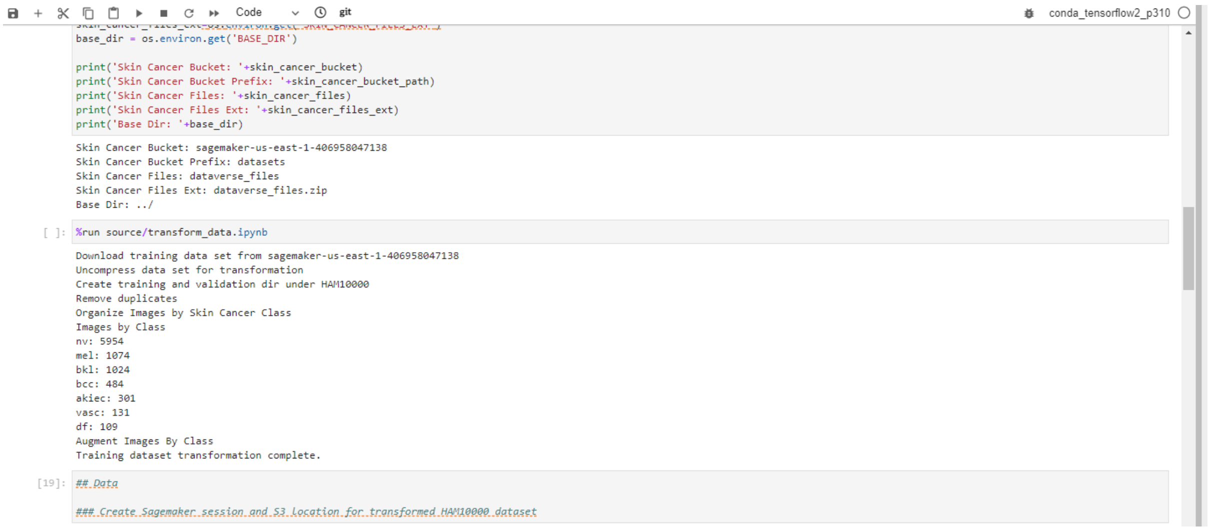


The code below will install MONAI framework and dependent packages and setup environment variables.

```
[16]: # Copyright 2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
      # SPDX-License-Identifier: MIT-0

      !pip install -r source/requirements.txt
```

```
Requirement already satisfied: monai==0.3.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 1)) (0.3.0)
Requirement already satisfied: torchtext==0.6.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 2)) (0.6.0)
Requirement already satisfied: python-dotenv==1.0.0 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from -r source/requirements.txt (line 3)) (1.0.0)
Requirement already satisfied: torch>=1.4 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from monai==0.3.0->-r source/requirements.txt (line 1)) (2.
1.0)
Requirement already satisfied: numpy>=1.17 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from monai==0.3.0->-r source/requirements.txt (line 1)) (1.
25.1)
Requirement already satisfied: tqdm in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (4.65.
0)
Requirement already satisfied: requests in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2))
(2.31.0)
Requirement already satisfied: six in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line 2)) (1.16.
0)
Requirement already satisfied: sentencepiece in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torchtext==0.6.0->-r source/requirements.txt (line
2)) (0.1.99)
Requirement already satisfied: filelock in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (lin
e 1)) (3.12.2)
Requirement already satisfied: typing-extensions in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirement
s.txt (line 1)) (4.5.0)
Requirement already satisfied: sympy in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line
1)) (1.12)
Requirement already satisfied: networkx in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (lin
e 1)) (3.1)
Requirement already satisfied: jinja2 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line
1)) (3.1.2)
Requirement already satisfied: fsspec in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/requirements.txt (line
1)) (2023.6.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r sour
ce/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r so
urce/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r sour
ce/requirements.txt (line 1)) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/re
quirements.txt (line 1)) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /home/ec2-user/anaconda3/envs/tensorflow2_p310/lib/python3.10/site-packages (from torch>=1.4->monai==0.3.0->-r source/r
```



```
base_dir = os.environ.get('BASE_DIR')

print('Skin Cancer Bucket: '+skin_cancer_bucket)
print('Skin Cancer Bucket Prefix: '+skin_cancer_bucket_path)
print('Skin Cancer Files: '+skin_cancer_files)
print('Skin Cancer Files Ext: '+skin_cancer_files_ext)
print('Base Dir: '+base_dir)
```

```
Skin Cancer Bucket: sagemaker-us-east-1-406958047138
Skin Cancer Bucket Prefix: datasets
Skin Cancer Files: dataverse_files
Skin Cancer Files Ext: dataverse_files.zip
Base Dir: ../
```

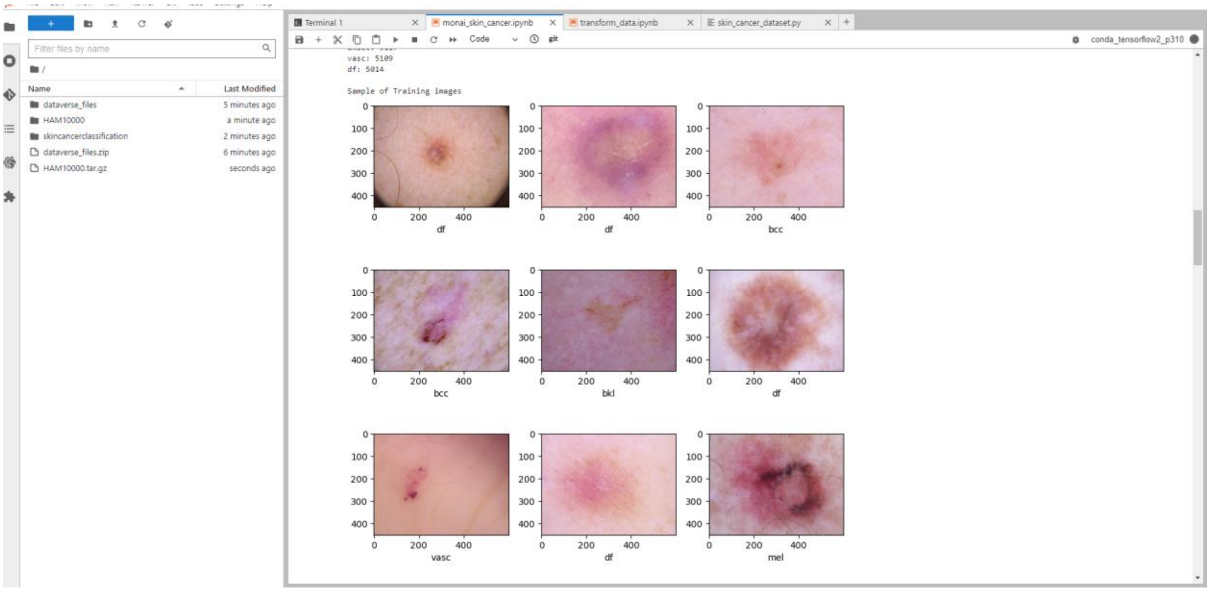
```
[ ]: %run source/transform_data.ipynb
```

```
Download training data set from sagemaker-us-east-1-406958047138
Uncompress data set for transformation
Create training and validation dir under HAM10000
Remove duplicates
Organize Images by Skin Cancer Class
Images by Class
nv: 5954
mel: 1074
bkl: 1024
bcc: 484
akiec: 301
vasc: 131
df: 109
Augment Images By Class
Training dataset transformation complete.
```

```
[19]: ## Data

      ### Create Sagemaker session and S3 Location for transformed HAM10000 dataset
```

Skin Cancer classification



Name	Last Modified
dataverse_files	5 minutes ago
HAM10000	a minute ago
skincancerclassification	2 minutes ago
dataverse_files.zip	6 minutes ago
HAM10000.tar.gz	seconds ago

```
vasc: 5109
df: 5014

Sample of Training images
```

Conclusion

Popular dermatoscopic image data set HAM10000 ("Human Against Machine with 10000 training images") is hosted by Harvard Dataverse and includes images from various demographics. The dataset comprises 10015 images that fall into various diagnostic categories, such as: dermatofibroma (df), melanoma (mel), melanocytic nevi (nv), basal cell carcinoma (bcc), actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), melanoma (mel), melanocytic nevi (nv), and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc).

This example will show you how to use Pytorch to link the MONAI framework with Amazon SageMaker. It will also include example code for MONAI pre-processing transforms that can help with picture transformations and imbalanced datasets. Additionally, the code to activate MONAI neural network topologies, like Densenet, will be demonstrated.