Week_2_Data_Ingestion

September 15, 2024

[1]:
```python
import zipfile
import os
import pandas as pd
import matplotlib.pyplot as plt
import h5py
import cv2
import numpy as np
```

0.1 Extract data from zipfile

[2]:
```python
import zipfile
import os

def extract_zip(zip_file_path: str, extract_to_directory: str):
    """
    Extracts a ZIP file to a specified directory if the ZIP file exists and the␣
↪files
    do not already exist.

    Parameters:
        zip_file_path (str): Path to the ZIP file.
        extract_to_directory (str): Directory where the ZIP file will be␣
↪extracted.
    """
    # Check if the ZIP file exists
    if not os.path.exists(zip_file_path):
        print(f"ZIP file {zip_file_path} does not exist. Skipping extraction.")
        return

    try:
        # Check if the extraction directory contains files
        if not os.path.exists(extract_to_directory) or not os.
↪listdir(extract_to_directory):
            with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
                zip_ref.extractall(extract_to_directory)
            print(f"Extracted {zip_file_path} to {extract_to_directory}")
        else:
            print(f"Files already exist in {extract_to_directory}. Skipping␣
↪extraction.")
    except zipfile.BadZipFile:
        print(f"Error: {zip_file_path} is not a valid ZIP file.")
    except Exception as e:
        print(f"An error occurred during extraction: {e}")

def remove_file(file_path: str):
    """
    Removes a file from the filesystem if it exists.

    Parameters:
        file_path (str): Path to the file to be removed.
    """
    if os.path.exists(file_path):
        os.remove(file_path)
        print(f"Removed {file_path}")
    else:
        print(f"File {file_path} does not exist.")

# Paths
zip_file_path = "../data/raw/train-metadata.zip"
extract_to_directory = "../data/raw/"

# Extract the ZIP file if it exists and the files are not already extracted
extract_zip(zip_file_path, extract_to_directory)

# Optionally, remove the ZIP file after extraction
remove_file(zip_file_path)
```

```
ZIP file ../data/raw/train-metadata.zip does not exist. Skipping extraction.
File ../data/raw/train-metadata.zip does not exist.
```

0.2 Load Metadata

[3]:
```
df_metadata = pd.read_csv("../data/raw/train-metadata.csv")
df_metadata.head(10)
```

```
/tmp/ipykernel_1668636/1128991036.py:1: DtypeWarning: Columns (51,52) have mixed
types. Specify dtype option on import or set low_memory=False.
  df_metadata = pd.read_csv("../data/raw/train-metadata.csv")
```

[3]:
```
        isic_id  target  patient_id  age_approx     sex anatom_site_general  \
0  ISIC_0015670       0  IP_1235828        60.0    male     lower extremity
1  ISIC_0015845       0  IP_8170065        60.0    male           head/neck
2  ISIC_0015864       0  IP_6724798        60.0    male     posterior torso
3  ISIC_0015902       0  IP_4111386        65.0    male      anterior torso
```

```
4  ISIC_0024200       0  IP_8313778        55.0    male      anterior torso
5  ISIC_0035502       0  IP_3026693        75.0  female           head/neck
6  ISIC_0051648       0  IP_0218255        65.0    male     upper extremity
7  ISIC_0051665       0  IP_7734648        50.0    male     posterior torso
8  ISIC_0051710       0  IP_1307115        50.0    male      anterior torso
9  ISIC_0051758       0  IP_2180091        60.0  female     upper extremity

   clin_size_long_diam_mm          image_type tbp_tile_type   tbp_lv_A  …  \
0                    3.04  TBP tile: close-up     3D: white  20.244422  …
1                    1.10  TBP tile: close-up     3D: white  31.712570  …
2                    3.40  TBP tile: close-up        3D: XP  22.575830  …
3                    3.22  TBP tile: close-up        3D: XP  14.242329  …
4                    2.73  TBP tile: close-up     3D: white  24.725520  …
5                    2.54  TBP tile: close-up     3D: white  22.129183  …
6                    3.74  TBP tile: close-up        3D: XP  14.319188  …
7                    4.31  TBP tile: close-up        3D: XP  20.102610  …
8                    3.17  TBP tile: close-up     3D: white  14.166805  …
9                    3.97  TBP tile: close-up        3D: XP  16.251490  …

    lesion_id  iddx_full  iddx_1  iddx_2  iddx_3  iddx_4  iddx_5  \
0         NaN     Benign  Benign     NaN     NaN     NaN     NaN
1  IL_6727506     Benign  Benign     NaN     NaN     NaN     NaN
2         NaN     Benign  Benign     NaN     NaN     NaN     NaN
3         NaN     Benign  Benign     NaN     NaN     NaN     NaN
4         NaN     Benign  Benign     NaN     NaN     NaN     NaN
5         NaN     Benign  Benign     NaN     NaN     NaN     NaN
6         NaN     Benign  Benign     NaN     NaN     NaN     NaN
7         NaN     Benign  Benign     NaN     NaN     NaN     NaN
8         NaN     Benign  Benign     NaN     NaN     NaN     NaN
9         NaN     Benign  Benign     NaN     NaN     NaN     NaN

   mel_mitotic_index  mel_thick_mm  tbp_lv_dnn_lesion_confidence
0                NaN           NaN                     97.517282
1                NaN           NaN                      3.141455
2                NaN           NaN                     99.804040
3                NaN           NaN                     99.989998
4                NaN           NaN                     70.442510
5                NaN           NaN                     99.619603
6                NaN           NaN                     99.918133
7                NaN           NaN                     99.972390
8                NaN           NaN                     99.818963
9                NaN           NaN                     99.999690

[10 rows x 55 columns]
```

0.3 Dataset Stats

[4]:
```
def df_stats(df: pd.DataFrame, include_all: bool = False):
    """
    Print statistics and null value counts for a pandas DataFrame.

    Parameters:
        df (pd.DataFrame): The DataFrame to analyze.
        include_all (bool): If True, include all columns in the descriptive␣
 ↪statistics; otherwise, include only numeric columns.

    Returns:
        None
    """
    if df.empty:
        print("The DataFrame is empty.")
        return

    # Print descriptive statistics
    print("Descriptive Statistics:")
    if include_all:
        print(df.describe(include='all'))
    else:
        print(df.describe(include=[np.number]))
    print("\n" + "-"*50 + "\n")  # Separator for clarity

    # Print the number of null values per column
    print("Null Value Counts:")
    print(df.isnull().sum())
    print("\n" + "-"*50 + "\n")  # Separator for clarity

    # Additional information: Percentage of null values per column
    print("Percentage of Null Values:")
    print(df.isnull().mean() * 100)
    print("\n" + "-"*50 + "\n")  # Separator for clarity

    # Number of rows and columns
    print(f"Number of rows: {df.shape[0]}")
    print(f"Number of columns: {df.shape[1]}")
    print("\n" + "-"*50 + "\n")  # Separator for clarity
```

[5]:
```
df_stats(df_metadata)
```

```
Descriptive Statistics:
              target     age_approx  clin_size_long_diam_mm        tbp_lv_A  \
count  401059.000000  398261.000000           401059.000000  401059.000000
mean        0.000980      58.012986                3.930827      19.974007
std         0.031288      13.596165                1.743068       3.999489
```

```
min          0.000000       5.000000                1.000000      -2.487115
25%          0.000000      50.000000                2.840000      17.330821
50%          0.000000      60.000000                3.370000      19.801910
75%          0.000000      70.000000                4.380000      22.304628
max          1.000000      85.000000               28.400000      48.189610

         tbp_lv_Aext       tbp_lv_B    tbp_lv_Bext       tbp_lv_C  \
count  401059.000000  401059.000000  401059.000000  401059.000000
mean       14.919247      28.281706      26.913015      34.786341
std         3.529384       5.278676       4.482994       5.708469
min        -9.080269      -0.730989       9.237066       3.054228
25%        12.469740      24.704372      23.848125      31.003148
50%        14.713930      28.171570      26.701704      34.822580
75%        17.137175      31.637429      29.679913      38.430298
max        37.021680      54.306900      48.372700      58.765170

         tbp_lv_Cext       tbp_lv_H  …  tbp_lv_radial_color_std_max  \
count  401059.000000  401059.000000  …                401059.000000
mean       30.921279      54.653689  …                     1.016459
std         4.829345       5.520849  …                     0.734631
min        11.846520      -1.574164  …                     0.000000
25%        27.658285      51.566273  …                     0.563891
50%        30.804893      55.035632  …                     0.902281
75%        33.963868      58.298184  …                     1.334523
max        54.305290     105.875784  …                    11.491140

         tbp_lv_stdL  tbp_lv_stdLExt  tbp_lv_symm_2axis  \
count  401059.000000   401059.000000      401059.000000
mean        2.715190        2.238605           0.306823
std         1.738165        0.623884           0.125038
min         0.268160        0.636247           0.052034
25%         1.456570        1.834745           0.211429
50%         2.186693        2.149758           0.282297
75%         3.474565        2.531443           0.382022
max        17.563650       25.534791           0.977055

       tbp_lv_symm_2axis_angle       tbp_lv_x       tbp_lv_y       tbp_lv_z  \
count            401059.000000  401059.000000  401059.000000  401059.000000
mean                 86.332073      -3.091862    1039.598221      55.823389
std                  52.559511     197.257995     409.819653      87.968245
min                   0.000000    -624.870728   -1052.134000    -291.890442
25%                  40.000000    -147.022125     746.519673      -8.962647
50%                  90.000000      -5.747253    1172.803000      67.957947
75%                 130.000000     140.474835    1342.131540     126.611567
max                 175.000000     614.471700    1887.766846     319.407000

       mel_thick_mm  tbp_lv_dnn_lesion_confidence
count     63.000000                  4.010590e+05
```

```
mean       0.670952                 9.716220e+01
std        0.792798                 8.995782e+00
min        0.200000                 1.261082e-16
25%        0.300000                 9.966882e+01
50%        0.400000                 9.999459e+01
75%        0.600000                 9.999996e+01
max        5.000000                 1.000000e+02

[8 rows x 37 columns]

--------------------------------------------------

Null Value Counts:
isic_id                         0
target                          0
patient_id                      0
age_approx                   2798
sex                         11517
anatom_site_general          5756
clin_size_long_diam_mm          0
image_type                      0
tbp_tile_type                   0
tbp_lv_A                        0
tbp_lv_Aext                     0
tbp_lv_B                        0
tbp_lv_Bext                     0
tbp_lv_C                        0
tbp_lv_Cext                     0
tbp_lv_H                        0
tbp_lv_Hext                     0
tbp_lv_L                        0
tbp_lv_Lext                     0
tbp_lv_areaMM2                  0
tbp_lv_area_perim_ratio         0
tbp_lv_color_std_mean           0
tbp_lv_deltaA                   0
tbp_lv_deltaB                   0
tbp_lv_deltaL                   0
tbp_lv_deltaLB                  0
tbp_lv_deltaLBnorm              0
tbp_lv_eccentricity             0
tbp_lv_location                 0
tbp_lv_location_simple          0
tbp_lv_minorAxisMM              0
tbp_lv_nevi_confidence          0
tbp_lv_norm_border              0
tbp_lv_norm_color               0
tbp_lv_perimeterMM              0
```

```
tbp_lv_radial_color_std_max          0
tbp_lv_stdL                          0
tbp_lv_stdLExt                       0
tbp_lv_symm_2axis                    0
tbp_lv_symm_2axis_angle              0
tbp_lv_x                             0
tbp_lv_y                             0
tbp_lv_z                             0
attribution                          0
copyright_license                    0
lesion_id                       379001
iddx_full                            0
iddx_1                               0
iddx_2                          399991
iddx_3                          399994
iddx_4                          400508
iddx_5                          401058
mel_mitotic_index               401006
mel_thick_mm                    400996
tbp_lv_dnn_lesion_confidence         0
dtype: int64

--------------------------------------------------

Percentage of Null Values:
isic_id                           0.000000
target                            0.000000
patient_id                        0.000000
age_approx                        0.697653
sex                               2.871647
anatom_site_general               1.435200
clin_size_long_diam_mm            0.000000
image_type                        0.000000
tbp_tile_type                     0.000000
tbp_lv_A                          0.000000
tbp_lv_Aext                       0.000000
tbp_lv_B                          0.000000
tbp_lv_Bext                       0.000000
tbp_lv_C                          0.000000
tbp_lv_Cext                       0.000000
tbp_lv_H                          0.000000
tbp_lv_Hext                       0.000000
tbp_lv_L                          0.000000
tbp_lv_Lext                       0.000000
tbp_lv_areaMM2                    0.000000
tbp_lv_area_perim_ratio           0.000000
tbp_lv_color_std_mean             0.000000
tbp_lv_deltaA                     0.000000
```

```
tbp_lv_deltaB                  0.000000
tbp_lv_deltaL                  0.000000
tbp_lv_deltaLB                 0.000000
tbp_lv_deltaLBnorm             0.000000
tbp_lv_eccentricity            0.000000
tbp_lv_location                0.000000
tbp_lv_location_simple         0.000000
tbp_lv_minorAxisMM             0.000000
tbp_lv_nevi_confidence         0.000000
tbp_lv_norm_border             0.000000
tbp_lv_norm_color              0.000000
tbp_lv_perimeterMM             0.000000
tbp_lv_radial_color_std_max    0.000000
tbp_lv_stdL                    0.000000
tbp_lv_stdLExt                 0.000000
tbp_lv_symm_2axis              0.000000
tbp_lv_symm_2axis_angle        0.000000
tbp_lv_x                       0.000000
tbp_lv_y                       0.000000
tbp_lv_z                       0.000000
attribution                    0.000000
copyright_license              0.000000
lesion_id                     94.500061
iddx_full                      0.000000
iddx_1                         0.000000
iddx_2                        99.733705
iddx_3                        99.734453
iddx_4                        99.862614
iddx_5                        99.999751
mel_mitotic_index             99.986785
mel_thick_mm                  99.984292
tbp_lv_dnn_lesion_confidence   0.000000
dtype: float64

--------------------------------------------------

Number of rows: 401059
Number of columns: 55

--------------------------------------------------
```

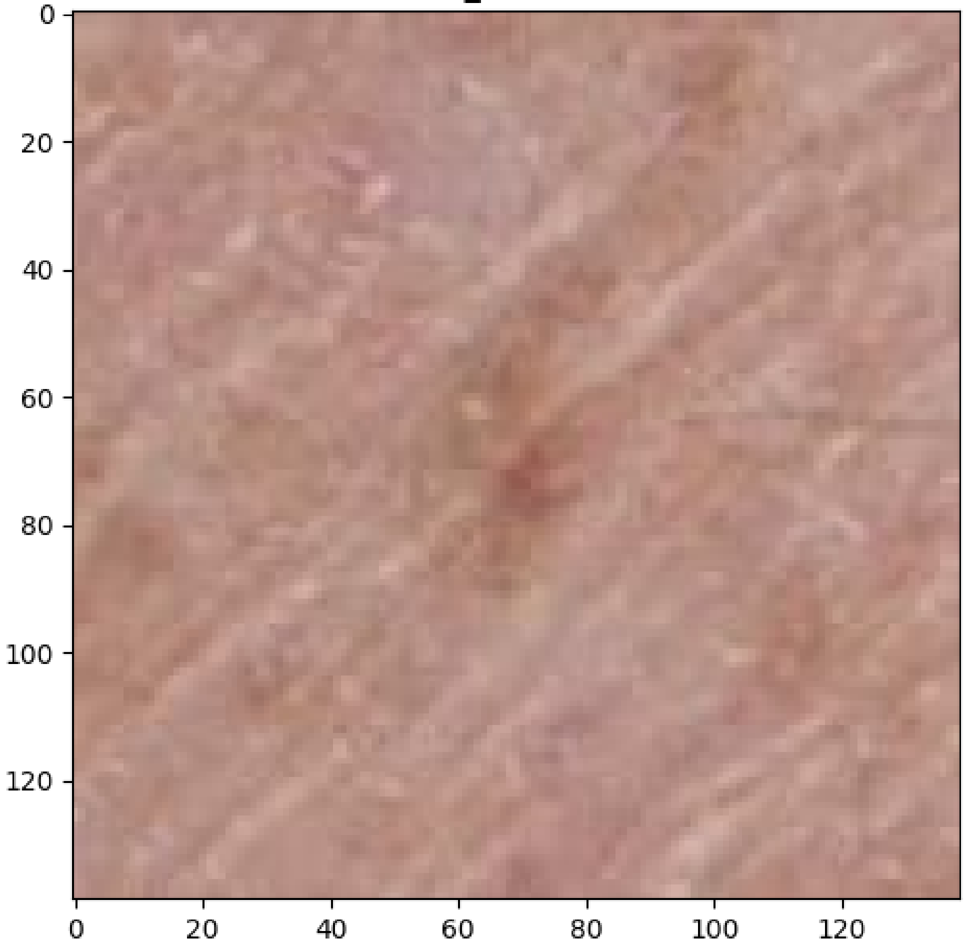
Load Image Byte String

[29]:
```python
import h5py

def load_image_from_hdf5(isic_id: str,
                         file_path: str = "../data/raw/train-image.hdf5",
                              n_channels: int = 3):
    # Handle the case where the isic_id is passed incorrectly
    if not isic_id.lower().startswith("isic"):
        isic_id = f"ISIC_{int(str(isic_id).split('_', 1)[-1]):>07}"

    # Open the HDF5 file in read mode
    with h5py.File(file_path, 'r') as hf:

        # Retrieve the image data from the HDF5 dataset using the provided ISIC␣
↪ID
        try:
            image_data = hf[isic_id][()]
        except KeyError:
            raise KeyError(f"ISIC ID {isic_id} not found in HDF5 file.")

        # Convert the binary data to a numpy array
        image_array = np.frombuffer(image_data, np.uint8)

        # Decode the image from the numpy array
        if n_channels == 3:
            # Load the image as a color image (BGR) and convert to RGB
            image = cv2.cvtColor(cv2.imdecode(image_array, cv2.IMREAD_COLOR),␣
↪cv2.COLOR_BGR2RGB)
        else:
            # Load the image as a grayscale image
            image = cv2.imdecode(image_array, cv2.IMREAD_GRAYSCALE)

        # If the image failed to load for some reason (problems decoding) ...
        if image is None:
            raise ValueError(f"Could not decode image for ISIC ID: {isic_id}")

        return image
```

[30]:

```
plt.figure(figsize=(6,6))
plt.title("ISIC_0015670", fontweight="bold")
plt.imshow(load_image_from_hdf5("ISIC_0015670"))
plt.show()
```

ISIC_0015670



```
[ ]:
```