```
import os
import pandas as pd
import numpy as np
import requests
import tensorflow as tf
from PIL import Image
from io import BytesIO
import matplotlib.pyplot as plt


# Define file paths and create directories
DATA_DIR = 'nga_dataset'
IMAGE_DIR = os.path.join(DATA_DIR, 'images')
PROCESSED_DIR = os.path.join(DATA_DIR, 'processed')

os.makedirs(DATA_DIR, exist_ok=True)
os.makedirs(IMAGE_DIR, exist_ok=True)
os.makedirs(PROCESSED_DIR, exist_ok=True)

# Function to filter portrait paintings
def filter_portrait_paintings(objects_df, terms_df, images_df):
    # Standardize column names
    objects_df.columns = [col.lower() for col in objects_df.columns]
    images_df.columns = [col.lower() for col in images_df.columns]

    # Filter only paintings
    paintings = objects_df[objects_df['classification'].str.contains('painting', case=False, na=False)]
    print(f"Found {len(paintings)} paintings")

    # Identify portraits
    portraits = paintings[paintings['title'].str.contains('portrait|self portrait', case=False, na=False)]
    print(f"Found {len(portraits)} potential portraits by title")

    # Identify object ID column in images dataset
    id_column = 'depictstmsobjectid' if 'depictstmsobjectid' in images_df.columns else 'objectid'

    # Merge portraits with images
    portrait_images = pd.merge(portraits, images_df, left_on='objectid', right_on=id_column, how='inner')

    # Filter for primary view images
    if 'viewtype' in portrait_images.columns:
        portrait_images = portrait_images[portrait_images['viewtype'] == 'primary']

    print(f"Final dataset contains {len(portrait_images)} portrait paintings with images")
    return portrait_images

# Load datasets
objects = pd.read_csv(os.path.join(DATA_DIR, 'objects.csv'), low_memory=False)
images = pd.read_csv(os.path.join(DATA_DIR, 'published_images.csv'), low_memory=False)
terms = pd.read_csv(os.path.join(DATA_DIR, 'objects_terms.csv'), low_memory=False)

# Filter portraits
portrait_data = filter_portrait_paintings(objects, terms, images)


# Download and preprocess images
def download_and_preprocess_images(portrait_data, target_size=(224, 224), max_images=100):
    portrait_data = portrait_data.head(max_images)
    processed_data = []

    for idx, row in portrait_data.iterrows():
        object_id = row['objectid']
        base_url = row['iiifurl'].rstrip("/")
        image_url = f"{base_url}/full/600,/0/default.jpg"

        image_filename = f"{object_id}.jpg"
        image_path = os.path.join(IMAGE_DIR, image_filename)

        if os.path.exists(image_path):
            print(f"Image {image_filename} already exists, skipping download.")
        else:
            try:
                response = requests.get(image_url, timeout=10)
                response.raise_for_status()

                img = Image.open(BytesIO(response.content))
```

```
                img.save(image_path)
                print(f"Saved image to {image_path}")
            except requests.exceptions.RequestException as e:
                print(f"Error downloading image for object {object_id}: {e}")
                continue

        processed_data.append({'objectID': object_id, 'title': row.get('title', 'Unknown'), 'image_path': image_path})

    return pd.DataFrame(processed_data)

processed_portraits = download_and_preprocess_images(portrait_data, max_images=None)
processed_portraits.to_csv(os.path.join(PROCESSED_DIR, 'portrait_dataset.csv'), index=False)


# Create TensorFlow dataset
def create_tf_dataset(df, target_size=(299, 299), batch_size=32):
    if df.empty:
        print("Error: Empty dataframe, cannot create dataset")
        return None

    def load_and_preprocess_image(image_path):
        img = tf.io.read_file(image_path)
        img = tf.image.decode_jpeg(img, channels=3)
        img = tf.image.resize(img, target_size)
        img = img / 255.0
        return img

    dataset = tf.data.Dataset.from_tensor_slices((df['image_path'].values, df['objectID'].values))
    dataset = dataset.map(lambda x, y: (load_and_preprocess_image(x), y), num_parallel_calls=tf.data.experimental.AUTOTUNE)
    return dataset.batch(batch_size).prefetch(tf.data.experimental.AUTOTUNE)

tf_dataset = create_tf_dataset(processed_portraits)

print("Dataset preparation complete!")
```

```
Found 4376 paintings
Found 253 potential portraits by title
Final dataset contains 197 portrait paintings with images
Image 19.jpg already exists, skipping download.
Image 25.jpg already exists, skipping download.
Image 38.jpg already exists, skipping download.
Image 49.jpg already exists, skipping download.
Image 59.jpg already exists, skipping download.
Image 89.jpg already exists, skipping download.
Image 1167.jpg already exists, skipping download.
Image 1206.jpg already exists, skipping download.
Image 1207.jpg already exists, skipping download.
Image 1235.jpg already exists, skipping download.
Image 37636.jpg already exists, skipping download.
Image 46496.jpg already exists, skipping download.
Image 46518.jpg already exists, skipping download.
Image 46628.jpg already exists, skipping download.
Image 50293.jpg already exists, skipping download.
Image 50722.jpg already exists, skipping download.
Image 51096.jpg already exists, skipping download.
Image 52257.jpg already exists, skipping download.
Image 166431.jpg already exists, skipping download.
Image 205907.jpg already exists, skipping download.
Image 224391.jpg already exists, skipping download.
Image 1095.jpg already exists, skipping download.
Image 1131.jpg already exists, skipping download.
Image 3209.jpg already exists, skipping download.
Image 34085.jpg already exists, skipping download.
Image 34070.jpg already exists, skipping download.
Image 34072.jpg already exists, skipping download.
Image 34083.jpg already exists, skipping download.
Image 34092.jpg already exists, skipping download.
Image 34096.jpg already exists, skipping download.
Image 34117.jpg already exists, skipping download.
Image 34130.jpg already exists, skipping download.
Image 34135.jpg already exists, skipping download.
Image 34146.jpg already exists, skipping download.
Image 34148.jpg already exists, skipping download.
Image 42444.jpg already exists, skipping download.
Image 42485.jpg already exists, skipping download.
Image 42486.jpg already exists, skipping download.
Image 42917.jpg already exists, skipping download.
Image 43445.jpg already exists, skipping download.
Image 46025.jpg already exists, skipping download.
Image 46587.jpg already exists, skipping download.
```

```
Image 46691.jpg already exists, skipping download.
Image 52380.jpg already exists, skipping download.
Image 56231.jpg already exists, skipping download.
Image 59907.jpg already exists, skipping download.
Image 65606.jpg already exists, skipping download.
Image 67384.jpg already exists, skipping download.
Image 67401.jpg already exists, skipping download.
Image 92215.jpg already exists, skipping download.
Image 92991.jpg already exists, skipping download.
Image 98594.jpg already exists, skipping download.
Image 98593.jpg already exists, skipping download.
```