Shape Check

This repository provides tools to process radiological medical images, extract shape features, and identify anomalies in synthetic images compared to real ones using machine learning (Isolation Forest). This tool was applied to check the shape of breast boundary in mammogram images.

Purpose

Synthetic data provides a promising solution to address data scarcity for training machine learning models; however, adopting it without proper quality assessments may introduce artifacts, distortions, and unrealistic features that compromise model performance and clinical utility. This work introduces a novel knowledge-based method for detecting network-induced shape artifacts in synthetic images. The method can detect anatomically unrealistic images irrespective of the generative model used and provides interpretability through its knowledge-based design. We demonstrate the effectiveness of the method for identifying network-induced shape artifacts using two synthetic mammography datasets. A reader study further confirmed that images identified by the method as likely containing network-induced artifacts were also flagged by human readers. This method is a step forward in the responsible use of synthetic data by ensuring that synthetic images adhere to realistic anatomical and shape constraints.

Tool Reference

Deshpande, R., Thompson, Y.L.E., and Zamzmi, G. (2025). ShapeCheck: Feature extraction tool to identify anomalies in synthetic images. https://github.com/DIDSR/ShapeCheck/

Disclaimer

About the Catalog of Regulatory Science Tools

The enclosed tool is part of the Catalog of Regulatory Science Tools, which provides a peer- reviewed resource for stakeholders to use where standards and qualified Medical Device Development Tools (MDDTs) do not yet exist. These tools do not replace FDA-recognized standards or MDDTs. This catalog collates a variety of regulatory science tools that the FDA's Center for Devices and Radiological Health's (CDRH) Office of Science and Engineering Labs (OSEL) developed. These tools use the most innovative science to support medical device development and patient access to safe and effective medical devices. If you are considering using a tool from this catalog in your marketing submissions, note that these tools have not been qualified as Medical Device Development Tools and the FDA has not evaluated the suitability of these tools within any specific context of use. You may request feedback or meetings for medical device submissions as part of the Q-Submission Program. For more information about the Catalog of Regulatory Science Tools, email OSEL_CDRH@fda.hhs.gov.

Installation

Install all dependencies using the provided requirements.txt file:

```
pip install -r requirements.txt
```

Required Packages

The following Python packages include:

- numpy
- scipy
- matplotlib
- pandas
- scikit-image
- scikit-learn

The full list of required packages can be found in requirements.txt. Project was tested on Python 3.9.4.

Project Structure

```
.
├── process_datasets.py       # Script to process all real/synthetic datasets and
save feature data
├── detect_shape_anomaly.py   # Script to detect anomalies using Isolation Forest
├── ImageProcessor.py         # Class to extract angular features from individual
images
├── offsets.py                # Utility toolbox to define offsets of 3 x 3 and 5 x
5 neignhors
├── requirements.txt          # Dependencies for the project
├── README.md                 # You are here
```

Input Format

Users can reference a folder for all input images or create an inputs folder within this working directoary. Images should be organized under by their dataset names:

```
inputs/
└── VinDrReal/         # Real images
└── VinDrSynthetic/    # Synthetic counterparts
```

All files are expected to be in .png format.

Software Usage

1. Extract Shape Features

Modify the dataset paths inside `process_datasets.py` (L35). Dataset should come in pairs, one with real dataset and one with the corresponding synthetic data that was generated using the real dataset. This tool was developed to compare the shape feature in the real dataset and the corresponding synthetic dataset. However, the code would still work if two real datasets or two synthetic datasets are provided to compare their shape feature. All images must be in PNG format.

Run the process_datasets.py script to process real and synthetic images:

```
python process_datasets.py --first_n_files 10 --verbose --outpath ./outputs/
```

This generates a pickled dictionary `data.p` file containing pixel-wise shape descriptors and angular gradient distributions.

2. Detect Shape Anomalies Use the processed data to detect anomalies:

```
python detect_shape_anomaly.py --data_path ./outputs/data.p --out_path ./outputs/
--do_plots
```

The `--data_path` points to the output from the previous step, and `--out_path` is where output csv and (optional) plots will be stored.

Optional flags:

- `--verbose`: Print detailed progress
- `--do_plots`: Save visual plots of results
- `--bad_percentile` / `--good_percentile`: Customize what qualifies as anomalous (default: 0.1 / 99.9)

Output

Two outputs are expected:

- `data.p`: coordinates of edge pixels and their angles, and the normalized angular gradient distributions
- `shape_anomaly_results.csv`: Per-image anomaly scores, percentiles, and rankings

When `--do_plots` is enabled, the following will be saved:

- extreme_images/: Visuals of the best and worst shape-quality synthetic images
- anomaly_score_distribution.png: Histogram of anomaly scores and edge feature distributions

How It Works

- ImageProcessor.py: Extracts 1-pixel-wide breast boundaries and computes angular gradients.
- process_datasets.py: Applies ImageProcessor to all images and prepares a feature set for modeling.

- detect_shape_anomaly.py: Trains an Isolation Forest on real data and flags anomalies in the synthetic dataset.

Example

All per-image processing are done using the ImageProcessor class. Here is a snippet of how it can be called within python.

```
# Use the ImageProcessor class directly
from ImageProcessor import ImageProcessor

processor = ImageProcessor('/path/to/image.png', 'VinDrReal', 'real')
processor.isMLO = True
processor.do_intermediate_plots = True
processor.do_plots = True
processor.build_angle_gradients()
print(processor.binned_angle_gradients)
```

Relevant Publications

- Deshpande, R., Lago, M., Subbaswamy, A., Kahaki, S., Delfino, J.G., Badano, A. and Zamzmi, G., In Medical Imaging with Deep Learning 2025. A knowledge-based method for detecting network-induced shape artifacts in synthetic images. https://openreview.net/forum?id=BAEwCzDmPB#discussion

Contact

For any questions/suggestions/collaborations, please contact Rucha Deshpande (rucha.deshpande@fda.hhs.gov) or Elim Thompson (yeelamelim.thompson@fda.hhs.gov).