PolyShape-2D: a synthetic control dataset for rotation equivariance

This repository contains the generation script for the **PolyShape-2D** dataset, a synthetic benchmark designed to rigorously test the properties of rotation-equivariant neural networks, specifically SFCNN from this paper

About the dataset

PolyShape-2D is a synthetic "toy problem" designed to provide a controlled environment for analyzing how SFCNN handle 2D rotation. The core philosophy is to create a clean, unambiguous unit test for rotation equivariance by isolating rotation as the sole significant variable.

Unlike real-world datasets (e.g., CIFAR-10) or more complex benchmarks (e.g., Rotated-MNIST), PolyShape-2D is free of confounding factors such as texture, illumination, intra-class shape variance, and semantic ambiguity. This allows for any observed difference in model performance to be attributed directly and unambiguously to its handling of rotation.

Dataset description

- **Classes**: The dataset consists of 8 classes of regular geometric shapes, chosen to span a spectrum of discrete (C_n) and continuous ($SO(2)$) symmetry groups:
    1. `0_triangle` (C_3)
    2. `1_square` (C_4)
    3. `2_pentagon` (C_5)
    4. `3_hexagon` (C_6)
    5. `4_heptagon` (C_7)
    6. `5_octagon` (C_8)
    7. `6_star` (a non-convex shape with C_5 symmetry)
    8. `7_circle` (approximated, representing $SO(2)$)
- **Image Properties**:
    - **Resolution**: 64×64 pixels
    - **Channels**: 1 (grayscale)
    - **Normalization**: All shapes are centered, scaled uniformly, and rendered as solid white on a black background.
    - **Rendering**: Vector-based rotation and supersampling are used to ensure high-quality, anti-aliased images.

Intended Use

This dataset is specifically designed to test two central hypotheses related to rotation-equivariant models like SFCNNs:

1. **H1 (Rotational Generalization)**: Can a model trained on a single, fixed orientation (e.g., "upright only") generalize to classify the same objects at novel, unseen angles?

2. **H2 (Sample Efficiency)**: Does an equivariant model require significantly fewer randomly rotated examples to achieve the same level of accuracy as a standard CNN?

Instructions

1. Installation

First, clone this repository to your local machine. The generation script requires a few standard Python libraries. It is recommended to create a virtual environment.

```
# Create and activate a virtual environment (optional but recommended)
python -m venv venv
source venv/bin/activate  # On Windows, use `venv\Scripts\activate`

# Install the required packages
pip install -r requirements.txt
```

2. Generating the dataset

The entire generation process is contained within the PolyShape-2D.ipynb Jupyter Notebook.

- Customize (Optional): At the top of the notebook (in the first code cell), you will find the NUM_SAMPLES dictionary. You can easily modify the values in this dictionary to change the number of images generated for each dataset split.

```
# Example from the notebook:
NUM_SAMPLES = {
    'train_h1': 10,    # Change to 5000 to generate more H1 training data
    'test': 30,        # Change to 10000 for a larger test set
    # ... and so on
}
```

- Run All Cells: Run all cells in the notebook from top to bottom. The script will create a root directory named PolyShape-2D and populate it with the specified dataset splits.

The repository already includes a small, pre-generated version of the dataset for quick visualization. Running the script will overwrite this with the splits defined by the NUM_SAMPLES variable.