PatchCore ViT for Anomaly Detection

This document serves as a detailed guide to help users run the `patchcore_vit.py` script for anomaly detection. The script is designed to process datasets for training, validation, and testing, and generates anomaly heatmaps stored in a specified output directory.

Overview

The `patchcore_vit.py` script implements an anomaly detection model using PatchCore with a Vision Transformer (ViT) backbone. It allows users to specify dataset paths and output directories via command-line arguments (CLI).

Features

- Train an anomaly detection model with a specified training dataset.
- Validate the model with normal and anomalous datasets to find the threshold.
- Test the model and generate anomaly heatmaps.
- Save the heatmaps to a user-specified directory.

Prerequisites

1. Environment Setup

Ensure you have Python installed (preferably Python 3.8 or higher). Install the required dependencies by running:

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

2. Directory Structure

Prepare your dataset directory with the following structure:

```
/training/
    /normal/         # Training dataset with normal images
/validation/
    /normal/         # Validation dataset with normal images
    /zflowers/       # Validation dataset with anomalous image
s
/testing/
    /normal/         # Testing dataset with normal images
    /zflowers/       # Testing dataset with anomalous images
```

Running the Script

Step 1: Command-Line Arguments

The script uses command-line arguments to accept dataset paths and output directories. Below is a description of each argument:

Argument	Description
`--train_dir`	Path to the training dataset.
`--val_normal_dir`	Path to the normal validation dataset.
`--val_anomalous_dir`	Path to the anomalous validation dataset.
`--test_normal_dir`	Path to the normal testing dataset.
`--test_anomalous_dir`	Path to the anomalous testing dataset.
`--output_dir`	Path to the output directory where anomaly heatmaps will be saved.
`--batch_size`	(Optional) Batch size for data loaders. Default is 4.

Step 2: Example Usage

To run the script, use the following command:

```
python patchcore_vit.py \
    --train_dir "/path/to/training/normal" \
```

```
    --val_normal_dir "/path/to/validation/normal" \
    --val_anomalous_dir "/path/to/validation/zflowers" \
    --test_normal_dir "/path/to/testing/normal" \
    --test_anomalous_dir "/path/to/testing/zflowers" \
    --output_dir "./output/anomaly_heatmaps" \
    --batch_size 4
```

Step 3: Output

- The script will train the model using the training dataset.

- It will validate the model and find the best threshold.

- During the testing phase, it will generate anomaly heatmaps for anomalous images and calculate metrics such as accuracy, confusion matrix, and AUROC.

- The heatmaps will be saved in the specified output directory (e.g., `./output/anomaly_heatmaps`).

Understanding the Outputs

Heatmaps

- Each heatmap highlights the areas in an image where anomalies are detected.

- Saved in the output directory with filenames like `anomaly_visual_0.png` , `anomaly_visual_1.png` , etc.

Logs

- The script prints the progress of training, validation, and testing phases in the terminal.

- Metrics such as accuracy, confusion matrix, and AUROC are displayed for performance evaluation.

Troubleshooting

1. **Missing Dependencies**:
   If you encounter errors related to missing libraries, ensure you have installed all dependencies from `requirements.txt` .

2. **Invalid Dataset Paths**:
   Double-check the paths provided in the CLI arguments. Ensure they point to valid directories containing images.

3. **Output Directory Not Found**:
   If the output directory does not exist, the script will create it automatically.

Additional Information

Supported Image Formats

- The script supports `.jpg` , `.jpeg` , and `.png` image formats.

Batch Size

- The batch size can be adjusted based on your system's memory capacity. A larger batch size may speed up training but requires more memory.