

VQ-Seg: an Unsupervised Anomaly Detection Method based on VQGAN

Description

In this work, we propose a method called VQ-Seg that combines the latent representation of VQGAN with a Transformer model and three masking strategies on the latent representation index sequences to enable unsupervised anomaly detection and segmentation.

Dataset Preparation

The dataset used must store images in the file directory and set them as folders for train, test, and val. In the test folder, image files such as defect, norm, and mask must be stored, and the suffix name should be ". PNG" or ". png"

Program Structure

-----VQ-Seg

-----/configs____: *Dataset Configuration File*

-----/Dataset____: *Data Images File*

-----/label____

-----/train_images

-----/test_images

-----/validation_images

-----/rec_imgs____: *Result Images File*

-----/taming____: *Network Training Tools File*

-----/tf-logs____: *Model Training Parameters File*

-----main.py____: *Runing File*

-----test.py__: *Testing File*

Runing Methods

If you want to train your own model, please enter the following code at the terminal:

```
input(python .\main.py --base .\configs\class10.yaml --train True --devices 1)

#model parameters saved in tf-logs dir
```

If you want to use the saved model parameters for direct testing, you can modify the model parameter file in the main file:

```
in main.py --
model.init_from_ckpt('tf-logs/2023-09-07T09-52-18_mvtec/check_points/last.ckpt')

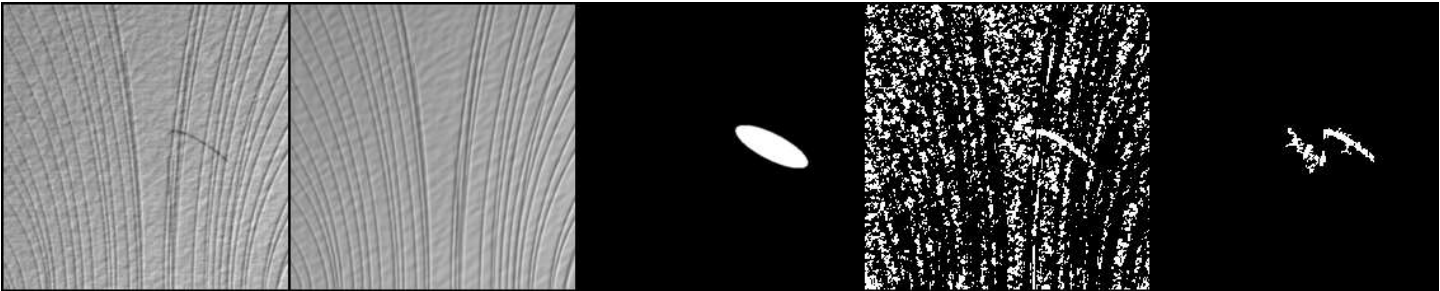
trainer.test(model, data)

in terminal input:
python .\main.py --base .\configs\class10.yaml --train False --devices 1
```

Result Images

Save output image to rec_imgs folder, each image from left to right represents "Ori, Rec, Mas, In-Seg, Otsu-Seg"

As shown below:



-
-
-

Model Framework

