CS782FP Report

Rui Hong

rhong5@gmu.edu

Computer Science Department

George Mason University

Contents

1 What the origin paper did

1.1 architecture

The paper [1] proposes a deep background subtraction method based on conditional Generative Adversarial Network (cGAN). The proposed model consists of two successive networks: generator and discriminator. The generator learns the mapping from the observing input (i.e., image and background), to the output (i.e., foreground mask). Then, the discriminator learns a loss function to train this mapping by comparing real foreground (i.e., ground-truth) and fake foreground (i.e., predicted output) with observing the input image and background.

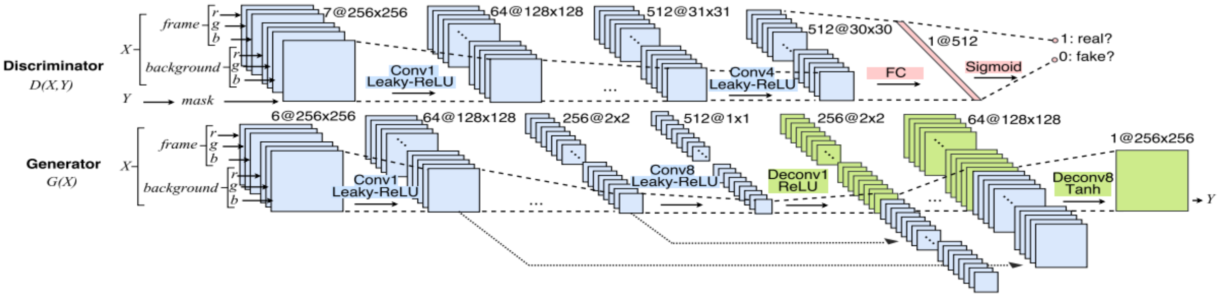


Figure 1 Architecture of the proposed cGAN model for background subtraction.

1.2 dataset and experiments

Evaluating the model performance with two public datasets, CDnet 2014 and BMC, shows that the proposed model outperforms the state-of-the-art methods.

1.2.1 Dataset

CDnet 2014

The CDnet 2014 dataset is a benchmark dataset for foreground/background segmentation in video sequences. It was created as part of the Change Detection Challenge (CDnet) held in 2014. The dataset consists of 11 video sequences captured in various scenarios, including indoor and outdoor environments, different lighting conditions, and different types of foreground objects.

Each video sequence in the CDnet 2014 dataset contains a ground truth annotation that specifies the foreground and background regions for each frame. The foreground regions indicate the objects of interest that need to be segmented from the background in change detection tasks. The total image number of CDnet 2014 is over 300000.

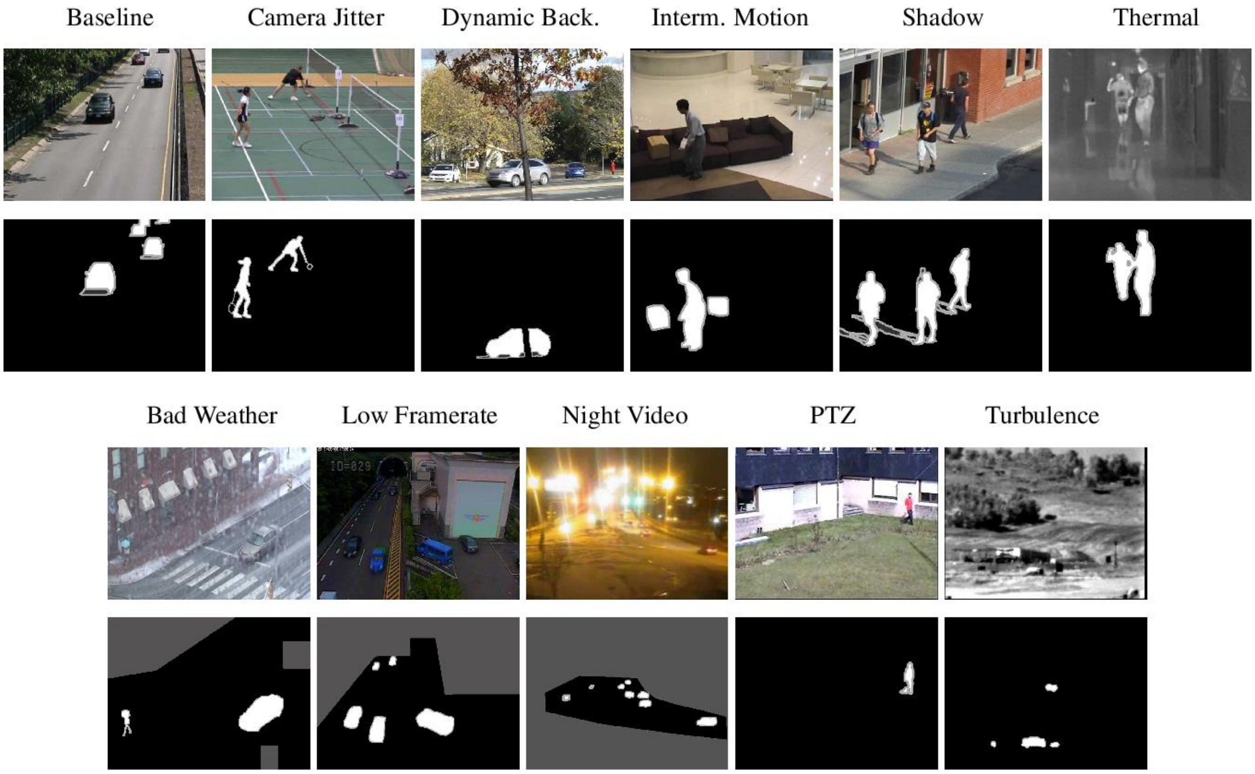


Figure 2. the example of CDnet 2014

There are background images for every category in the dataset except for foreground samples as shown below.

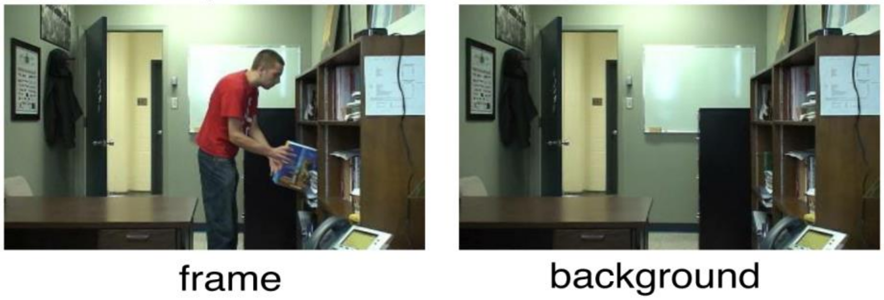


Figure 3. the foreground and background examples of CDnet 2014

BMC (Background Models Challenge)

It is a benchmark dataset and evaluation that contains synthetic videos representing urban scenes acquired from a static camera. It focuses on outdoor situations with weather variations such as wind, sun or rain. This dataset is composed of 20 synthetic urban videos.

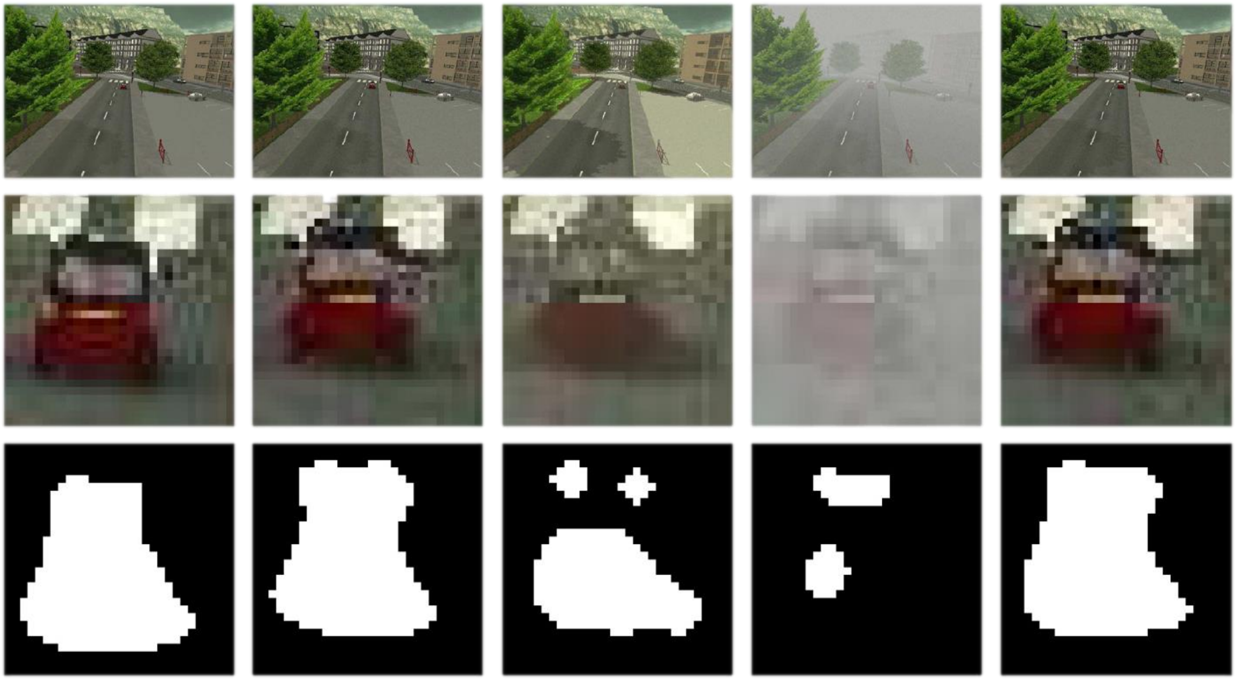


Figure 4. the example of BMC (Background Models Challenge) dataset

Similarly, there are background images for every category in the.

1.2.2 experiments

On the two datasets, the proposed method calculated the metric F-mesure.

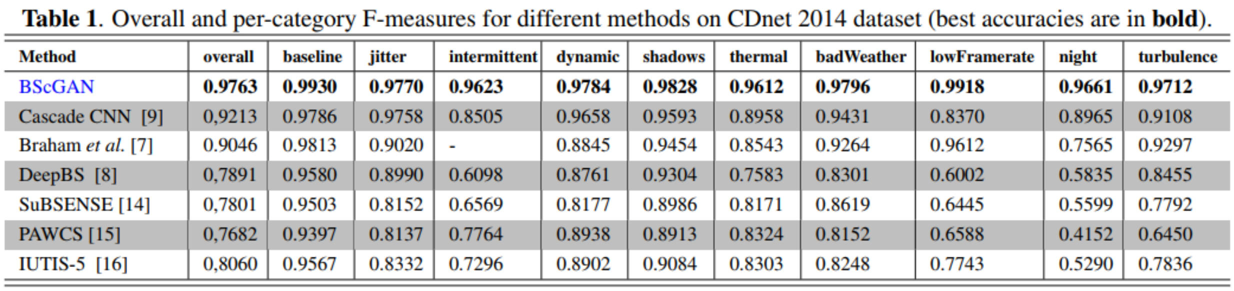


Table 1. Overall and per-category F-measures for different methods on CDnet 2014 dataset (best accuracies are in **bold**).

Method	overall	baseline	jitter	intermittent	dynamic	shadows	thermal	badWeather	lowFramerate	night	turbulence
BScGAN	**0.9763**	**0.9930**	**0.9770**	**0.9623**	**0.9784**	**0.9828**	**0.9612**	**0.9796**	**0.9918**	**0.9661**	**0.9712**
Cascade CNN [9]	0,9213	0.9786	0.9758	0.8505	0.9658	0.9593	0.8958	0.9431	0.8370	0.8965	0.9108
Braham *et al.* [7]	0.9046	0.9813	0.9020	-	0.8845	0.9454	0.8543	0.9264	0.9612	0.7565	0.9297
DeepBS [8]	0,7891	0.9580	0.8990	0.6098	0.8761	0.9304	0.7583	0.8301	0.6002	0.5835	0.8455
SuBSENSE [14]	0,7801	0.9503	0.8152	0.6569	0.8177	0.8986	0.8171	0.8619	0.6445	0.5599	0.7792
PAWCS [15]	0,7682	0.9397	0.8137	0.7764	0.8938	0.8913	0.8324	0.8152	0.6588	0.4152	0.6450
IUTIS-5 [16]	0,8060	0.9567	0.8332	0.7296	0.8902	0.9084	0.8303	0.8248	0.7743	0.5290	0.7836

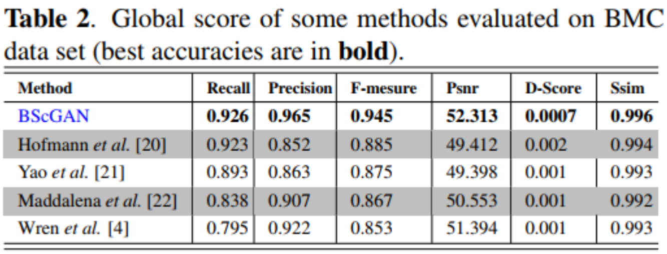


Table 2. Global score of some methods evaluated on BMC data set (best accuracies are in **bold**).

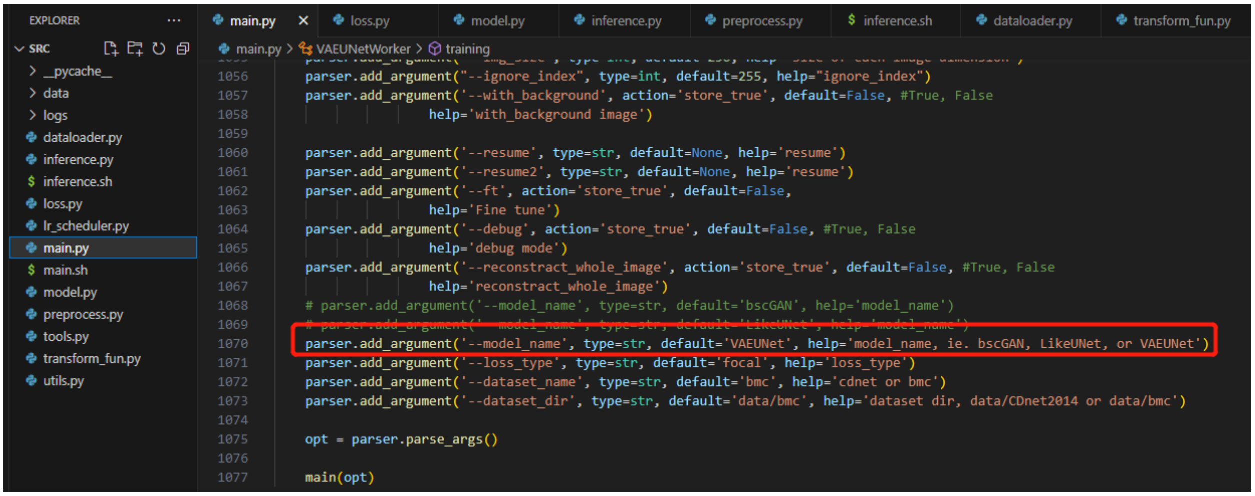
Method	Recall	Precision	F-mesure	Psnr	D-Score	Ssim
BScGAN	**0.926**	**0.965**	**0.945**	**52.313**	**0.0007**	**0.996**
Hofmann *et al.* [20]	0.923	0.852	0.885	49.412	0.002	0.994
Yao *et al.* [21]	0.893	0.863	0.875	49.398	0.001	0.993
Maddalena *et al.* [22]	0.838	0.907	0.867	50.553	0.001	0.992
Wren *et al.* [4]	0.795	0.922	0.853	51.394	0.001	0.993

2 what I did

2.1 implement the original method.

Because no code of the original proposed method is provided, I implemented the project with Pytorch for training and evaluating the proposed method, comparing with my proposal methods.

The implementation code can be found here, https://github.com/hongrui16/VAESegNets.git



```
    parser.add_argument('--img_size', type=int, default=256, help='size of each image dimension')
    parser.add_argument("--ignore_index", type=int, default=255, help="ignore_index")
    parser.add_argument('--with_background', action='store_true', default=False, #True, False
                    help='with_background image')

    parser.add_argument('--resume', type=str, default=None, help='resume')
    parser.add_argument('--resume2', type=str, default=None, help='resume')
    parser.add_argument('--ft', action='store_true', default=False,
                    help='Fine tune')
    parser.add_argument('--debug', action='store_true', default=False, #True, False
                    help='debug mode')
    parser.add_argument('--reconstract_whole_image', action='store_true', default=False, #True, False
                    help='reconstract_whole_image')
    # parser.add_argument('--model_name', type=str, default='bscGAN', help='model_name')
    # parser.add_argument('--model_name', type=str, default='LikeUNet', help='model_name')
    parser.add_argument('--model_name', type=str, default='VAEUNet', help='model_name, ie. bscGAN, LikeUNet, or VAEUNet')
    parser.add_argument('--loss_type', type=str, default='focal', help='loss_type')
    parser.add_argument('--dataset_name', type=str, default='bmc', help='cdnet or bmc')
    parser.add_argument('--dataset_dir', type=str, default='data/bmc', help='dataset dir, data/CDnet2014 or data/bmc')

    opt = parser.parse_args()

    main(opt)
```

Figure 5. the hyperparameters of implementation project

Set the hyperparameter “model_name” to “bscGAN”. It is the proposed model in the paper [1].

2.2 my proposed new method -- LikeUNet

After setting the hyperparameter “model_name” to “bscGAN”, we will get “LikeUNet” model.

As its name suggests, its structure is like to U-Net, the structure is as follows:

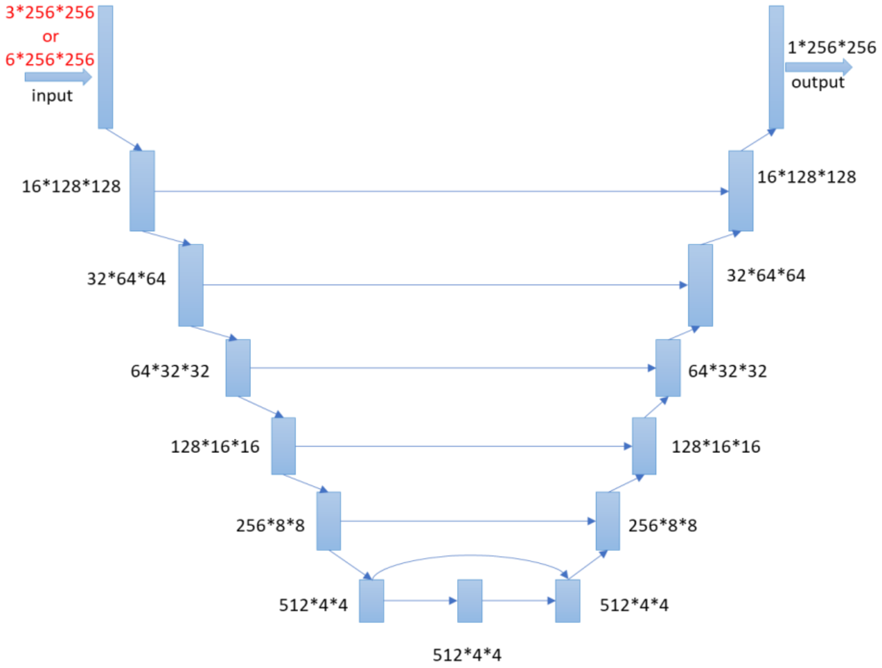


Figure 6. the structure of “LikeUNet”

The input of LikeUNet has two modes: one is a combination of a foreground and a background image for six channels, the other is to use a 3-channel RGB image as input.

Since the negative samples outnumber the positive samples in the dataset, there exists an unbalanced problem between foreground and background categories.

To solve this problem, focal loss is introduced to calculate segmentation loss.

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t).$$

2.3 my proposed new method – VAEUNet

As its name suggests, the model “VAEUNet” is a combination of VAE and LikeUNet. The blue part of the model is the encoder, the green part is decoder of segmentation, and yellow part is the VAE decoder.

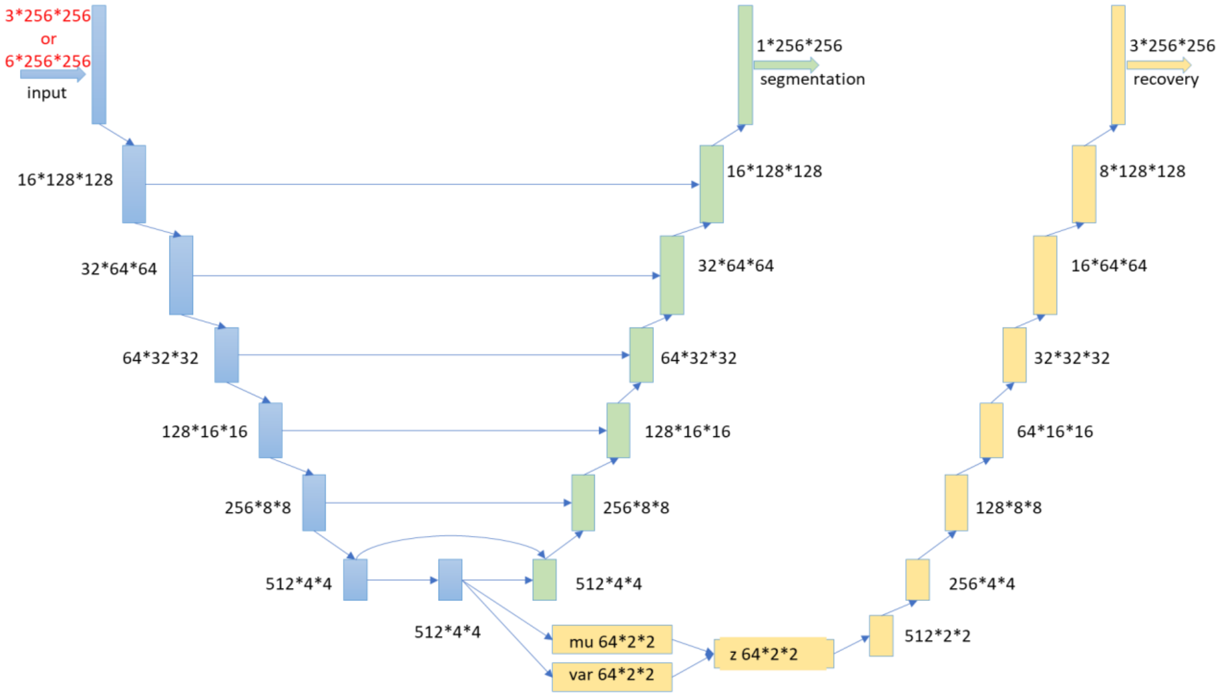


Figure 7. the structure of "VAEUNet"

VAE and segmentation model share the same encoder. The latent vector of VAE is sampled from the output of the encoder.

The idea of using a VAE is to enhance the encoder's ability to learn meaningful representations from input data.

For VAEUNet, the total loss is composed of three parts, segmentation loss, KL loss, and reconstruction loss.

The segmentation loss also used focal loss.

The KL loss is defined as:

$$D_{KL}(P||Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)}$$

The reconstruction loss is Mean Squared Error (MSE), also called L2 Loss, defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

To prevent the segmentation loss from being overshadowed by the VEA loss due to significant differences in their values, the weight ratio is introduced. The three weight ratios can be dynamically adjusted based on the training epoch.

*lambda_seg = 1000 - 50*epoch if 1000 - 50*epoch > 500 else 500*

lambda_kl = 0.01(10*epoch + 1) if 0.01*(10*epoch + 1) <= 10 else 10* *lambda_rec = 0.01*

*loss = seg_loss*lambda_seg + rec_loss*lambda_rec + kl_loss*lambda_kl*

where seg_loss, rec_loss, and kl_loss stand for segmentation loss, KL loss, and reconstruction loss, respectively.

2.4 training and evaluation experiments

As there are over three hundred thousand samples of CDnet 2014, I cannot use my laptop with a single GPU to do all experiments. Uploading the whole dataset to Colab and Kaggle for training the models are also full of problems. So, I select six thousand samples and their corresponding ground truth from CDnet 2014 to make a mini dataset which is called “mini CDnet”, to do experiments.

Similarly, the whole dataset of BMC is also too large to train and evaluate on my laptop. After converting the video into images, I have selected 3200 pairs of samples from the original training dataset to create a new training dataset. Additionally, I have chosen 1600 pairs of samples from the original evaluation dataset to form a new test dataset. This newly created dataset is referred to as "mini BMC."

In the proposed paper, F-measure is calculated, which is also called F1 score.

F-measure = 2TP / (2TP+FP+FN)

Apart from the F-measure, I also calculated the mIoU, which is a widely used metric in segmentation.

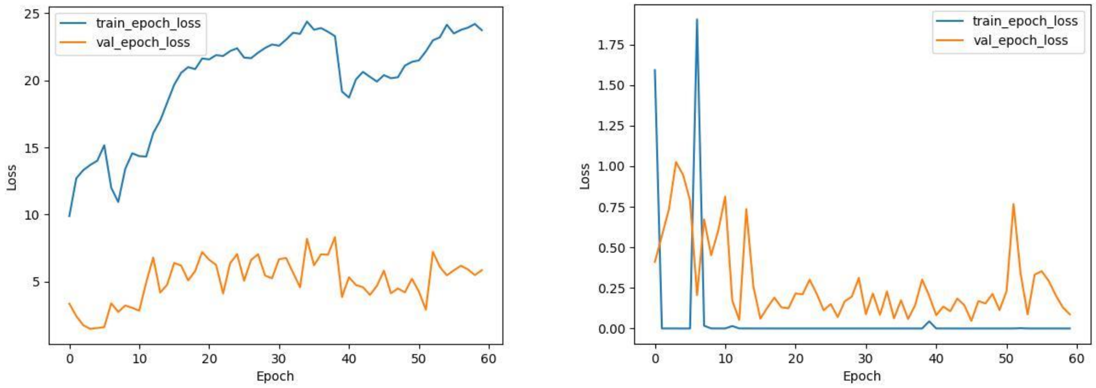
IoU = TP / (TP+FP+FN)

mIoU means the average IoU over all categories.

Due to the high proportion of negative sample (background) pixels in the dataset, in the experiment, I only consider metrics such as F-measure and mIoU for the positive samples (foreground).

2.4.1 training and evaluation of bscGAN

Firstly, I did the training and evaluation of “bscGAN” on the mini CDnet. The loss figures are as follows:

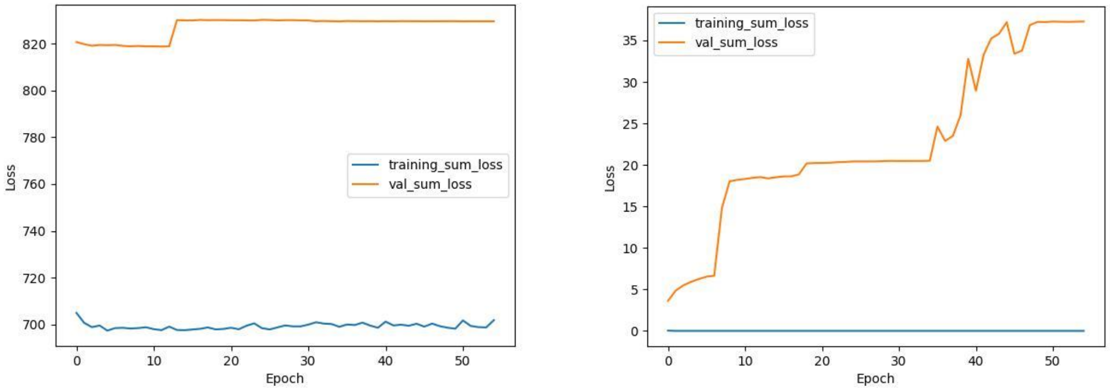


a: the loss of the generator.

b: the loss of the discriminator.

Figure 8. the loss curves of bscGAN on mini CDnet

I also did training and evaluation for “bscGAN” on “mini BMC.”



a: the loss of the generator.

b: the loss of the discriminator.

Figure 9. the loss curves of bscGAN on mini BMC

Due to the significant percentage of background pixels in the dataset, we only calculate the foreground categories metrics.

Table 3. the metrics of bscGAN on the two datasets.

dataset	F-measure	mIoU
mini CDnet	0.190300	0.133200
mini BMC	0.121100	0.097000

Unfortunately, the bscGAN did not converge on the two mini datasets.

2.4.2 training and evaluation of LikeUNet

LikeUNet has two types of input: one is 3-channel RGB image, the other one is 6-channel data, which is composed of an RGB image with foreground objects and an RGB background image.

a. Experiment with background images as the input

This experiment is done by taking a 6-channel input image, the training and validation loss decrease gradually generally. The model also converged soon and got satisfactory results.

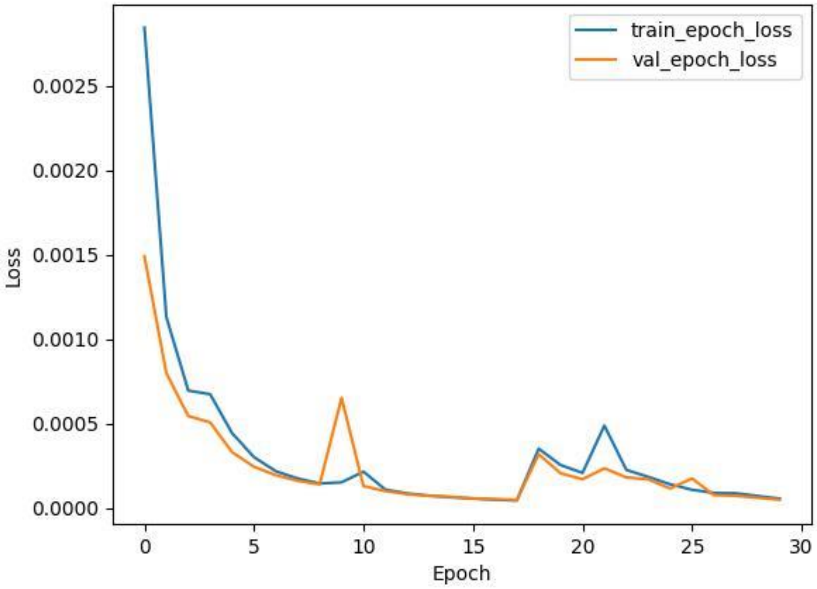


Figure 10. the loss curves of LikeUNet on mini CDnet.

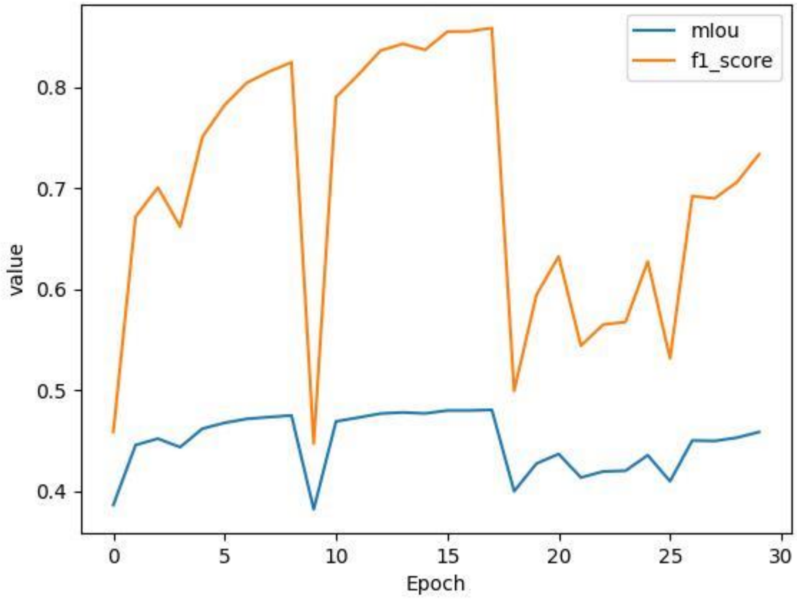


Figure 11. the metric curves of LikeUNet on mini CDnet.

Where f1_score equals to F-measure.

The best evaluation results are as follows:

Table 4. the metrics of LikeUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.480300	0.858800

b. Experiment without background images as the input.

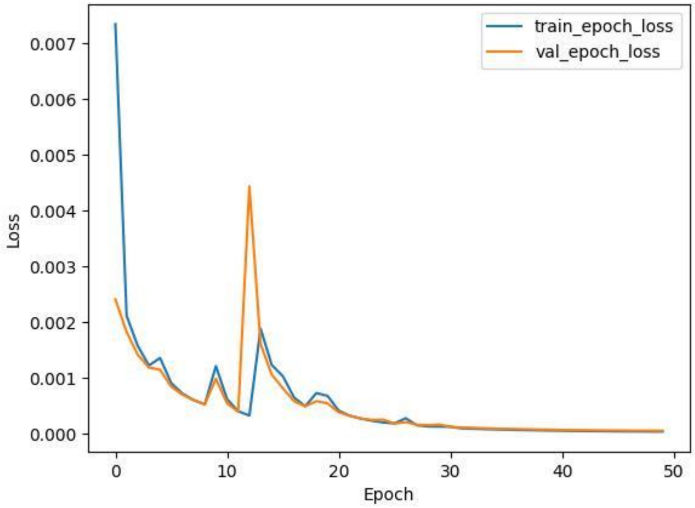


Figure 12. the loss curves of LikeUNet on mini CDnet.

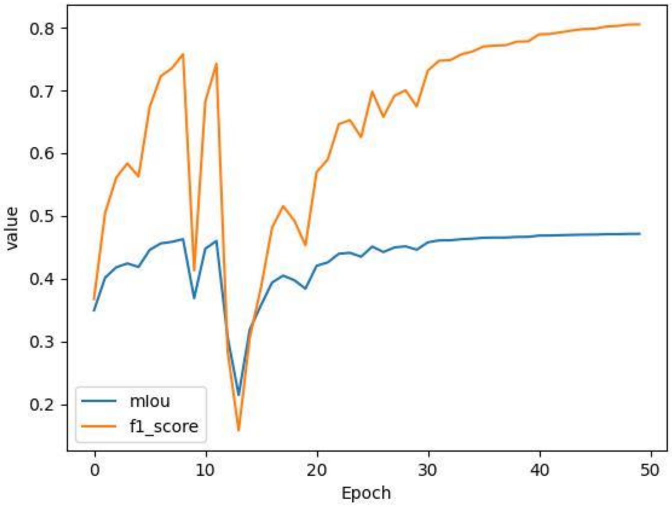


Figure 13. the metric curves of LikeUNet on mini CDnet.

Table 5. the metrics of LikeUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.471500	0.805200

Compared with experiment ‘a’ result, it is a little worse, but it still outperforms the bscGAN.

2.4.3 training and evaluation of VAEUNet

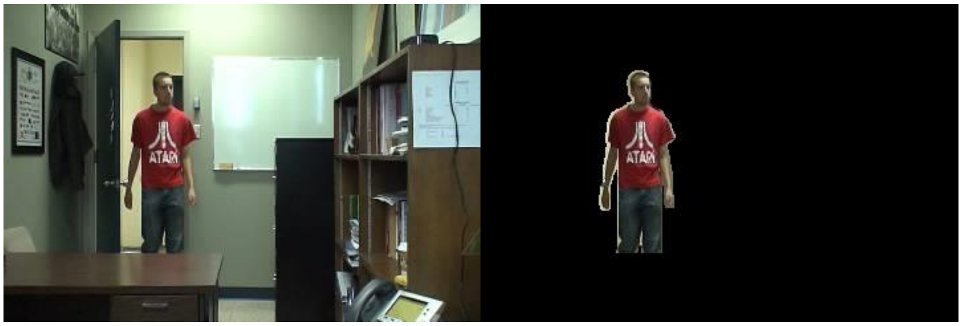
The input of VAEUNet has two modes as LikeUNet, a 3-channel and a 6-channel.

VAE reconstruction also has two modes:

1) decoder of VAE reconstructs a complete image and every pixel counts when the reconstruction loss is calculated.

2) only reconstruct the foreground objects and the foreground reconstruction loss is calculated.

So, there were four experiments done to verify the hyperparameters with different selections.



a: a complete image reconstruction. b: a reconstruction with only foreground objects.

Figure 14. an example of illustrating reconstruction.

a. Experiment with background images as the input and with only foreground objects reconstruction

A foreground object reconstruction means the decoder of VAE only reconstructs the foreground object.

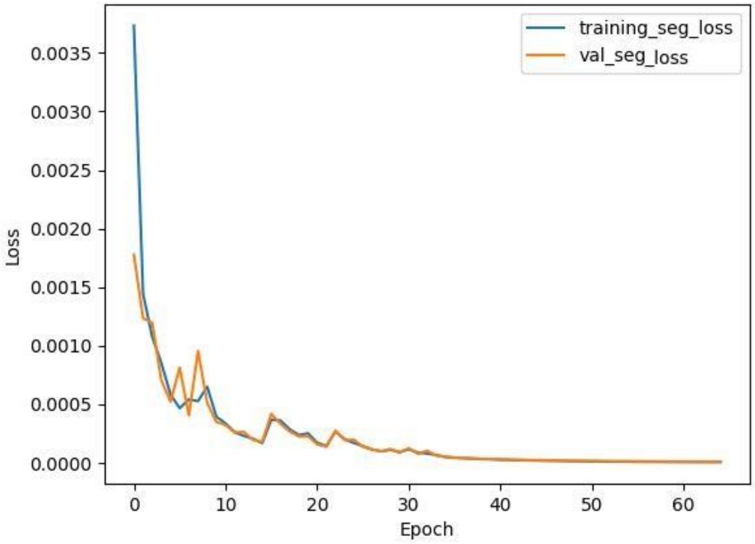


Figure 15. the segmentation loss curves of VAEUNet on mini CDnet.

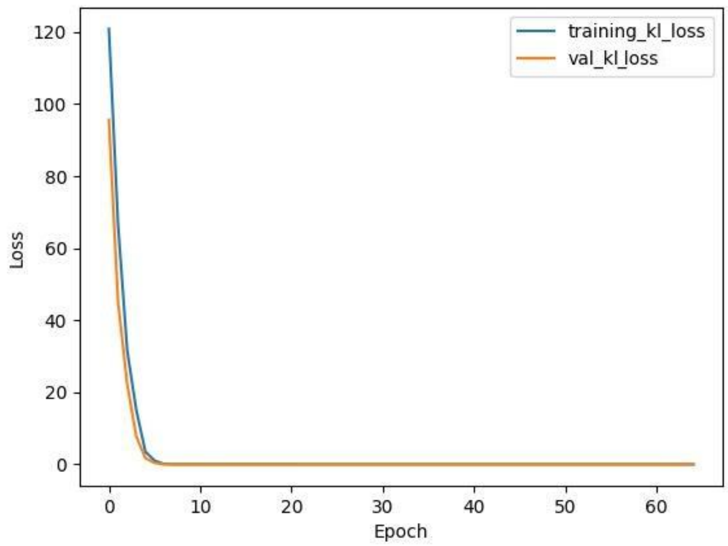


Figure 16. the KL loss curves of VAEUNet on mini CDnet.

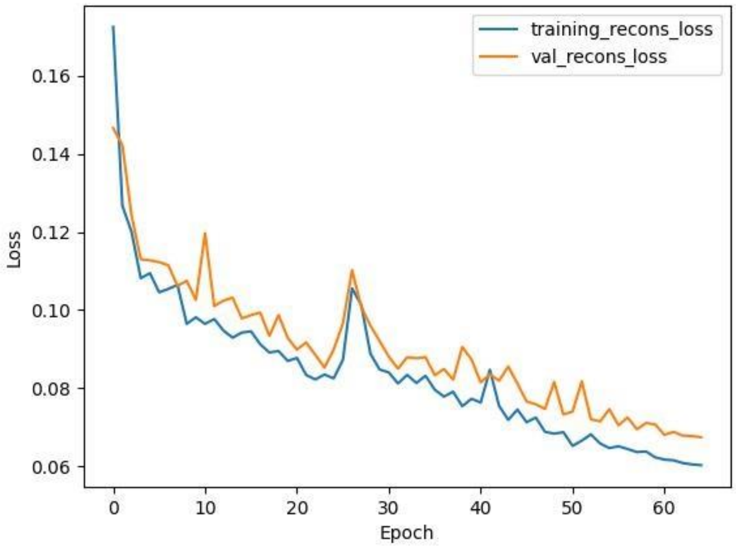


Figure 17. the reconstruction loss curves of VAEUNet on mini CDnet.

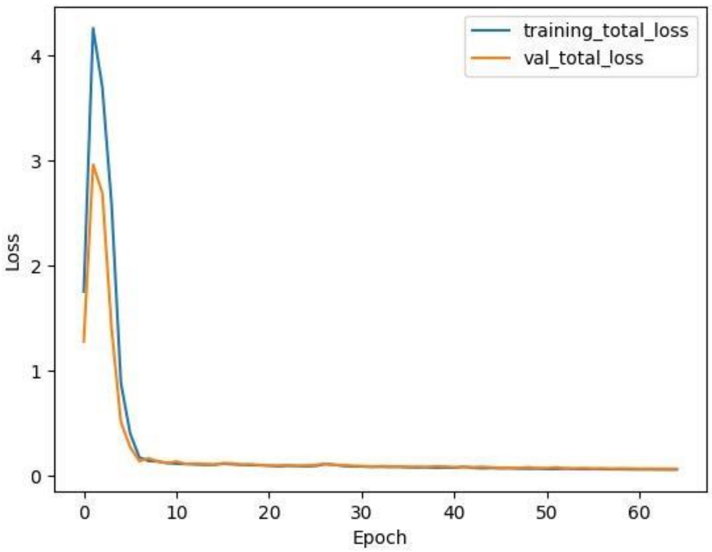


Figure 18. the total loss curves of VAEUNet on mini CDnet.

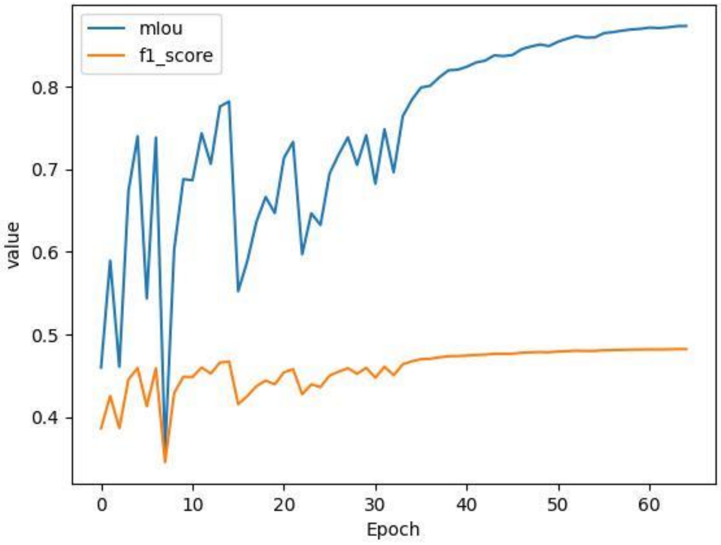


Figure 19. the metric curves of VAEUNet on mini CDnet.

Table 6. the metrics of VAEUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.482500	0.873300

b. Experiment without background images as the input and with only foreground objects reconstruction

This experiment is done without background, and the input is a 3-channel RGB image. The decoder of VAE reconstructs the foreground objects.

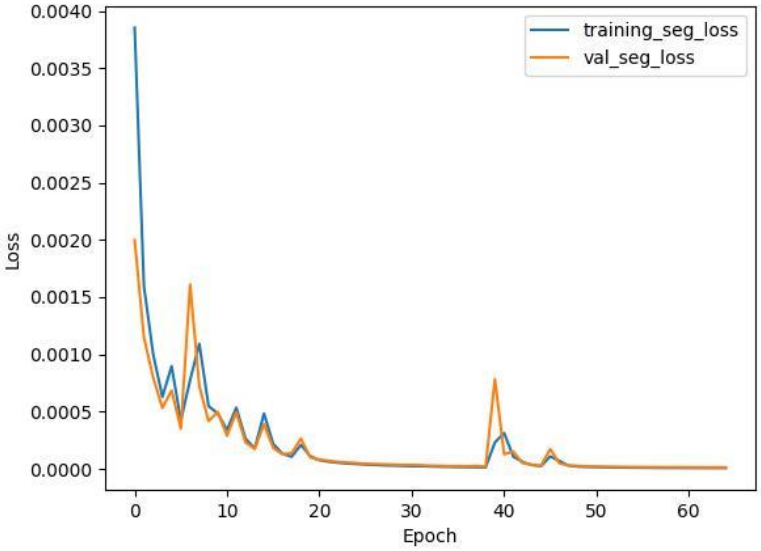


Figure 20. the segmentation loss curves of VAEUNet on mini CDnet.

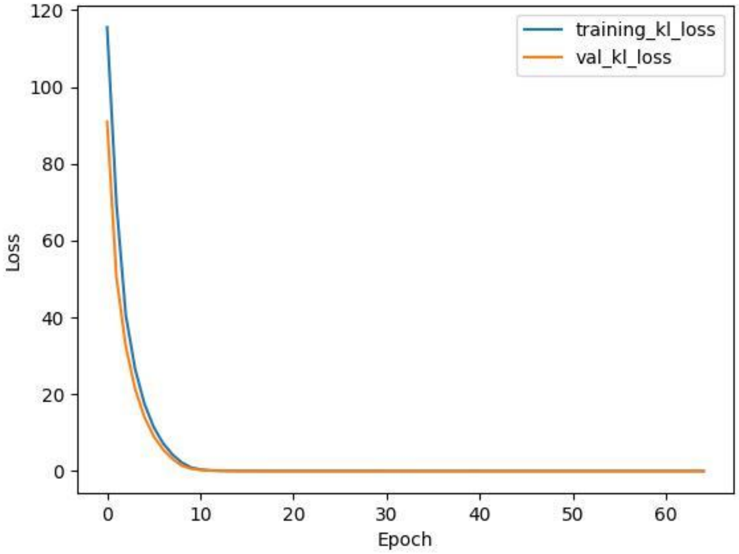


Figure 21. the KL loss curves of VAEUNet on mini CDnet.

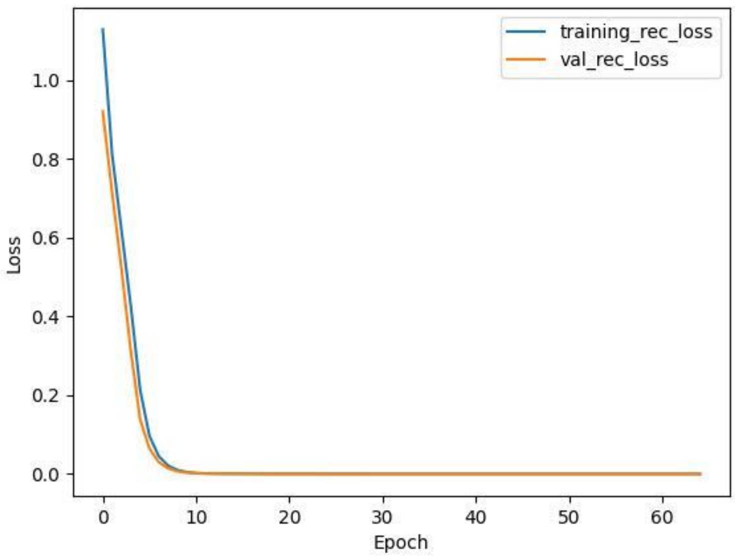


Figure 22. the reconstruction loss curves of VAEUNet on mini CDnet.

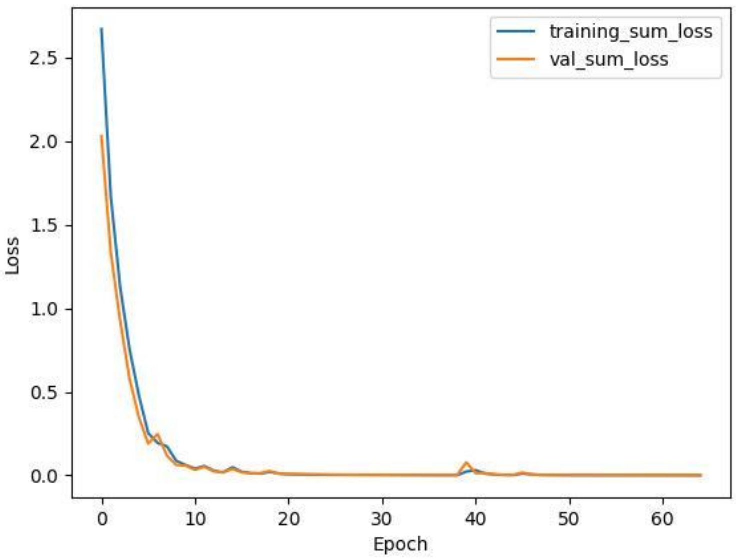


Figure 23. the total loss curves of VAEUNet on mini CDnet.

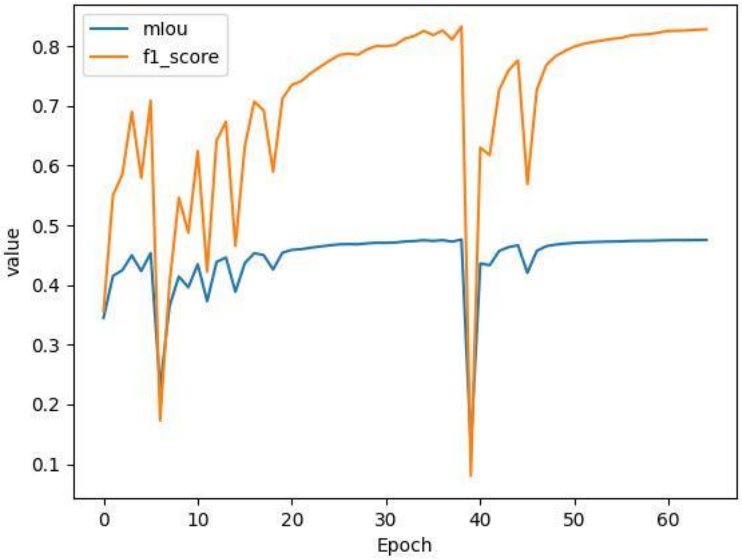


Figure 24. the metric curves of VAEUNet on mini CDnet.

Table 7. the metrics of VAEUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.475000	0.826400

c. Experiment with background images as the input and with a complete reconstruction

This experiment is done with background images and foreground input, and the decoder of VAE reconstructs a complete image. Every pixel reconstruction loss is calculated.

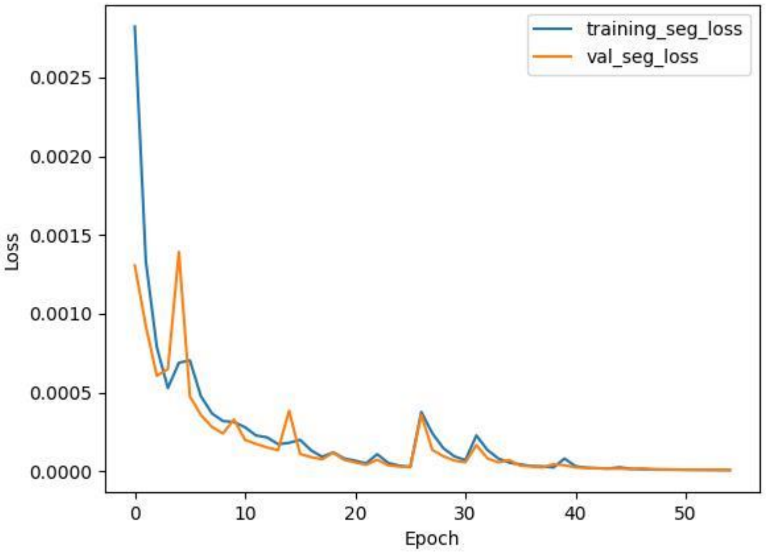


Figure 25. the segmentation loss curves of VAEUNet on mini CDnet.

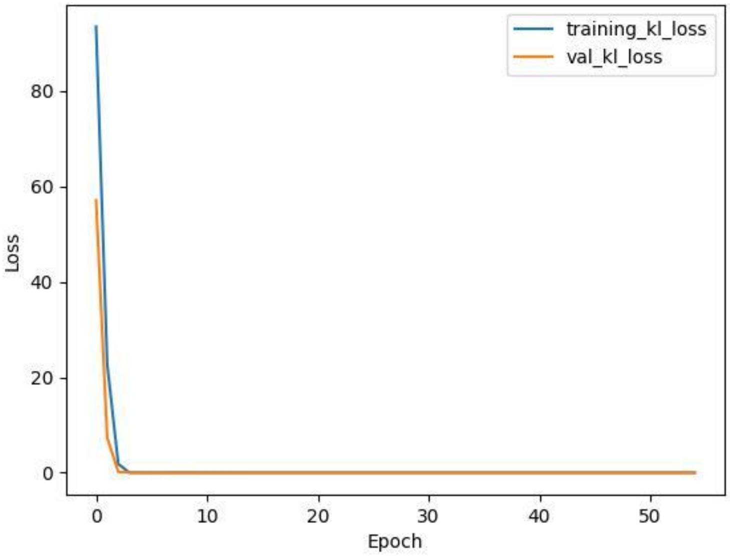


Figure 26. the KL loss curves of VAEUNet on mini CDnet.

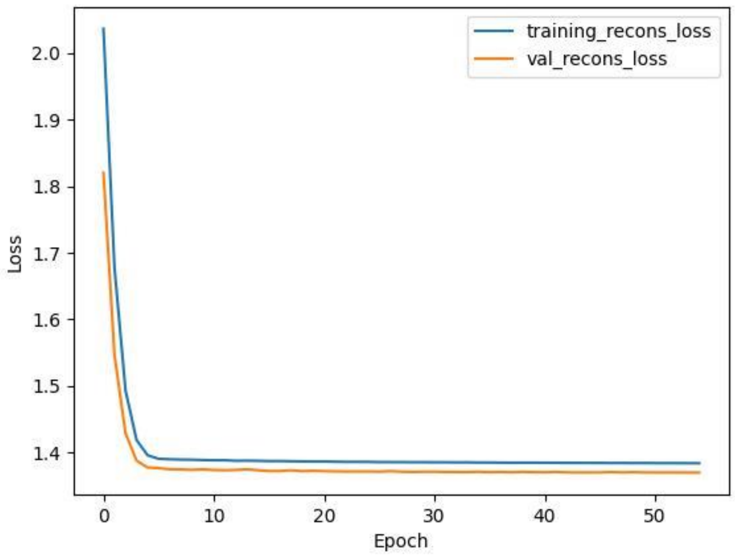


Figure 27. the reconstruction loss curves of VAEUNet on mini CDnet.

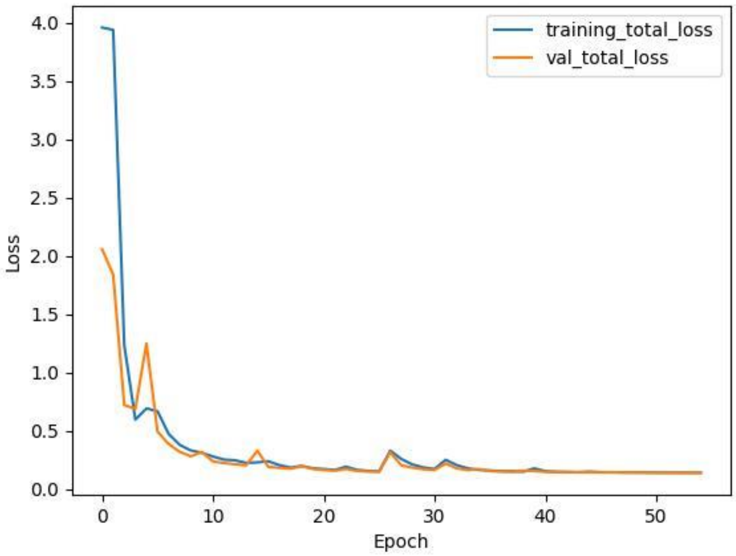


Figure 28. the total loss curves of VAEUNet on mini CDnet.

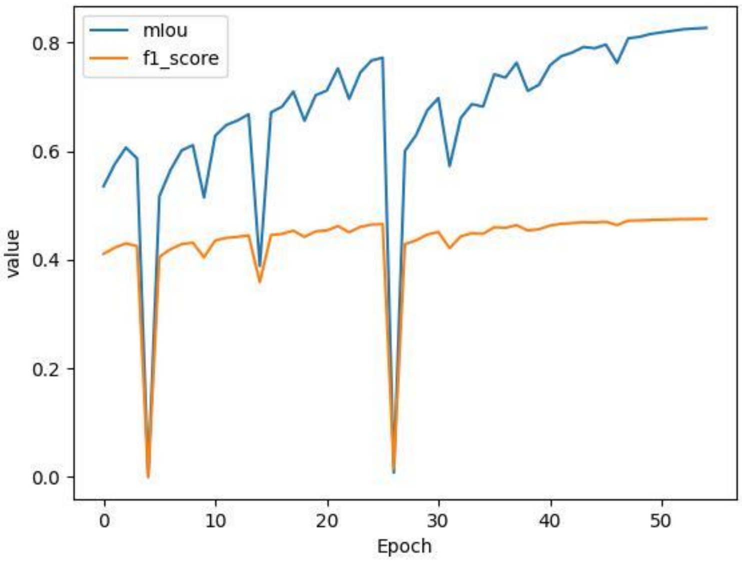


Figure 29. the metric curves of VAEUNet on mini CDnet.

Table 8. the metrics of VAEUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.475200	0.827000

d. Experiment without background images as the input and with a complete reconstruction

This experiment is done without background images, and the decoder of VAE reconstructs a complete image. Every pixel reconstruction loss is calculated.

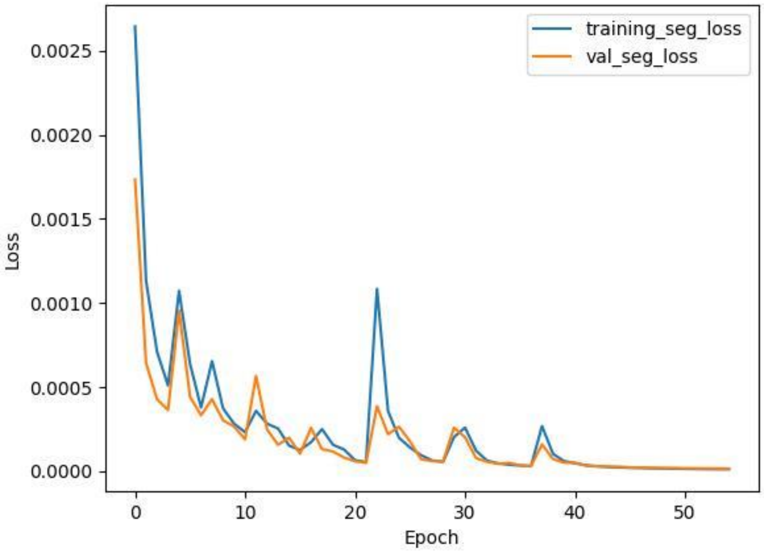


Figure 30. the segmentation loss curves of VAEUNet on mini CDnet.

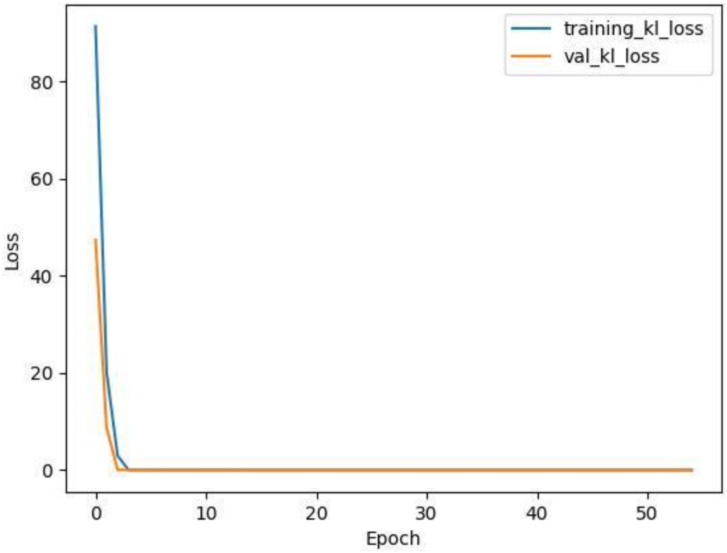


Figure 31. the KL loss curves of VAEUNet on mini CDnet.

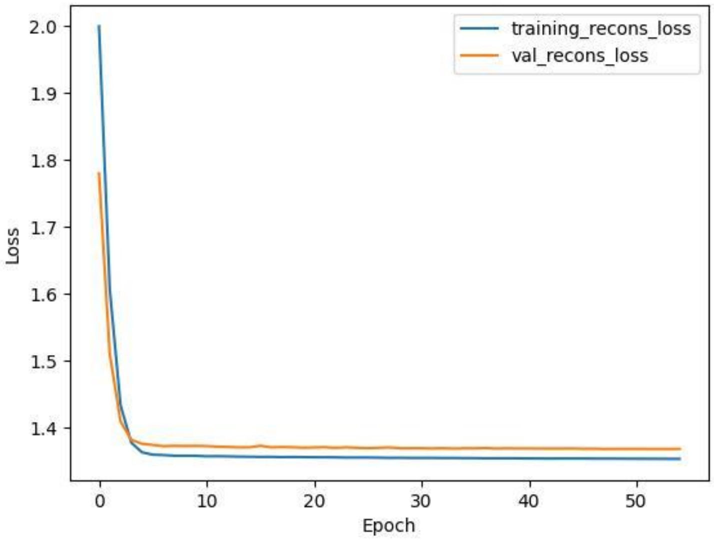


Figure 32. the reconstruction loss curves of VAEUNet on mini CDnet.

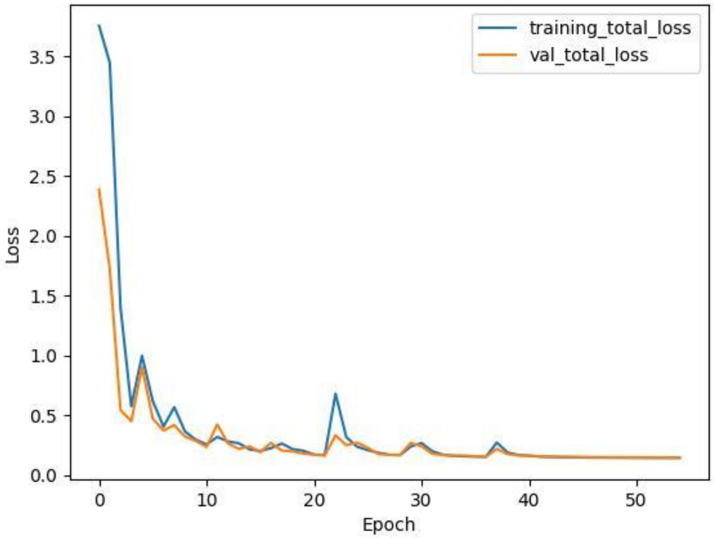


Figure 33. the total loss curves of VAEUNet on mini CDnet.

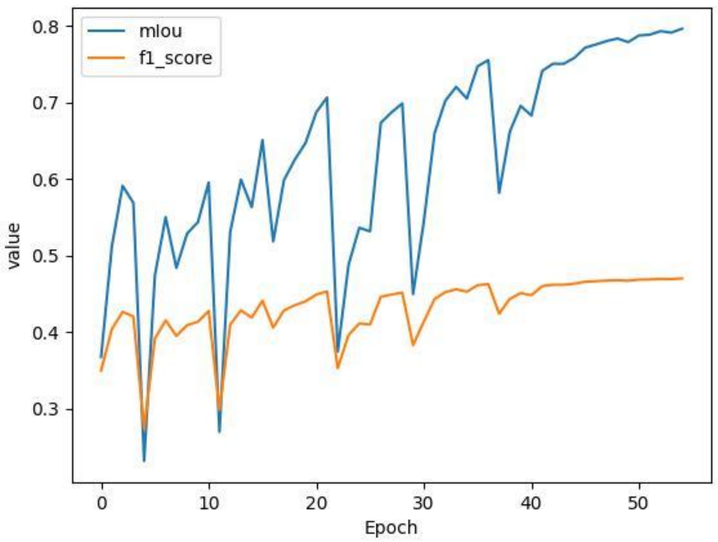


Figure 34. the metric curves of VAEUNet on mini CDnet.

Table 9. the metrics of VAEUNet on mini CDnet.

Dataset	F-measure	mIoU
mini CDnet	0.469900	0.796000

2.5 experiments results comparison and conclusion

After the above training and evaluation of different methods, we have data and figures to analyze and verify my ideas.

2.5.1 comparison

The segmentation examples of the same input image by different models and different hyperparameters are shown below.

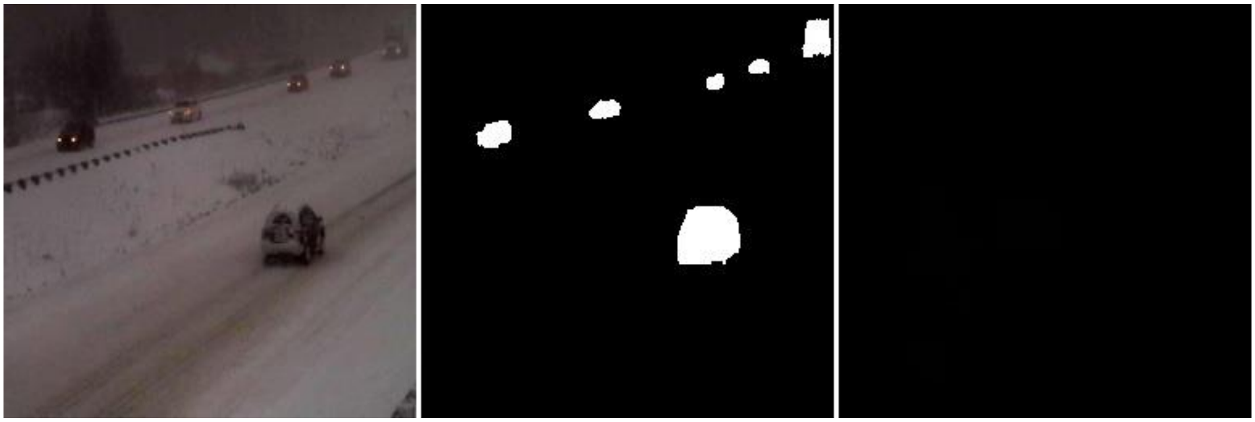


Figure 35. the generated segmentation of bscGAN (from left to right: input, GT, segmentation).

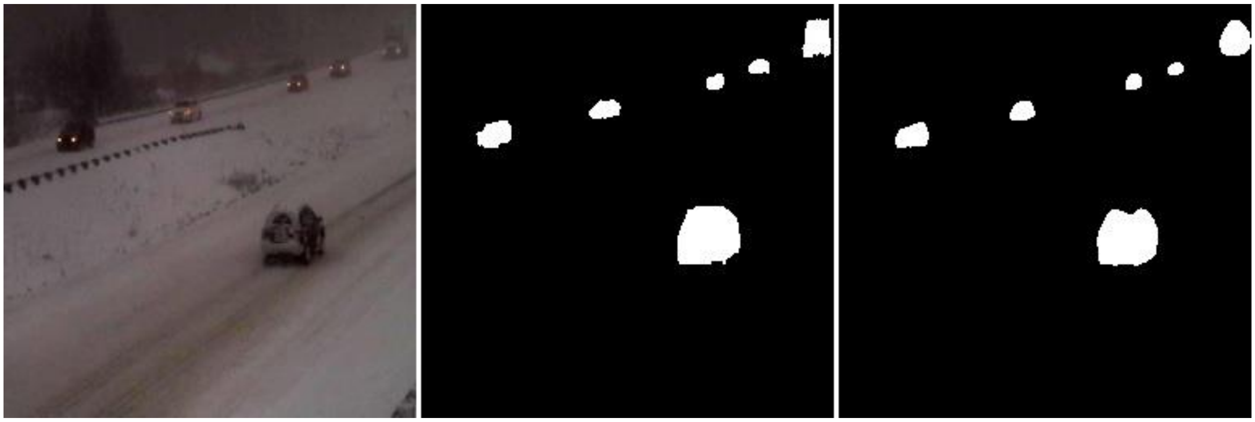


Figure 36. the segmentation of LikeUNet with background input (from left to right: input, GT, segmentation).

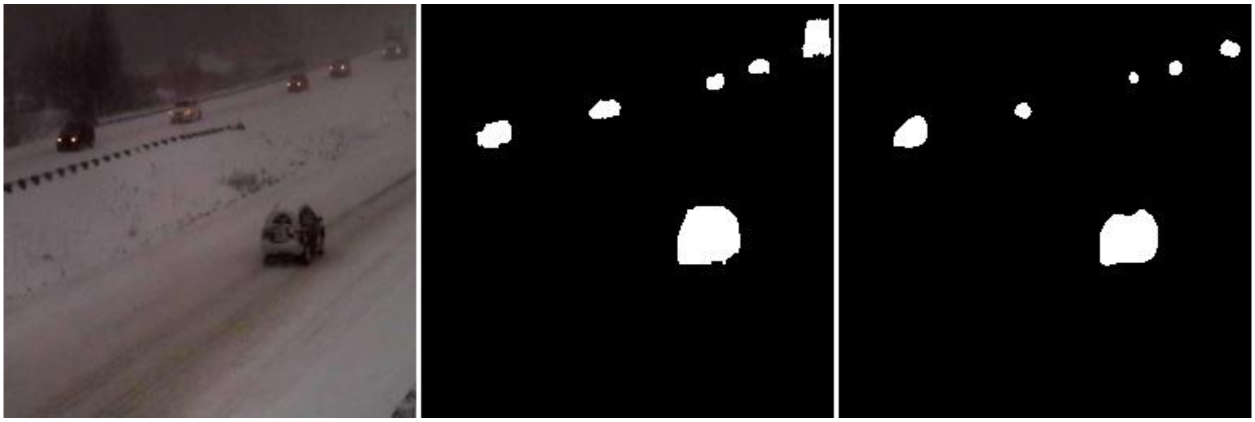


Figure 37. the segmentation of LikeUNet without background input (from left to right: input, GT, segmentation).

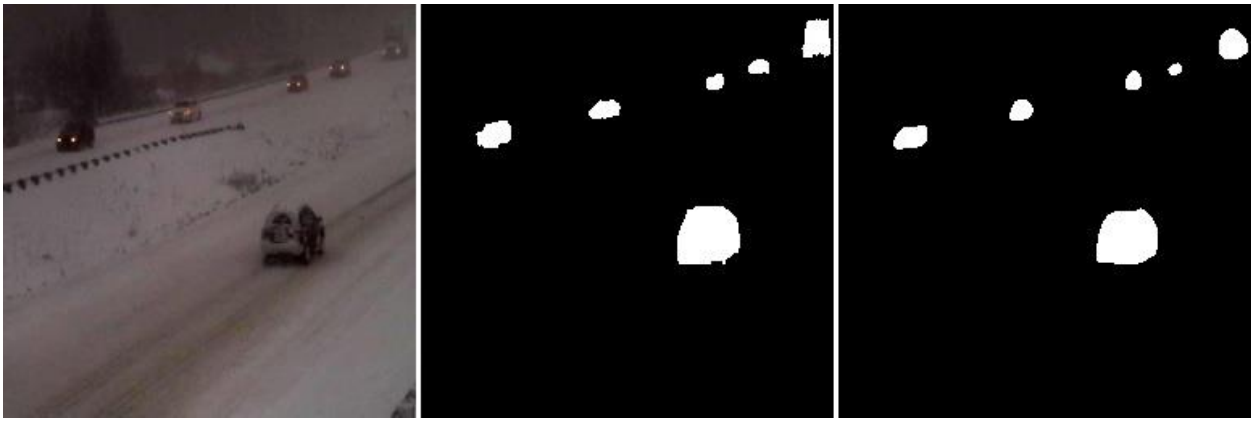


Figure 38. the segmentation of VAEUNet with background input and reconstruction of foreground objects (from left to right: input, GT, segmentation).

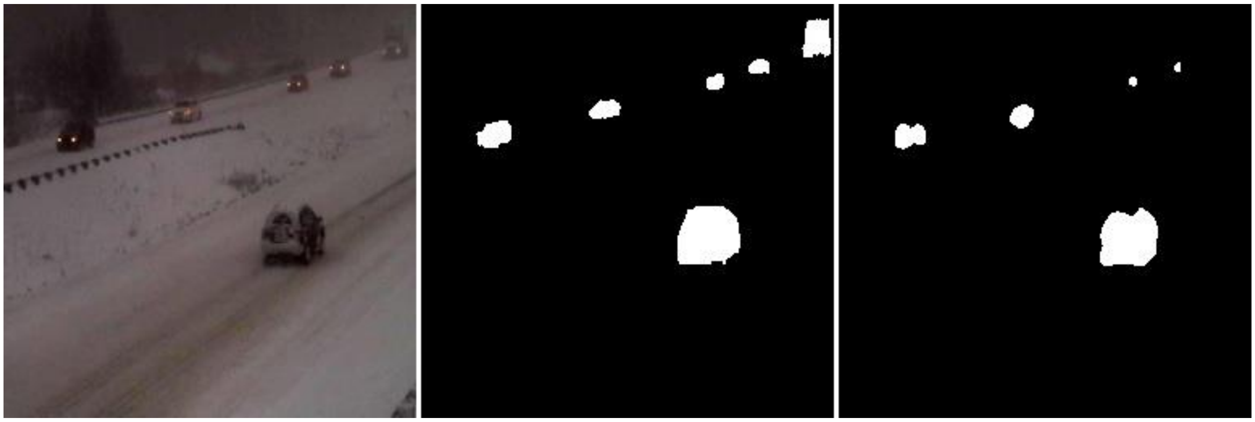


Figure 39. the segmentation of VAEUNet without background input and reconstruction of foreground objects (from left to right: input, GT, segmentation).

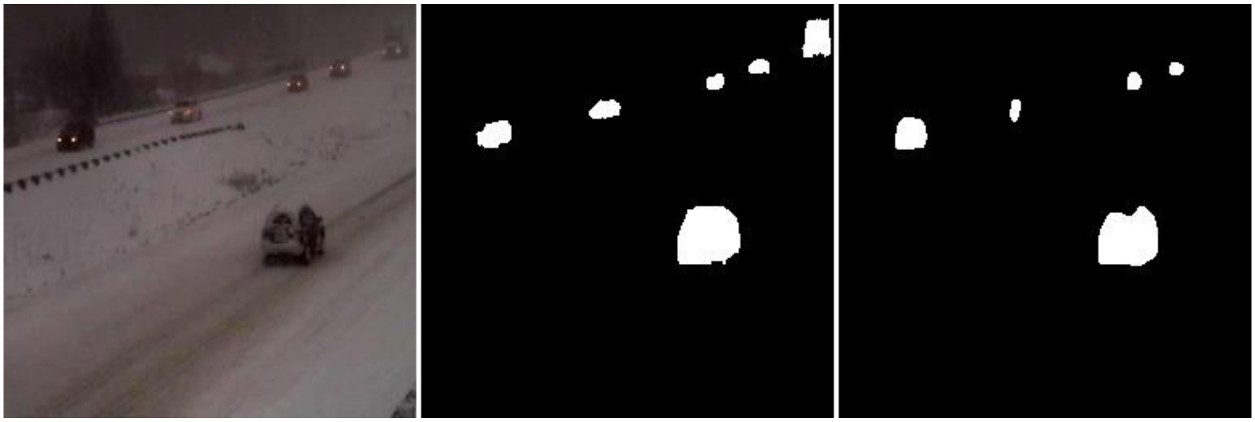


Figure 40. the segmentation of VAEUNet with background input and reconstruction of the whole image (from left to right: input, GT, segmentation).

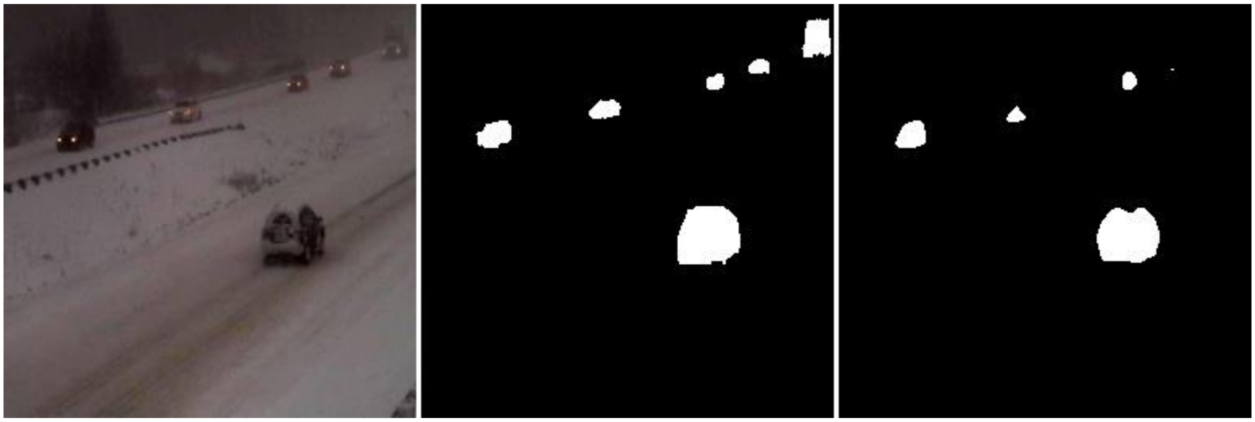


Figure 41. the segmentation of VAEUNet without background input and reconstruction of the whole image (from left to right: input, GT, segmentation).

The best metrics of the different methods discussed above on mini CDnet are shown in table 10 below.

Table 10. The performance of different methods on mini CDnet

Code	F-measure	mIoU
Q1	0.190300	0.133200
Q2	0.480300	0.858800
Q3	0.471500	0.805200
Q4	**0.482500**	**0.873300**
Q5	0.475000	0.826400
Q6	0.475200	0.827000
Q7	0.469900	0.796000

Where Q1~Q7 stand for the below in order:

bscGAN,

LikeUNet with background input,

LikeUNet without background input,

VAEUNet with background input and reconstruction of foreground objects,

VAEUNet without background input and reconstruction of foreground objects,

VAEUNet with background input and reconstruction of the whole image,

VAEUNet without background input and reconstruction of the whole image.

2.5.2 conclusion

From figure 35 to figure 41 and table 10, we can get the conclusion that the method of VAEUNet with background input and reconstruction of foreground objects is the best and outperforms the other methods. This is due to the following points:

a. The skip connection in its U-Net structure can help gain texture information in the low dimensions.
b. The background image can be a good reference to help the model understand foreground objects even though the input image is not aligned 100% correctly with the background.
c. The sharing encoder for VAE (only reconstructs foreground objects) can improve the representation learning of encoder. The reconstruction of foreground can help encoder focus more on foreground objects, while the whole image reconstruction can distract the attention of encoder and have a negative impact on the encoder.
d. The bscGAN cannot converge on the small dataset.

3 prospects

There are several tasks that need to be completed, including training, and evaluating the proposed methods (Q2~Q7) on the "mini BMC" dataset. I have successfully created the "mini BMC" dataset and finished training and evaluating the method (Q1) proposed in the paper on it, but unfortunately, I do not have enough time and resources to complete the validation of Q2~Q7 on the "mini BMC" dataset before the deadline. If this part of the work had been completed, it would have provided strong support for my ideas.

Reference

[1] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig and Y. Ruichek, "BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 4018-4022, doi: 10.1109/ICIP.2018.8451603.