EPFL

Summer Project Report

Biocycling project : Generate data with GAN

CAMILLE CHALLIER

Camille Challier

I-INTRODUCTION

A- Motivation

B- Context

C- Timeline of the Project

A- Motivation

- Lots of impurities (plastics, aluminium, ceramic, …) are present in biowastes.
- Solution : compute volume ratio of impurities for legal requirements.

Camille Challier

B-Context

- Acquiring RGB and/or hyperspectral pictures of compost
- Segmenting impurities
- Regressing ratio of impurities volume
- Generative data augmentation of plastic anomalies in biodegradable waste
- Generate RGB + mask pairs with a GAN

Fachhochschule Nordwestschweiz

Camille Challier

C- Timeline of the Project

Tsinghua-Daimler Cyclist Dataset

Make the code work for the cyclist dataset.

Add the segmentation mask to the images

Adapt the code to a 4-channel images.

1 2 3 4 5

Pedestrian-Synthesis GAN

Learn about GANs, the pix2pix and PS-GAN project.

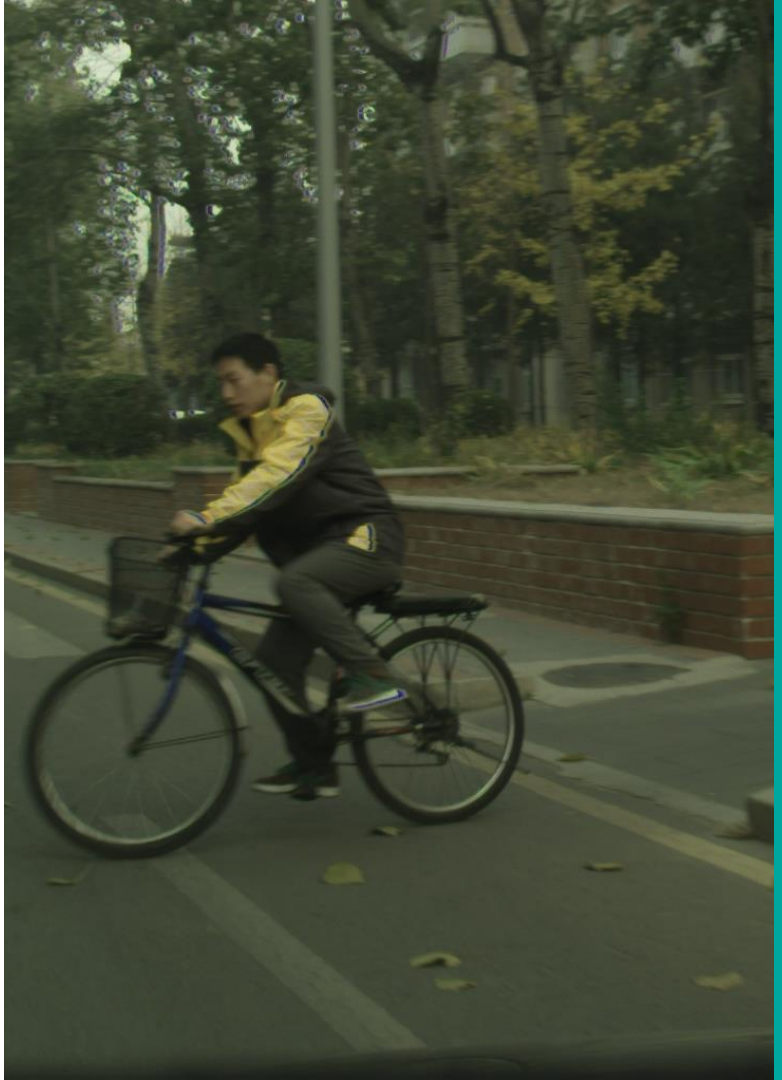
UAVVaste Dataset

Make the code work for the biowaste dataset.

Analysis

Analyse the generated images and add the code to the pipeline.

Camille Challier



II- Tsinghua-Daimler Cyclist Dataset

A- Steps

B- Results

Camille Challier

A- Steps

1 image training

Select only one image and try to make the algorithm converging to this image

Test different types of noise

- Random noise
- Gaussian noise
- Color/ Black-and-white

Increase the number of images

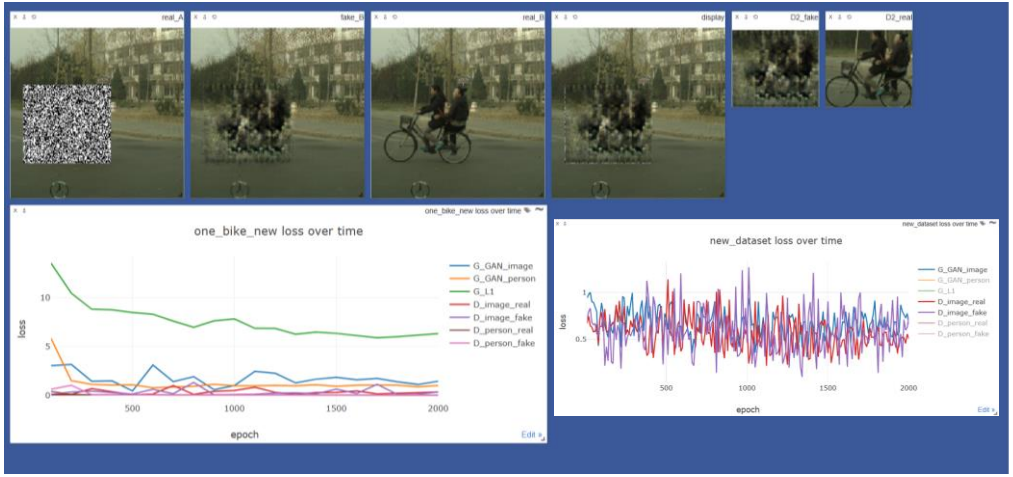
Look at the trend of the loss curves when doubling the number of images in the training set.

All dataset

Create new cyclist by training the entire dataset.

Image nb = 1, random noise, epochs = 2000

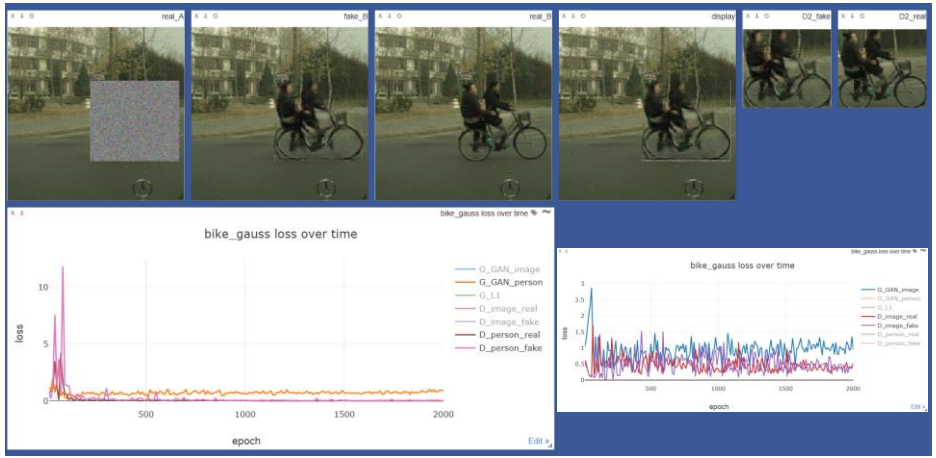
- The training step optimizes the parameters of the decoder and the generator. When training the model with one image, we observe that the generator needs to be trained more than the decoder. Here are the results for training the generator 1 time per 1 decoder training and 3 times per 1 decoder training.



Camille Challier

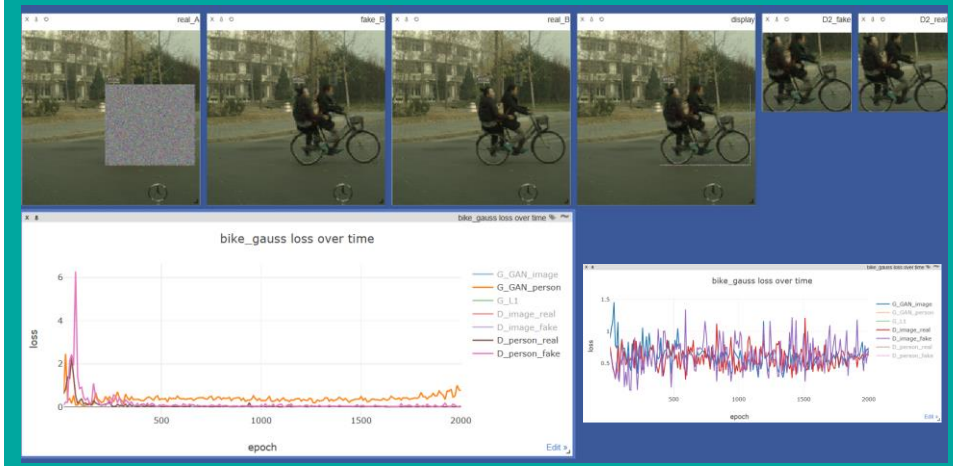
Image nb = 1, gaussian noise, epochs = 2000

Generator training : 3 times



- Higher Image-Generator loss compared to random noise.
- Same quality of the generated image.

Generator training : 5 times

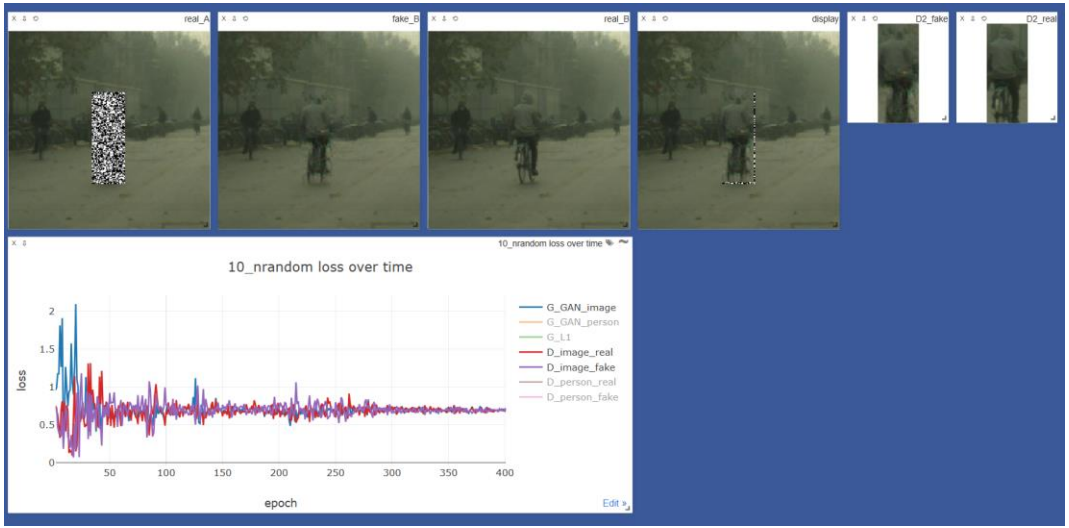


- Improved image quality by increasing generator training.

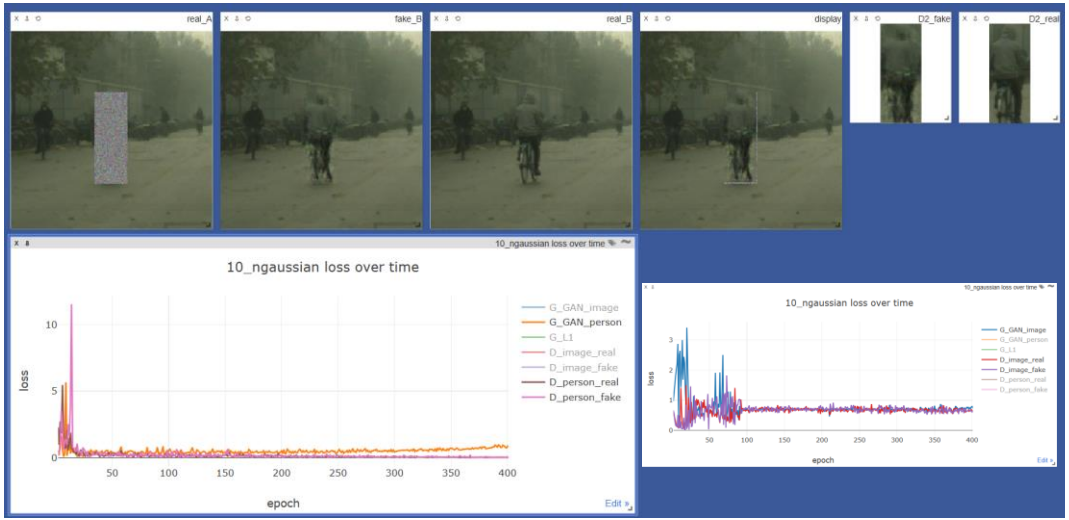
Camille Challier

Image nb = 10, epochs = 400, G training = 3

- Random noise :



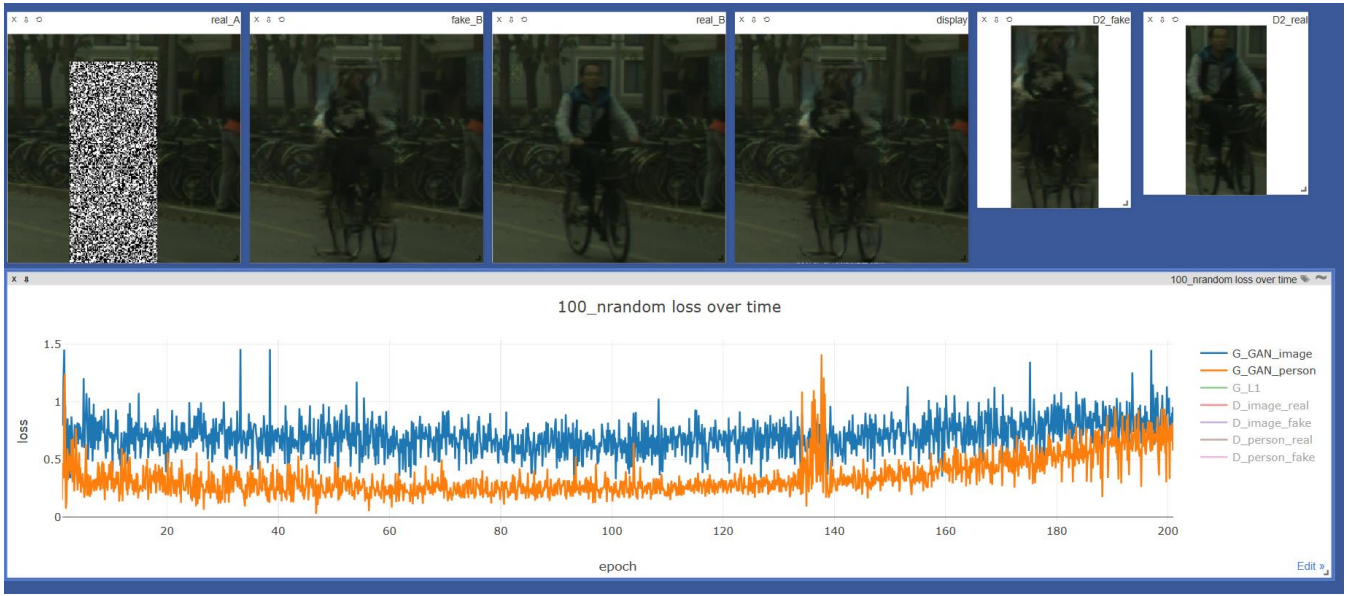
- Gaussian noise :



=> Very similar results with gaussian and random noise.

Camille Challier

Image nb = 100, epochs = 200, G training = 3



- The increase in generator's loss decreases as the training of G is increased.

III- UAVWaste Dataset

A- Steps

B- Results

Camille Challier

A- Steps

Adaptation of the code

- Select specific size of bounding boxes
- Create dataset

1 image training

Select only one image and try to make the algorithm converging to this image

Increase the number of images

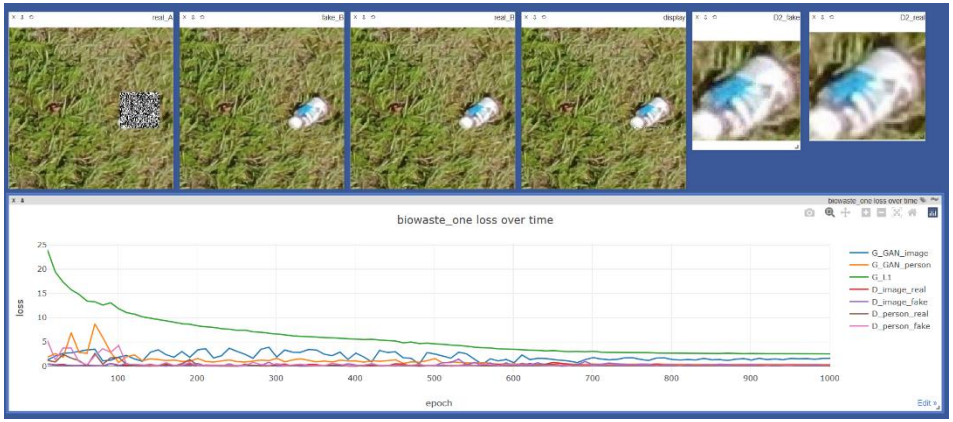
Look at the tendance of the losses curves when increasing by 2* the number of images in the training set.

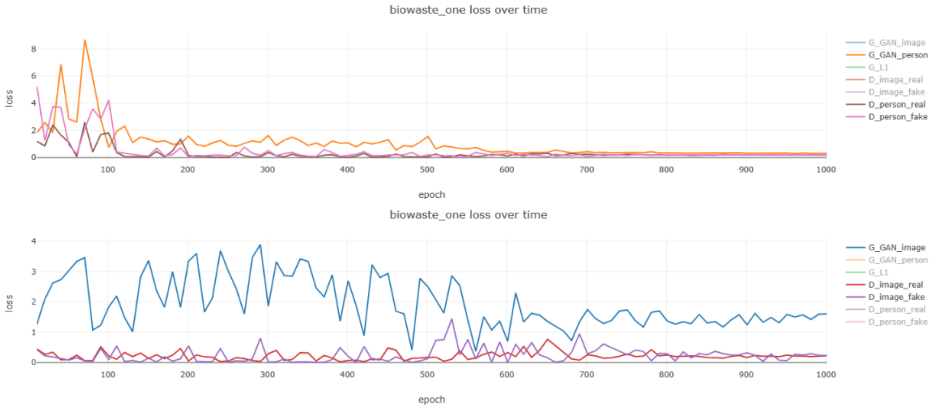
All dataset

Create new plastics by training the entire dataset.

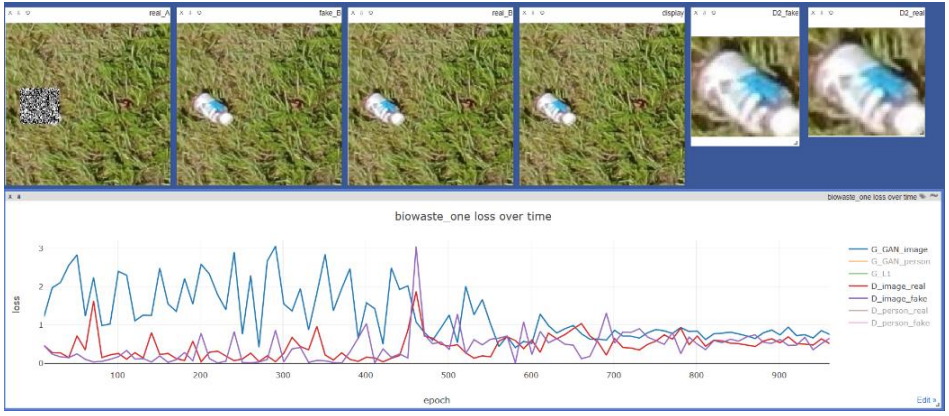
Image nb = 1, epoch =1000, random noise

Generator training : 1 time





Generator training : 3 times



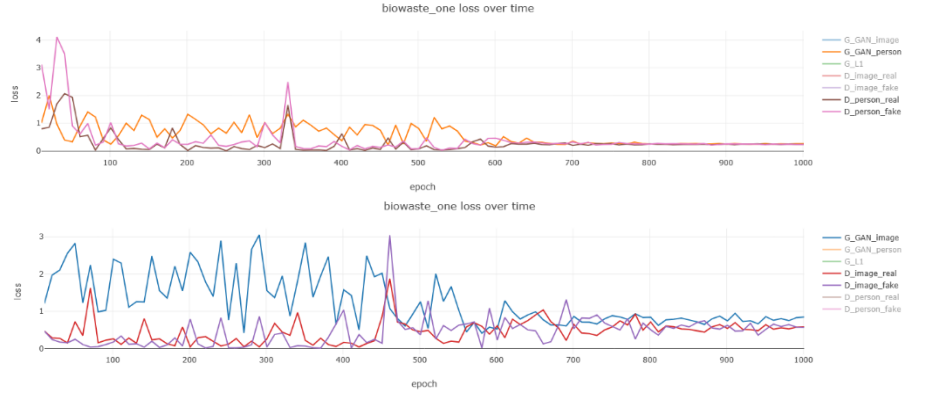
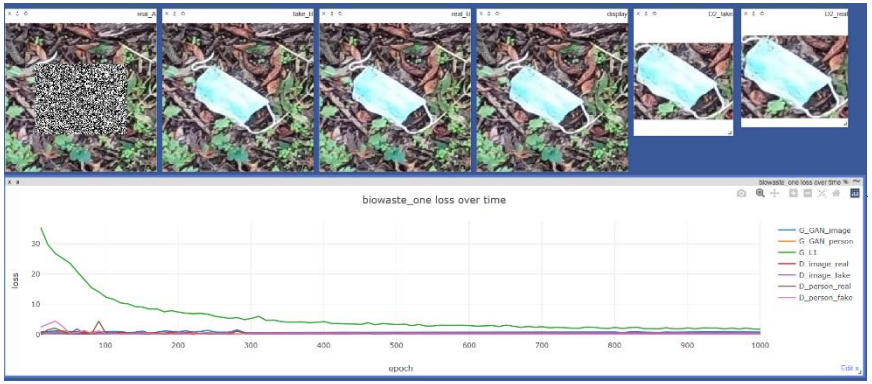
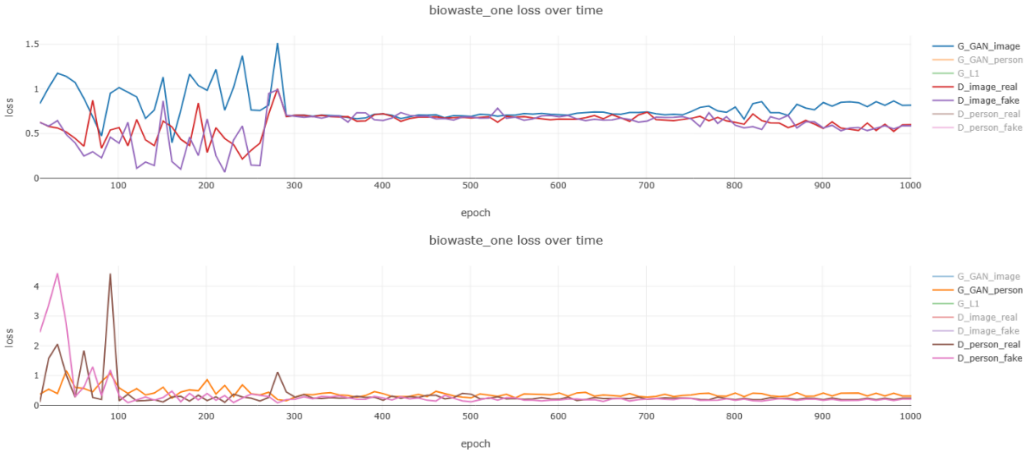


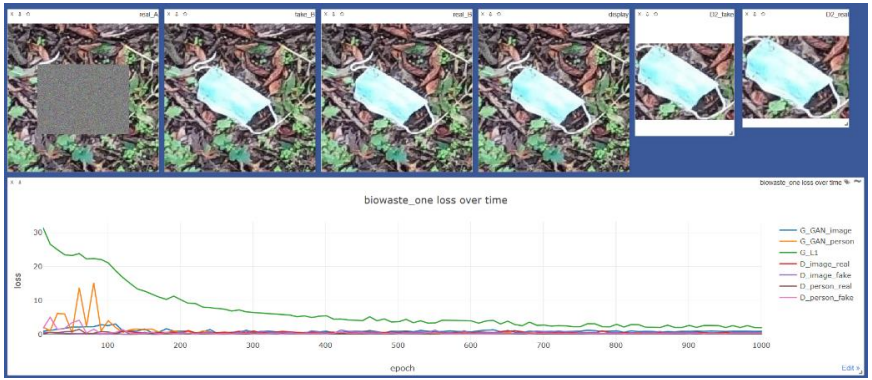
Image nb = 1, epoch =1000, Generator training = 3

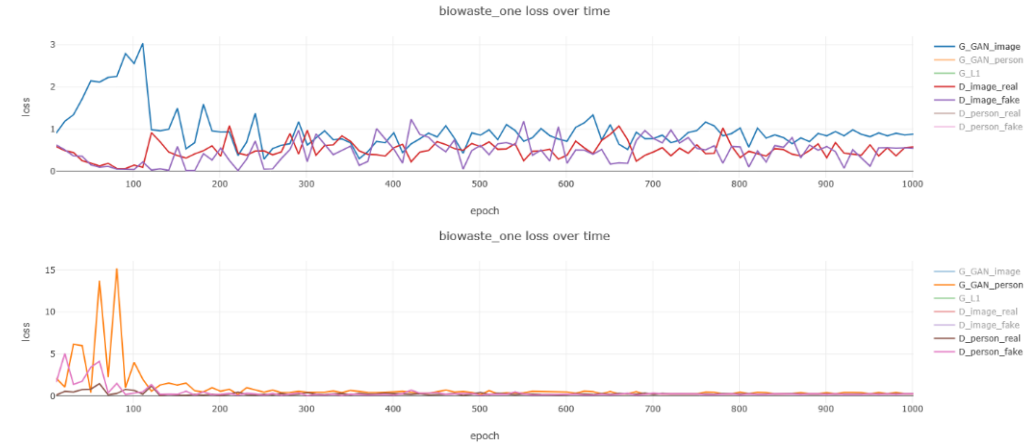
Random noise





Gaussian noise

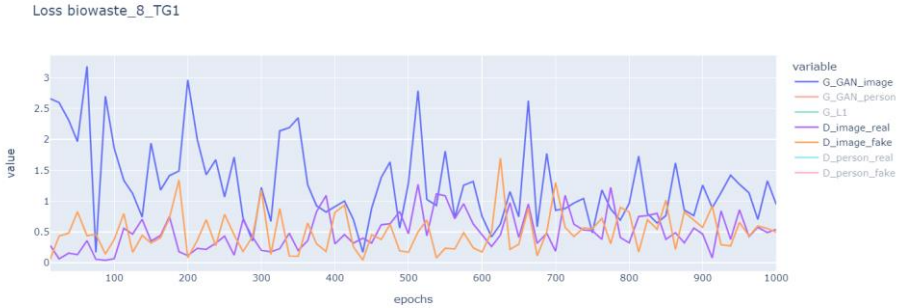


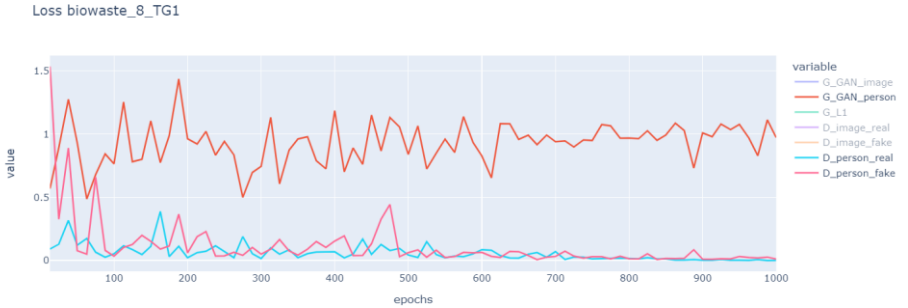


=> Very similar results with gaussian and random noise. Choose the use random noise

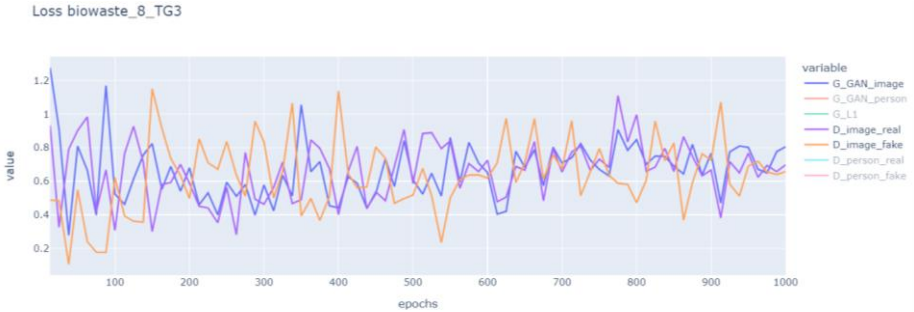
Image nb = 8, epoch =1000, random noise, G training = 1,3

Generator training : 1 times





Generator training : 3 times



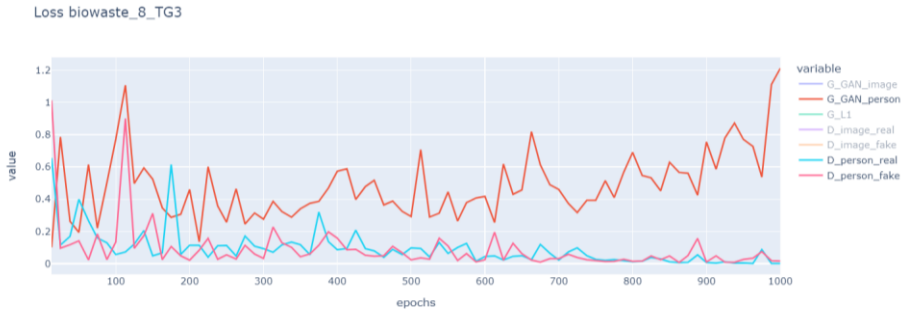
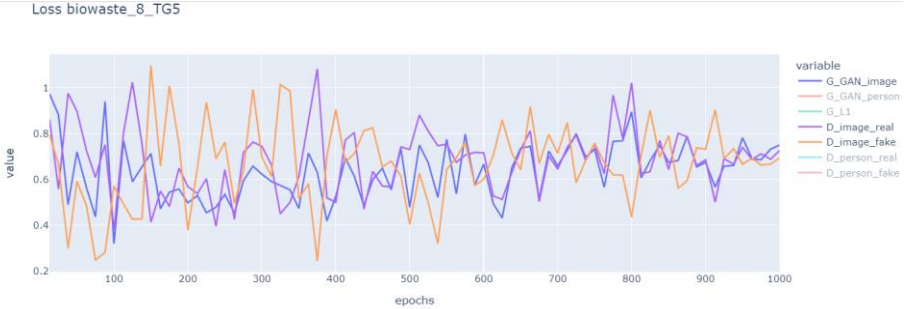
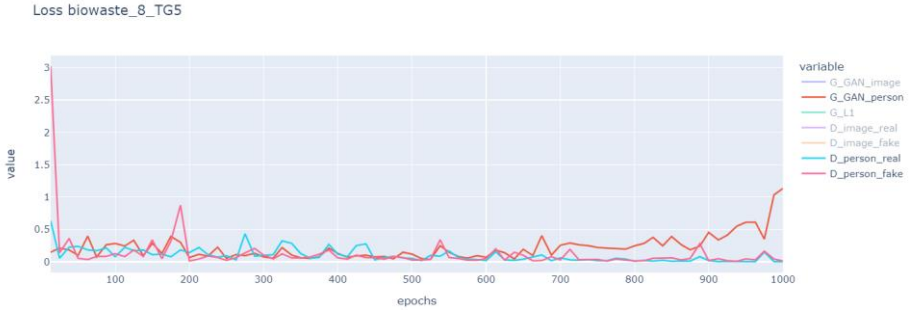


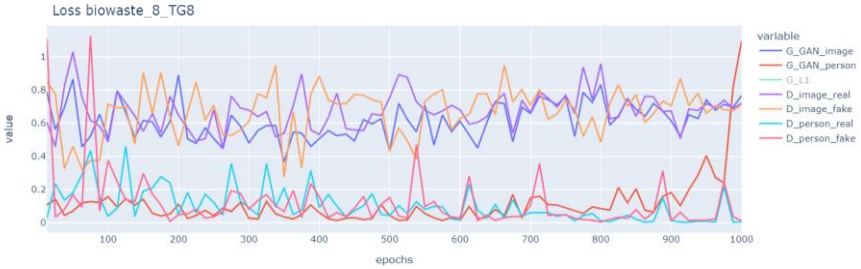
Image nb = 8, epoch =1000, random noise, G training = 5,8

Generator training : 5 times





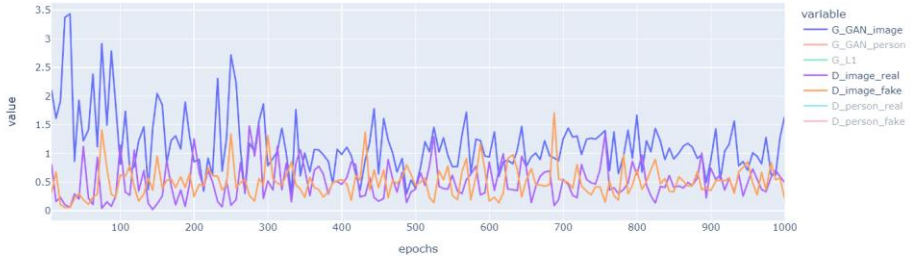
Generator training : 8 times



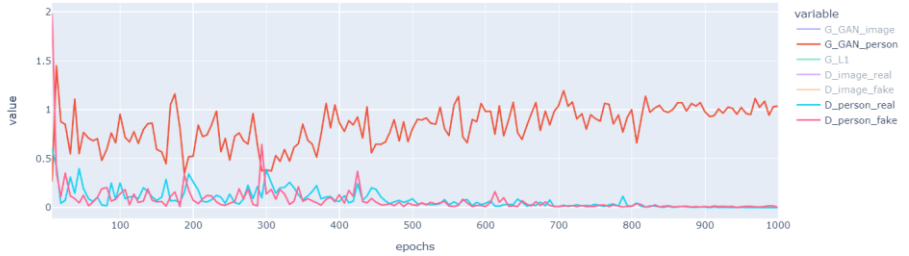
- The Generator's loss decreases as the training of G is increased.
- However, we still observe an increase in the generator's person loss at the end of the epochs for high generator training.

Image nb : 16, epoch =1000, random noise, G training = 1,3,5

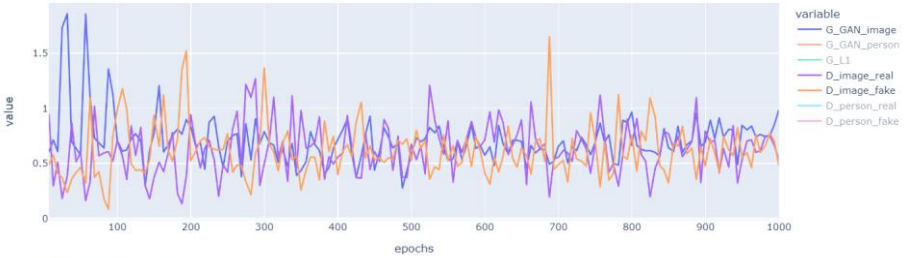
Loss biowaste_16_TG1



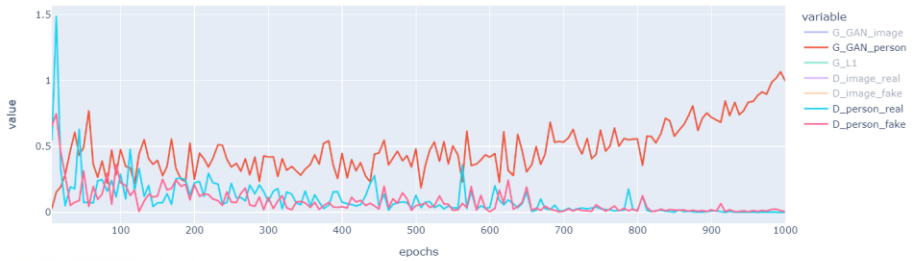
Loss biowaste_16_TG1



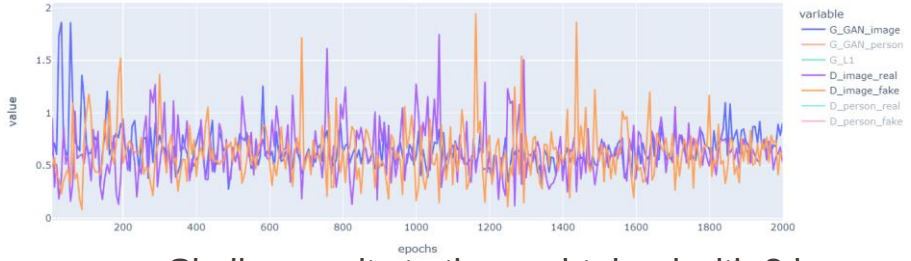
Loss biowaste_16_TG3



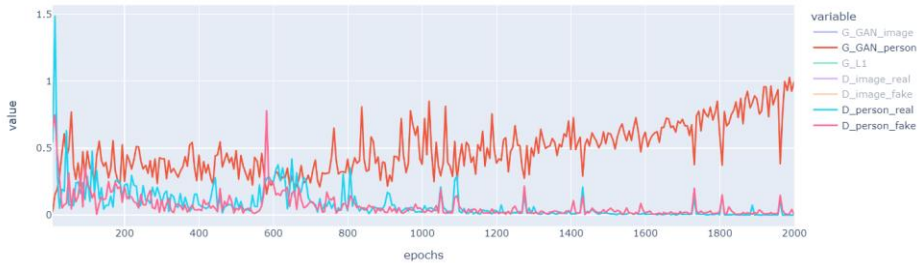
Loss biowaste_16_TG3



Loss biowaste_16_TG5



Loss biowaste_16_TG5



=> Similar results to those obtained with 8 images.

Image nb = all, epoch = 200, random noise, G training = 3, 5

=> More clearly defined generated cyclist with three times the generator training

Camille Challier

III- UAVWaste and mask

A- Steps

B- Results

Camille Challier

A- Steps

Generate mask

Adapt the algorithm to work on 4 channels images.

Training

Increase the number of images in the training set and observe the tendances.

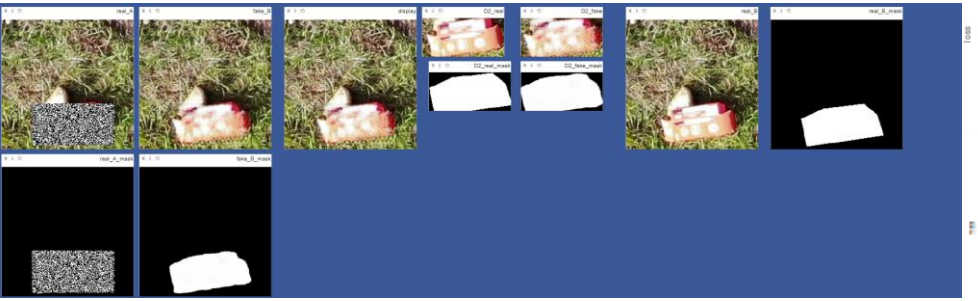
Testing

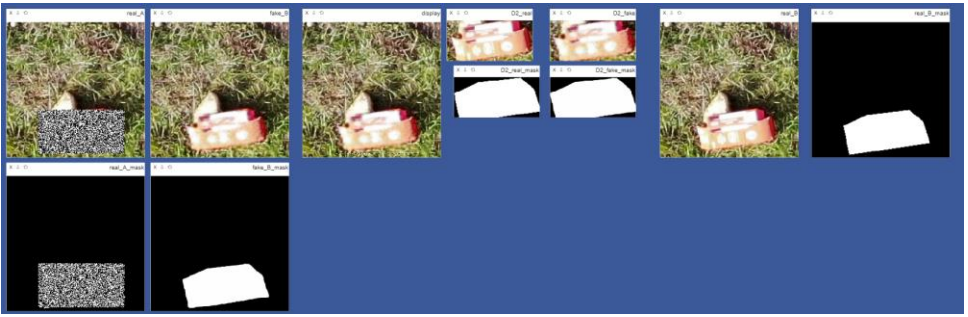
- Generate new plastic and their segmentation at new places or replace existing plastics.
- Paste the generated plastic back to the entire image.

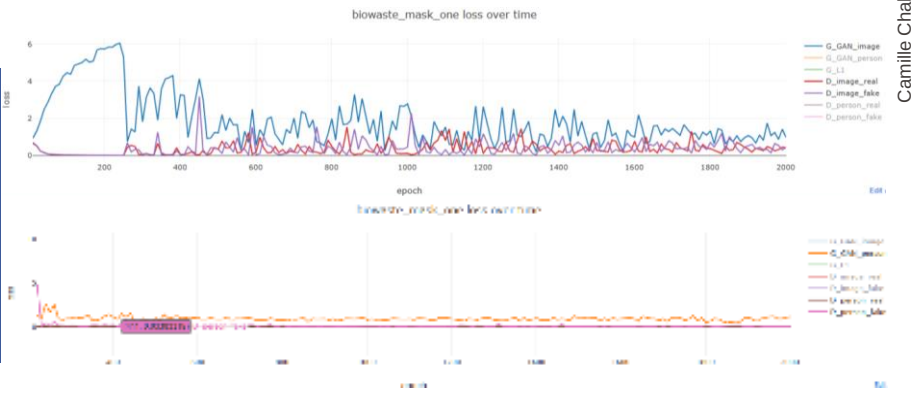
Improvements

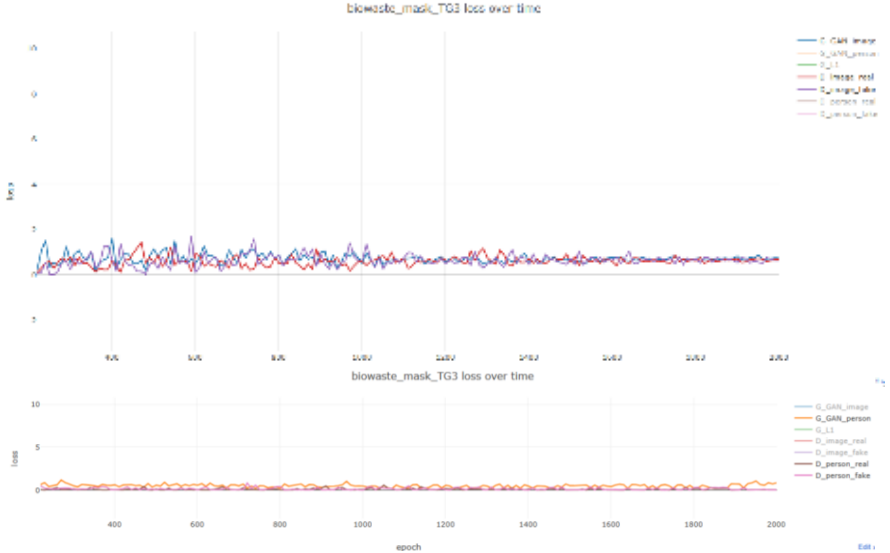
- Change the addition of noise.
- Change the implementation of the testing.

Image nb = 1, epoch = 2000, random noise, G training = 1, 3









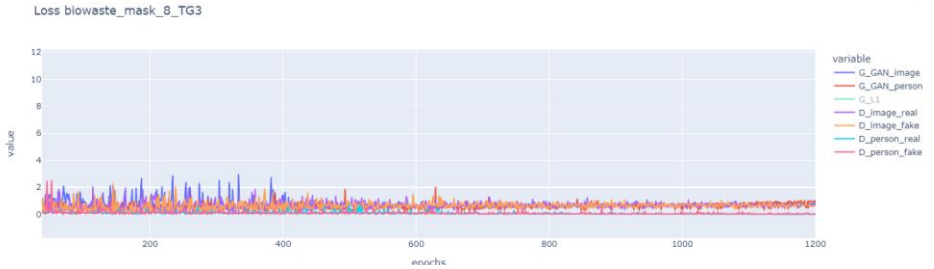
Camille Challier

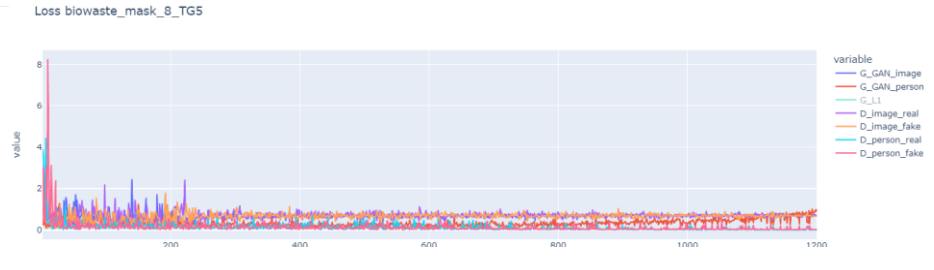
Change the addition of noise

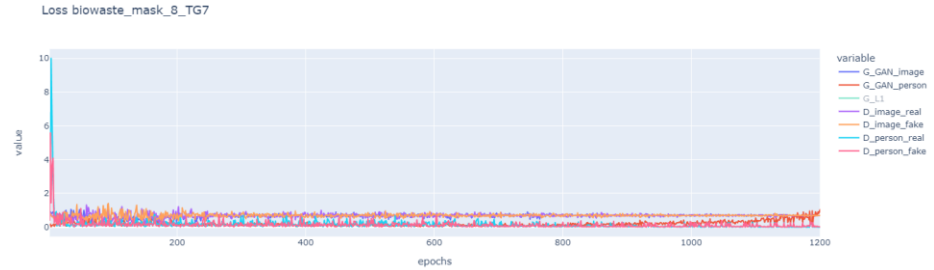
- The noise is added before the training for each images.
- **Problem :** In each epoch, the image is resized randomly, leading to occasional inaccuracies in the bounding box placement. This error becomes evident as a line of pixels around the overlaid generated object on the original image.
- **Solution :** Add noise after resizing in each epoch. Set the seed of the generator with the index of the object to ensure consistent noise for the same images.

Camille Challier

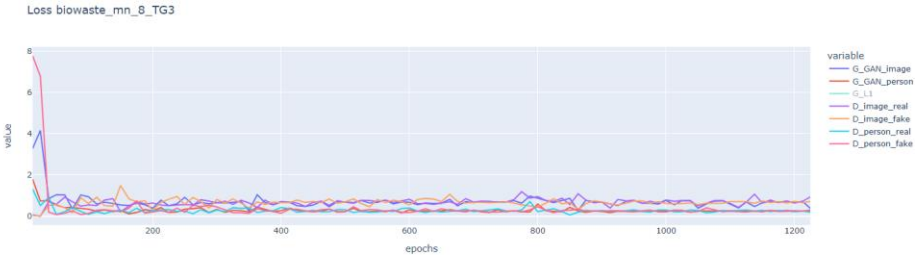
Image nb = 8, epoch =1200, random noise, G training = 3,5,7







Noise added at each epoch :

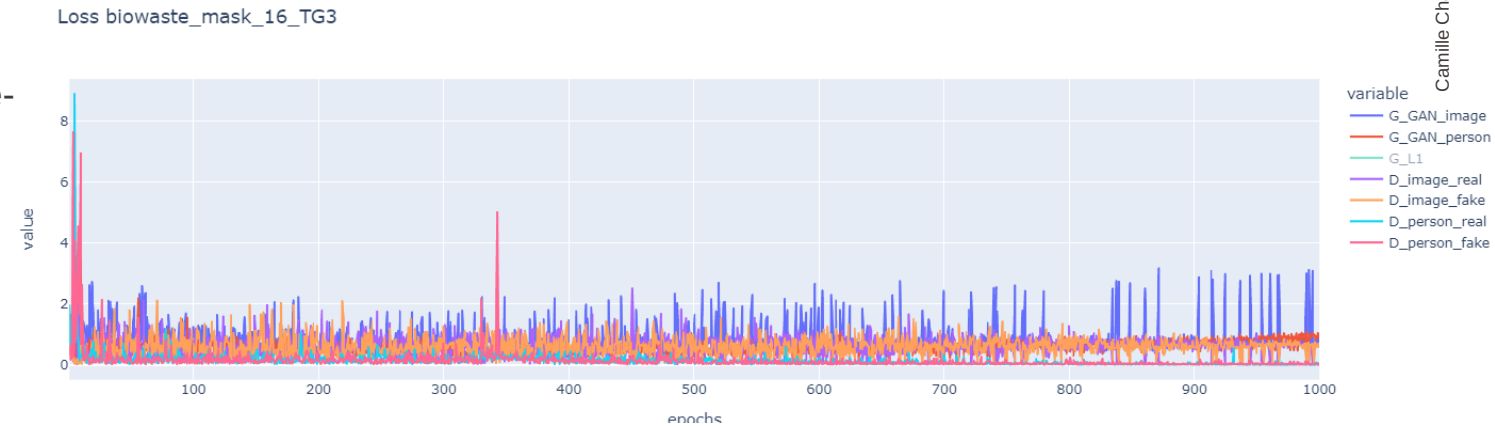


- The Generator's loss decreases as the training of G is increased (especially for the red curve).
- However, we still observe an increase in the generator's person loss at the end of the epochs for high generator training.
- This increase does not appear when adding noise at each epoch.

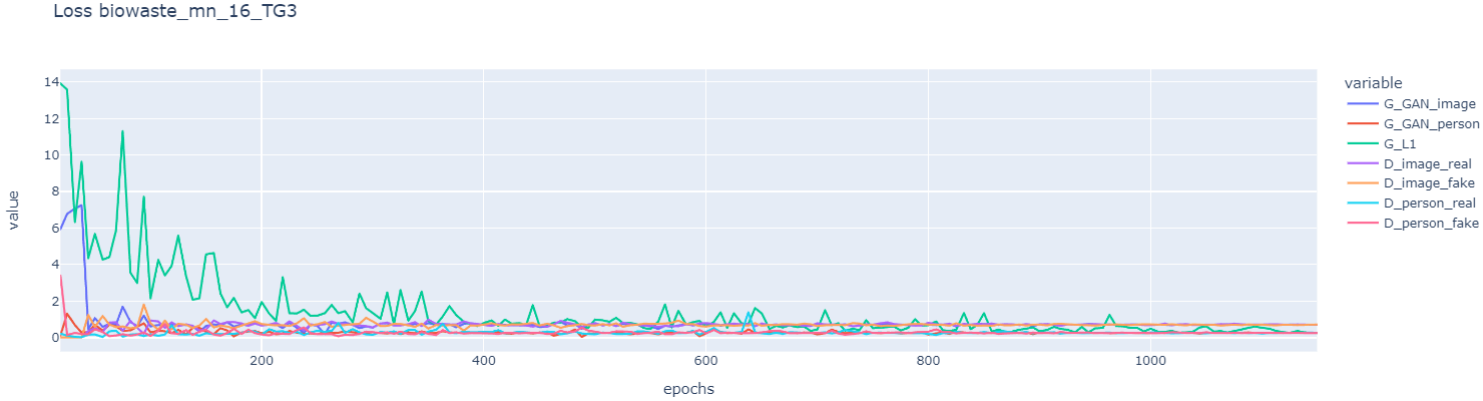
Camille Challier

Image nb = 16, epoch = 1000, random noise, G training = 3,5

- Add noise during pre-processing step:



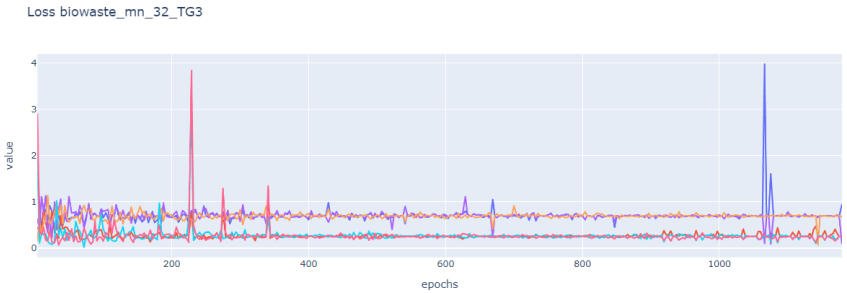
- Add noise at each epoch :

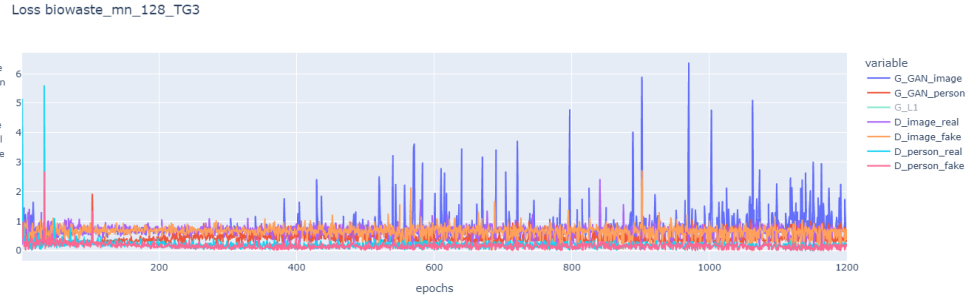


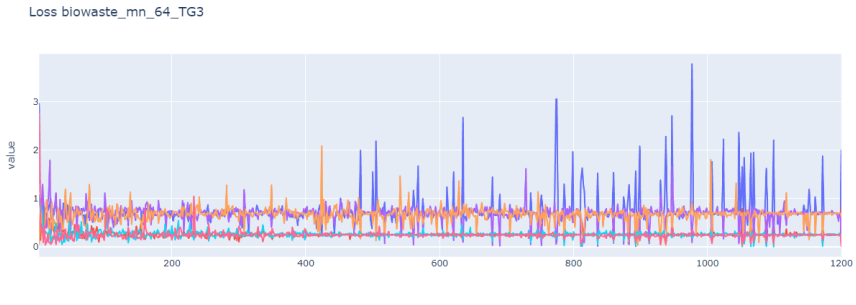
- => The increase in the generator's person loss does not appear when adding noise at each epoch.

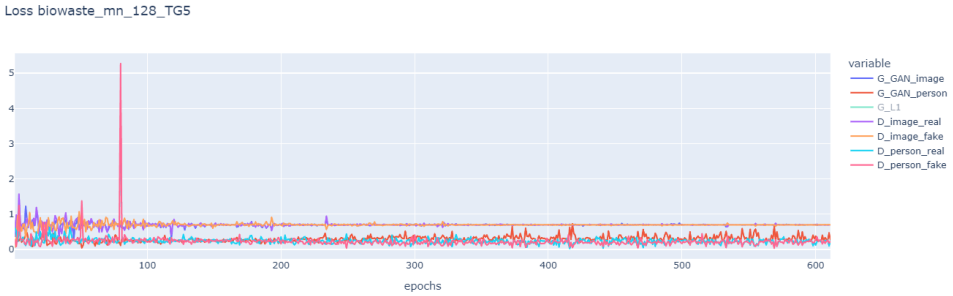
Camille Challier

Image nb = 32/64/128, epochs =1200, random noise, G training = 3 and 5







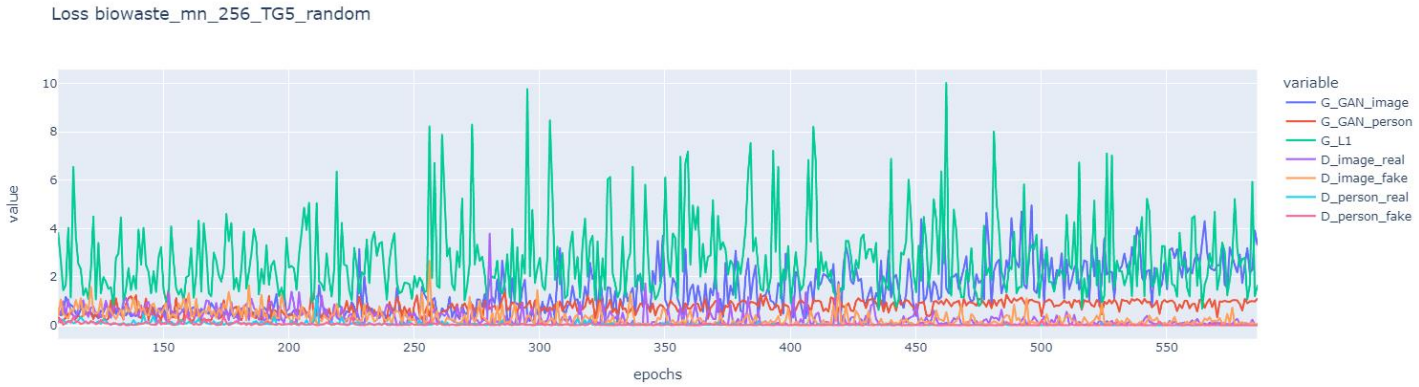


=> Training the generator three times is sufficient for 32 and 64 images to achieve a low generator loss. However, with 128 images, we need to increase it to five times to achieve a smaller loss.

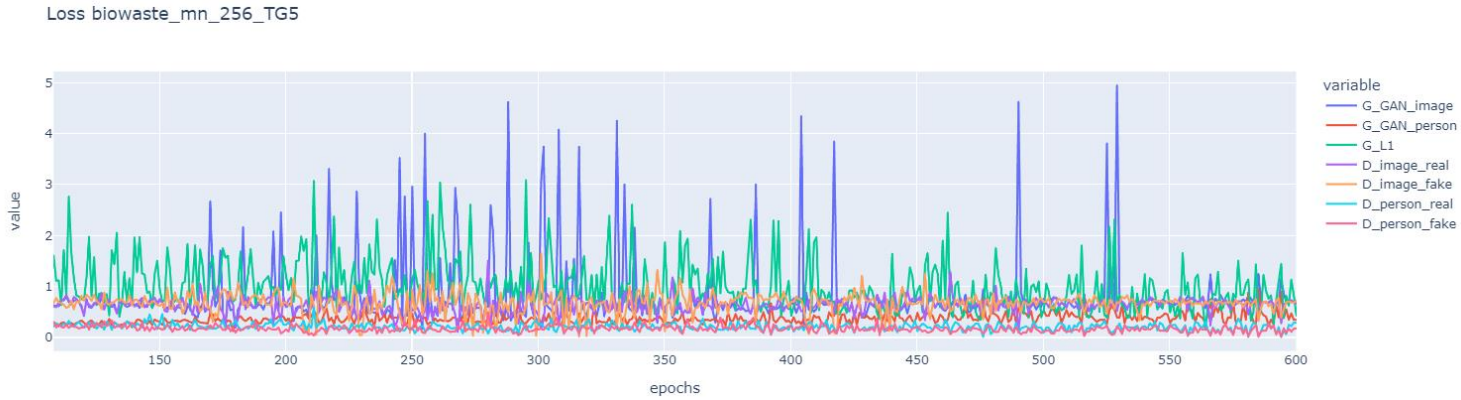
Camille Challier

Image nb = 256, epoch = 600, random noise, G training = 5

Comparison between random noise (seed not set) and same noise at each epoch for the same image



- Image-Generator loss curve do not converge
- Higher Person-Generator loss
- Higher L1 loss

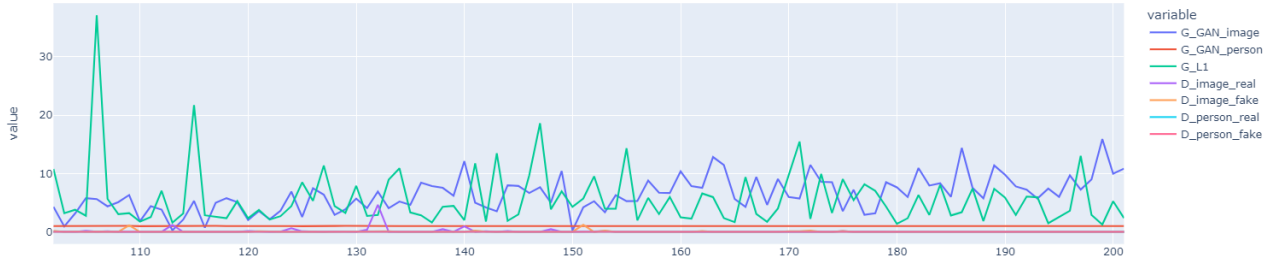


=> Better to use the same noise box for an image at each epoch.

Image nb : all 2,5K, epoch =200, random noise, G training = 1, 3

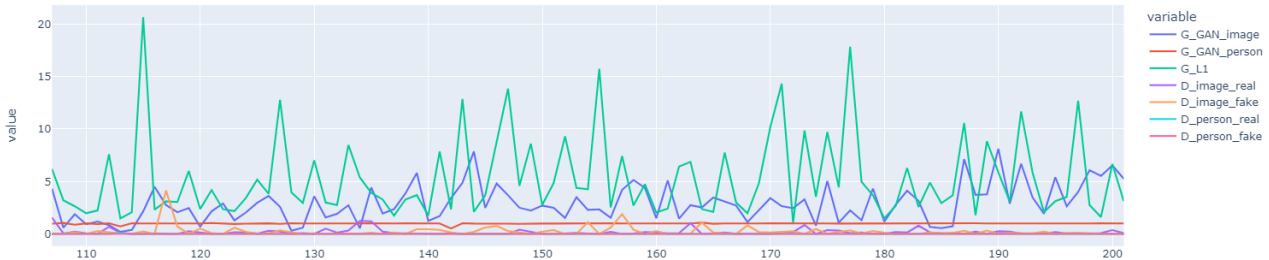
Camille Challier

Loss biowaste_mn_all_TG1



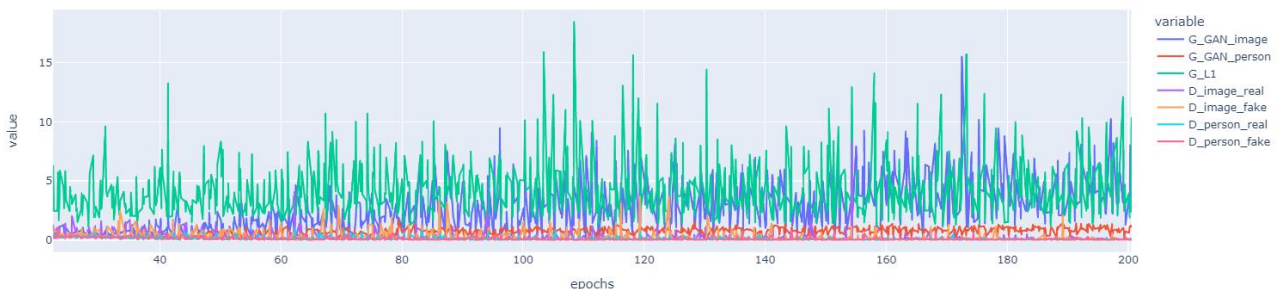
- Image-Generator loss curve do not converge very well to 10.
- Person-Generator loss converge to 1.

Loss biowaste_mn_all_TG3



- Image-Generator loss curve is smaller, converge to 5.
- Person-Generator loss converge to 1.

Loss biowaste_mn_all_TG3_same_background

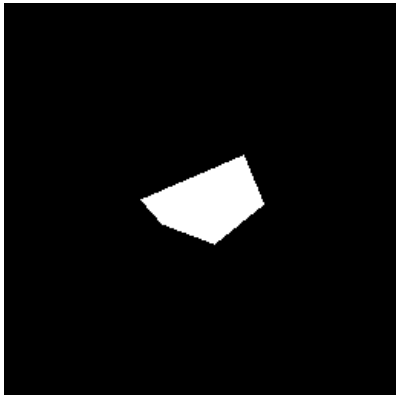


- Selecting cameras with the same green background due to poor results of generated images on a gray background.
- Same results.

Camille Challier

Exemple of generated waste

- UAVVaste all images, epochs = 200, random noise, G training = 1
  - Initial image :



  - Output:

Generated wastes

- All images, epochs = 200, random noise, G training = 1
- All images, epochs = 200, random noise, G training = 3
- Grass background images (1,2k), epochs = 200, random noise, G training = 3

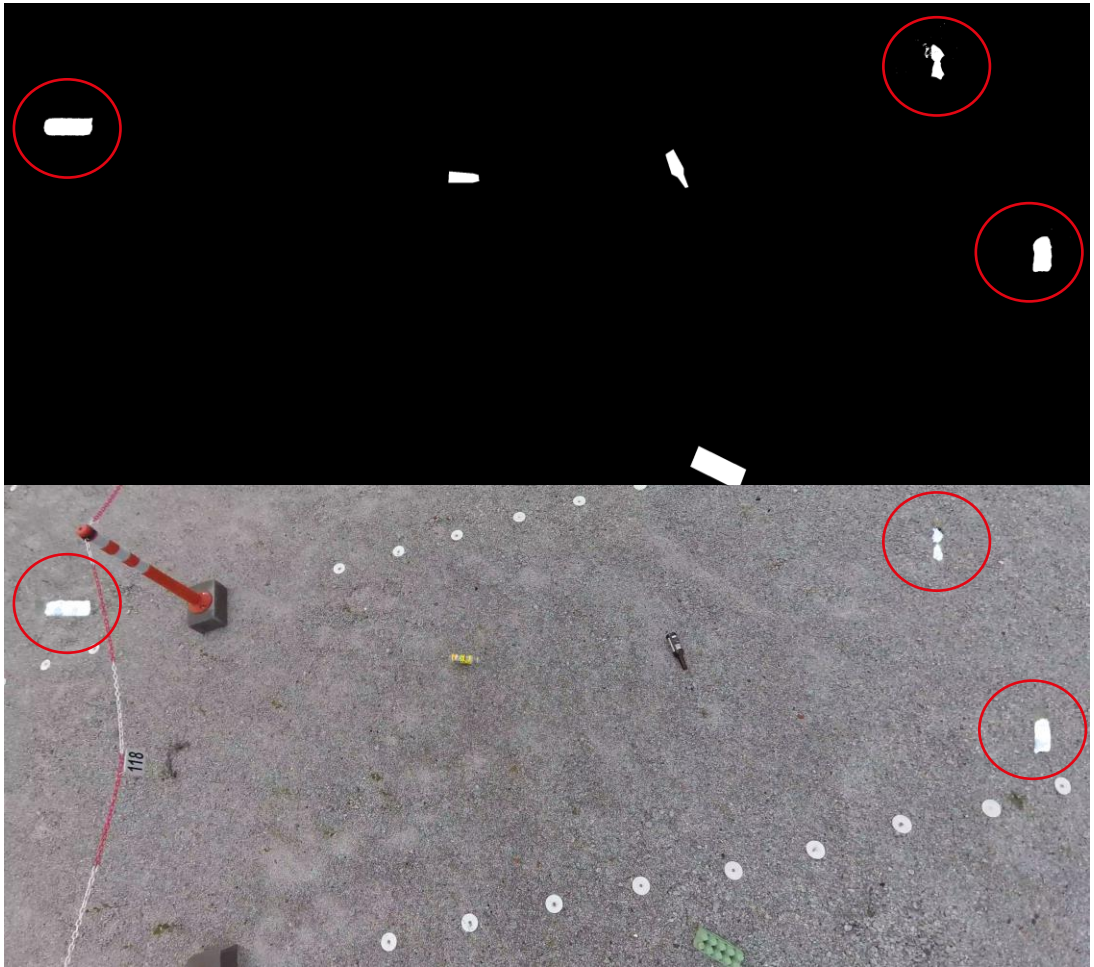
Test results Analysis

Camille Challier

- No significant differences in the generated plastic when training the generator 1 time or 3 times.
- Improved results with a grass background compared to a grey background, mainly due to a larger number of training images with green/brown backgrounds.
- An attempt was made by selecting only videos with green backgrounds. However, the results were not as good, possibly due to a smaller dataset and the presence of non-green photos.

Final result

Newly generated plastics



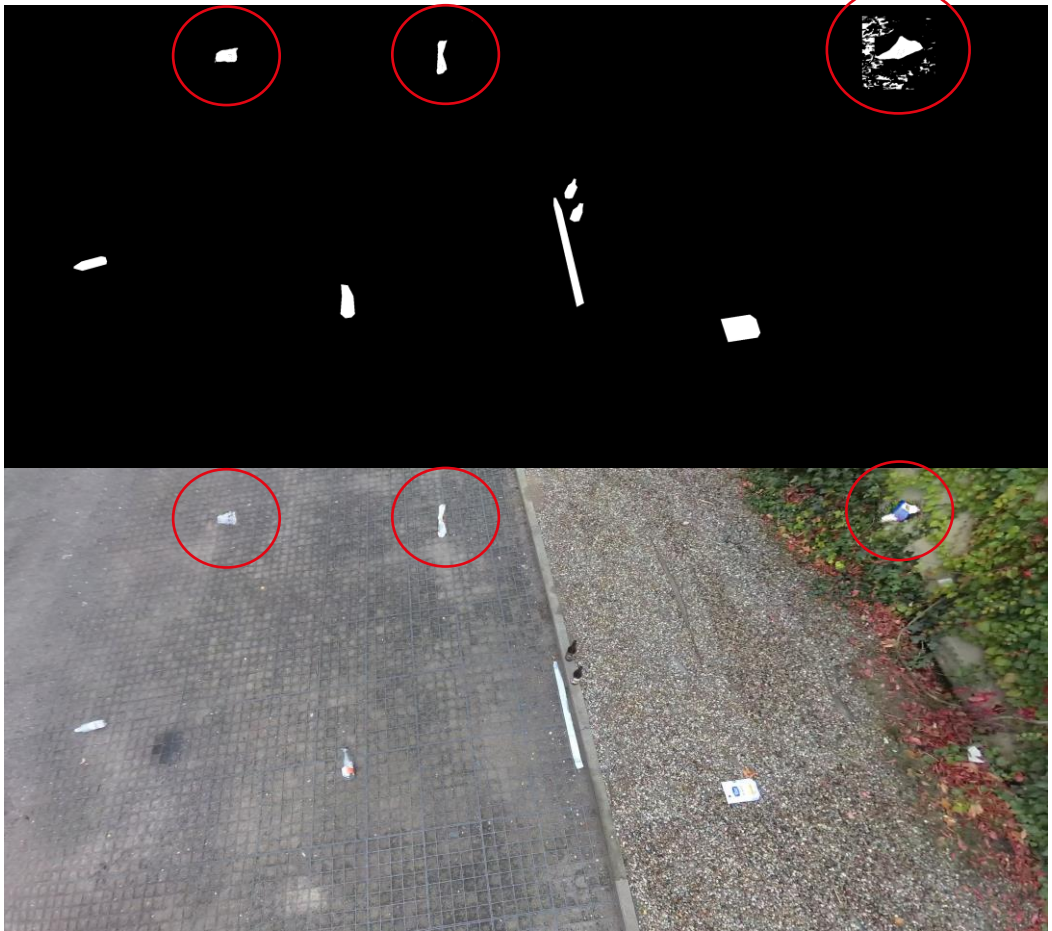
Camille Challier

Camille Challier

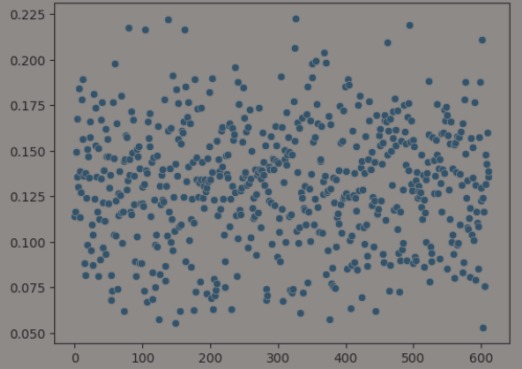
Final result

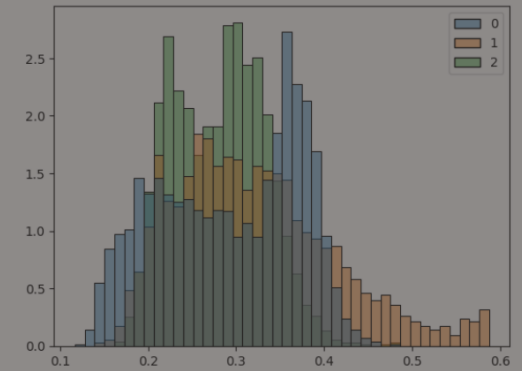
Have not fully converged

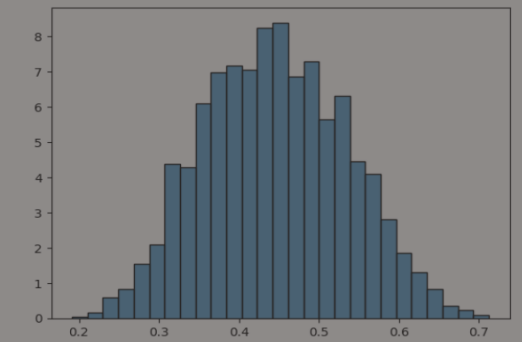
Newly generated plastics



Camille Challier







IV- Analysis of the generated images

Goal : observe similarities between generated images and the training set by using :

1) Pixel-wise comparison with l1 loss

2) Structural Similarity Index

3) VGG16

Camille Challier

Process :

Pre-processing for each pair of generated and original image :

1. Determine the larger length of the 2 images to set the *new size* for both images.

Process 1 (smaller_crop) :

1. Resize the images, ensuring that the smaller side of both images matches the *new size* while maintaining the original aspect ratio.
2. Crop the images to transform each into a square with a side length equal to the *new size*.

Process 2 (larger_resize) :

1. Alternatively, resize the images so that the larger side of both images matches the *new size* while preserving the aspect ratio.
2. Apply the addition of black pixels to expand the image to the *new size*.

Camille Challier

Comparison functions :

Computation of the images differences by using :

1. **Pixel-wise comparison with l1 loss :**

Quantify the difference in the values of each of the corresponding pixels between the original and the generated image.

```
transform_list = transforms.ToTensor()
loss = torch.nn.L1Loss()
output = loss(transform_list(image1), transform_list(image2))
```

2. **The Structural Similarity Index from image_similarity_measures** :

Quantify the difference between 2 images by taking into account the interdependencies between spatially close pixels.

```
from image_similarity_measures.quality_metrics import ssim
output = ssim(org_img=np.array(image1), pred_img=np.array(image2))
```

Comparison functions :

3. VGG16 and cosine similarity

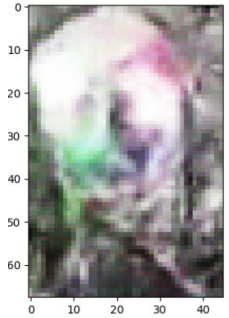
Use VGG16 pretrained model to identify similarities between images.

By using this model, we can extract high-level features from different images and compare them to identify similarities.

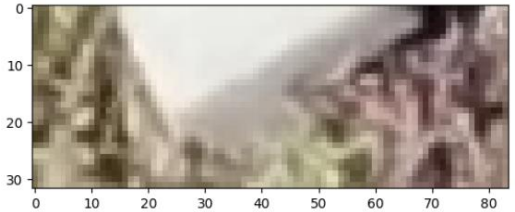
```python
def get_image_embeddings(object_image, shape):

    """
      ----------------------------------------------------
      convert image into 3d array and add additional dimension for model input
      ----------------------------------------------------
      return embeddings of the given image
    """

    image_array = np.expand_dims((object_image), axis = 0)
    vgg16 = VGG16(weights='imagenet', include_top=False,
                  pooling='max', input_shape=(shape, shape, 3))
    image_embedding = vgg16.predict(image_array)

    return image_embedding

def get_similarity_score(first_image, second_image,shape):

    first_image_vector = get_image_embeddings(first_image, shape)
    second_image_vector = get_image_embeddings(second_image, shape)

    similarity_score = cosine_similarity(first_image_vector, second_image_vector).reshape(1,)

    return similarity_score


output = get_similarity_score(image1,image2, new_size)
```

Most similar images found :

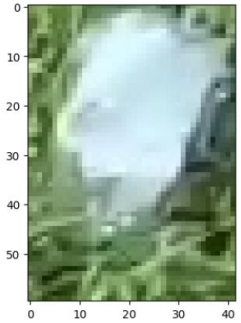
- Generated image :



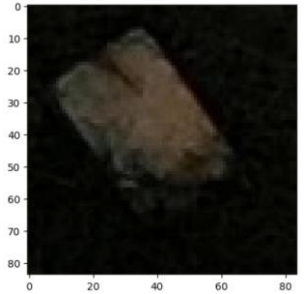
- L1 + smaller_crop



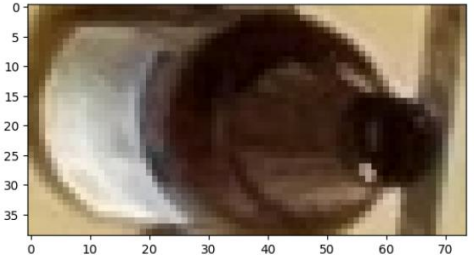
- L1 + larger_resize :



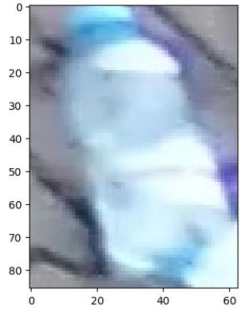
- SSIM + smaller_crop :



- SSIM + larger_resize :

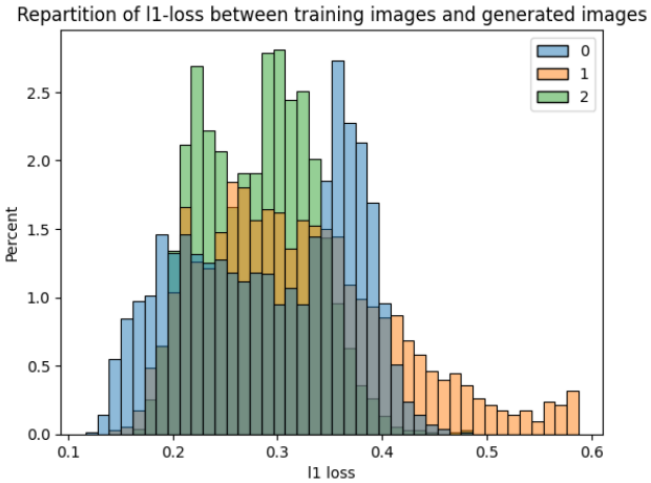


- VGG16 + smaller_crop :



Camille Challier

L1 pixel wise comparison



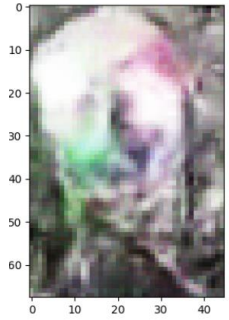
- Calculation of L1 Pixel-wise comparison between 3 generated image and the entire training set.

- For each generated image, plot the smaller loss found by pixel wise comparison over the whole training set.

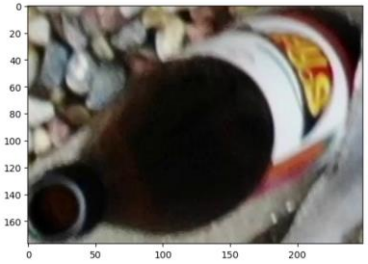
Camille Challier

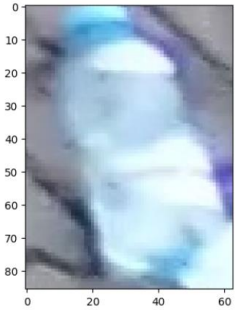
VGG16 comparison

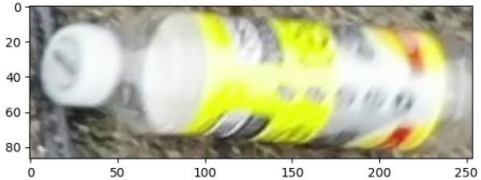
- Generated image :

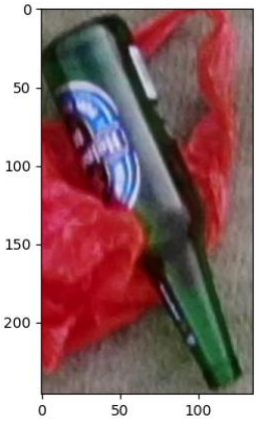


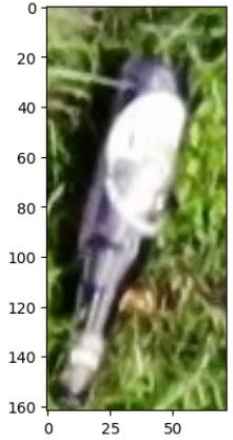
- Fifth more similar images :





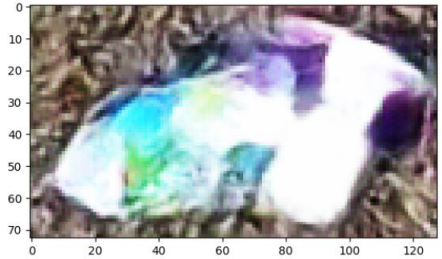




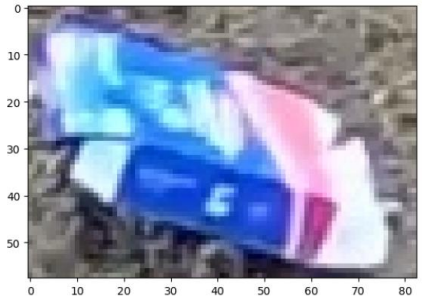


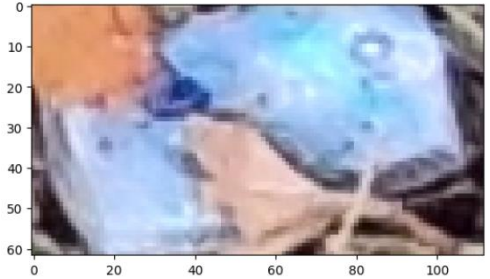
VGG16 comparison

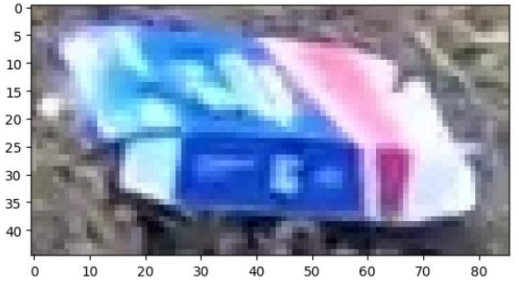
- Generated image :

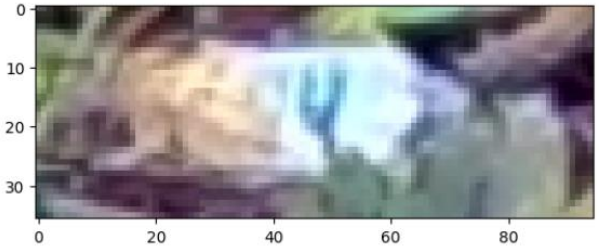


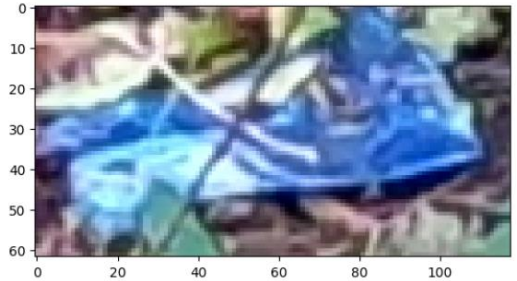
- Fifth more similar images :







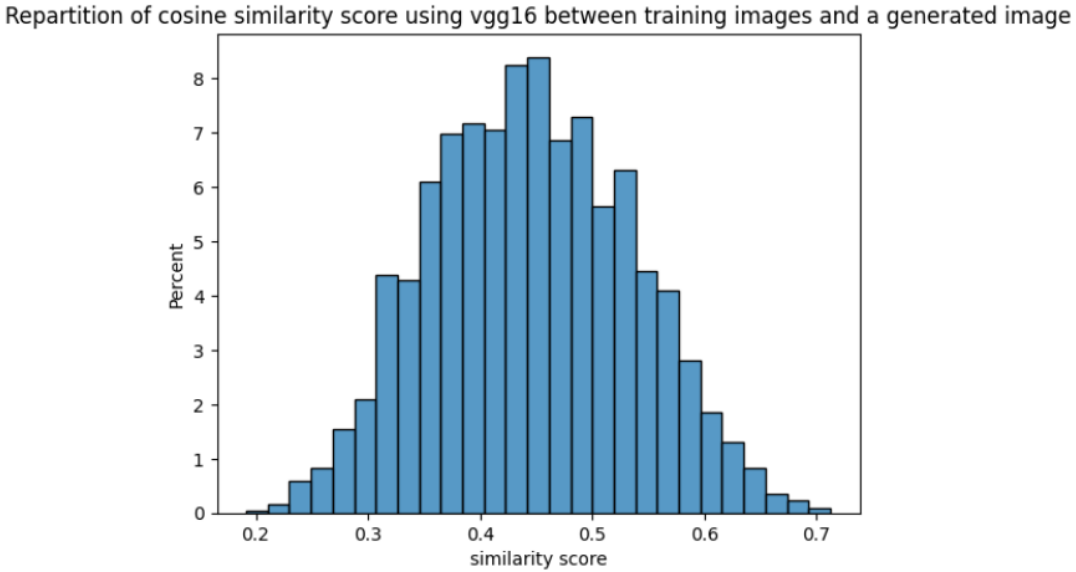




Camille Challier

Repartition of Similarity Scores with VGG16

- Using one generated image and computing all similarity score with vgg16 model.



Conclusion

- The generated mask consistently aligns accurately with the generated waste.
- The background integration of the waste has been effectively executed across the majority of cases.
- Regarding the quality of the generated waste, there exists a range of diversity. Certain instances exhibit inconsistencies, possibly stemming from the diverse backgrounds encompassed within the dataset. We will observe if there are any variations with the utilization of the new dataset.
- The synthesized plastics are distinct from those present in the training set, hailing from a novel source.

Thank you for everything :)

Bibliography

- https://github.com/yifanjiang19/Pedestrian-Synthesis-GAN
- https://uavvaste.github.io/

Repository : https://github.com/CamilleChallier/LSP5_compost