MIDTERM PRESENTATION

Dokmak Mahmoud & Ogay Xavier

Bachelor Project Spring 2023

24/04/2023

DE-OCCLUSION OF OCCLUDED VEHICLE IMAGES FROM DRONE VIDEO

Urban Transport Systems Laboratory

Teacher: Pr. Geroliminis Nikolaos

Supervised by Ms Yura Ta ,& Mr Robert Fonod

PLAN

- Introduction
- Literature review
- First Model Explanations
- Second Model Explanations
- Results
- Comparisons
- Conclusion

INTRODUCTION: DEFINITIONS

What does de-occlusion mean?

De-occluding is the process of removing obstacles or more generally speaking occlusions that obstruct the view of a target object.

What is inpainting?

Inpainting consists in reconstructing damaged or missing parts of an image using surrounding information.

What is our goal?

To improve the detection capabilities of an UAV by removing the different obstacles from the targeted area.

LITERATURE REVIEW

The new methods improve the capabilities of de-occlusion and inpainting thanks to probabilistic and stochastic calculations. In fact, the development of probability theory permits a huge improvement in the diversity and the quality of the generated outputs.

Another major improvement is the development of more precise datasets using different techniques as using in-game pictures of GTA-V or advanced mathematical techniques/mappings.

The majority of studies are specific to facial recognition.

PD-GAN: Probabilistic Diverse GAN for Image Inpainting

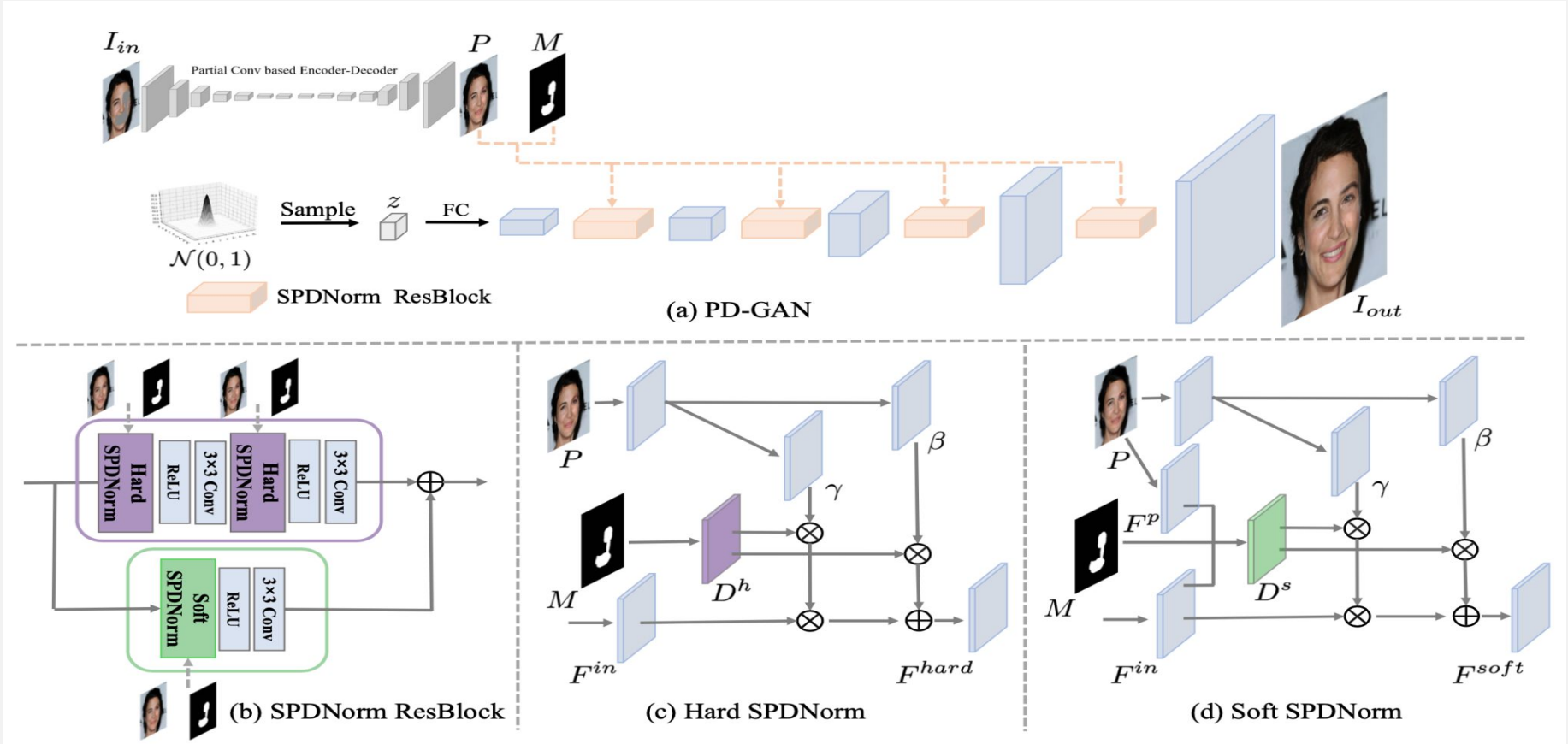
Based on the paper : PD-GAN: Probabilistic Diverse GAN for Image Inpainting

What's the main technique described in the study?

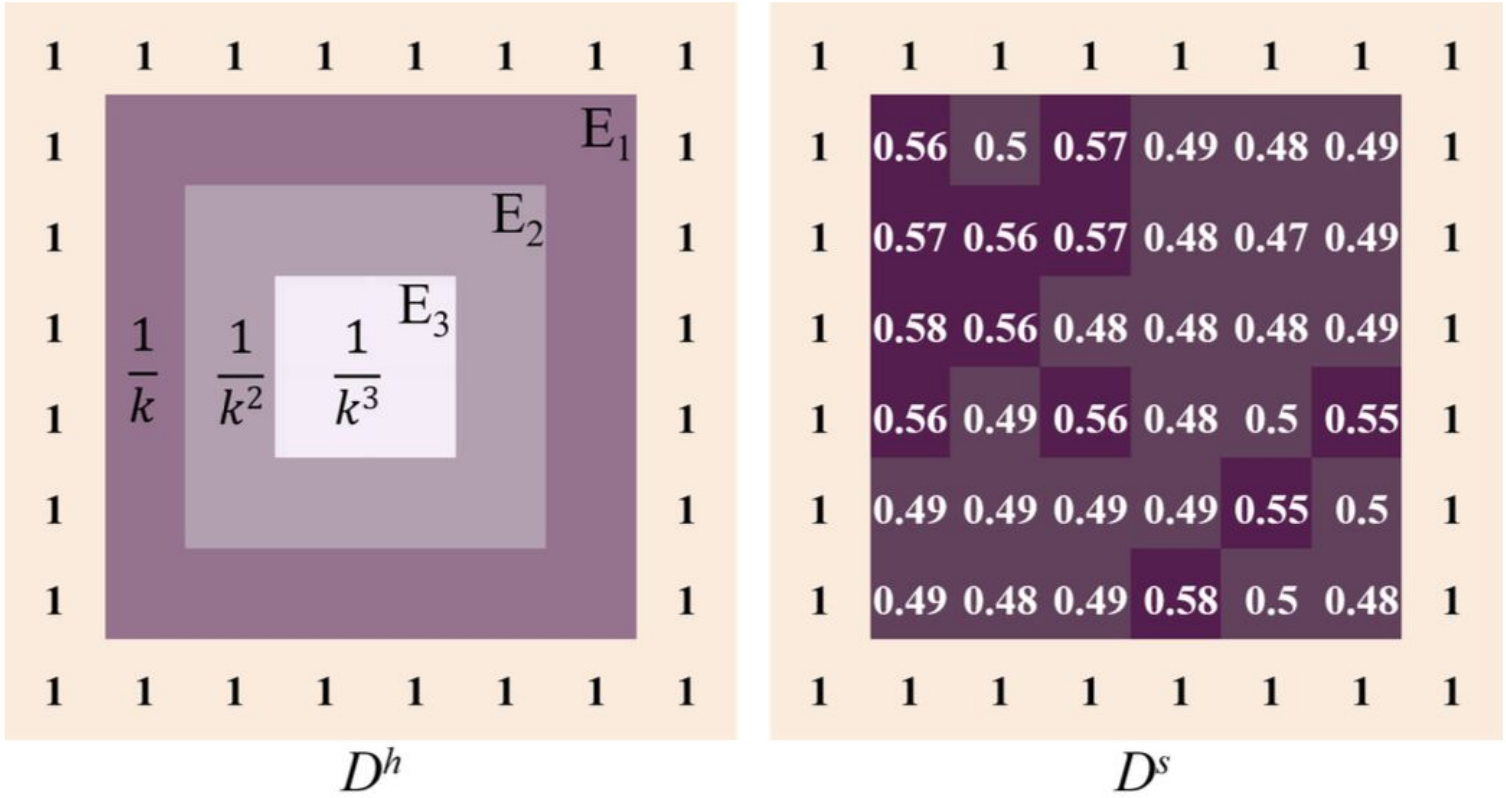
PD-GAN : Probabilistic Diverse GAN : A variant of the vanilla GAN that uses more powerful probability properties to generates multiple and more meaningful outputs to inpaint missing regions.

What's new in it?

Unlike a classic GAN generator which generates images based on random noise, PD-GAN generator uses a SPDNorm matrix that captures informations about the required generation.



PD-GAN use a pretrained partial convolution model that creates an SPDNorm matrix. The SPDNorm matrix extracts lots of informations about the kind of generation needed. The SPDNorm ResBlock consists of SPDNorm Hard and SPDNorm Soft. The Hard SPDNorm controls the probability according to the distance between the pixel and hole boundary, while the Soft SPDNorm learns the probability in an adaptive process.



The Hard probabilistic diversity map D^h is only determined by the mask and transcribes the following observation: The pixels at the border have high dependence on the known one and the pixels in the middle can be much more diverse. The Soft probabilistic diversity map D^s is an adaptive map which is obtained by the input feature and coarse prediction through a learning process.

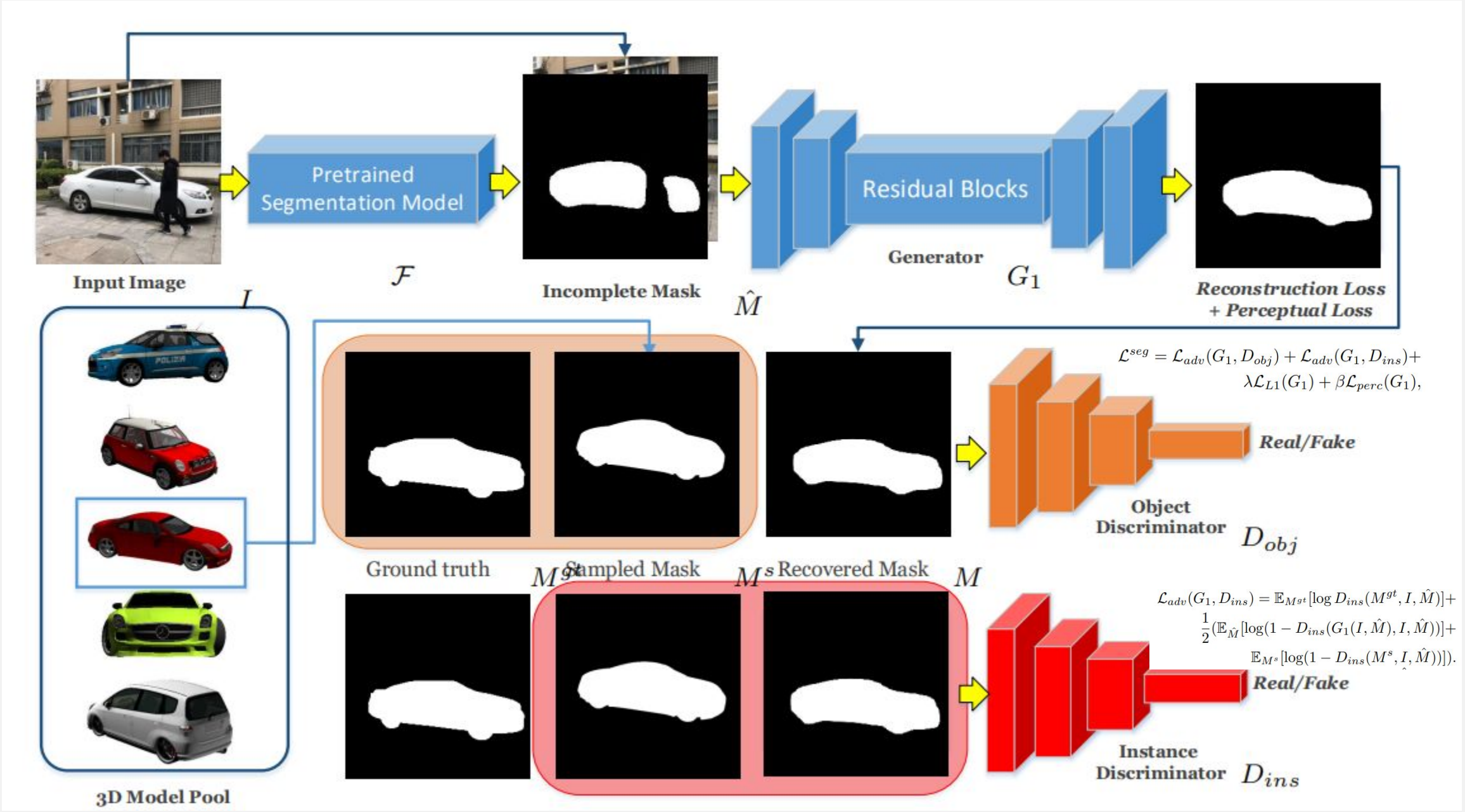
Visualizing the Invisible: Occluded Vehicle Segmentation and Recovery

This model is specifically aimed at vehicle de-occlusion.

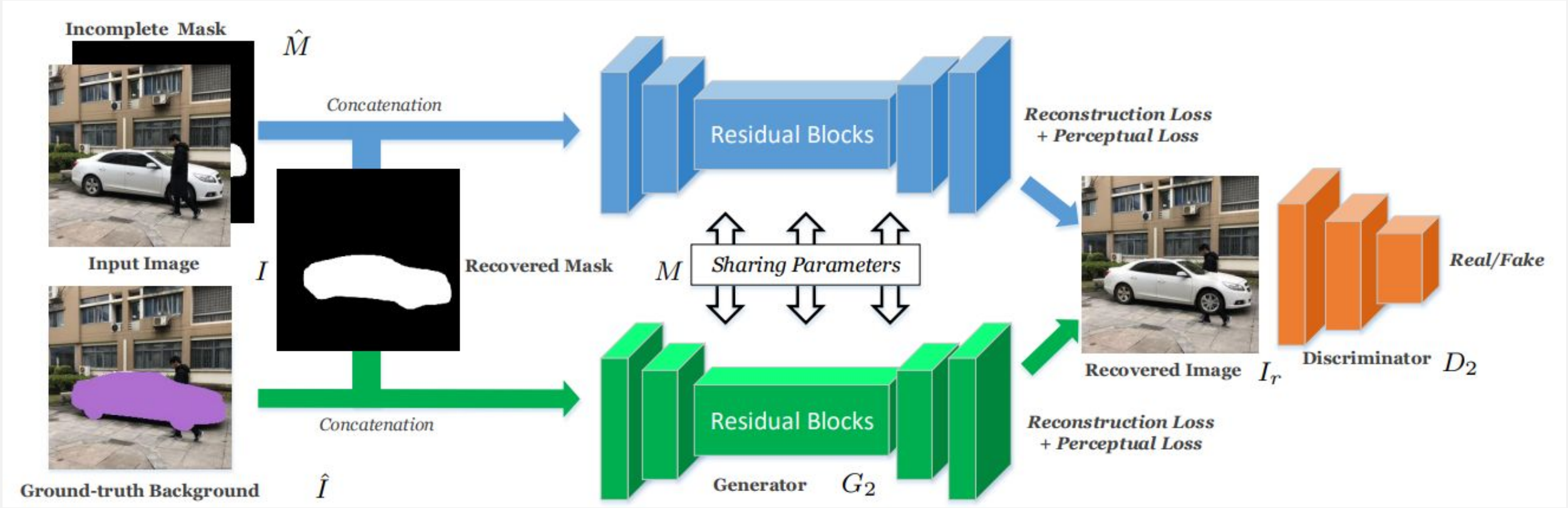
Made of two module:

- The segmentation completion network
  - To create a realistic mask of a vehicle
- The appearance recovery network
  - To recover the occluded appearance

Segmentation completion network



Appearance recovery network



$$\begin{aligned}\mathcal{L}^{app} =& \mathcal{L}_{adv}(G_2(I,\hat{M},M),D_2) + \mathcal{L}_{adv}(G_2(\hat{I},M,\phi),D_2)+\\ & \lambda_1\mathcal{L}_{L1}(G_2(I,\hat{M},M)) + \beta_1\mathcal{L}_{perc}(G_2(I,\hat{M},M))+\\ & \lambda_2\mathcal{L}_{L1}(G_2(\hat{I},M,\phi)) + \beta_2\mathcal{L}_{perc}(G_2(\hat{I},M,\phi)).\end{aligned}$$

Visualizing the Invisible: Occluded Vehicle Segmentation and Recovery
Why not this one?

This last model was specifically designed for vehicle and introduced a novel iterative framework to obtain the segmentation of a vehicle and recover its appearance.

Un-hopefully, the code is not available and we would need to manually label the segmentation of each of the vehicle on the ground truth dataset to create a training dataset.

Also the constraint applied to the mask segmentation is great but not as useful as we always will have a UAV POV.

Other papers

Image Completion with Heterogeneously Filtered Spectral Hints - SH GAN :

Introduce a new Spectral transform strategies:
Heterogeneous Filtering and Gaussian Split for large scale free form missing region inpainting.

Code available but with no training pipeline.

Can GAN Hallucinate Occluded People with a Plausible Aspect? :

Present the use of GANs for image enhancing in people attributes classification with occlusion.
Introduce an innovative way of creating a dataset with generated video games images

No code available.

Large Scale Image Completion Via Co-Modulated Generative Adversarial Networks

it allows the generator network to adaptively adjust its feature maps based on the conditioning vector, which enables it to capture high-level information about the image being generated.

No code available.

Complete Face Recovery GAN: Unsupervised Joint Face Rotation and De-Occlusion from a Single-View Image

Very powerful Technique that uses the 3D Morphable Model (3DMM) a statistical model for faces shape and texture to create a 3D image with the correct shape and texture. By creating the 3D model, the model can deal with occlusions and face rotations. This technique is very powerful for face recognition.

Not applicable for vehicles because there is no model similar to 3DMM for vehicles..

Older papers for understanding:

Occlusion-Aware Gan For Face De-Occlusion in the Wild

Context Encoders: Feature Learning by Inpainting

FIRST MODEL : REPAINT

Based on the paper :"RePaint: Inpainting using Denoising Diffusion Probabilistic Models"

What's the main technique(s) described in the article?

RePaint : An inpainting technique similar to the Denoising Diffusion Probabilistic Models (DDPM)

How does it work?

It creates several noisy version of our image by adding i.i.d. Gaussian noise at each iteration until it is totally noised. A neural network revert it by predicting the cumulative noise and feeding back with samples from the training that is the closest. We do so until having a clear picture.

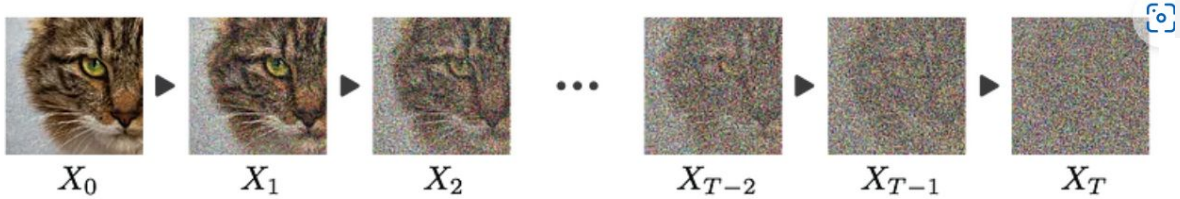
Denoising Diffusion Probabilistic Models

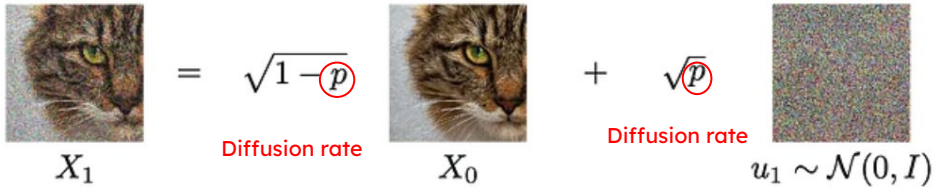
To understand RePaint we news to look at DDPM papers as RePaint does not modify or condition the original DDPM network itself.

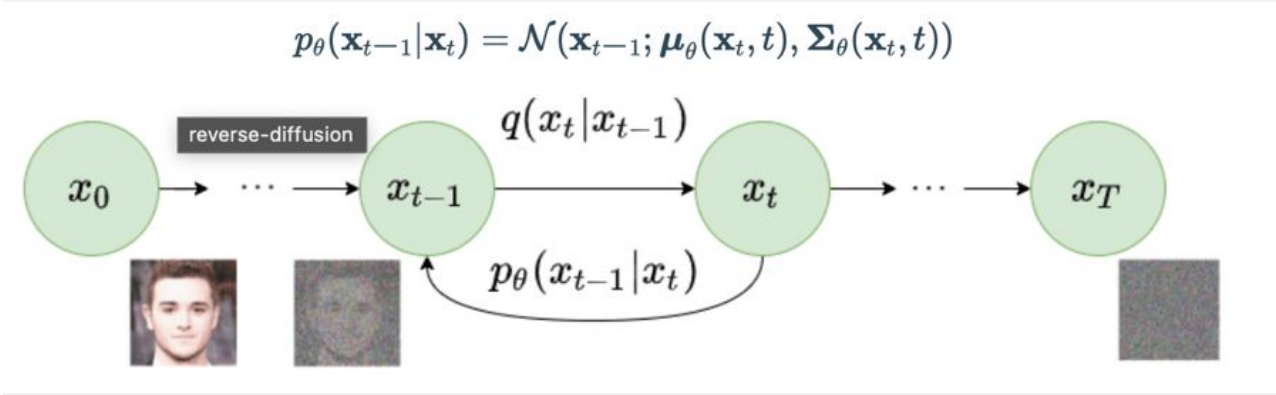
Learning a function (represented by a neural network) that takes some gaussian noise and output data from a targeted complex distribution can be pretty hard.

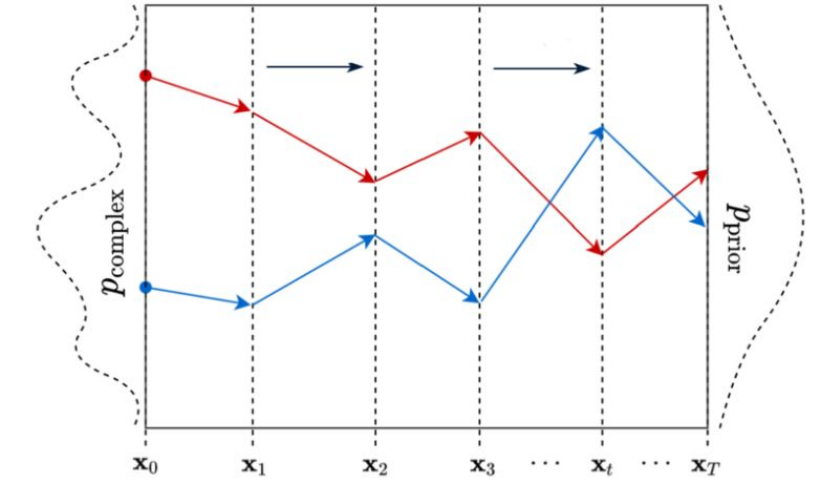
The idea behind Diffusion Probabilistic Models is to learn the reverse process of a diffusion, expected to go progressively from some gaussian noise to a complex distribution by removing noise.

From a complex distribution to a simple isotropic gaussian noise

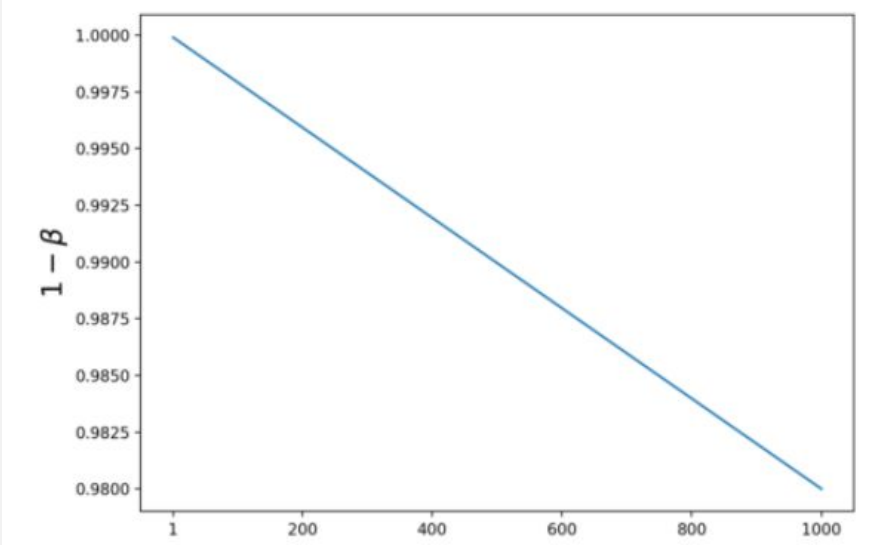


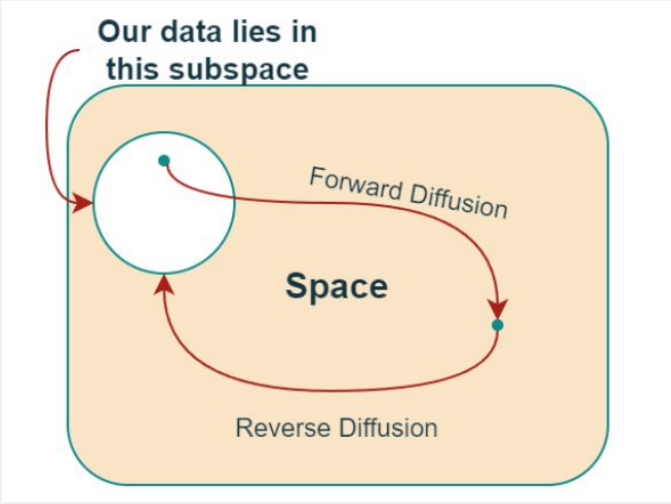


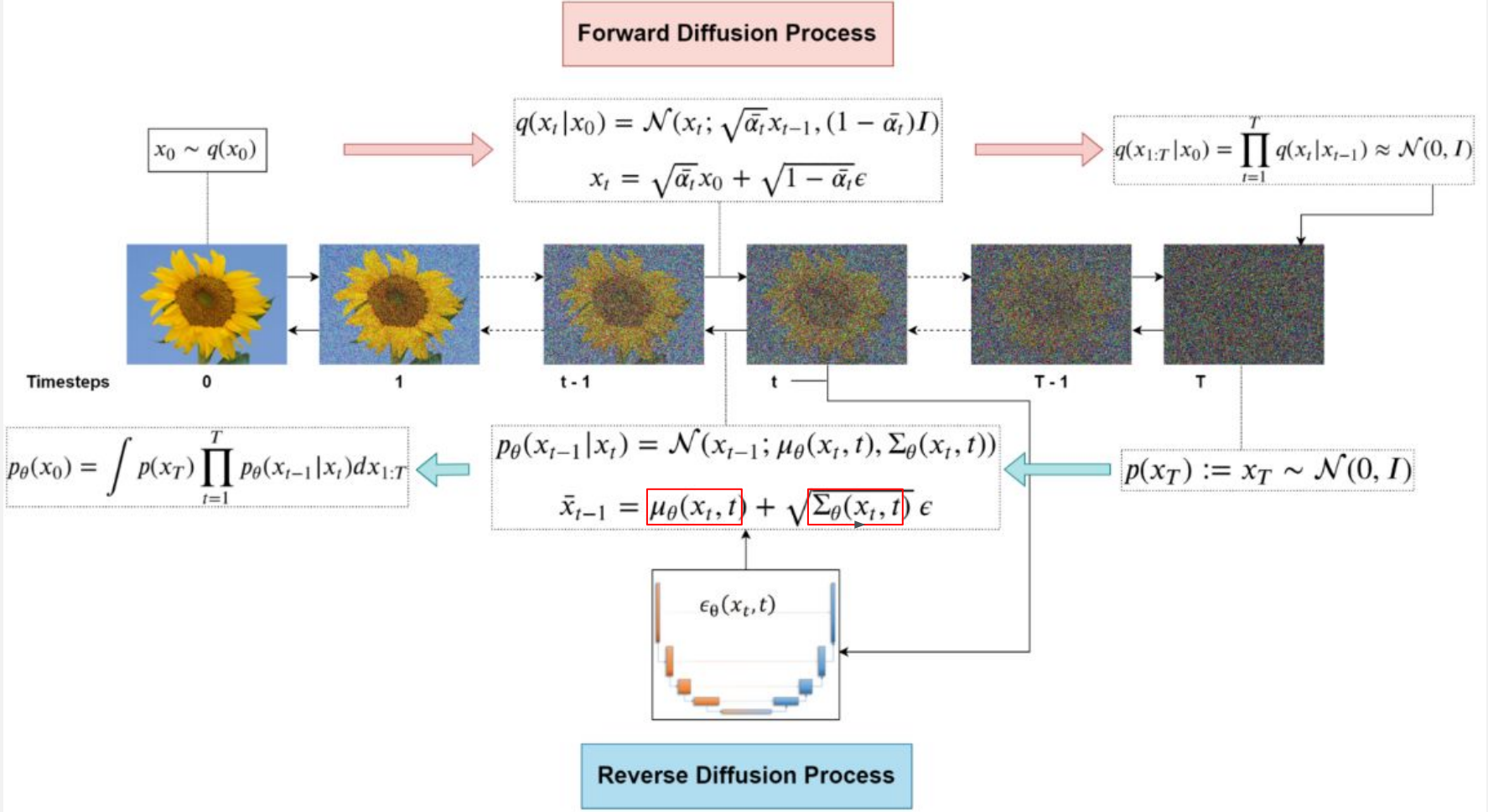




Linear Variance scheduler







Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

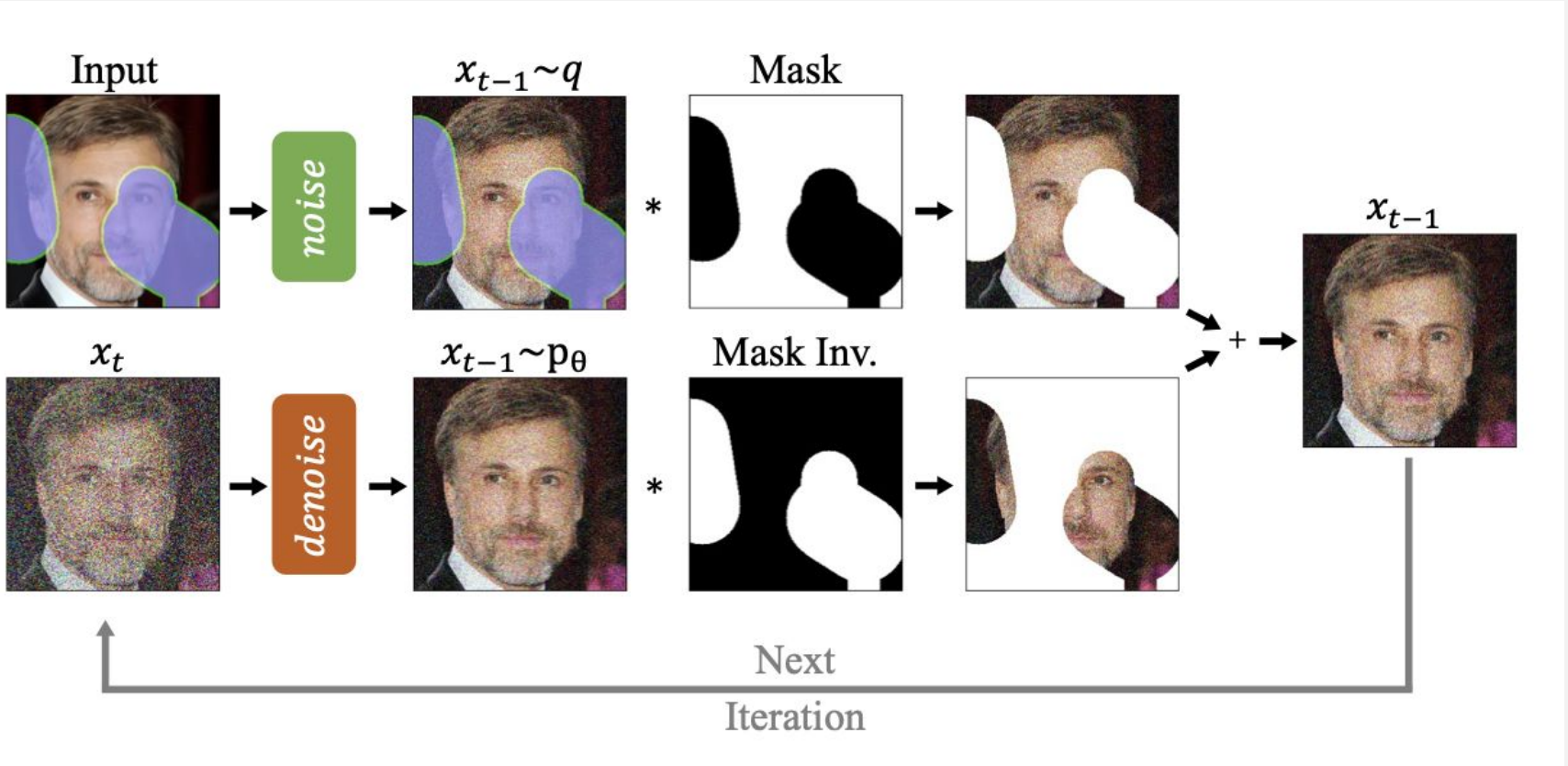
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

RePaint

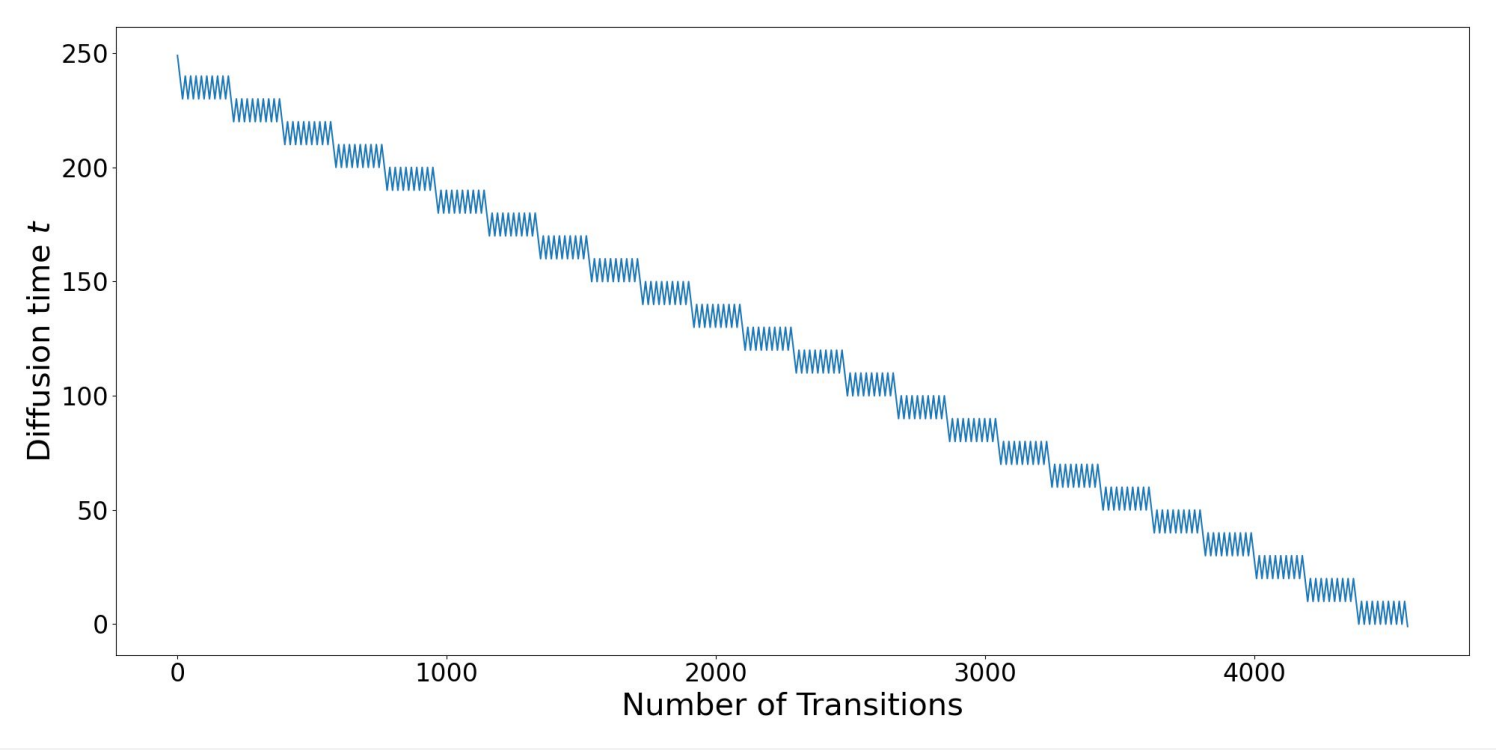
Algorithm 1 Inpainting using our RePaint approach.

1: $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: **for** $u = 1, \dots, U$ **do**
4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon = \mathbf{0}$
5: $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
6: $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
7: $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
8: $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
9: **if** $u < U$ and $t > 1$ **then**
10: $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}} x_{t-1}, \beta_{t-1}\mathbf{I})$
11: **end if**
12: **end for**
13: **end for**
14: **return** x_0



By assuming that the reverse transition function p is also Gaussian, we only need to find the mean and the variance of the next image. The Neural Network is going to add new points based on the current state and the proximity of its distribution. The training outputs the joint distribution between an image and its noised version.

Variance scheduler



Good points (+)

1. Requires no mask-specific training and generalize to any mask even big ones.
2. Produces sharp, highly detailed and semantically meaningful images
3. More flexible and diverse in the generation
4. Harmonize generation (no border delimitation)

Defaults (-):

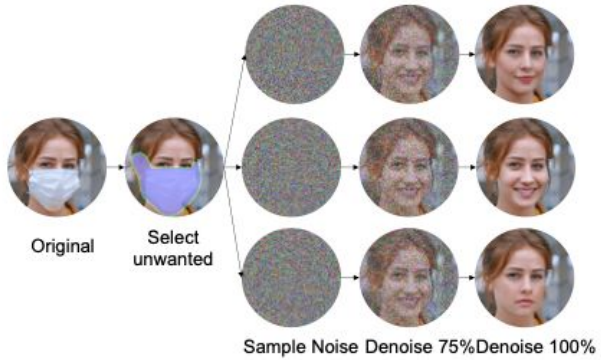
1. More computational expensive than other techniques. Sampling requires multiple steps, meaning that the generative process will be longer than it is for GANs or VAEs
2. Slow optimization process which makes it currently difficult to apply it for real-time applications
3. RePaint can produce realistic images completions that are very different from the Ground Truth image (Hallucination)

RePaint: Inpainting using Denoising Diffusion Probabilistic Models

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, Luc Van Gool
Computer Vision Lab, ETH Zurich

GitHub
git.io/RePaint

Introduction



Motivation

- Current Autoregressive and GAN Inpainting Methods:
  - Limited generative capabilities lead to failure for large masks.
  - Design for specific masks lead to fail on sparse masks.
- Diffusion Models showed good generative capabilities.
- The conditioning process for Diffusion Model Inpainting lacked harmonization of the known and generated part.

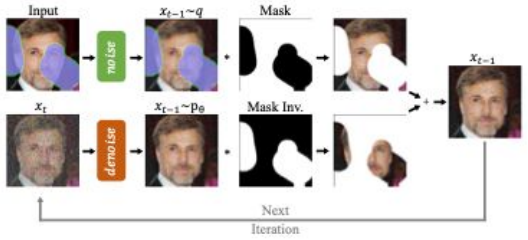
Contribution

- Method to condition an unconditionally trained Diffusion Models.
- Inference schedule generalizes to any inpainting mask.
- Generate semantically meaningful image completions.
- Harmonize generated and known part for inpainting.
- Analysis of inpainting algorithms on six different masks.

Method

Overview

- Condition inference of Diffusion Model
- No training or finetuning of the model
- Harmonization of known and generated content
  - Go forward and backward in diffusion time
  - Using larger jumps improve perceptual quality



Conditioning

$$x_{t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$
$$x_{t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$$
$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1-m) \odot x_{t-1}^{\text{unknown}}$$

Resampling

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  for $u = 1, \dots, U$ **do**
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon = \mathbf{0}$
    $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t}x_0 + (1-\bar{\alpha}_t)\epsilon$
    $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
    $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right) + \sigma_t z$
    $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1-m) \odot x_{t-1}^{\text{unknown}}$
    if $u < U$ and $t > 1$ **then**
      $x_t \sim \mathcal{N}(\sqrt{1-\beta_{t-1}}x_{t-1}, \beta_{t-1}\mathbf{I})$
    end if
  end for
end for
return x_0

Ablation Study

Resampling vs Slowing Down

	T	r	LPIPS	T	r	LPIPS	T	r	LPIPS	T	r	LPIPS
Slowing down	250	1	0.168	500	1	0.167	750	1	0.179	1000	1	0.161
Resampling	250	1	0.168	250	2	0.148	250	3	0.142	250	4	0.134

Jump Length

	$j=1$		$j=5$		$j=10$	
r	LPIPS	Votes [%]	LPIPS	Votes [%]	LPIPS	Votes [%]
5	0.075	42.50±7.7	0.072	46.88±7.8	0.073	53.12±7.8
10	0.088	42.50±7.7	0.073	45.62±7.8	0.068	56.25±7.8
15	0.065	46.25±7.8	0.063	53.12±5.5	0.065	53.75±7.8

Number of Resampling

Experiments

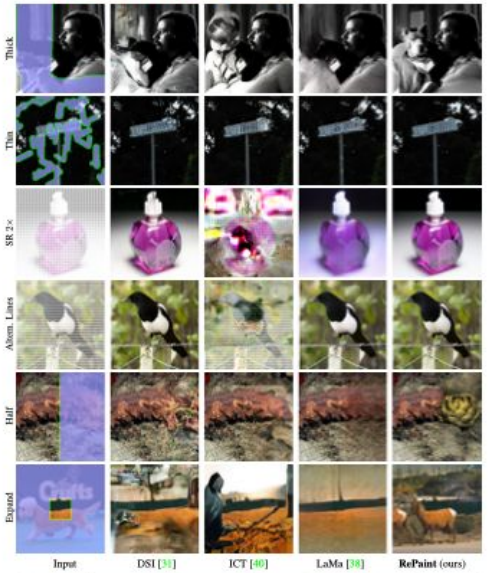
Experiments

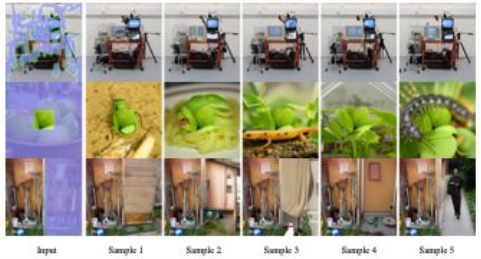
- Datasets: CelebA-HQ, ImageNet, Places2
- Masks: Thin, Thick, Generate Half, Expand, Every Second Line, Super-Resolution
- Class Conditional Inpainting
- Extensive use study

SOTA Comparison

ImageNet	Wide		Narrow		Super-Resolve 2×		Altern. Lines		Half		Expand	
Methods	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]
DSI [31]	0.117	31.7 ± 2.9	0.072	28.6 ± 2.8	0.153	26.9 ± 2.8	0.069	23.6 ± 2.6	0.283	31.4 ± 2.9	0.583	9.2 ± 1.8
ICT [40]	0.107	42.9 ± 3.1	0.073	33.0 ± 2.9	0.708	1.1 ± 0.6	0.620	6.6 ± 1.5	0.255	51.5 ± 3.1	0.544	25.6 ± 2.7
LaMa [38]	0.105	42.4 ± 3.1	0.061	33.6 ± 2.9	0.272	13.0 ± 2.1	0.121	9.6 ± 1.8	0.254	41.1 ± 3.1	0.534	20.3 ± 2.5
RePaint	0.134	*Reference*	0.064	*Reference*	0.183	*Reference*	0.089	*Reference*	0.304	*Reference*	0.629	*Reference*

Visual Examples





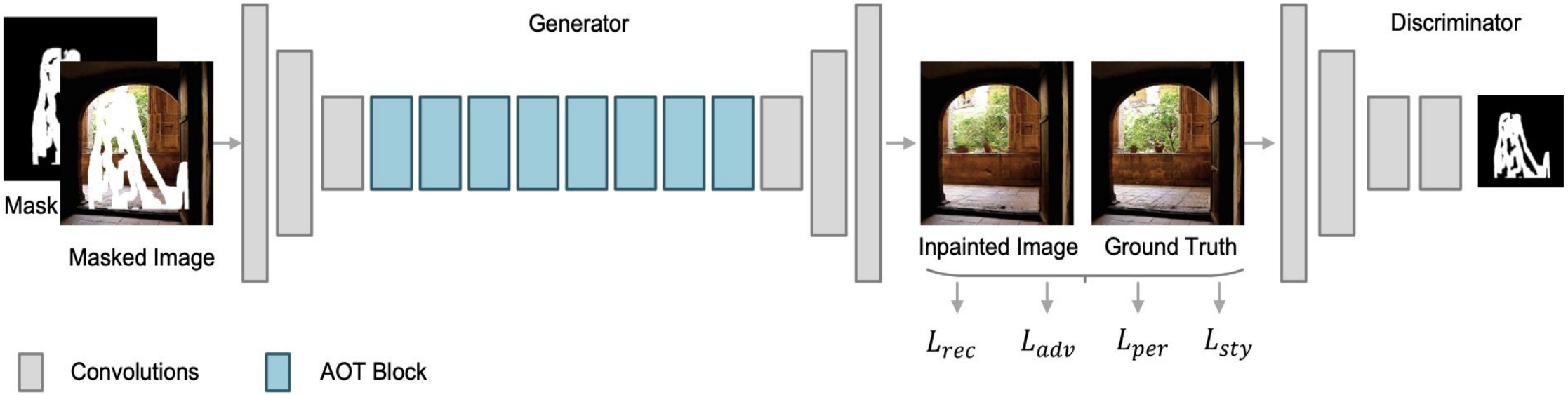
Class Conditional Inpainting

SECOND MODEL : AOT-GAN

Based on the paper : "Aggregated Contextual Transformations for High-Resolution Image Inpainting"

What is a GAN?

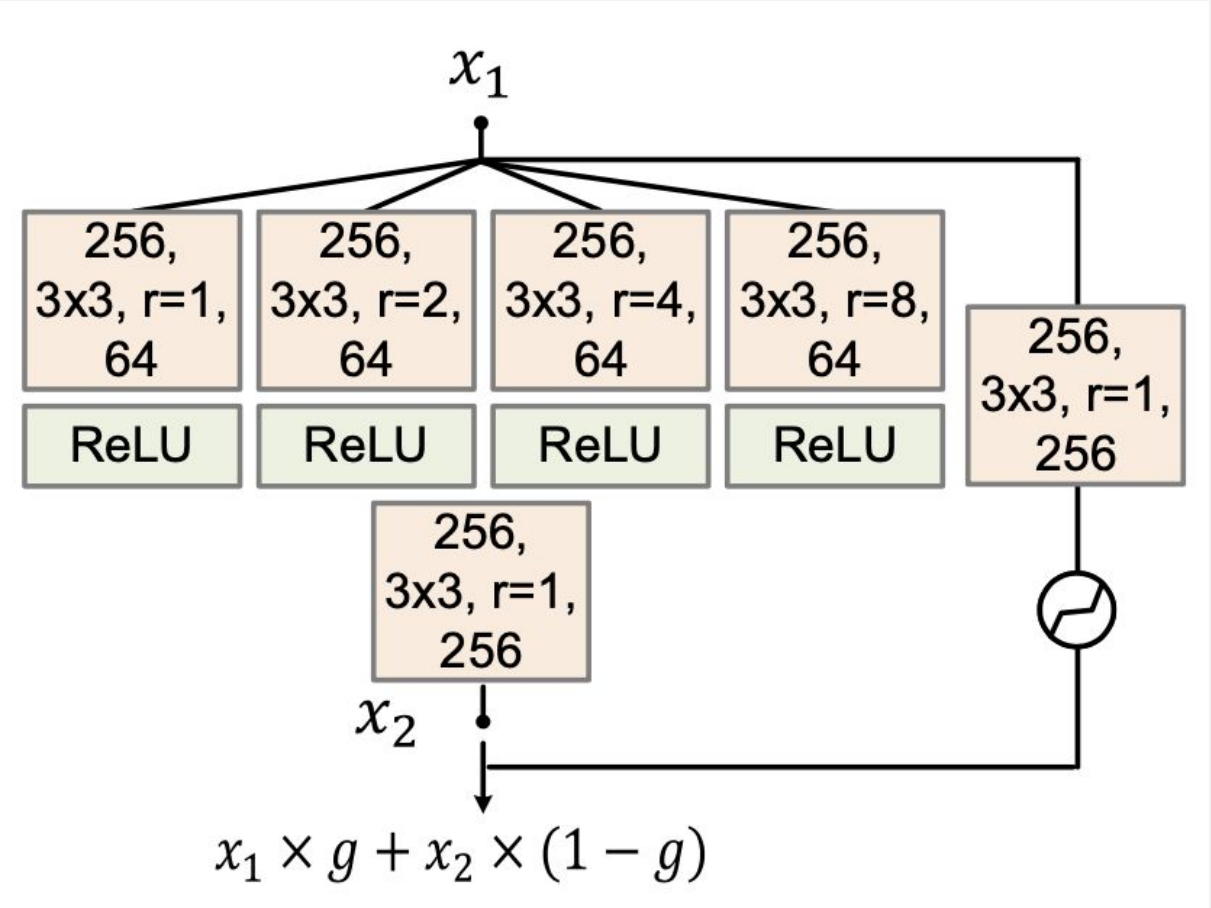
A GAN Model that consists of two neural networks called generator and discriminator trained in an adversarial manner. During training the generator with the help of a random noise (standard normal Gaussian) creates new images and gives it to discriminator. The discriminator tries to determine if the image is real or generated. The operation is repeated until the discriminator is fooled ie. it can't differentiate between the generated and the real image.



What does the AOT-GAN modify?

OAT-GAN is an optimization of Patch-GAN. Patch-GAN is a well known GAN model that gives the following modification from the classical one : The discriminator compares patch-wisely all patches in the inpainted image. OAT-GAN modifies the PATCH-GAN discriminator (that already works well) to perform comparisons only on the inpainted region. This optimization lets us to reduce the computational time and synthesize more realistic fine-grained textures.

OAT-GAN gets also an other modification in its generator part leading to better context reasoning for missing regions. This is done by using AOT Block. AOT Blocks performs various dilation rates which is going to capture meaningful context informations (even from far inputs) by aggregating multiple transformation results.



AOT blocks adopt the split-transformation-merge strategy by three steps. (i) Splitting : AOT block splits the kernel of a standard convolution into multiple sub-kernels, each of which has fewer output channels. For example, splitting a kernel with 256 output channels into four sub-kernels makes each sub-kernel has 64 output channels. (ii) Transforming : each sub-kernel performs a different transformation of the input feature x_1 by using a different dilation rate. (iii) Aggregating: the contextual transformations from different receptive fields are finally aggregated by a concatenation followed by a standard convolution for feature fusion. Such a design allows the AOT block to predict each output pixel through different views. Through the above three steps, an AOT block is able to aggregate multiple contextual transformations for enhancing context reasoning.

Good points (+)

1. Works with large images
2. Works with very large masks
3. Produces sharp, highly detailed and semantically meaningful images
4. Harmonized generation

Defaults (-):

1. More computational expensive than other GAN techniques
2. Cannot work on a set of image of different size.

INTERMEDIATE RESULTS

	GAN	Diffusion
Pros	Fast Sampling rate. High sample generation quality.	High sample generation quality. Diverse sample generation
Cons	Unstable training, low sample generation diversity (Mode Collapse)	Low sampling rate

Next are some results on the models respectively trained as on their release on their papers.
Image of 256x256 where used.

AOT GAN was trained on Places2 (building) and RePaint was trained on ImageNet (general but a lot of dog).

Small masks

AOT

GT

RePaint

AOT

GT

RePaint

Bigger Mask

AOT

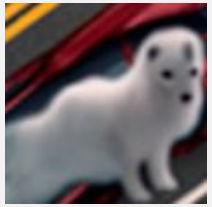
GT

RePaint

AOT

GT

RePaint



HOW TO MEASURE PERFORMANCE?

- L1/Manhattan Norm : Compute the mean absolute error between the generated image and the original one to determine the per-pixel reconstruction accuracy.
- L2/Euclidean Norm : Compute the mean squared error between the generated image and the original one to determine the per-pixel reconstruction accuracy.
- SSIM : It's a metric based on the human perception that outputs a number between 0 and 1 based on the similarity of three aspects : luminance, contrast & structure. An output of 1 means it's the same image.
- PSNR : It computes the ratio between the maximum possible power of the image and the power of the noise that is introduced during the reconstruction. The output is expressed as the logarithm of the PSNR ratio in db. A greater output means better performance.
- LPIPS : LPIPS is used to judge the perceptual similarity between two images. LPIPS essentially computes the similarity between the activations of two image patches for some pre-defined network. This measure has been shown to match human perception well. A low LPIPS score means that image patches are perceptual similar.
- MAE : measures the average magnitude of the errors in a set of predictions ie. the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.
- FID : The Fréchet distance compute the distance between two probability distributions : the distribution of the reconstructed images and the real images. A smaller output is a sign of good performance.

DIFFERENCES OF PERFORMANCE

Mean of metrics on previous results

l1 -> 0
l2 -> 0
ssim -> 1
psnr -> inf
lpips -> 0.0
mae -> 0
fid -> 0

	AOT small	RP small	AOT big	RP big
l1	0.0906	0.1469	0.1364	0.1997
l2	0.0258	0.0683	0.046	0.086
ssim	0.7409	0.6573	0.531	0.4253
psnr	18.769	15.9563	14.2656	11.3369
lpips	2.3577	2.3411	2.7835	2.9692
mae	0.1316	0.2055	0.1895	0.2625
fid	337	322.99	311.92	316.922

GENERATING MULTIPLE OUTPUTS

RePaint model, as said, is capable of generating multiple output. It is a very interesting property but at this stage of our work, we don't think it will be useful for us. In fact, with RePaint we just need to iterate the model for generating another image. So we don't lose anything to not work on it. AOT-GAN doesn't offer this option.

We hope to finetune a model which generate vehicle inpainting and we are going to train our model to this purpose. This should, by itself, condition well and prevent the generation of unrealistic outputs. But in the big picture, this ability to create multiple outputs from the same input could help to avoid some outliers. For example by making multiple outputs from the same image/mask and calculate the mean of probability of being "this" vehicle. Unfortunately as DDPM inferences are pretty expensive it looks pretty unrealistic compared to the earning it adds.

CONCLUSION

- The GAN models trained with Places2 seems to outperform RePaint trained on ImageNet. But not really relevant as not trained on vehicle nor on the same dataset

- Upcoming Work : -Create a dataset with masks to train our models
  -Finetune the model to perform on vehicle UAV POV