Landscape Generation

Project for the course of Deep Learning 2022-2023

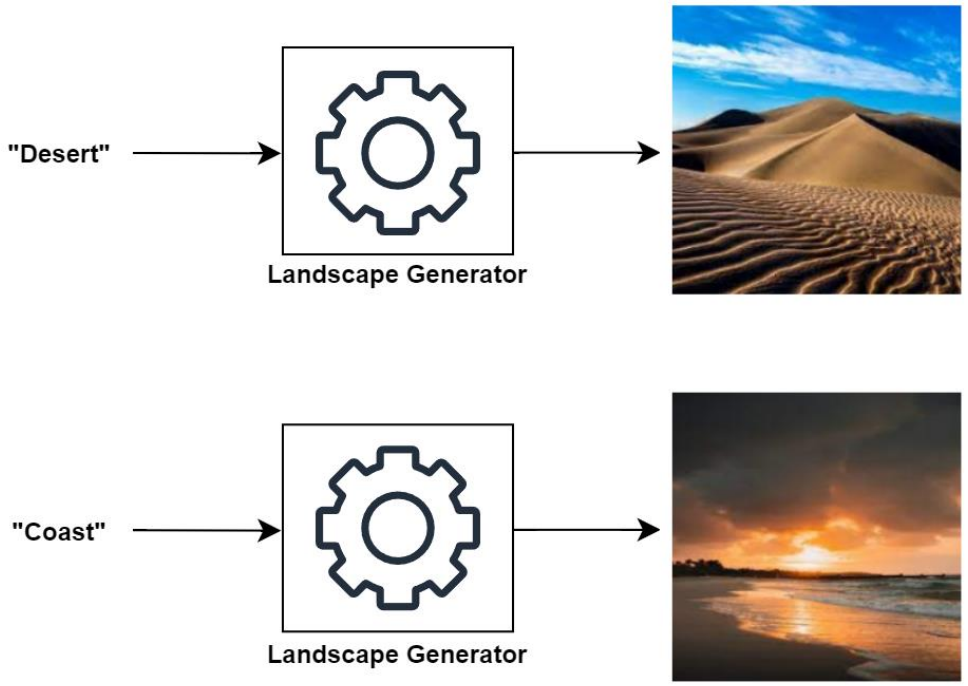
Jahrim Gabriele Cesario
jahrim.cesario2@studio.unibo.it

Repository: https://github.com/jahrim/dl-project
Colab: https://colab.research.google.com/drive/1vof-GBruVmq2ZhjEyifVLhNtbRypOBzc?usp=sharing

Business Understanding

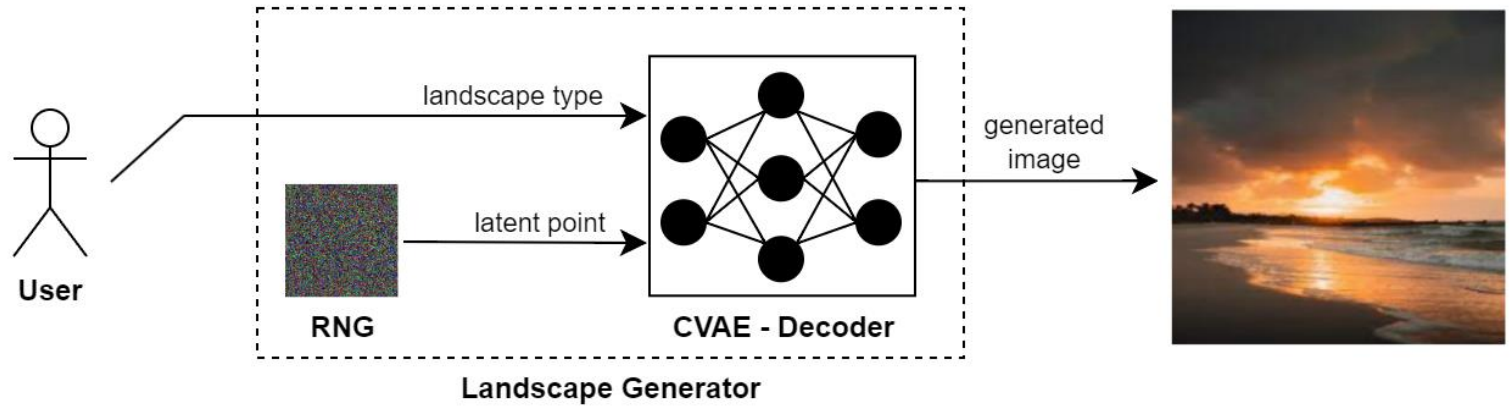
Problem

The goal of this project is to train a generative model capable of creating images representing different kinds of landscapes.



Strategy

To let the user choose the kind of landscape to generate during prediction, the type of generative model trained in this project will be the **Decoder** of a **Conditional Variational AutoEncoder (CVAE)**.



Data Understanding

Dataset

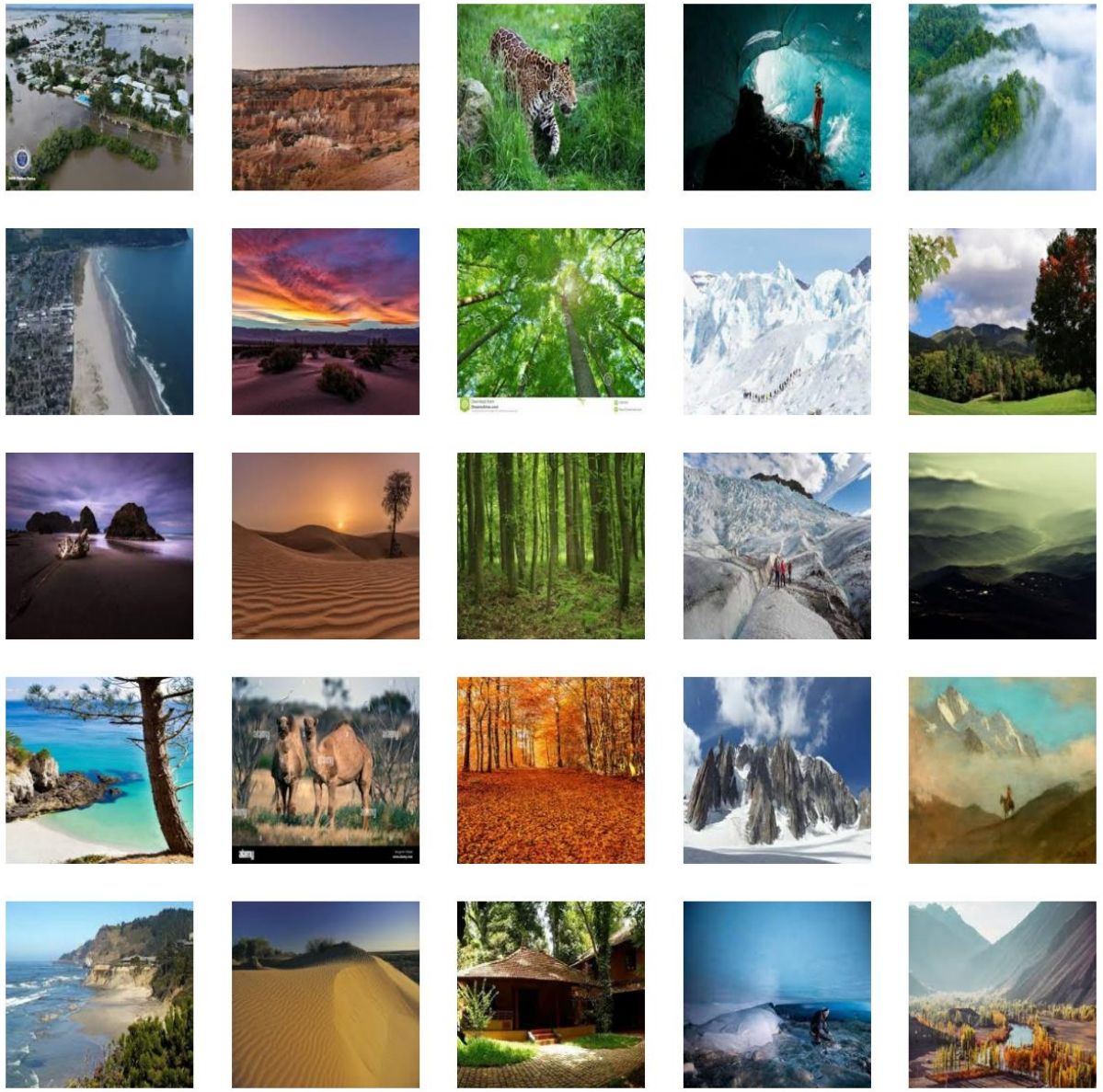
The training will be based on the Landscape Recognition Dataset available on **Kaggle**, which contains **12.000** images of **Coasts**, **Deserts**, **Forests**, **Glaciers** and **Mountains**. The dataset is **balanced** with respect to the landscape types.

Each object of the dataset is a TFRecord containing a **256x256 RGB** image bound to a **numerical** label.

256

256

Label Name	Label Id
Coast	0
Desert	1
Forest	2
Glacier	3
Mountain	4



F1. A sample of 25 images extracted from the dataset. Each column contains some pictures of a specific landscape type. In order: *Coasts*, *Deserts*, *Forests*, *Glaciers* and *Mountains* (as labelled within the dataset).

Observations (1/3)

- **Mixed Landscapes**: landscapes are often a mix of different landscape types. The model will have to deal with **outliers**.

Forest

Forest

Mountain

Observations (2/3)

- **Secondary Subject**: landscapes are often a background for the main subjects of the pictures. The model will have to deal with **noise**.

Desert

Forest

Forest

Observations (3/3)

- **High Variance**: each landscape type can be expressed by a large pool of **shapes**, **colors** and **perspectives**. The model will have to learn all these features for each landscape type.

Coast

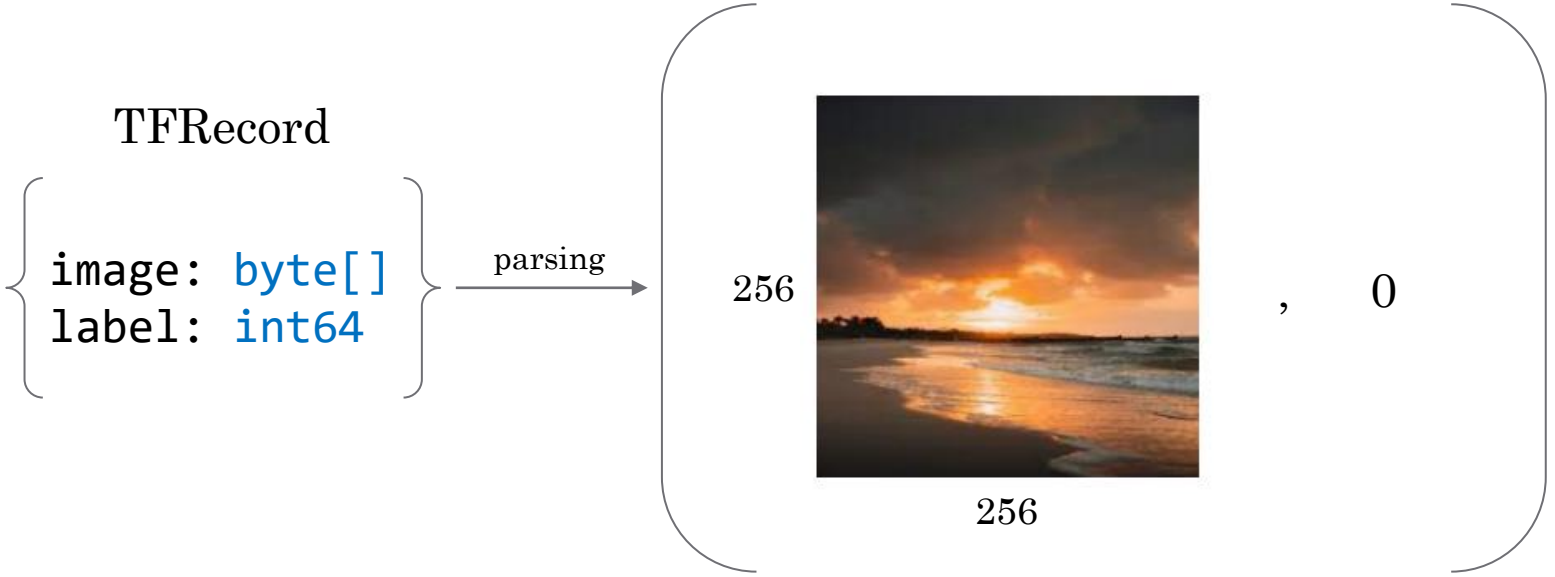
Coast

Coast

Data Preparation

Parsing

The TFRecords have been **parsed to extract the images and labels** of the dataset as tensors.

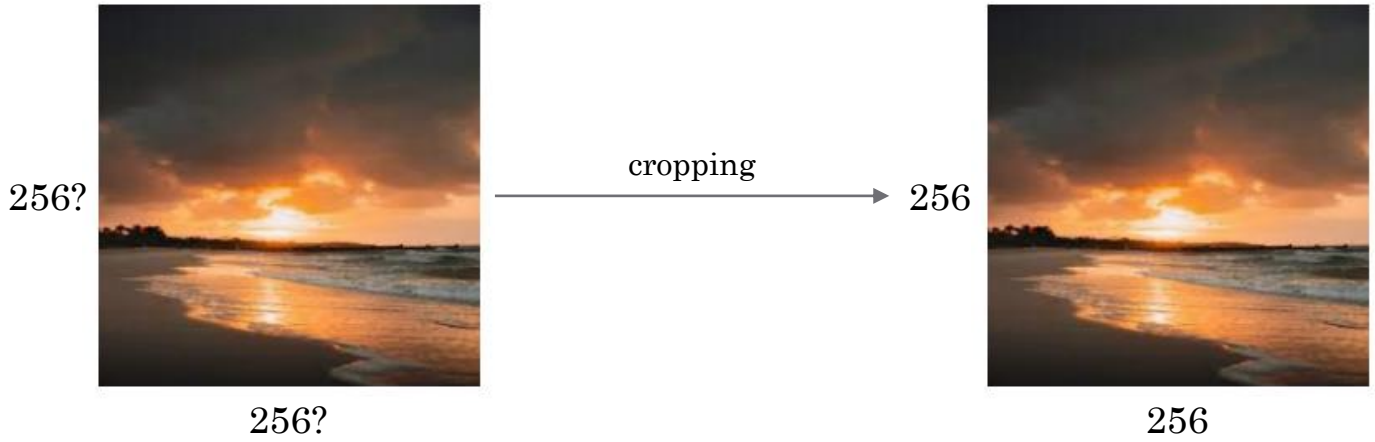


Filtering

In some experiments, the landscape types have been **filtered to reduce the variance** in the training set, analyzing the performances of the model in different conditions.

Cropping

The images have been **cropped to make sure that they all had the same size**.

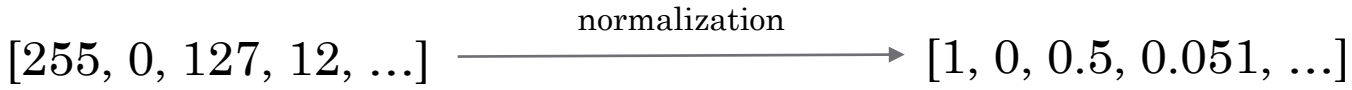


Resizing

In some experiments, the images have also been **resized to reduce the complexity** of the generation problem.

Normalization

The images have been **normalized** from values in {0..255} to values in [0; 1] **to increase the training stability**.



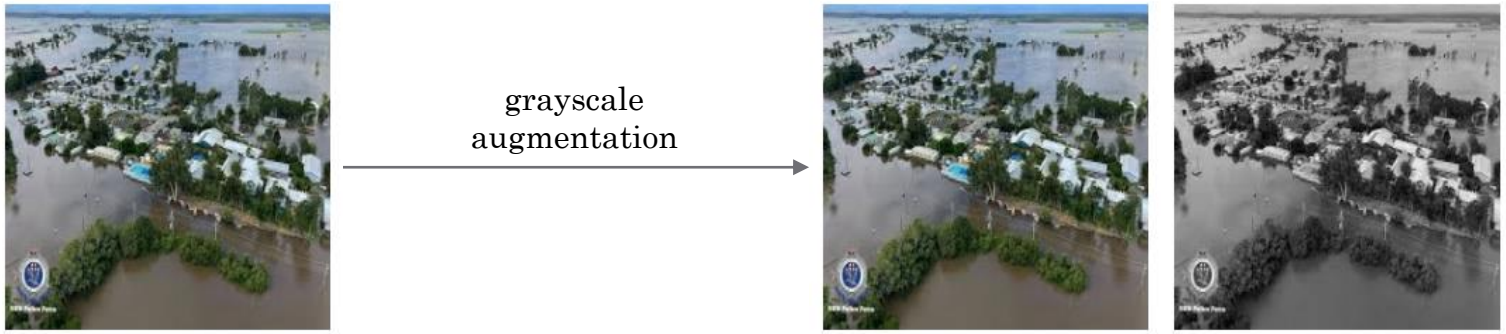
One-Hot Encoding

The labels of the dataset have been transformed using **one-hot encoding to forbid the model from inferring similarities** between the labels from the similarities between their identifiers.

Label Name	Label Id	D(0,y)	One-Hot Encoding	$D(0_{enc}, y_{enc})$
Coast	0	0	[1, 0, 0, 0, 0]	0
Desert	1	1	[0, 1, 0, 0, 0]	$\sqrt{2}$
Forest	2	2	[0, 0, 1, 0, 0]	$\sqrt{2}$
Glacier	3	3	[0, 0, 0, 1, 0]	$\sqrt{2}$
Mountain	4	4	[0, 0, 0, 0, 1]	$\sqrt{2}$

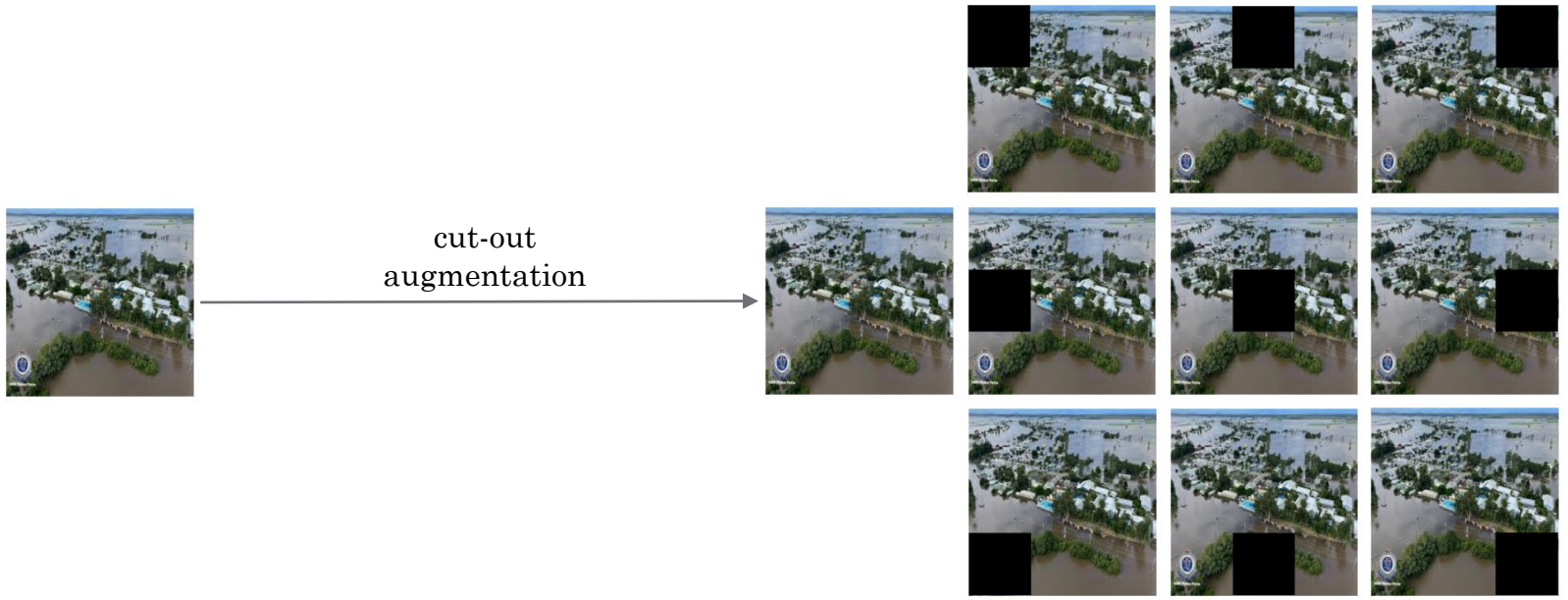
Grayscale Augmentation

In some experiments, the training set has been augmented by transforming the images into **grayscale images**, with the intention of **reducing the amount of focus that the model puts in the colors** for distinguishing the different landscape types.



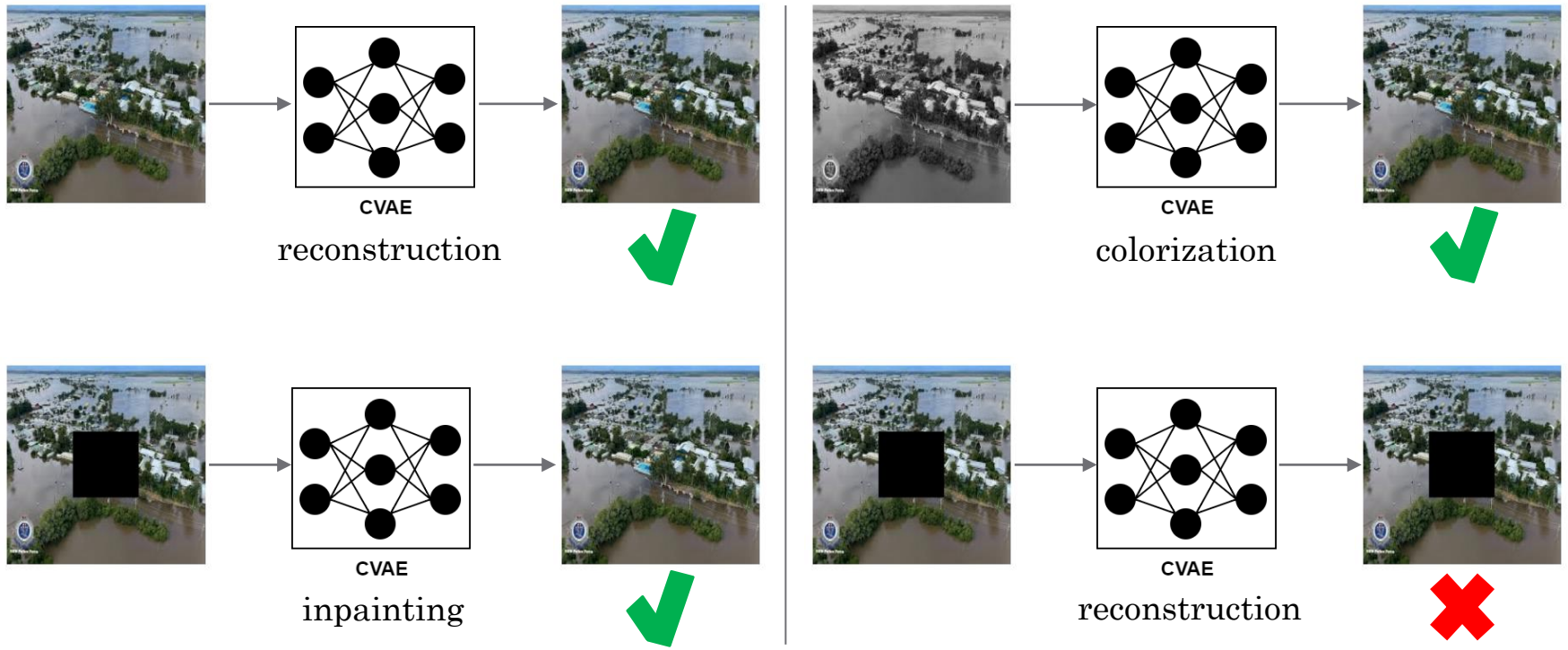
Cut-Out Augmentation

In some experiments, the training set has been augmented by using the **Cut-Out** method, with the intention of **forcing the model to focus on different parts of the images during training**.



Observations

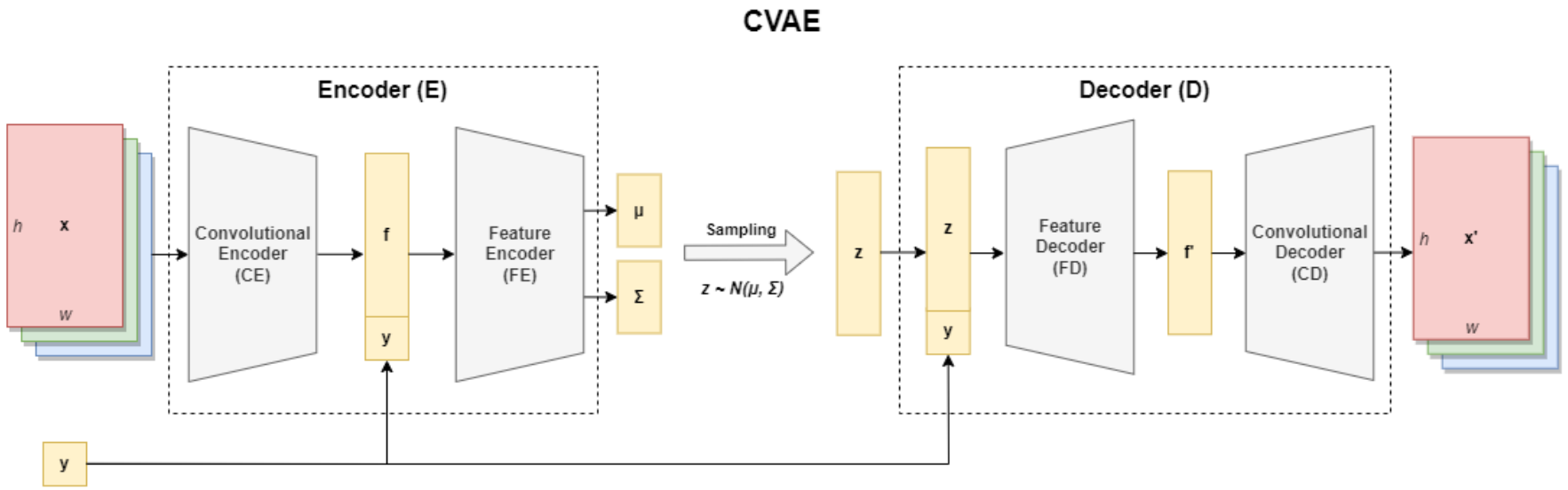
In retrospective, the effectiveness of the augmentations could have been improved by **training the model on different tasks** other than reconstructing the original images.



Modelling

CVAE

In the first experiments, the **CVAE** included a **Convolutional Encoder (Decoder)** to map images into features (viceversa) and a **Feature Encoder (Decoder)** to map such features into points in the latent space (viceversa).



Narrow Variant

- **CE:** deep CNN
- **FE & FD:** deep FFNN
- **CD:** reverse of **CE**
- **Features**: ~ 1024 - 2048
- **Weights**: ~ 1 - 15M
- **Example Graph:** link

AlexNet Variant

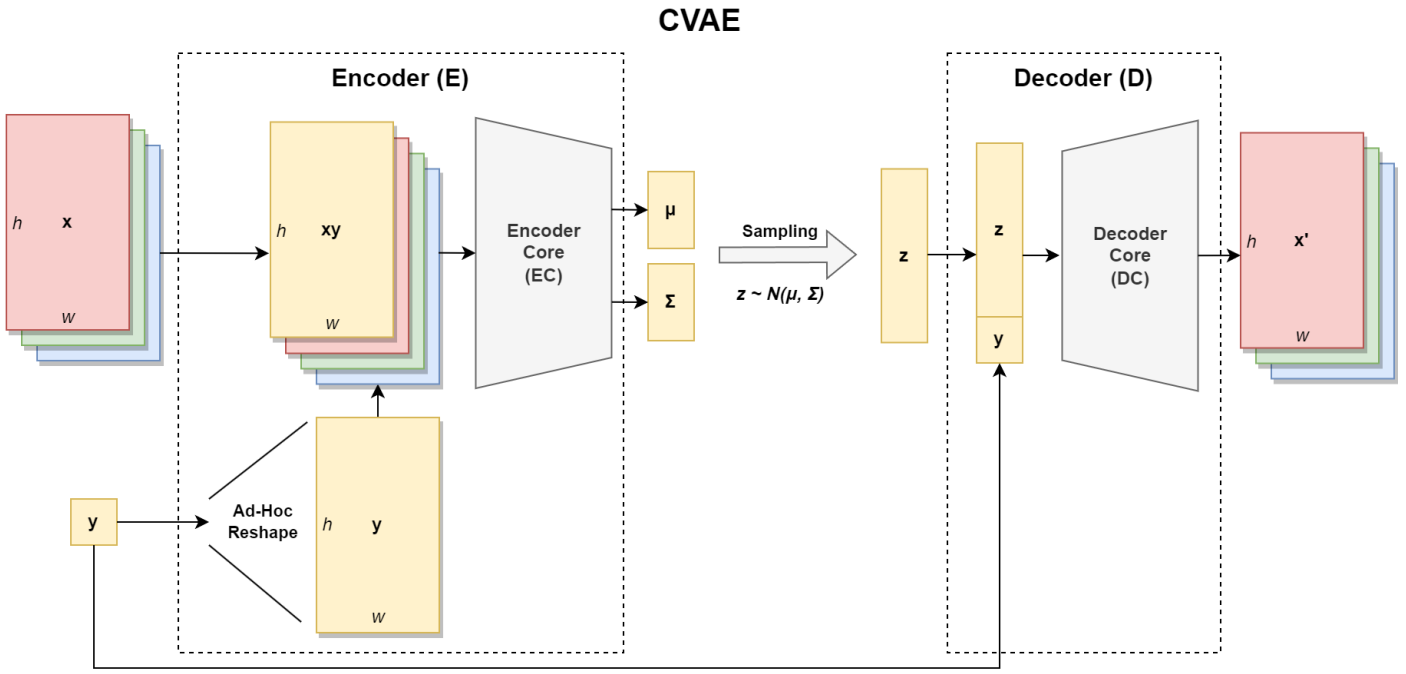
- **CE:** AlexNet with modified outer layers
- **FE & FD:** deep FFNN
- **CD:** reverse of **CE**
- **Features**: ~ 4096 - 9216
- **Weights**: ~ 120M - 320M
- **Example Graph:** link

Wide Variant

- **CE**: deep CNN, no pooling layers, little compression
- **FE & FD:** simple FFNN, only output layers
- **CD:** reverse of **CE**
- **Features**: ~ 65536 - 1.048.576
- **Weights**: ~ 10M - 600M
- **Example Graph:**
  - First version: link
  - Improved version: link

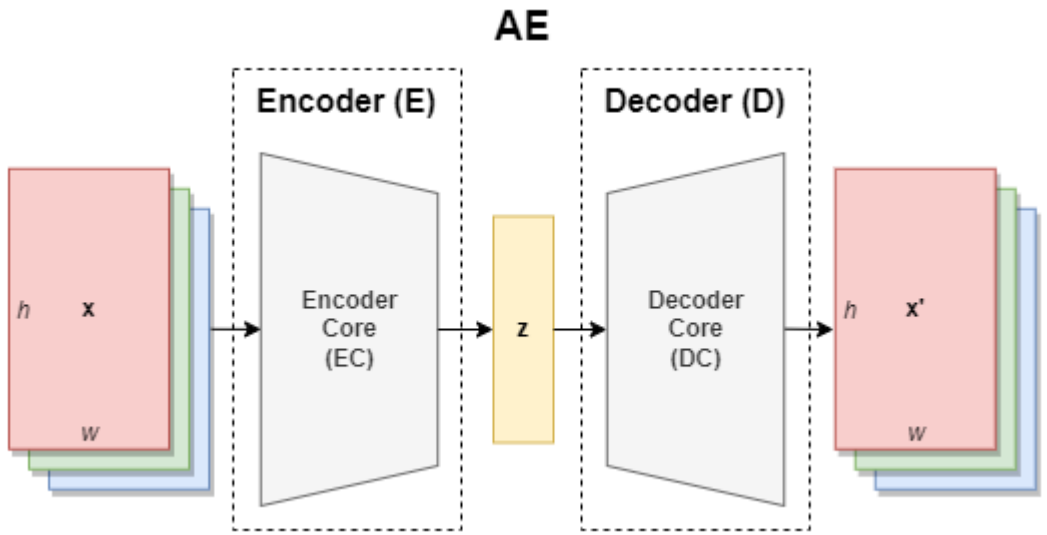
Improved CVAE

In later experiments, the **Wide Variant** of the **CVAE** was simplified, including a single component for the **Encoder** and for the **Decoder**. Moreover, the features extracted from the images were dependent on the **condition**.



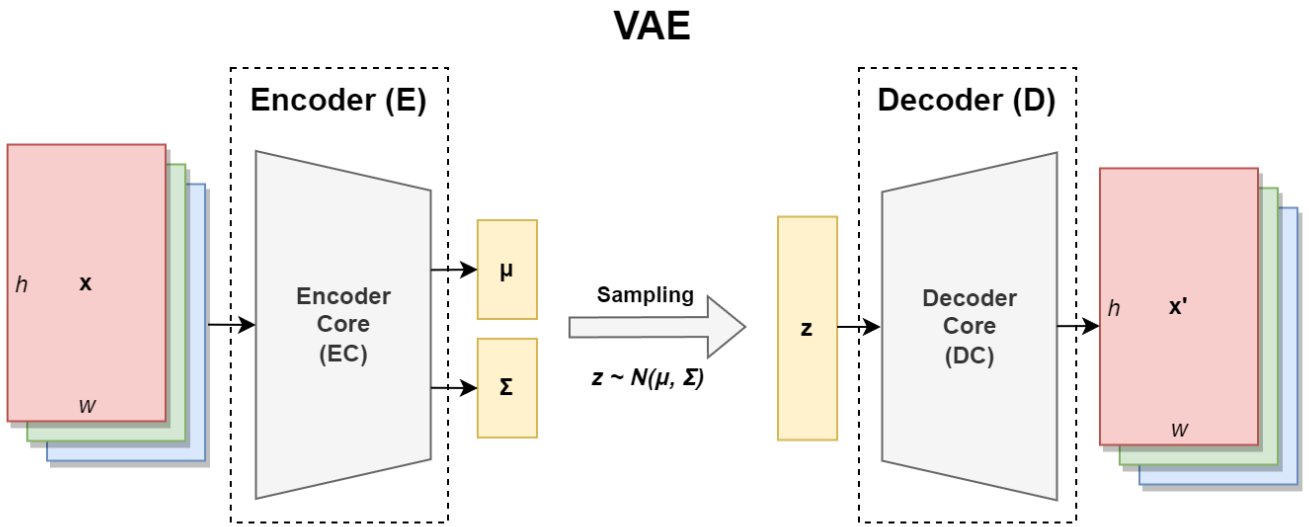
AE

In order to test the reconstruction capabilities of the model, an **Undercomplete AutoEncoder** with a similar architecture has been trained as well.



VAE

For further testing involving also the generative capabilities of the model, a **Variational AutoEncoder** with a similar architecture has been trained as well.



Evaluation

Reconstruction

The first metric used for evaluating the model performance is the **Reconstruction Error**, which measures **how unfaithful the generated images are with respect to the original images**.

$$E_{recon} = MSE(x, x') \cdot |T| = L_2(x, x')$$

This was also **used for training** the model.

Regularization

The second metric used for evaluating the model performance is the **Regularization Error**, which measures **the degree of incompleteness and discontinuity of the latent space.**

$$E_{regul} = \lambda \cdot KL[N(\mu, \sum)||N(0^k, I_k)]$$

This was also **used for training** the model.

Generation Quality

Finally, as the last metric, the **quality** of the generated images was measured as the **accuracy achieved on them by the best landscape classifier for the dataset**.

In particular, the quality of the generated images has been evaluated **for each landscape type separately**.

However, such measure is not complete, as it does not take in consideration the **variance** of the images generated by the model. Moreover, it is **biased** towards the features learnt by the landscape classifier.

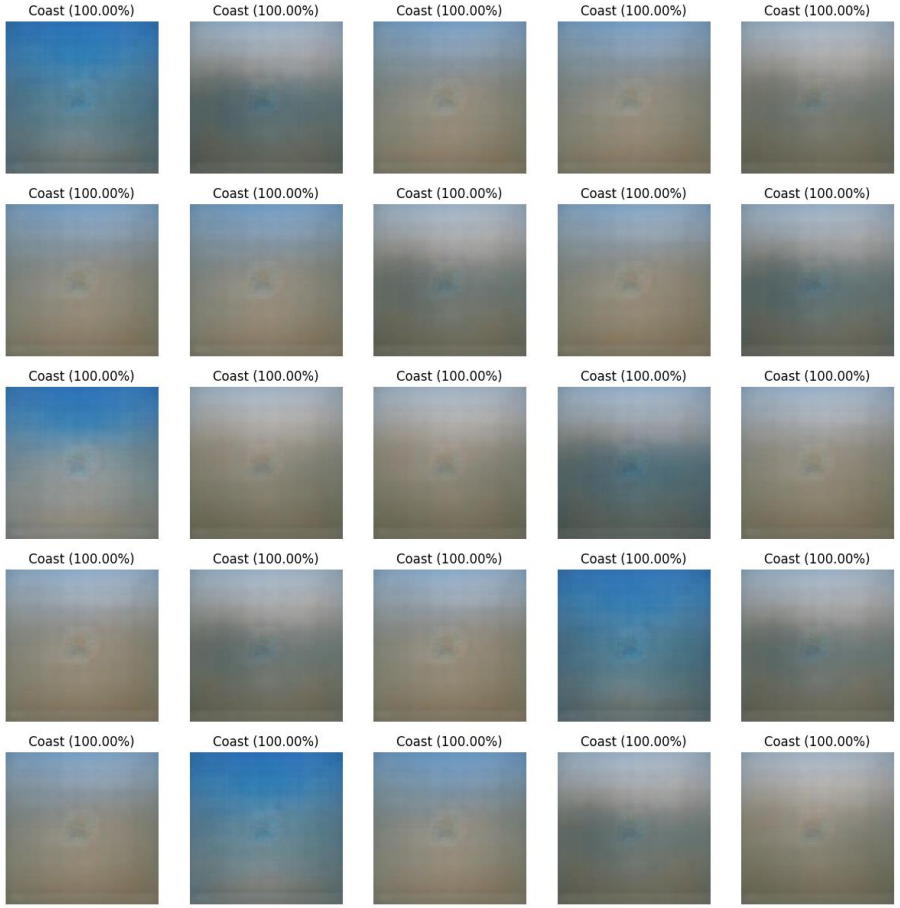
Results

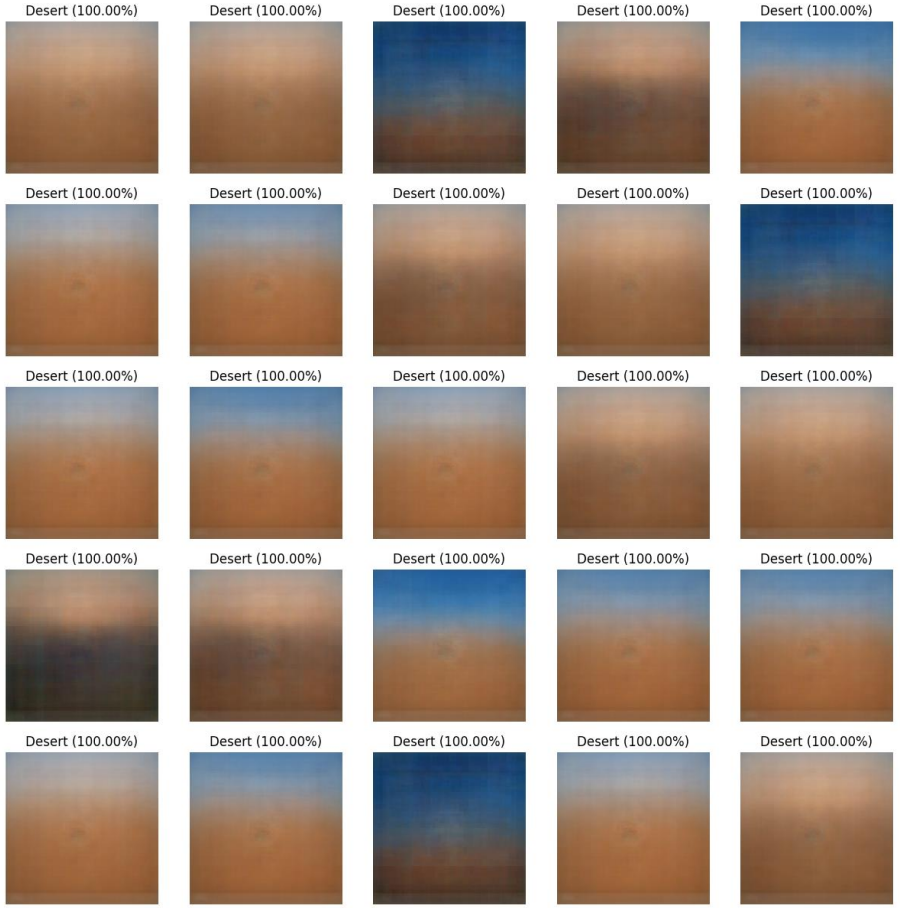
Performances

Variant	E	F	D	W	T	LA	λ	L	E_{recon}	E_{regul}	Q_{avg}
Narrow	100	2048	2	14M	1h	linear	1	**430**	**n/a**	**n/a**	**35%**
AlexNet	100	4096	4096	166M	1.5h	linear	1	**278**	**n/a**	**n/a**	**37%**
Wide	100	1048576	2	13M	2h	tanh	1	**481**	**480**	**1**	**44%**
Wide	100	1048576	10	38M	2h	tanh	1	**355**	**349**	**6**	**39%**
Wide	100	1048576	100	321M	2.5h	tanh	1	**207**	**156**	**51**	**36%**
Wide +	100	1048576	200	636M	4.5h	linear	0.1	**57 (291)**	**31**	**26 (260)**	**28%**
Wide +	100	1048576	200	636M	3.5h	linear	1	**164**	**103**	**61**	**34%**
Wide +*	1000	65536	1024	202M	9.5h	linear	1	**160**	**112**	**48**	**41%**
Wide +*	1000	1048576	64	208M	22h	linear	1	**89**	**42**	**47**	**43%**

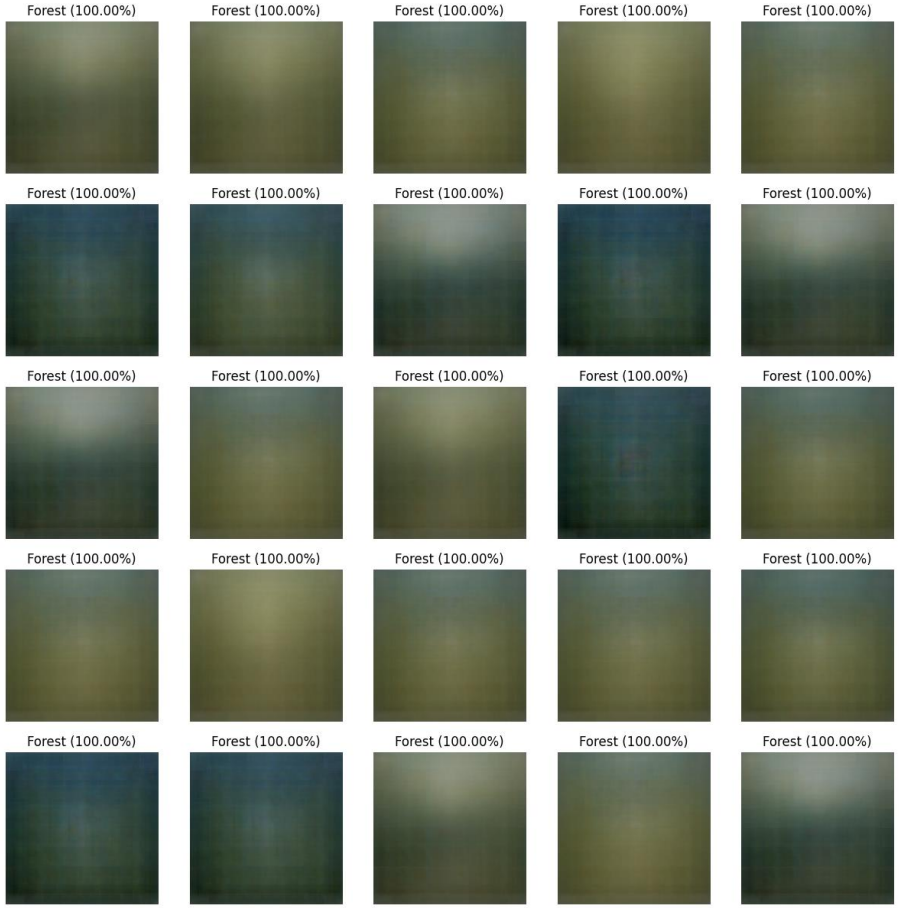
T1. The performances of some of the best configurations tried for each variant of the model.
E: epochs; **F:** features; **D:** code size; **W:** weights; **T:** training time; **LA:** latent activation; **L:** loss;
Q_{avg}: average quality; *: improved wide variant with images resized to 128x128

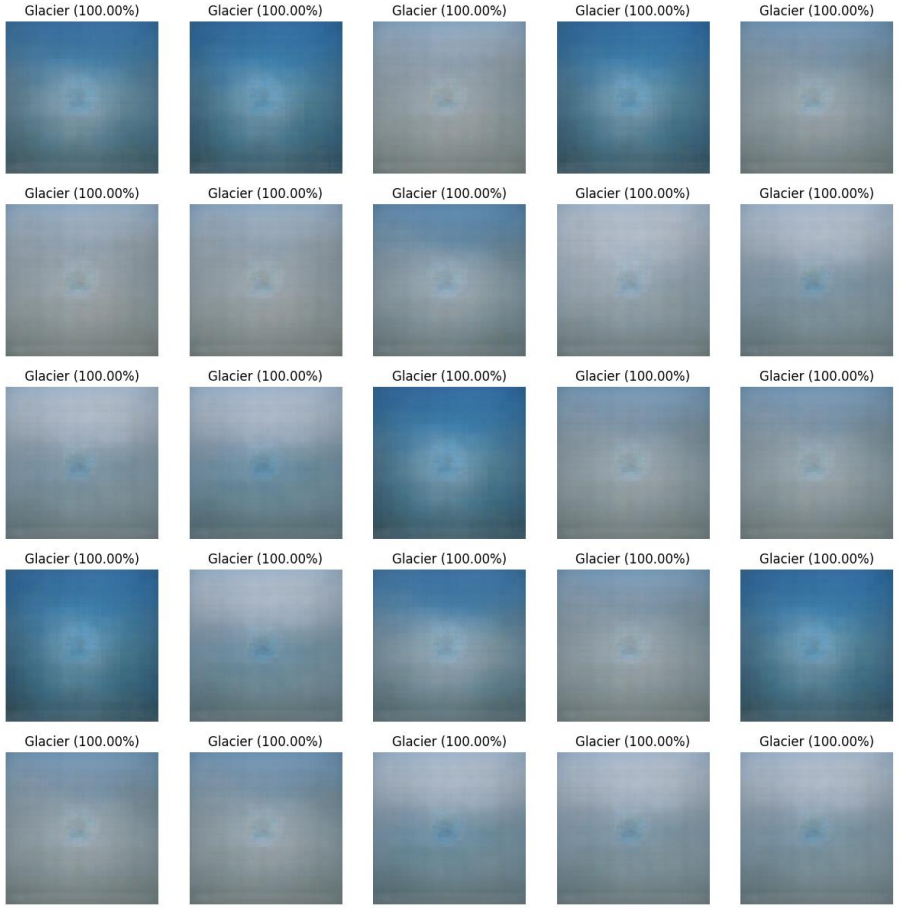
Narrow Variant (1/3)



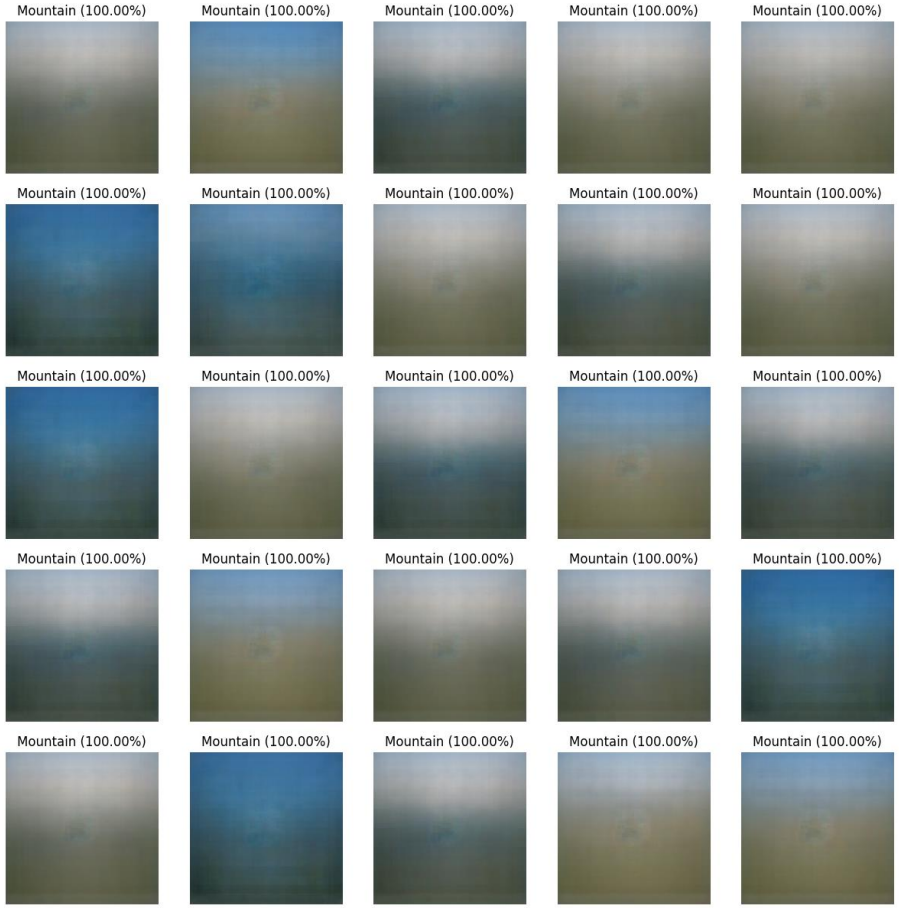


Narrow Variant (2/3)





Narrow Variant (3/3)



- Landscapes are mostly **combinations of colors**
- Usually there is a **terrain** and a **sky**
- It can be either **day** or **night**

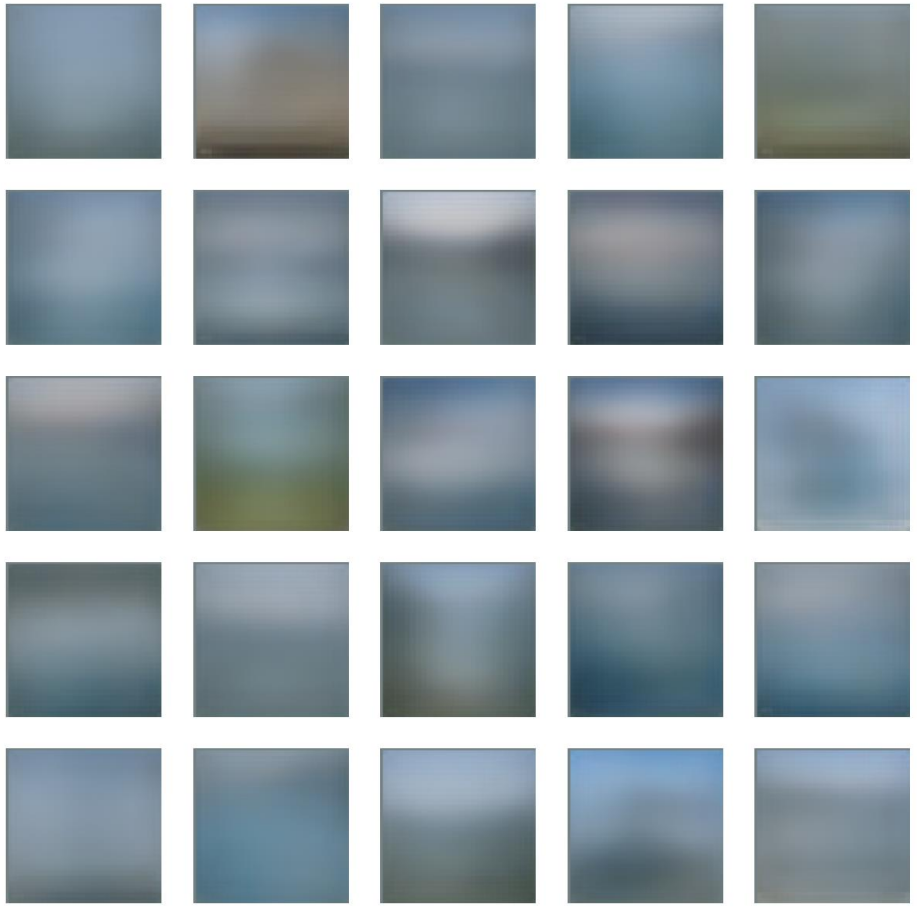
AlexNet Variant (1/3)

coasts

deserts

AlexNet Variant (2/3)

forests



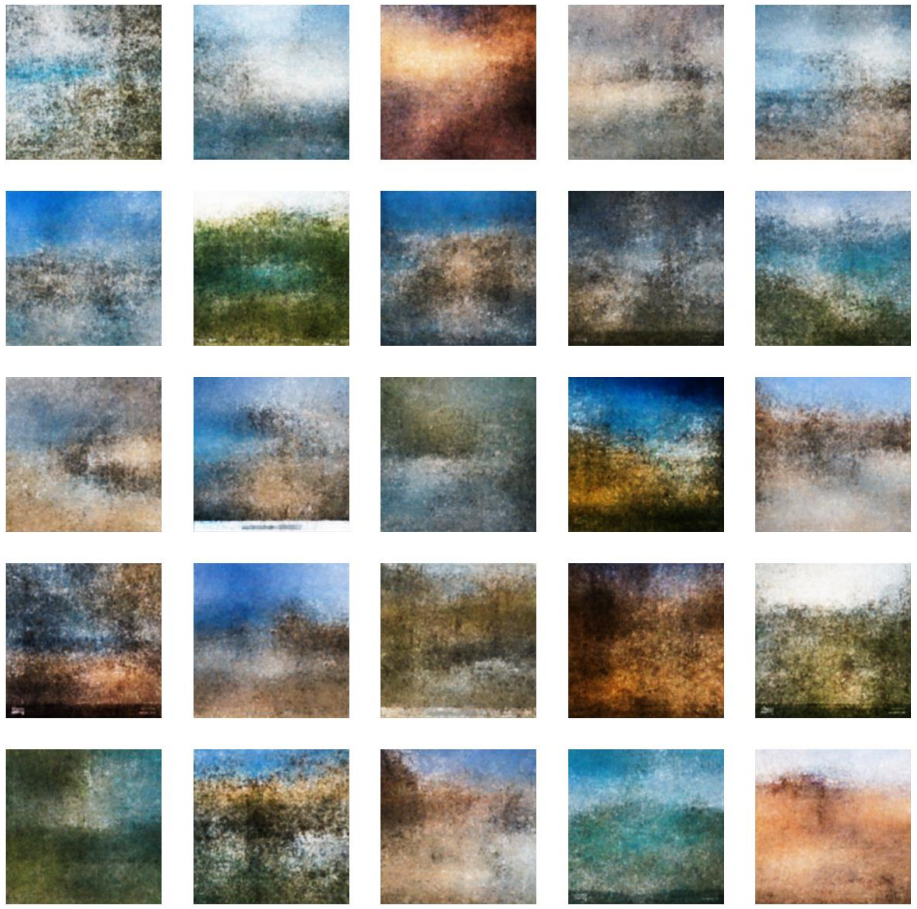
glaciers

AlexNet Variant (3/3)

mountains

- Landscapes are also **combinations of shapes**
- Some pictures seem possibly realistic, but **very out of focus**

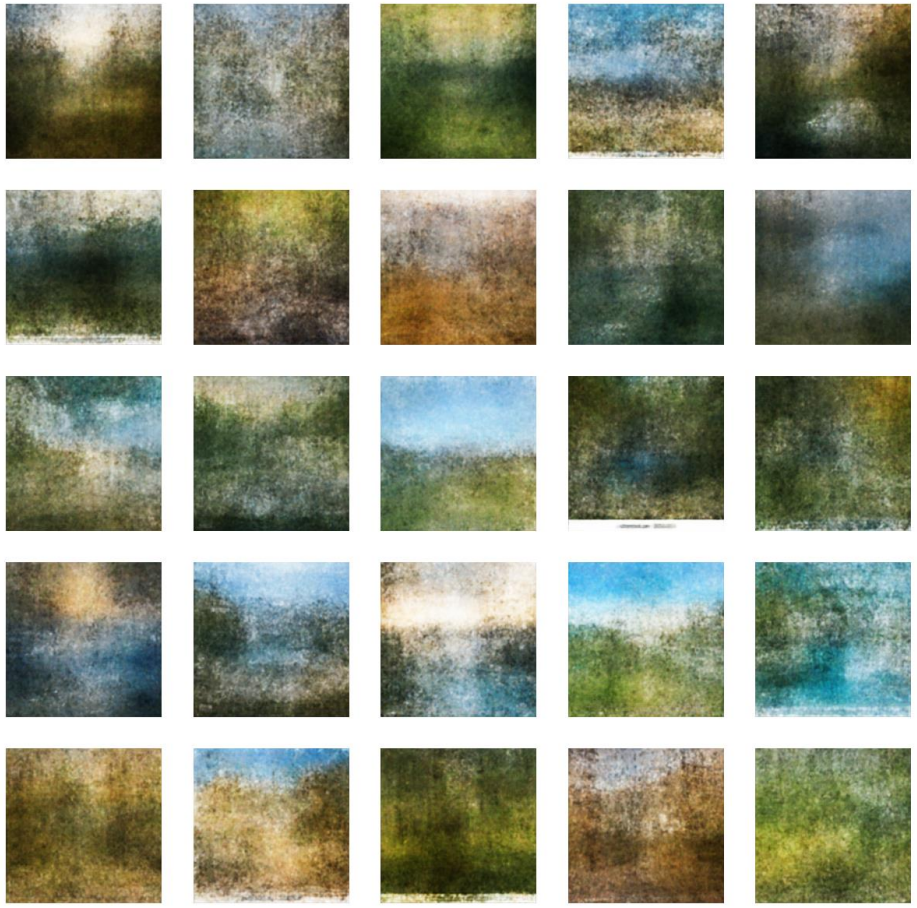
Improved Wide Variant (1/3)



coasts

deserts

Improved Wide Variant (2/3)



forests

glaciers

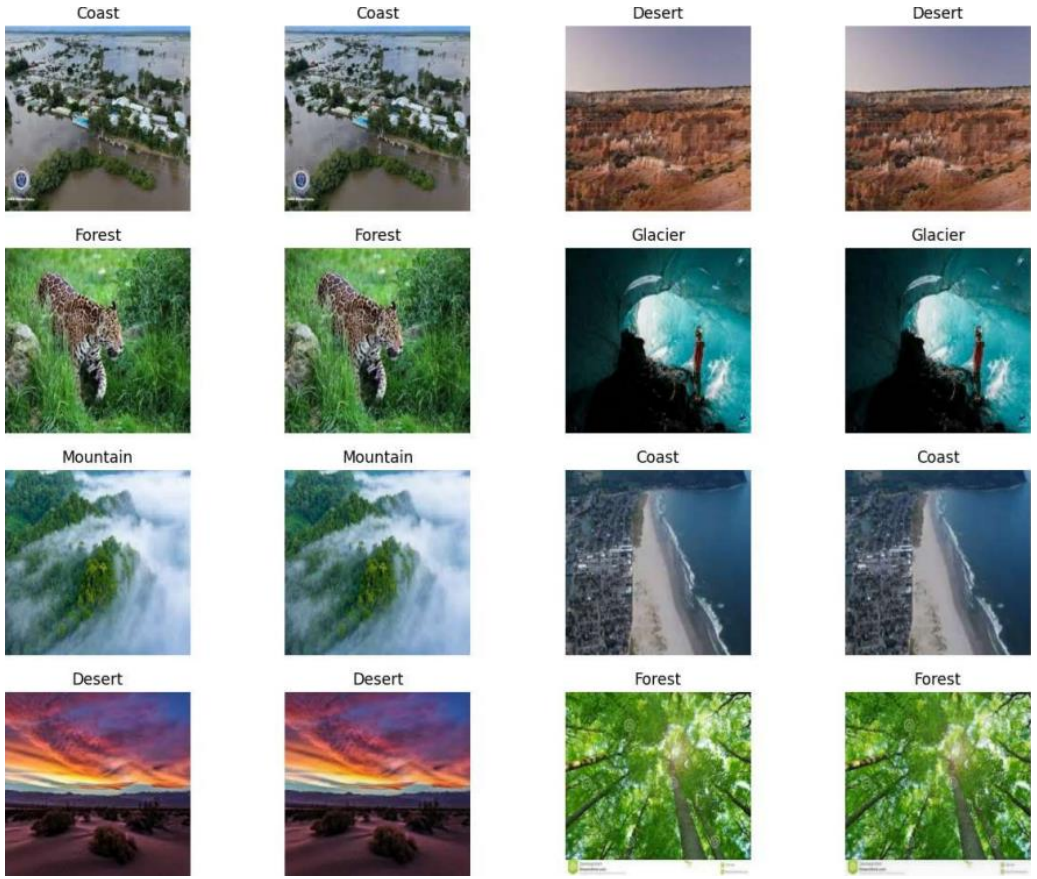
Improved Wide Variant (3/3)

mountains

- The pictures are no longer **out of focus**
- Some pictures seem like **impressionistic** representations of landscapes
- There's still **lack of details** in the images
- More **outliers** in the generated images

AE

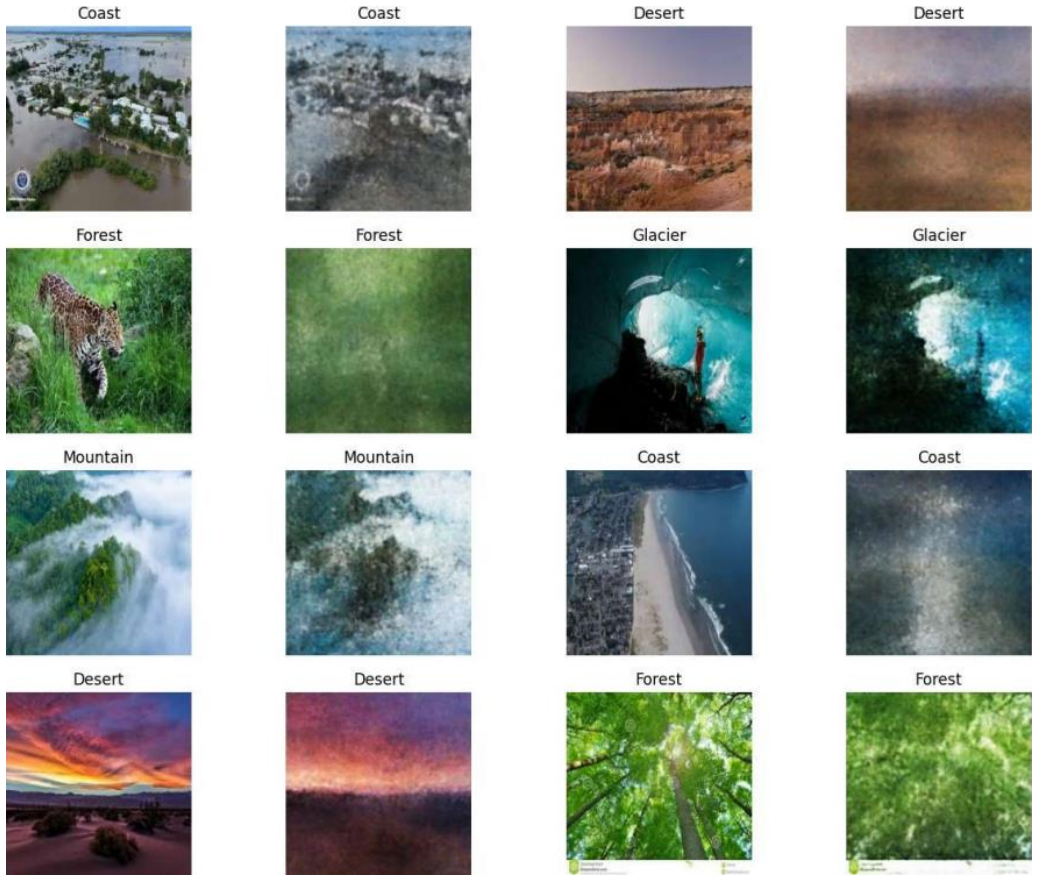
The **AutoEncoder** could reconstruct the original images flawlessly.



real generated real generated

VAE

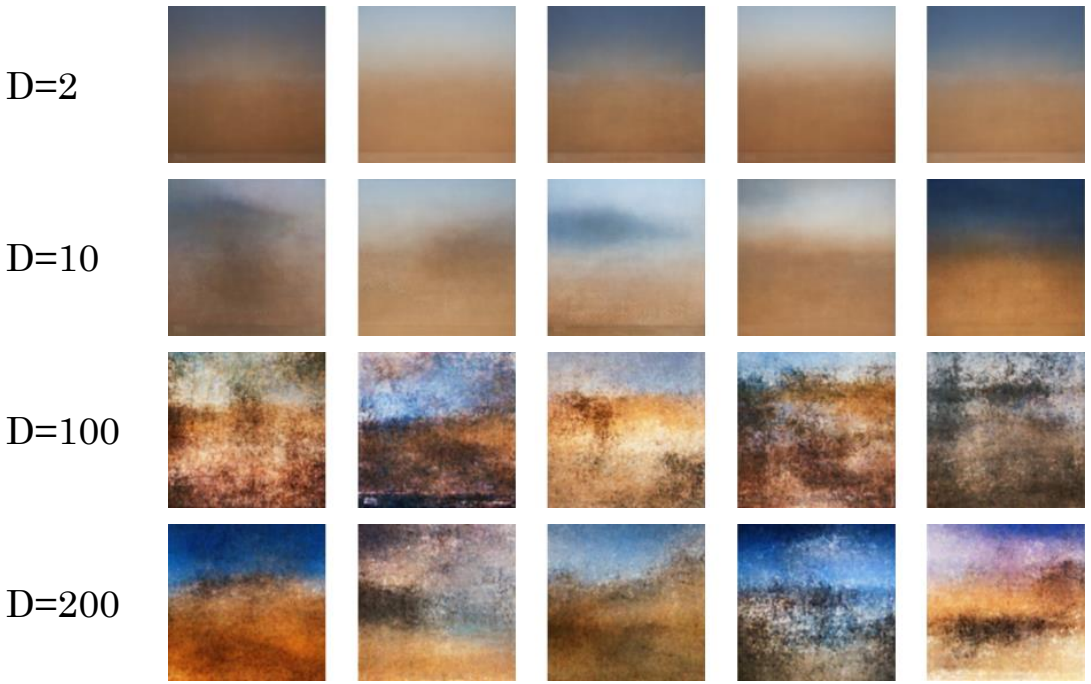
The **VAE** and the **CVAE** could **not** reconstruct the original images flawlessly.



real generated real generated

Conclusions (1/4)

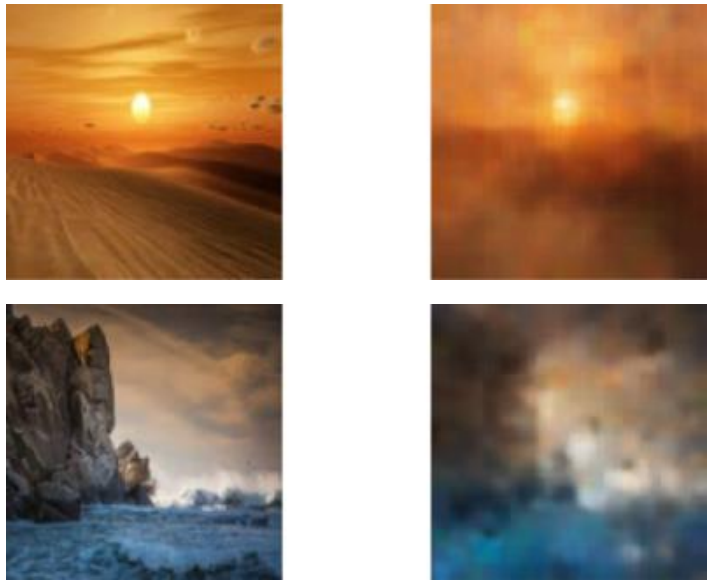
- **Increasing the dimension of the latent space increases the amount of details** in the images, up to a certain limit. However, it also **increases the amount of outliers** in the generated images;



deserts

Conclusions (2/4)

- **Increasing the number of features** extracted from the original images **increases the amount of details** in the generated images, up to a certain limit;



reconstruction
F=65536 D=1024 F·D=67108864

reconstruction
F=1048576 D=64 F·D=67108864

Conclusions (3/4)

- **Decreasing λ increases the quality** of the generated images, however it also **increases the amount of outliers** in the generated images;

high-quality
mountain

outlier
mountain

Conclusions (4/4)

- **Reducing the size of the images increases the reconstruction performances of the model**, but it **doesn't increase the amount of details** in the generated images.

coasts
128x128

Further Explorations

- *The amount of information contained in the condition of the CVAE may be too little*, making the task of generating a landscape much more difficult (e.g. to reduce the impact of mixed landscapes, it may be possible to adopt fuzzy labels…);
- *It may be too difficult to find an explicit distribution that well represents the variety of landscapes in this dataset*. The results of this project should be compared with the result obtained with an implicit generative model (e.g. GAN…);
- *More time could be invested to experiment with more advanced auto-encoding techniques…*

Thank you.

End