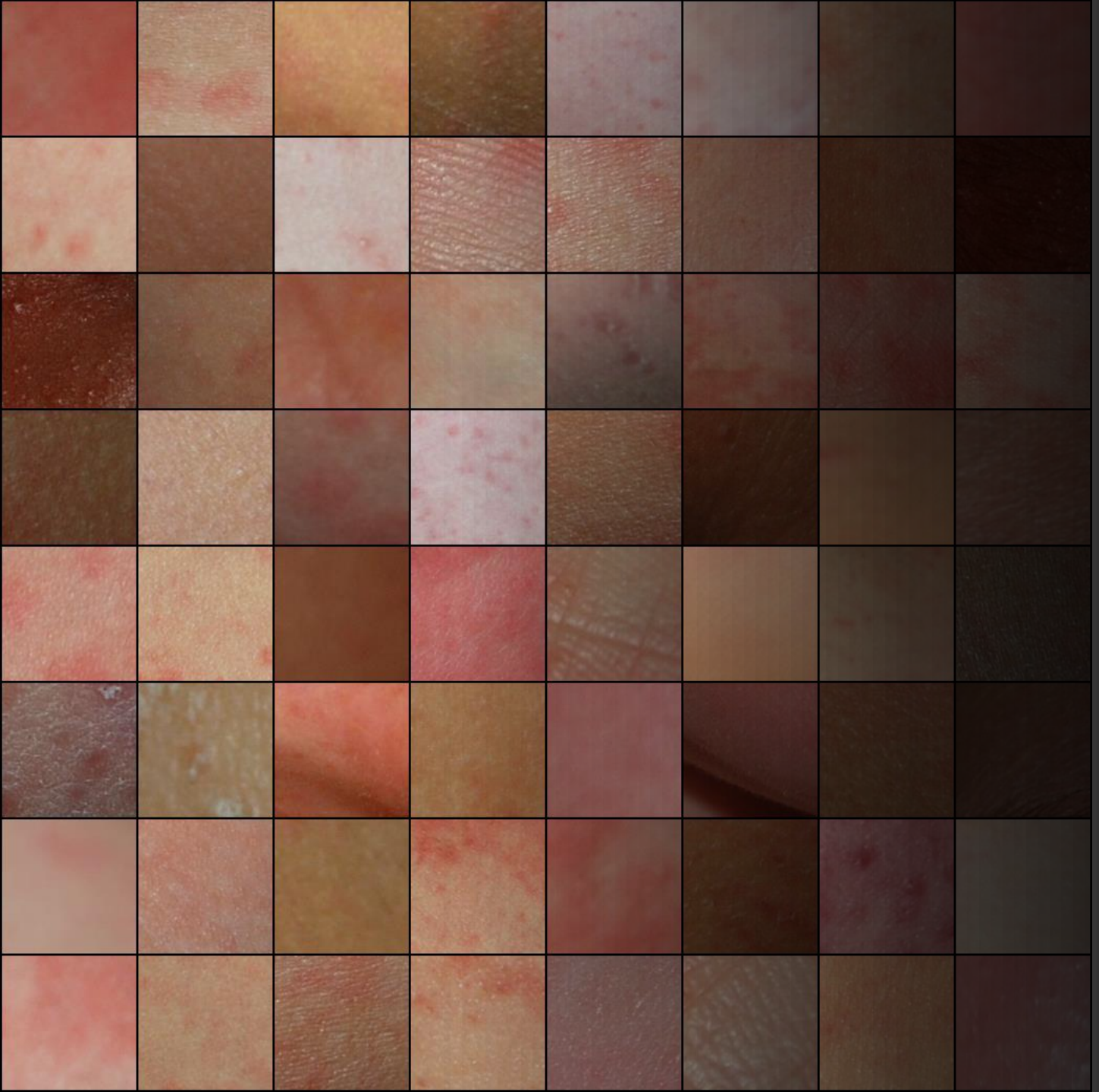


Skin Disease Image Generation

Artificial Intelligence in Industry Exam

A. D'Amico, R. Murgia, M. Moeini Feizabadi

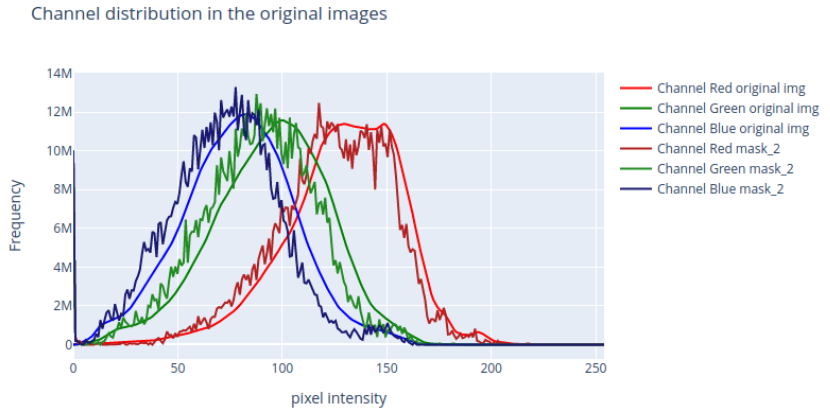
Purpose

- Generate synthetic images of skin illnesses.
- The generated images are meant to be used for **augmentation.**
- The augmented dataset (original images + synthetic images) should be big enough to train a skill illness detector.

Dataset

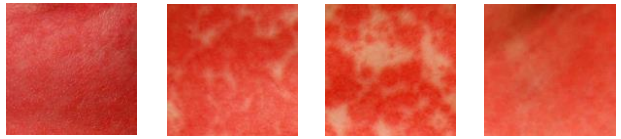
- Acquired from italian hospitals, images captured "in the wild" (not by professional instruments or photographers, but in an "amateur" fashion).
- 9 kinds of skin illnesses, we take just 1: **esantema maculo-papuloso**
  - crops are extracted from bigger original images of varying size
  - each crop is of dimension **256x256**
- A **binary mask** is provided to isolate ill skin area (can be useful for both downstream and upstream tasks). However, by cropping, mask utility for downstream tasks is partially lost.

Dataset Analysis

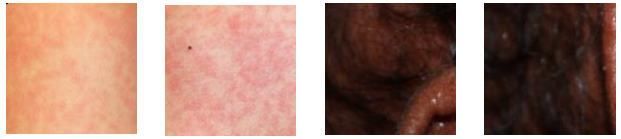


crops

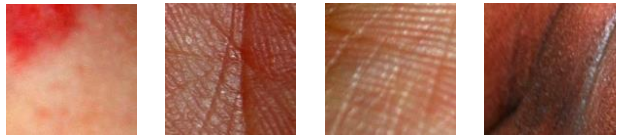
Colors are distributed on 3 gaussians, but normalizing them to have zero mean would make it difficult to reverse the transformation for the new generated images



Some images with higher saturation were found (using HSV colorspace)



As well as some with low and high luminosity (using LAB colorspace)



And some images are difficult to learn (we trained a convolutional autoencoder)

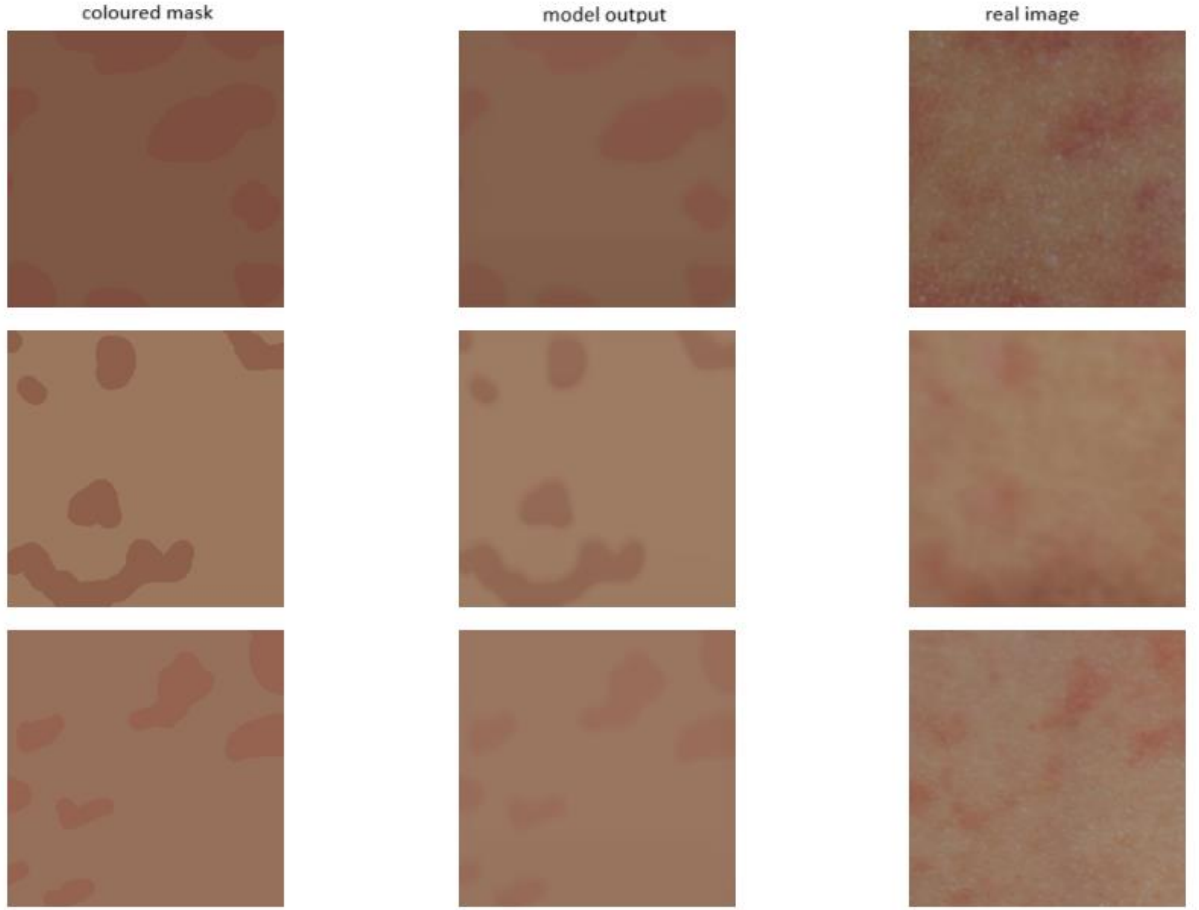
Models

- Previously implemented in this project:
  - DermGAN
  - DermDiff

  } Using coloured input masks from train at test time
- Models in this work:
  - DCGAN
  - CVAE
  - GLIDE

DermGAN

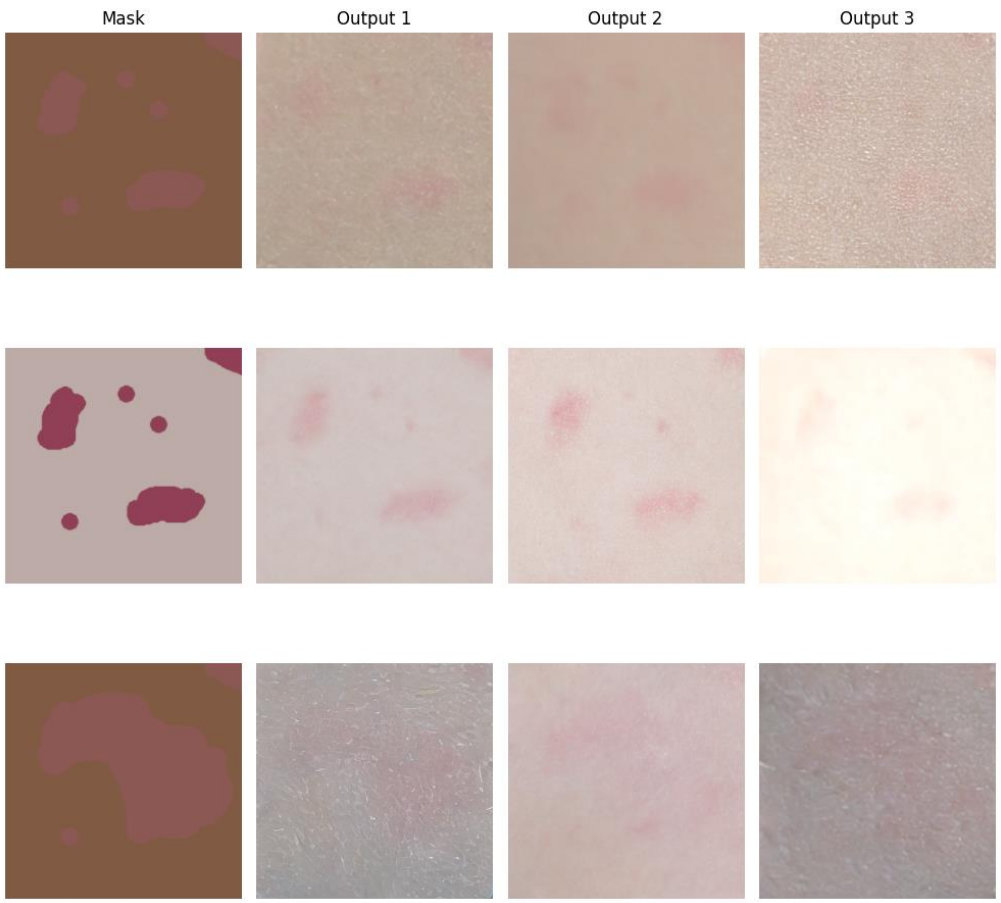
- a conditional Pix2Pix-based GAN, reimplemented by scratch by our colleagues following the original paper by Google researchers.
- Uses colored masks as input.
- The output is not satisfying: the model ends up in a simple gaussian blur effect, not able to produce realistic images.



FID =311

DermDiff

- A diffusion model presented by our course colleagues.
- Uses colored masks as input.
- The output is much better than DermGAN.
- But the results are not fully convincing, not realistic enough.



FID = 74

Limitations of previous approaches

- Usage of **masks**:
  - Extracting binary masks is *useful* for downstream tasks (i.e. training a detector with higher localization capabilities), but is not a crucial requirement to do so.
  - Binary masks in our dataset are not always accurate.
  - **Using masks is not a flexible/generalizable approach** (to increase the dataset we would need to have b/w masks coupled to each sample).
  - Extracting **color masks increases the preprocessing time**
  - If the models need the mask to generate new samples from masks (like DermGAN and DermDiff), the output may not be realistic.
    Our task is to provide an ***augmentation*** method, not to create brand new images.

Original
256x256 crop

B/W mask
256x256 crop

Color mask
256x256 crop

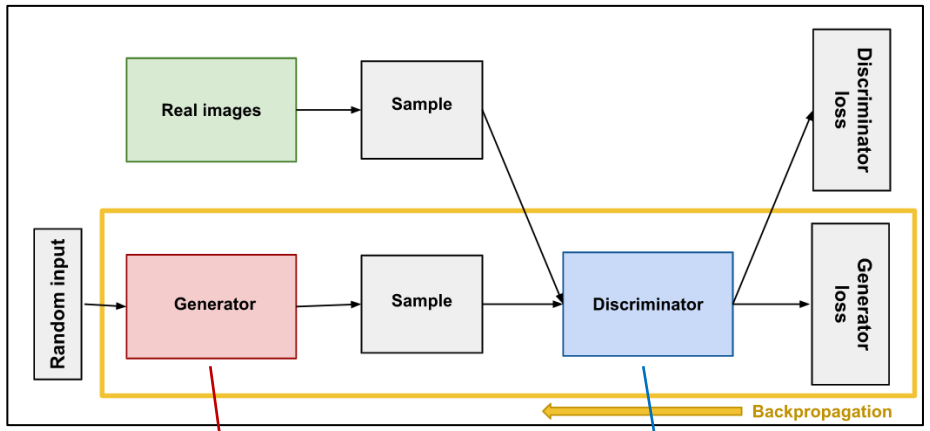
Our baselines

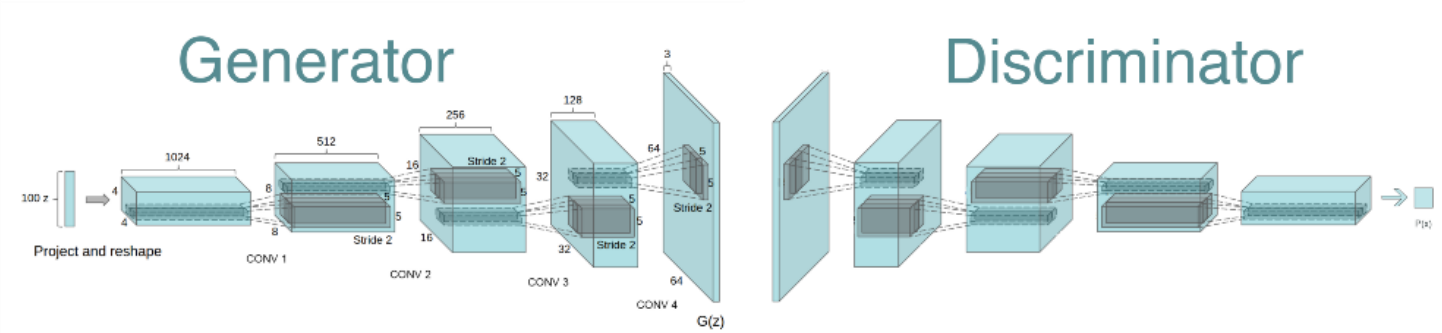
We proceed building two model baselines (DCGAN and CVAE), following the **Occam Razor principle**:

- Simpler models can be used with low resources (both at train and at test time).
- Simpler models are easier to tune and debug (in theory!).
- We can have a better understanding of the limitations that characterize our task.
- We **avoid using masks**.

DCGAN

- ***Generator*** plays against Discriminator. Generator tries to create undistinguishable samples from real images.
- ***Discriminator*** tries to recognize real from fake images.



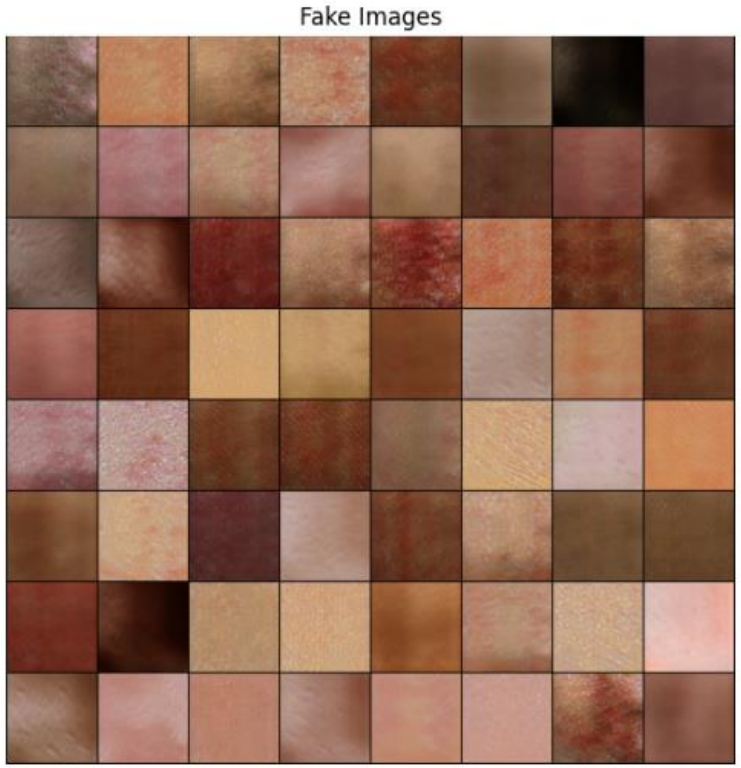


- Architecture is fully convolutional.
- Resize-convolutions used instead of Transpose Convolutions (to avoid checkerboard patterns).
- One sided label smoothing for the descriminator (1 →0.9), to avoid descriminator overconfidence.

```
maximize log(D(G(z)))
```

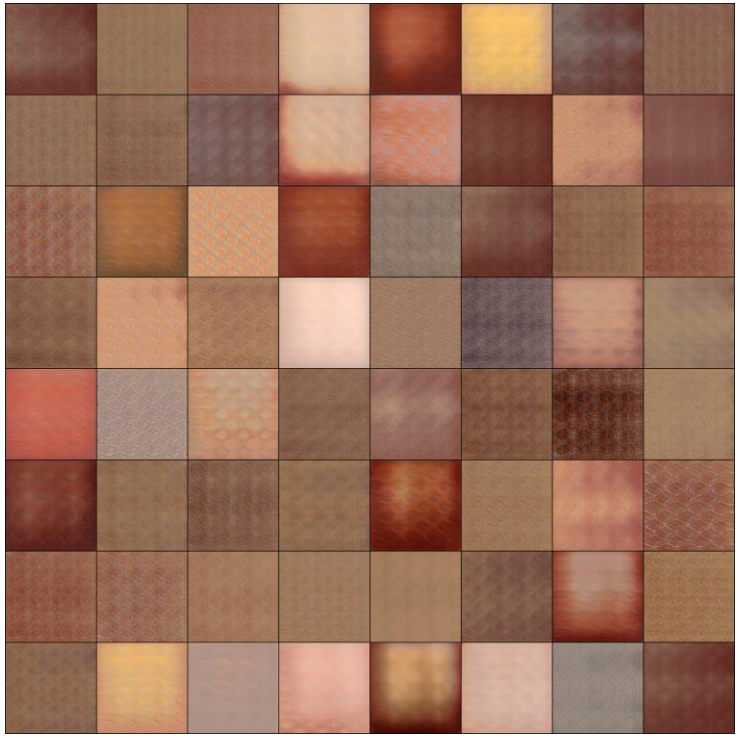
```
maximize log(D(x)) + log(1 - D(G(z)))
```

DCGAN



128x128 patches

0.08 sec/image (Nvidia T4)
FID = 98



256x256 patches

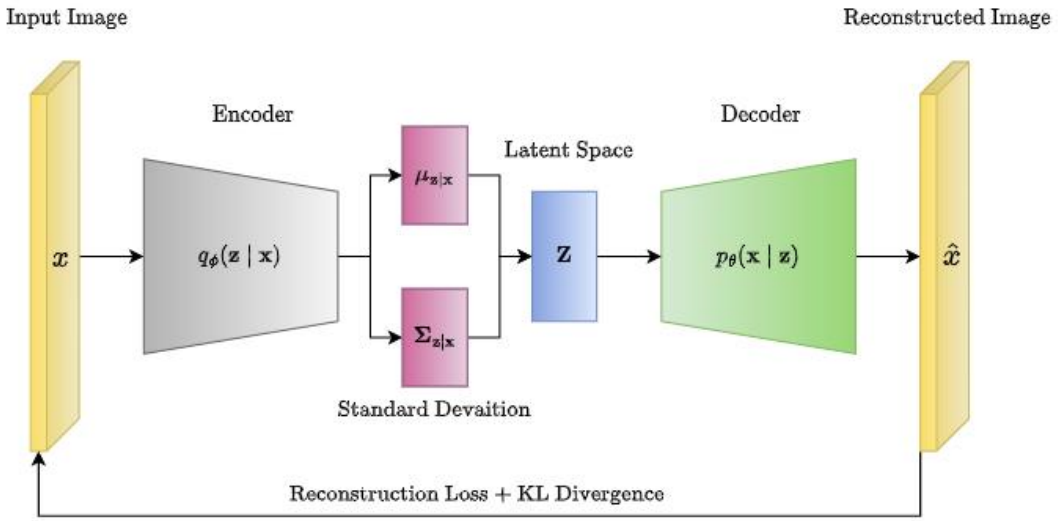
0.10 sec/image (Nvidia T4)
FID = 358

Problems:

- Bad result for 256x256 images.
- Unstable training, very sensitive to hyperparams.
- higher resource need for larger resolutions (1xT4 may be not enough).
- We're really generating images from scratch (the model had to learn the latent space)

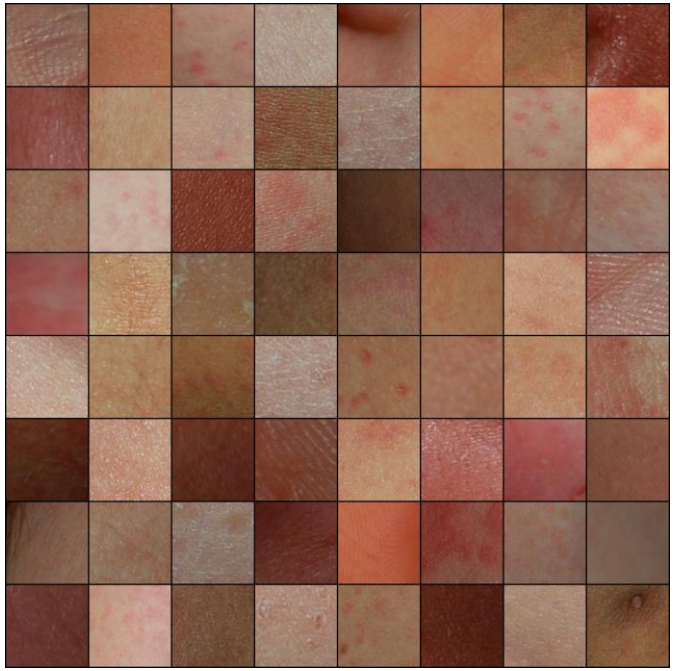
CVAE

- Class of neural networks used to learn probabilistic representations of complex data in a continuous latent space.
- Composed by two main parts **Encoder** and **Decoder**.
- Encoder Composed of Convolutional Blocks and Dense Layers.
- Decoder composed of Dense Layers and Resize-Convolutional Blocks.
- Two additional Dense Layers (estimate mean and std)
- The Loss function is composed by two terms the Reconstruction term and Kullback-Leibler Divergence.
- Reconstruction terms quantifies the discrepancy between the predicted and actual pixels using Mean Squared Error.
- The Kullback-Leibler Divergence between the learned distribution in the latent space and a multivariate standard Gaussian. It ensures that the latent space maintains a structured and organized form.

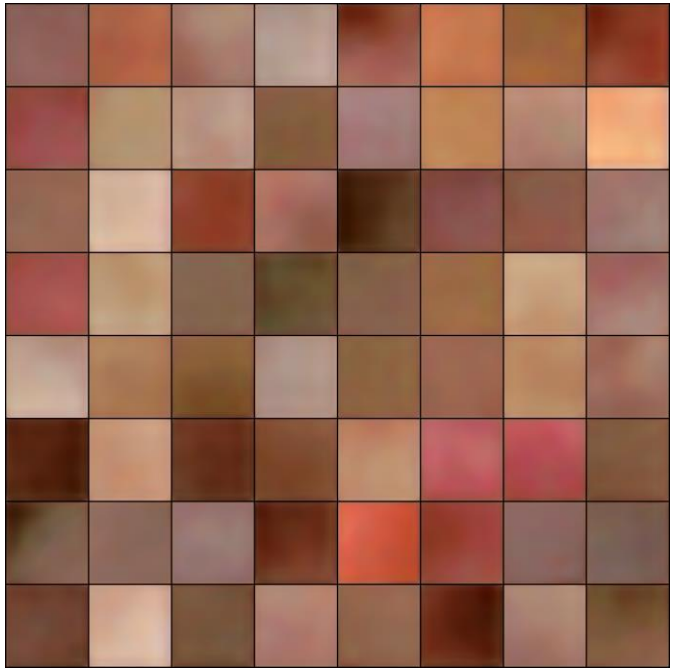


$$L(\theta, \phi, x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \beta \cdot D_{\mathrm{KL}}(q_\phi(z|x)||p(z))$$

CVAE

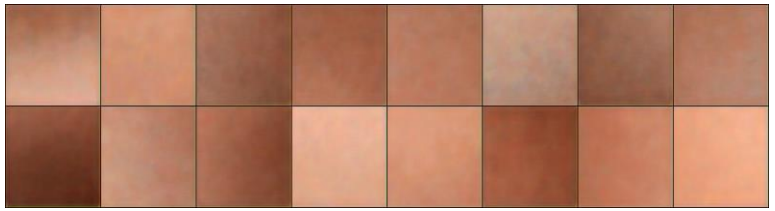


128x128 patches
(train, original)



128x128 patches
(train, reconstructed)

256x256 patches
(train, original)



256x256 patches
(test, reconstructed)

0.09 sec/image (Nvidia 4060)

Limitations of Variational Autoencoders

- **Hyperparameter Sensitivity:**

A grid search methodology has been employed to evaluate various structures and Hyperparameters.

- **Blurry Outputs:**

Because of the inherent characteristics of the model and the loss function employed in its training (as discussed in Towards a Deeper Understanding of Variational Autoencoding Models and Generating Images with Perceptual Similarity Metrics based on Deep Networks).

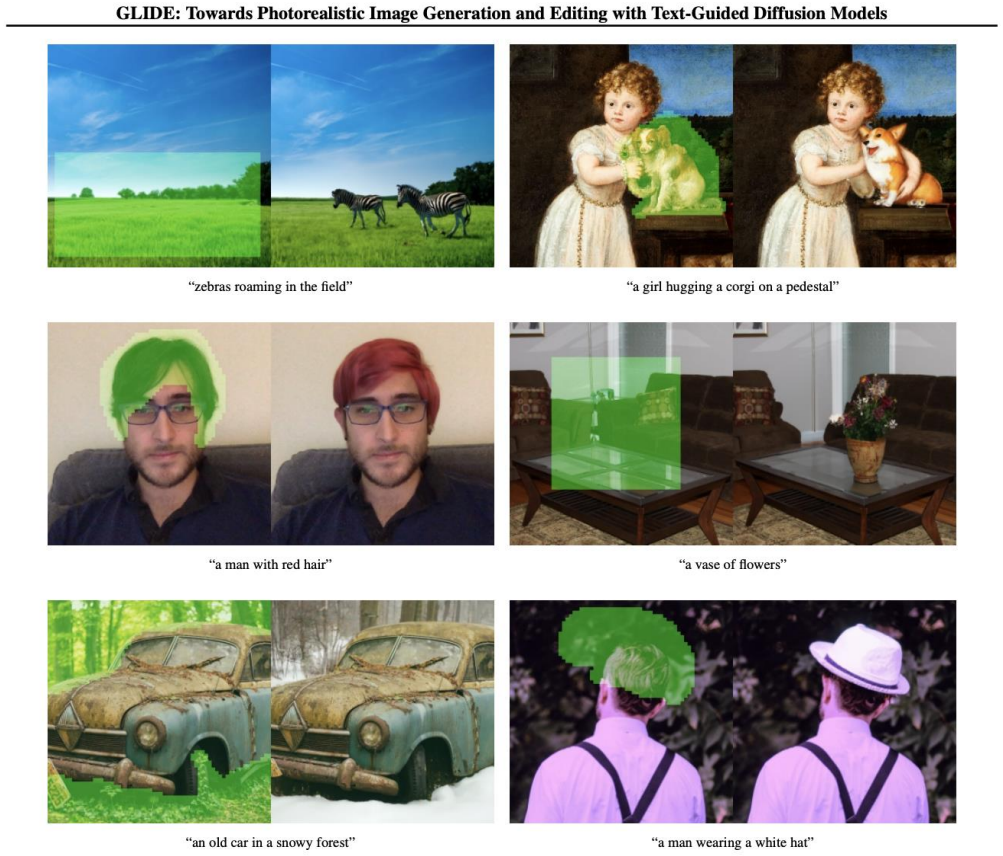
  - We tried adding GAN discriminator as loss to mitigate this blurry effect (no improvement at all).

- **Generating Images from scratch**

We are really generating images from scratch (the model had to learn the latent space).

GLIDE

GLIDE (Guided Language to Image Diffusion for Generation and Editing) is a text-guided diffusion model.
The model is provided with editing capabilities and it was the first model to use transformer-encoded text to generate specific images.
Ironically for our task, we don't use text guidance.
GLIDE is finetuned to preform image inpainting, give it good zero shot capabilities for editing.



ITERATIVE INPAINT CON GLIDE

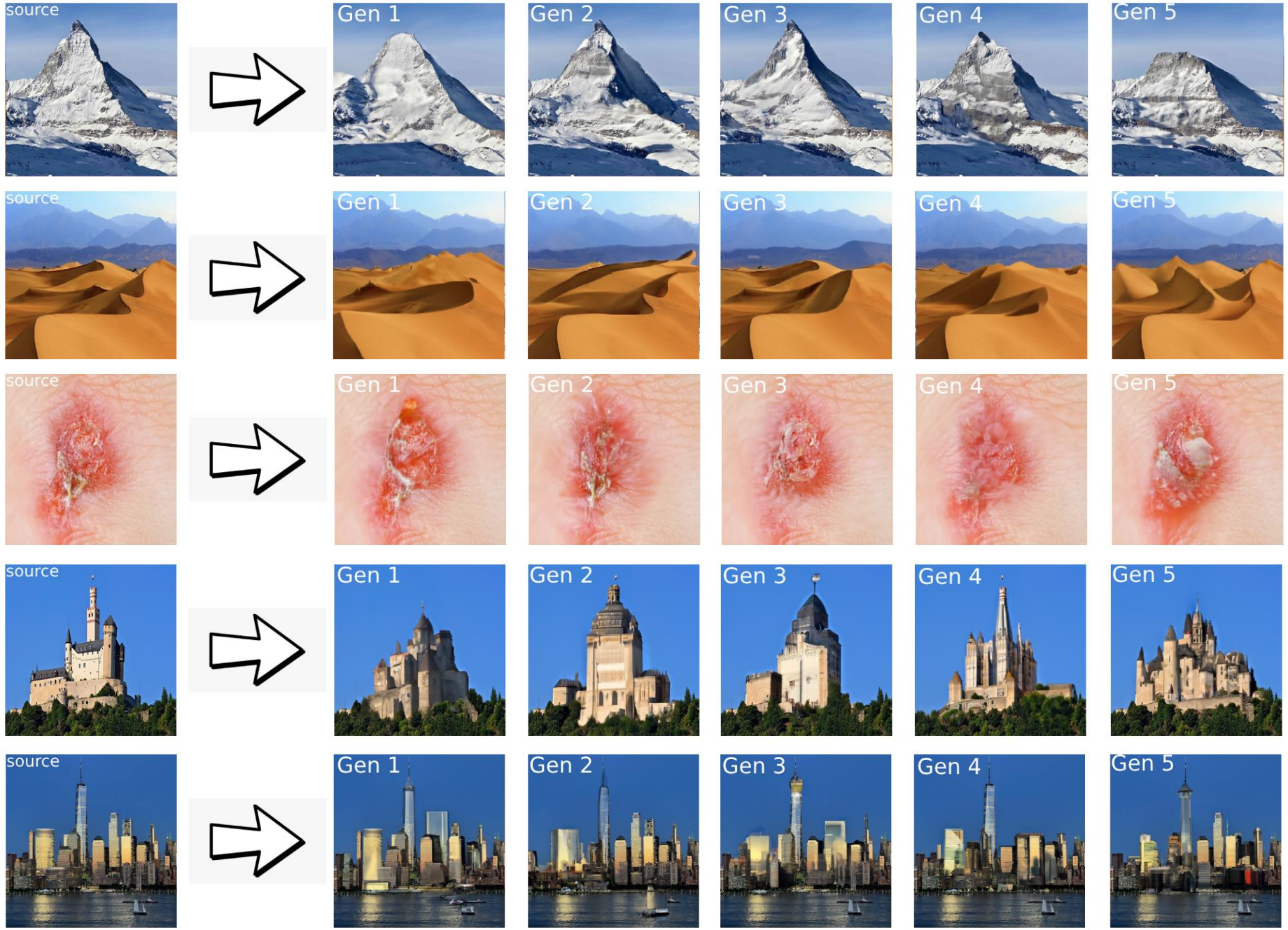
We propose **Iterative Inpaint**: Where in each step a small portion of the image is masked off and inpainted, over the course of 9-12 steps a brand-new image of the same content is created.

GLIDE NOTES

- Inpainting uses the rest of the image as conditional prior, therefore making the box too big we may lose context.
- Making the box too small cause practically no meaningful change but rather blurring.
- The number of steps should also be finetuned to not cause boxes to overlap so often. This can create a sort of xerox effect only causing more blurring.
- Dynamically changing the box size and sample position to avoid overlapping should improve results.

GLIDE ALGORITHM TERMS

- ***Runs*** take an original source image and create x many different outputs.
- ***Steps*** are the number of times we mask and inpaint an image iteratively.
- **100 diffusion steps** happens in one ***Step*** of this algorithm.
- Then its uses fast27 which is the fastest upsampling algorithm.


source
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
source
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
source
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
source
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5
source
Gen 1
Gen 2
Gen 3
Gen 4
Gen 5

How Does AI Describe These Images?

Matterhorn With Snow On It

A Mountain With Snow On It

A Mountain With Snow On It

A Mountain With Snow On It

A Mountain With Snow On It

A Mountain With Snow On It

How Does AI Describe These Images?

A Sand Dunes In The Desert

A Sand Dunes In The Desert

A Sand Dunes In The Desert

A Sand Dunes In The Desert

A Sand Dunes In The Desert

A Sand Dunes In The Desert

How Does AI Describe These Images?

A Castle on a Hill

A Castle on a Hill

A Castle on a Hill with Tibidabo in the Background

A Building on a Hill

A Building with a Dome on Top

A Castle on Top of a Hill

GLIDE Mask Maintenance

- We think it's impossible to maintain the hand-labeled segmentation masks for image-to-image translation while doing inpainting. Inpainting has no idea of segmentation and if it was possible to automatically segment the images, they would have done that.
- However, we are certain about where the image has been edited which is valuable information if a new mask was to be created.
- We don’t get to choose if inpainting removes or adds healthy skin.

Quirks

- Is it necessary to mask 100% of the image, considering the total area covered eventually gets masked?
- Excessive repetition in masking the same pixel too often leads to blurring.
- Run time is an important factor; at this stage, the algorithm is too slow.
- The size and number of boxes are the two biggest parameters.
- All global details remain the same, such as content, lighting, seasons, and hard borders (horizons). Don't expect these to change.

Different Types of Masks

We tried changing the masks from normal squares to discrete Gaussians, this is because inpaint masks have to be binary. The idea being that it will have smoother inpaints. However, not much changed, the masks are 256x256 in resolution and when shrunk down to 64x64, they practically form squares.

Some important factors to think about:

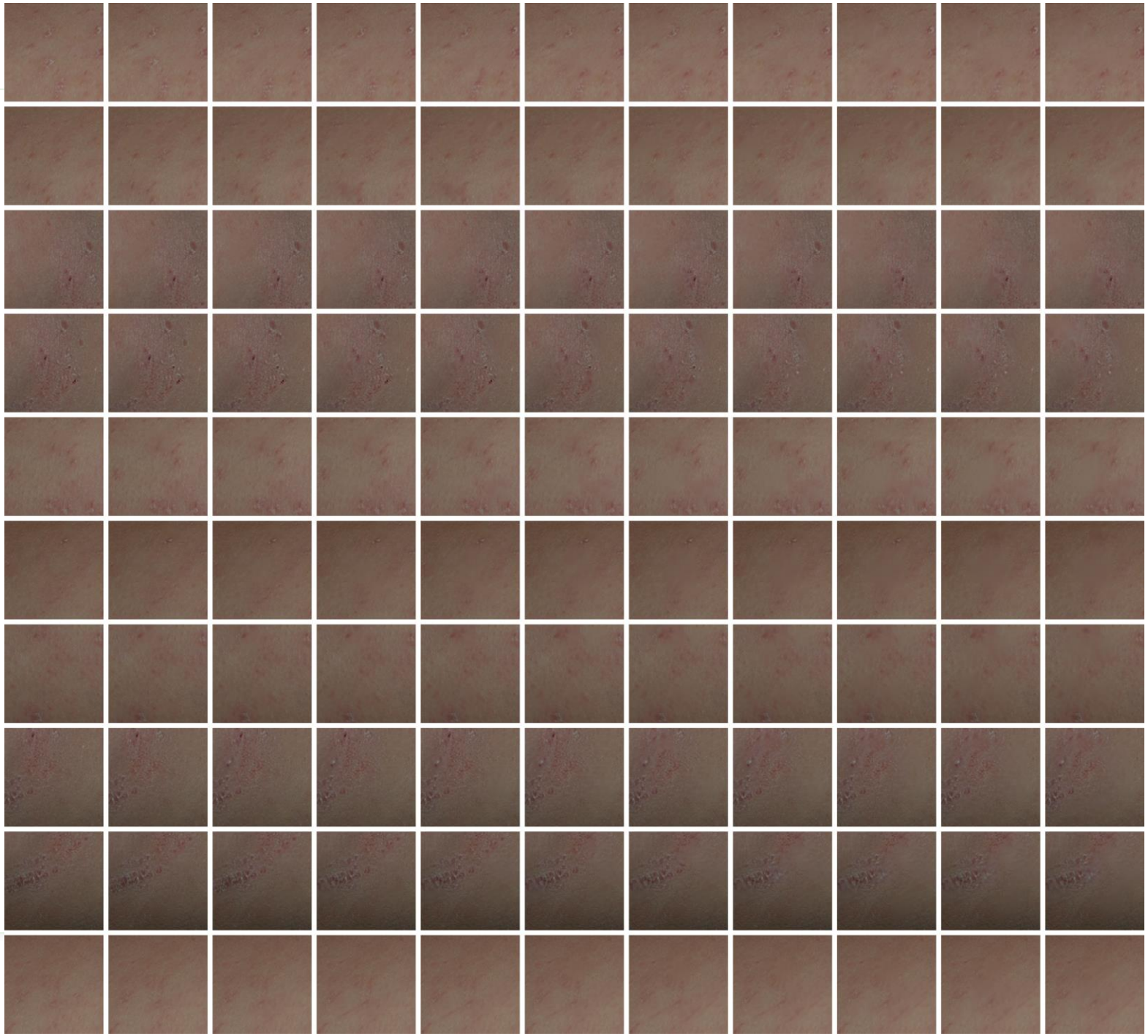
- Total area covered that eventually gets masked, it is not necessary to mask 100% of the image.
- How often is the same pixel masked repeatedly, too often leads to blurring.
- Run time is an important factor at this stage the algorithm is too slow.
- The size and number of boxes are the two biggest parameters.
- All global details remain the same, such as content, lighting, seasons, hard borders (horizons). Don't expect these to change.

Practicality

- GLIDE was mainly used because it's the simplest and cheapest to run in colab. Other options such as Stable Diffusion XL would lead to better results with less blurring.
- On an A100, a step takes ~8 seconds, and with 10 steps per image, it takes 1:20 minutes per run (new image).
- To generate synthetic images for a whole dataset it would require a cluster of A100, if would be wise to focus only on underrepresented classes.
- With ~12000 images it would take ~11 days generate a new synthetic image per each original image.

Results

- The first column is the original image, the other ones are generated in sequence by adding diffusion steps (left to right)
- We got a Fréchet Inception Distance (FID) score of 154, which is considered bad. However, we feel that this does not represent the capabilities of the generation.
- Human evaluation is our primary metric and is also used in many of the modern diffusion papers.



Results

Different runs starting from the same image

Original crop

GLIDE

Run 0 – 5 steps

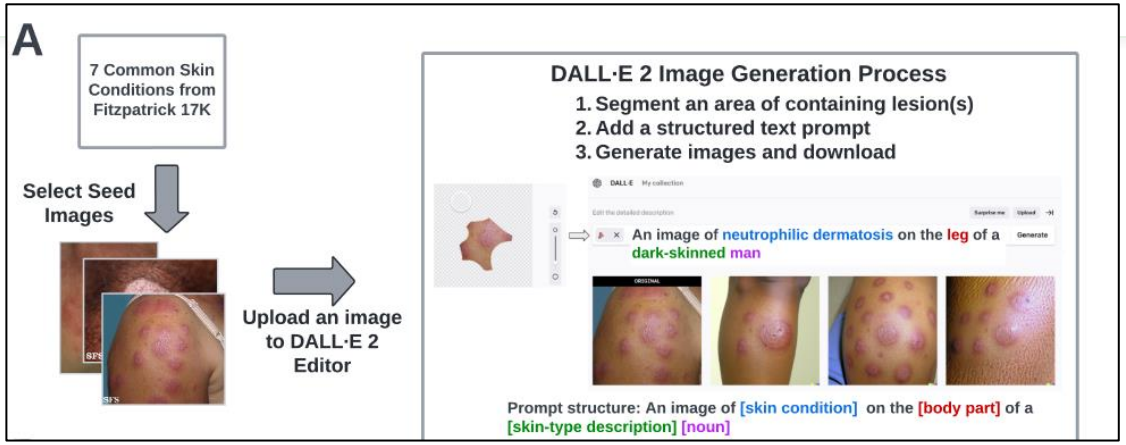
Run 2 – 5 steps

Run 1 – 5 steps

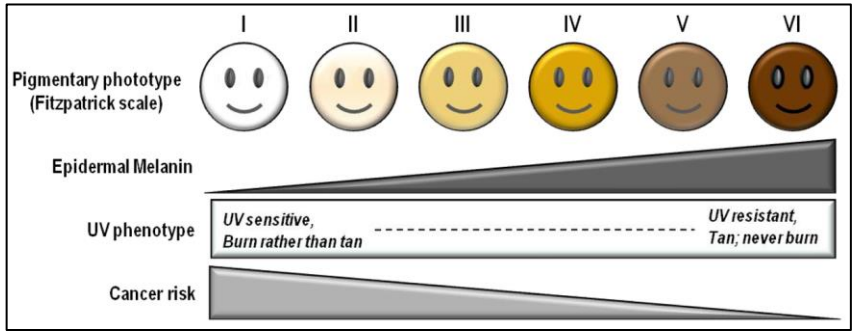
Run 3 – 5 steps

Limitations and future steps

- **Augmentation for other skin groups other than white** (which now represent the majority if not the totality)
  - It worked with Dall-E 2 (with training off to avoid data leaks)
  - Cost evaluation TBD:
    - Using API: 0.016 for 256x256 imgs – DALL-E 2)
    - Using ChatGPT-4: 20 euros/month (02/2024)
- **Labeling the dataset by skin color** (for instance, using Fitzpatrick skin types)
- **Evaluating** the improvement on the test performance of the skin illness **detector**



Improving dermatology classifiers across populations using images generated by large diffusion models (Sagers et. al)



Fitzpatrick skin types

Thanks for your attention!

Appendix

- How is the FID measured?

FID is measured by computing the differences between the representations of features, such as edges and lines, and higher-order phenomena, such as the shapes of eyes or paws that are transformed into an intermediate latent space. FID is calculated using the following steps:

1. **Preprocess the images**. Ensure the two images are compatible using basic processing. This can include resizing to a given dimension size, such as 640x480 pixels, and then normalizing pixel values.
2. **Extract feature representations**. Pass the real and generated images through the Inception-v3 model. This transforms the raw pixels into numerical vectors to represent aspects of the images, such as lines, edges and higher-order shapes.
3. **Calculate statistics**. Statistical analysis is performed to determine the mean and covariance matrix of the features in each image.
4. **Compute the Fréchet distance**. Compare the difference between each image's computed mean and covariance matrixes.
5. **Obtain the FID**. Compare the Fréchet distance between the real and generated images. Lower numbers indicated the images are more similar.