Fighting Class Imbalance with Textual Inversion

SAPIENZA
UNIVERSITÀ DI ROMA

Bahareh Najafi - 2042940
Anton Kutsenko - 2186960

Advanced Machine Learning 2024-2025

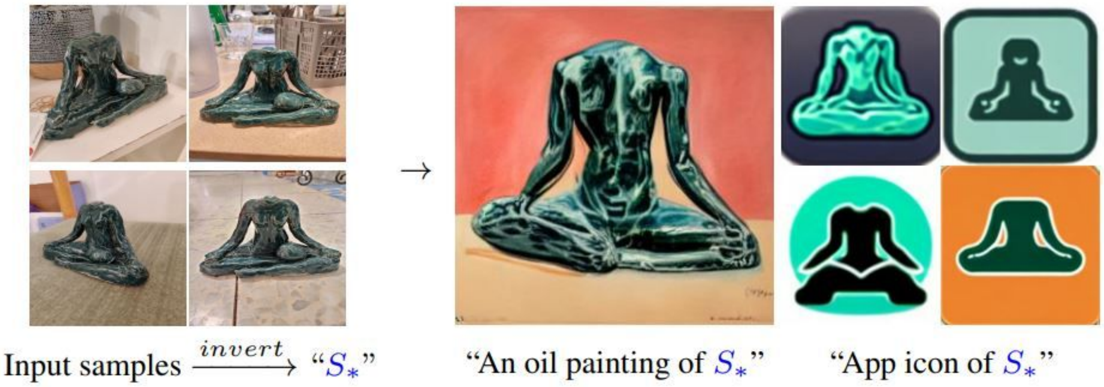
Quick reminder

Task:

Use **Textual Inversion** technique to generate synthetic data for underperforming classes and use it to improve classifier performance.

High-level Steps:

- **Target class identification**: Identify the underperforming or underrepresented class
- **Select Images**: Choose 5-10 diverse, high-quality images from this class.
- **Train Embedding** for special **<token>** representing the class using **Textual Inversion**.
- **Sample Images**: Generate new images and evaluate their quality.
- **Augment Data and Improve Classifier**: Use generated images to retrain/finetune the classifier.



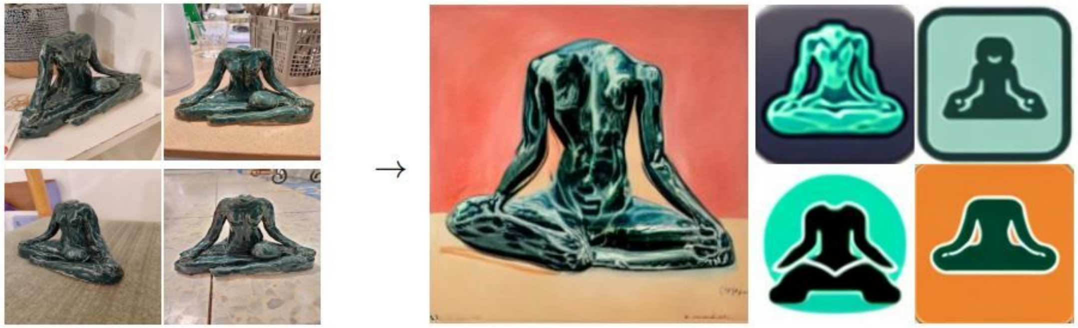
Main focus

Task:

Use **Textual Inversion** technique to generate synthetic data for underperforming classes and use it to improve classifier performance.

High-level Steps:

- **Target class identification**: Identify the underperforming or underrepresented class
- **Select Images**: Choose 5-10 diverse, high-quality images from this class.
- **Train Embedding** for special **<token>** representing the class using **Textual Inversion**.
- **Sample Images**: Generate new images and evaluate their quality.
- **Augment Data and Improve Classifier**: Use generated images to retrain/finetune the classifier.



Input samples $\xrightarrow{invert}$ "S_*" "An oil painting of S_*" "App icon of S_*"

Dataset, Literature & Sources

Dataset

- Original 102 Category Flower Dataset
- Oxford 102 Flower Dataset from Kaggle - labels to names conversion

Data information

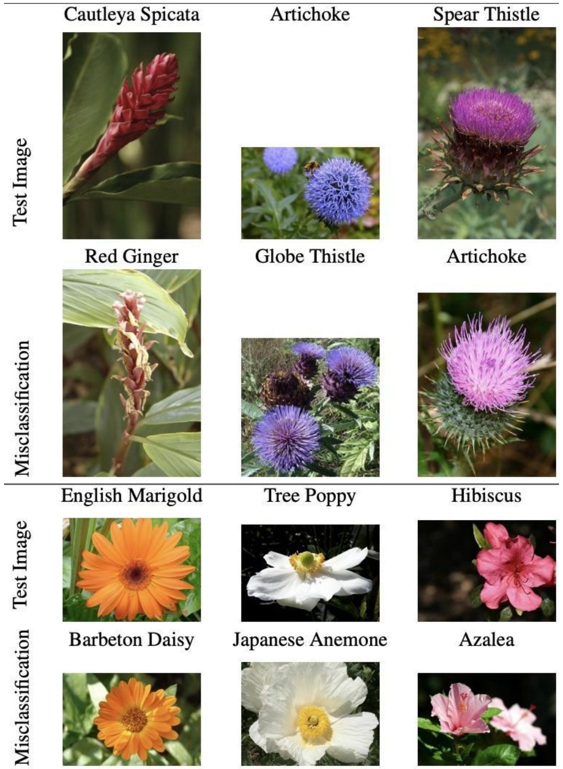
- **Training Set**: 1020 images (10 images per class)
- **Validation Set**: 1020 images (10 images per class)
- **Test Set**: 6149 images (remainder of the dataset)

Literature

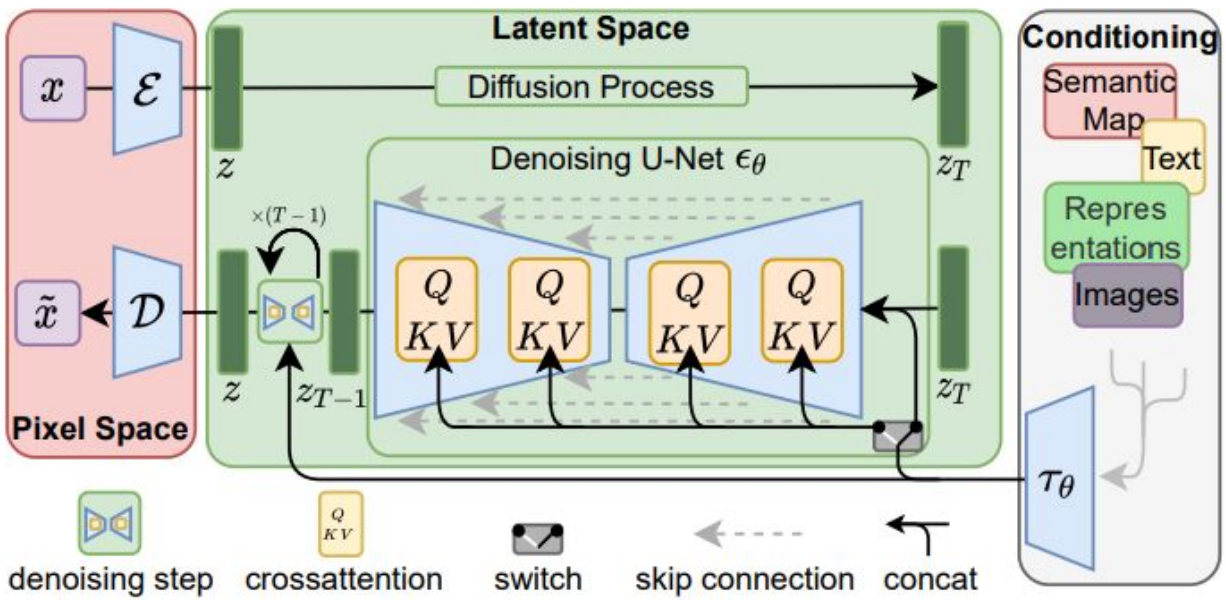
- An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion
- High-Resolution Image Synthesis with Latent Diffusion Models
- Learning transferable visual models from natural language supervision.

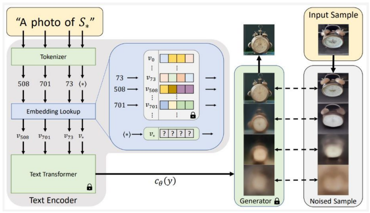
Code baseline

- Textual Inversion from Hugging Face diffusers



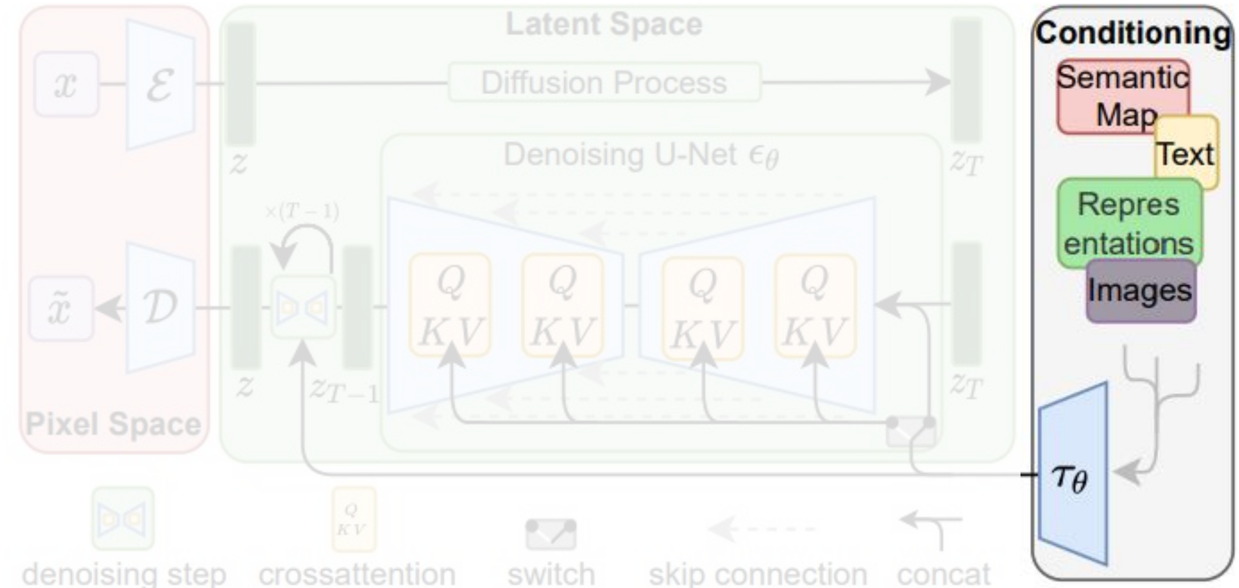
Deeper into Textual Inversion

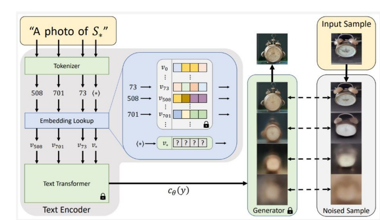




Rombach et al "High-resolution image synthesis with latent diffusion models." (2022)

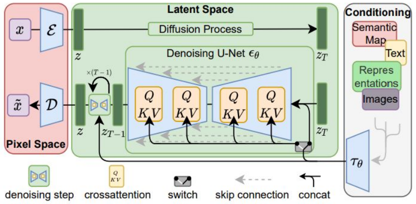
Deeper into Textual Inversion

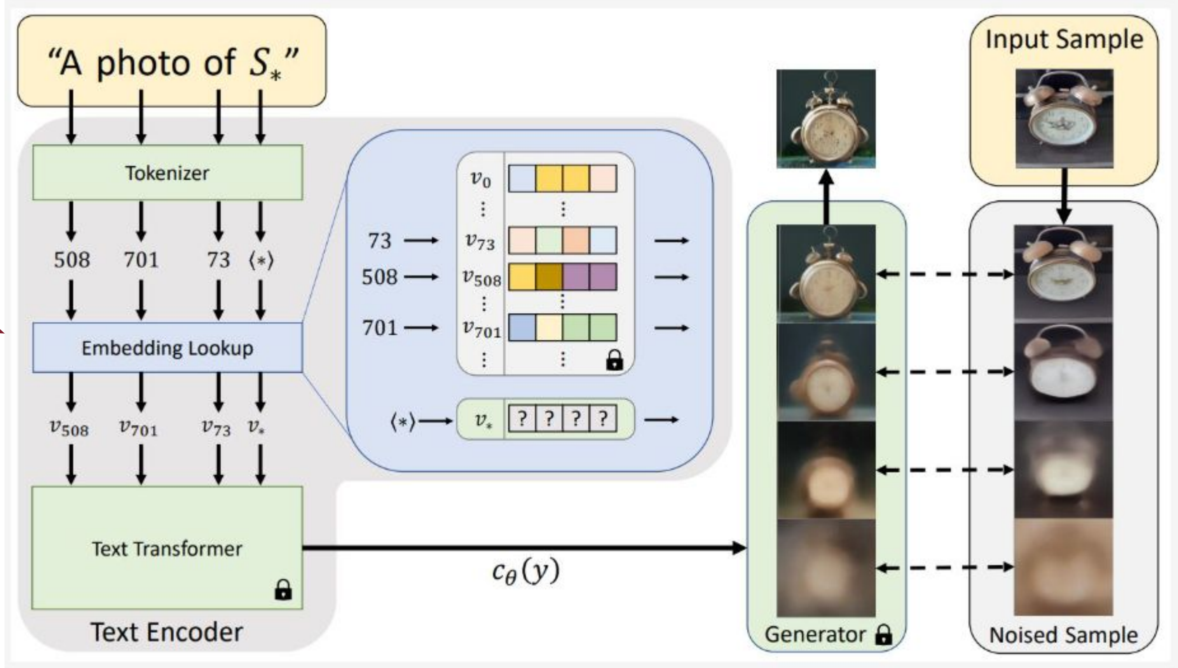




Rombach et al "High-resolution image synthesis with latent diffusion models." (2022)

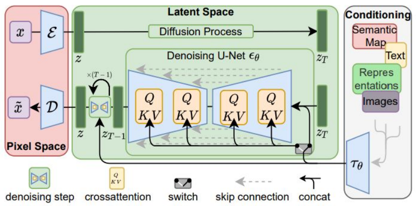
Deeper into Textual Inversion



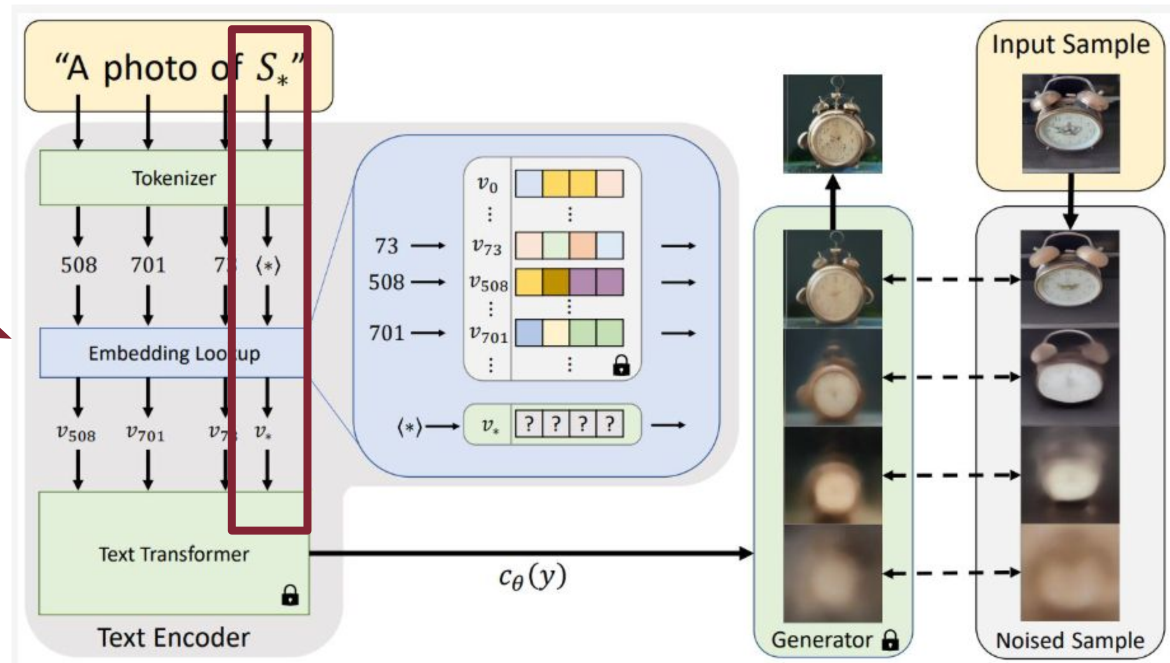


Gal et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion." (2022)

Deeper into Textual Inversion



The only part that we are training



Gal et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion." (2022)

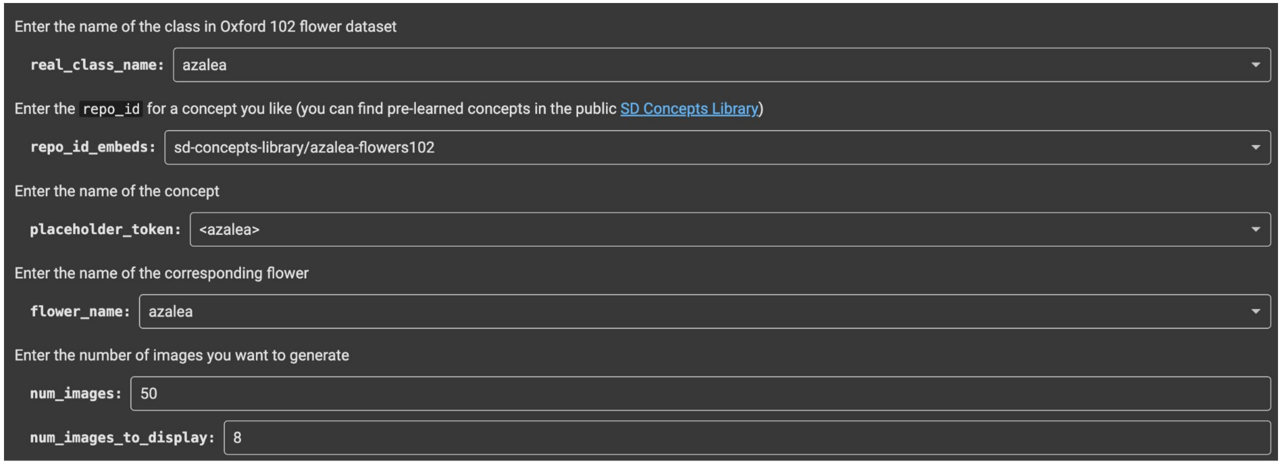
Our code

Github repo

- https://github.com/ontenkutsenko/AML_course/tree/main/Project

Main blocks

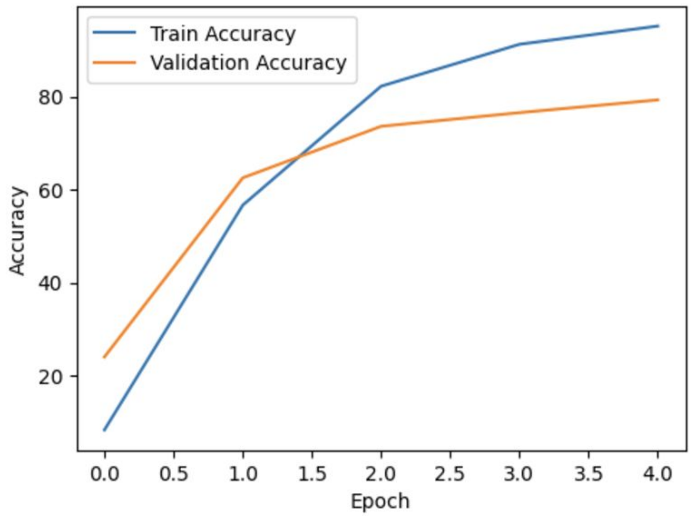
- **Modules -** logical blocks of useful functions and classes
- **Whole pipeline -** notebook with all steps to run for one concept
- **Demo** **-** a demo functionality we can try to use

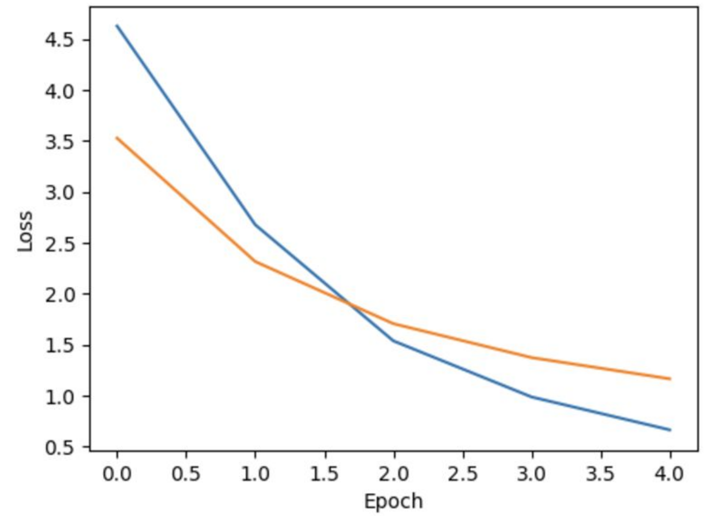


Main findings

Classification part

- **Dataset is very easy** due to a good quality. We ~0.8 validation accuracy/F1 after ~5 epochs of fine tuning fully connected layer of resnet50 and starts to overfit.
- **Small validation and training sizes -** a lot of variability in results with multiple training iterations.
- **No strong need in advanced data augmentation**. With that metrics simple augmentations and some regularization
- **Training one concept takes around ~2 hours** with 2000 training steps, so we could train **only 3** for dataset with 102 classes, which is small number to improve classification.

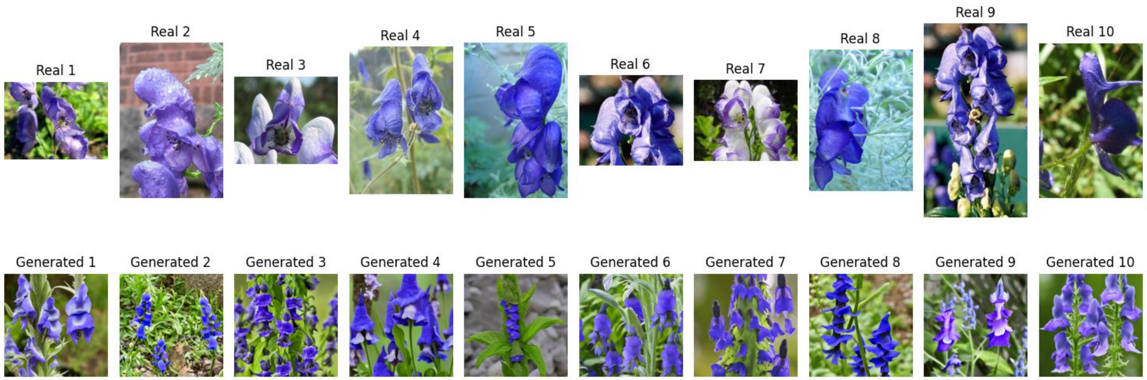




Main findings

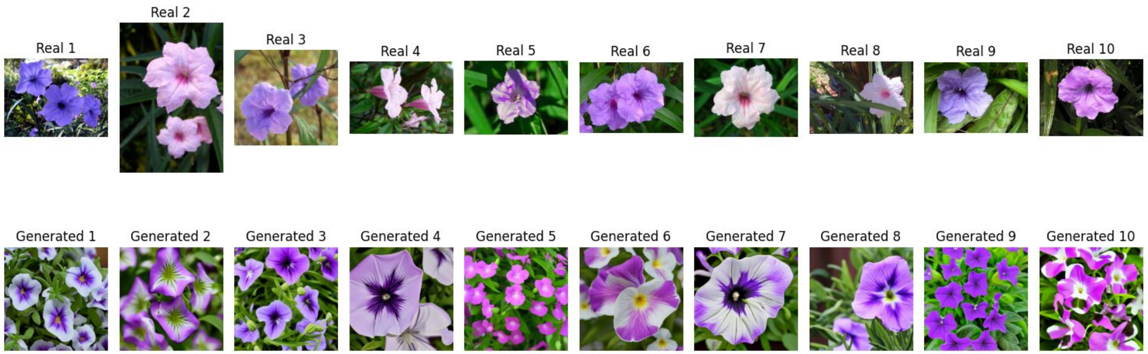
Raw Model Generation

- Using Stable Diffusion **without Textual Inversion** with class names of flowers
- Model already **knows many classes** as they were appeared in the test set



Monkhood

Mexican petunia

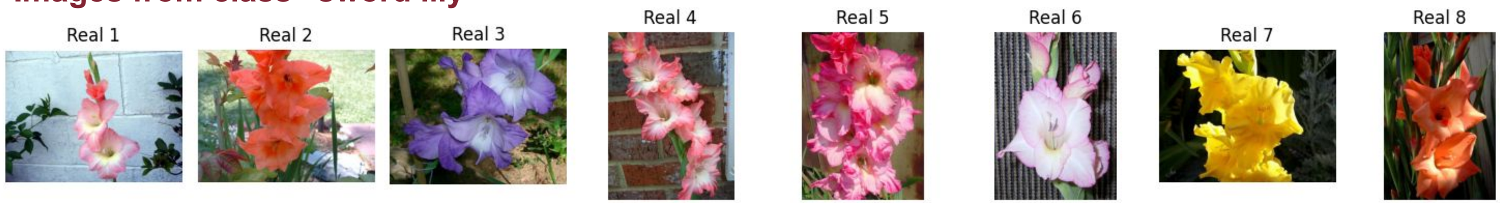


Main findings

Raw Model Generation

- Using Stable Diffusion **without Textual Inversion** with class names of flowers
- Model already **knows many classes** as they were appeared in the test set
- **Generations have a lot of variability -** doesn't fit our augmentation purpose for this data
- Model **doesn't recognise some names** of the flowers or knows them by other names
  - sword lily → gladiolus
  - flower prince of wales feathers → Amaranthus hypochondriacus

Images from class "sword lily"



Prompt: "A photo of sword lily"

Prompt: "A photo of gladiolus"

Main findings

Textual inversion

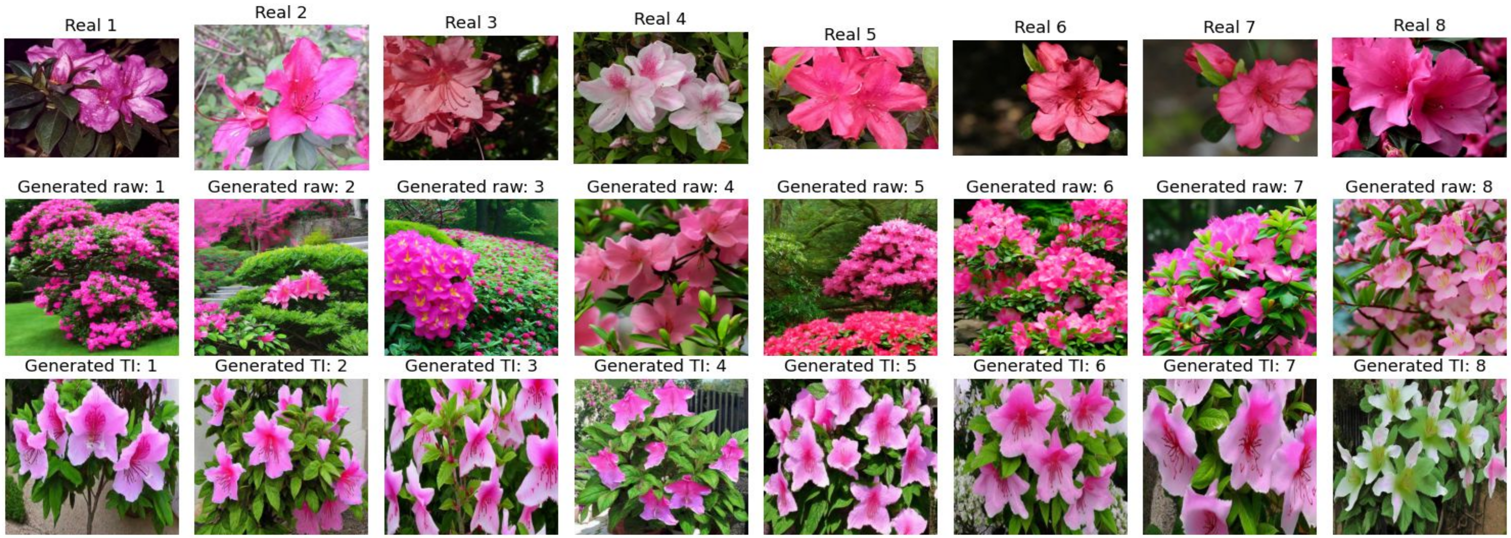
- Trained **3 tokens** for different flowers
  https://huggingface.co/sd-concepts-library/azalea-flowers102
  https://huggingface.co/sd-concepts-library/sword-lily-flowers102
  https://huggingface.co/sd-concepts-library/canna-lily-flowers102
- **Two different techniques** for training
  - Starting from token “flower” (sword lily and canna lily)
  - Starting from token matching flower name (azalea)

Comparisons

- **Using Frechet Inception Distance** to compare test+validation images, images generated from raw Stable Diffusion and after Textual Inversion
- **Comparing trained tokens embedding** with embeddings from **CLIP vocabulary:**
  - For all three token the closest ones are the other two due to similarity of data: close and high quality photos of flowers.
  - For <azalea> trained token original “azalea” token is close as well as it was starting token. Original “flower” token that we started with for other two was not similar
  - Some other “plant-related” tokens : planted, hibiscus, agawa, fleur, flourish.

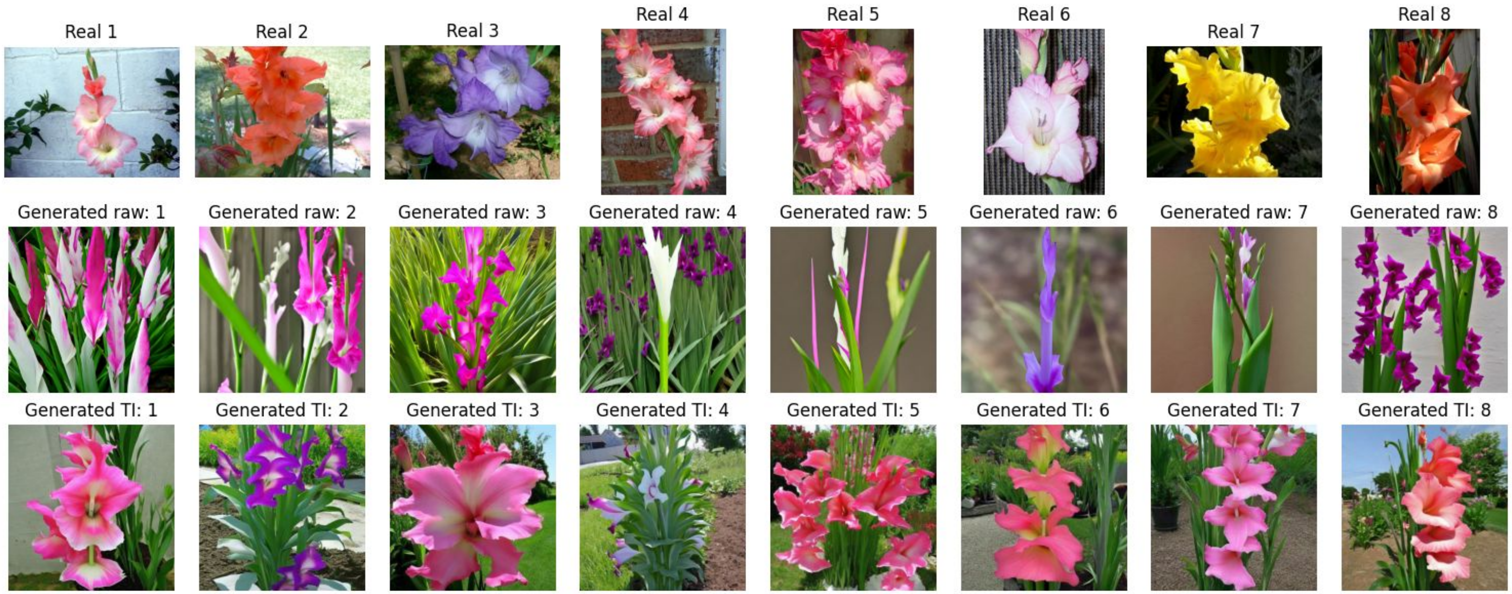
Results

Azalea
FID raw - real: **198.1**
FID trained - real: **127.7**



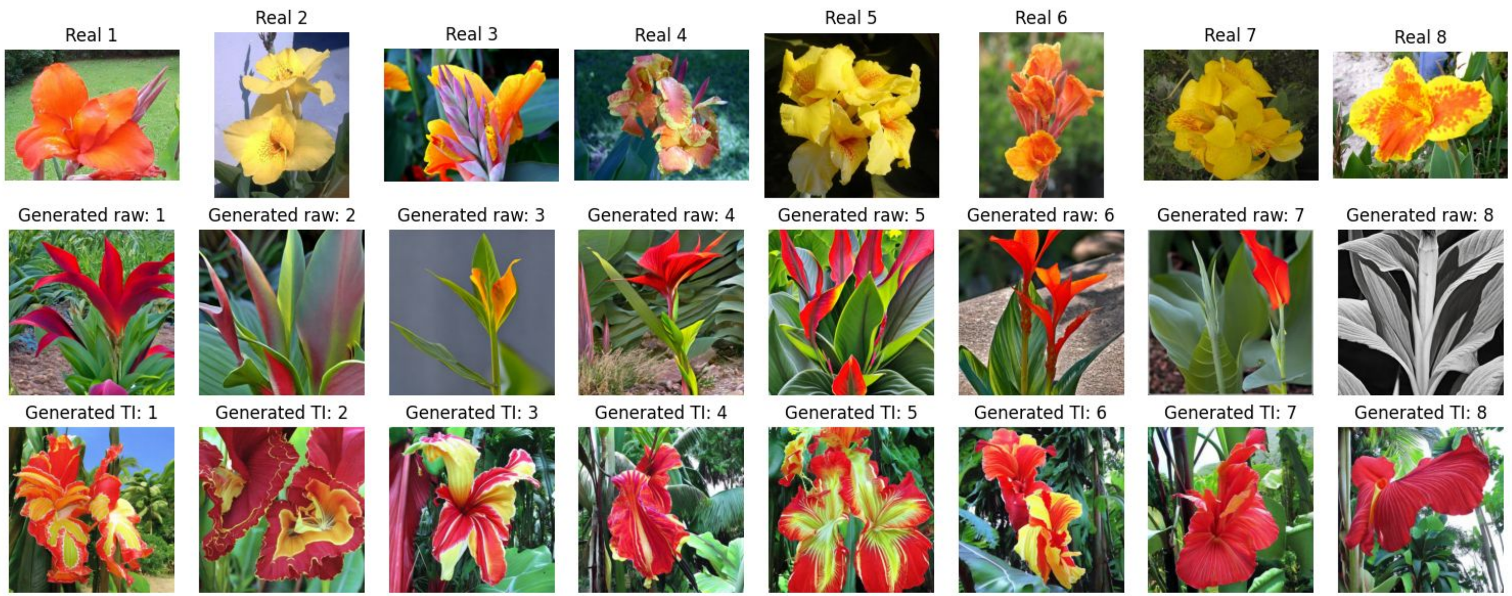
Results

Sword lily
***FID raw - real:* 206.6**
***FID trained - real:* 146.6**



Results

Canna lily
***FID raw - real:* 210.3**
***FID trained - real:* 161.9**



Final remarks

Textual inversion

- Fighting Class imbalance with Textual Inversion is good if:
  - Classes represent rare and very specific concepts - otherwise use raw model
  - Classes do not have much variability of object appearance
  - Number of underrepresented classes is small (i.e. binary classification)

Limitations and future work

- Try to compare results not only to target class but with all classes from dataset to see if generated images really represent specific class better
- More focus on prompts:
  - Quality of prompts may increase diversity and help to find perfect image structure for the given dataset
- Try to train token starting from empty or random token but for longer - to give model more freedom
- Use other techniques like LoRA or DreamBooth

Thank you for the attention!

References

- *An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion*
- *High-Resolution Image Synthesis with Latent Diffusion Models*
- Learning transferable visual models from natural language supervision