Our github link

Bringing old photos back to life

Original github link

Team 5

Chaeeun Ryu: 2018312824
Sofia Vega Zambada: 2021318437

Overview

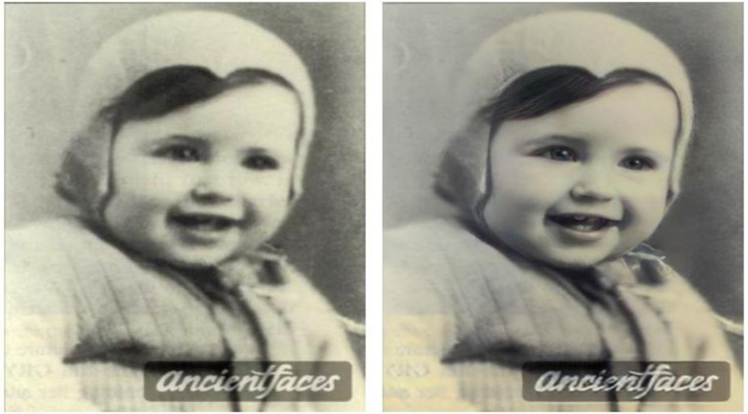
01 Introduction

02 Method

03 Dataset

04 Code demo

05 Evaluation



06 Conclusion

07 Challenges

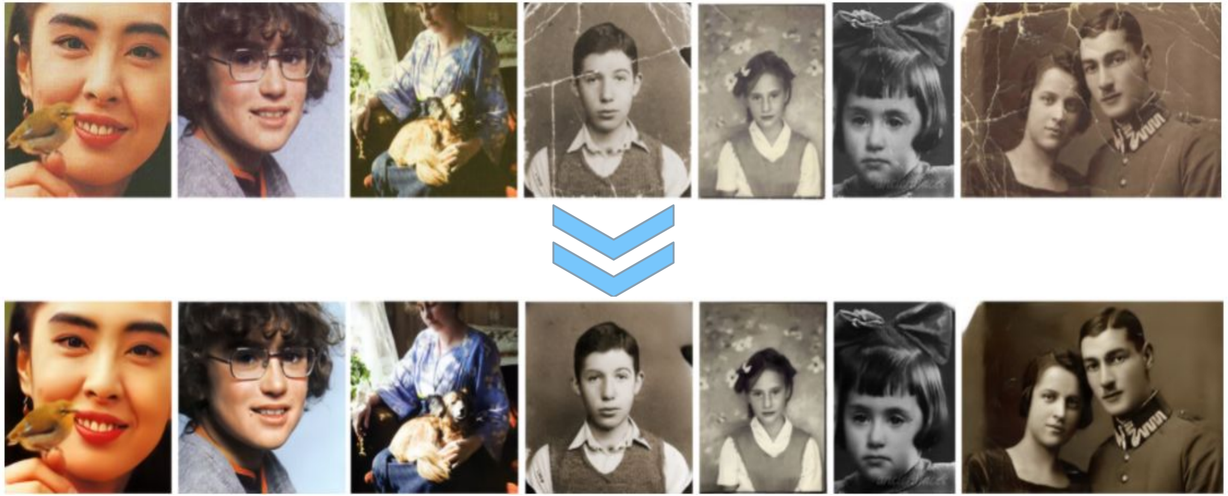
01

Introduction

Photo restoration

BEFORE

AFTER



Ziyu Wan Bo Zhang Dongdong Chen Pan Zhang Dong Chen Jing Liao Fang Wen

City University of Hong Kong Microsoft Research Asia Microsoft Cloud AI USTC

Two Types of degradations

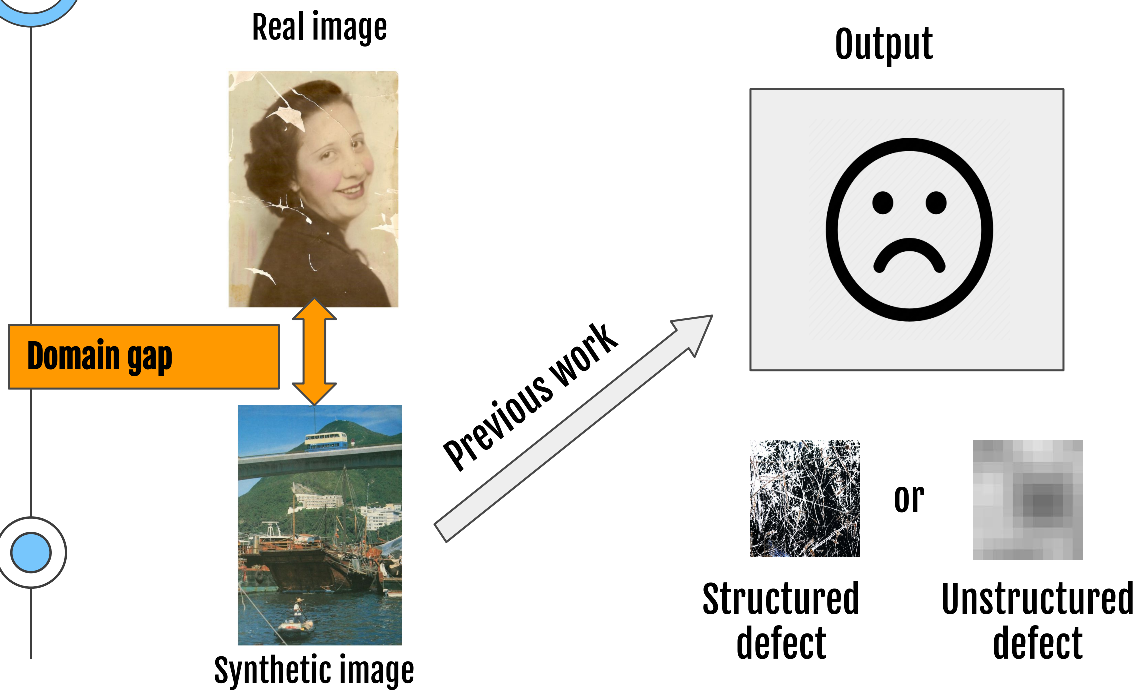
Unstructured

1. Blurriness
2. Noise
3. Low resolution
4. Sepia issue

Structured

1. Scratches
2. Holes
3. Spots

Previous work



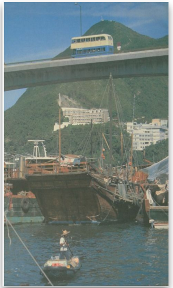
- Focused on either structured or unstructured errors
- Used synthetic data only for training
- Unsatisfied output

This paper

1

Real image

2



Synthetic image

3

Corresponding ground truth

Structured defect

&

Unstructured defect

- This project proposes **a triplet domain** translation network to restore the mixed degradation in old photos.
- Deals with **both structured and unstructured** defects

.

02
Method

Two problems and respective solution

01

Generalization issue

There always exists a **domain gap between synthetic and real photos.**

Real ≠ Synthetic

02

Mixed degradation issue

The defects of old photos are a **compound** of multiple degradations. Different types of defects necessitate **different methods** to solve.

film noise, blurriness and color fading …

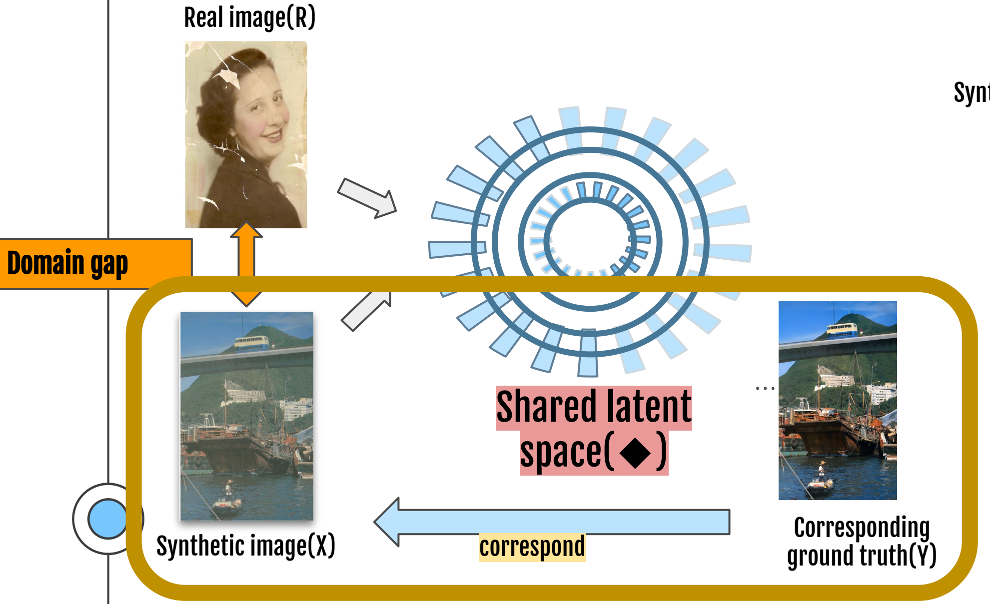
scratches and blotches …

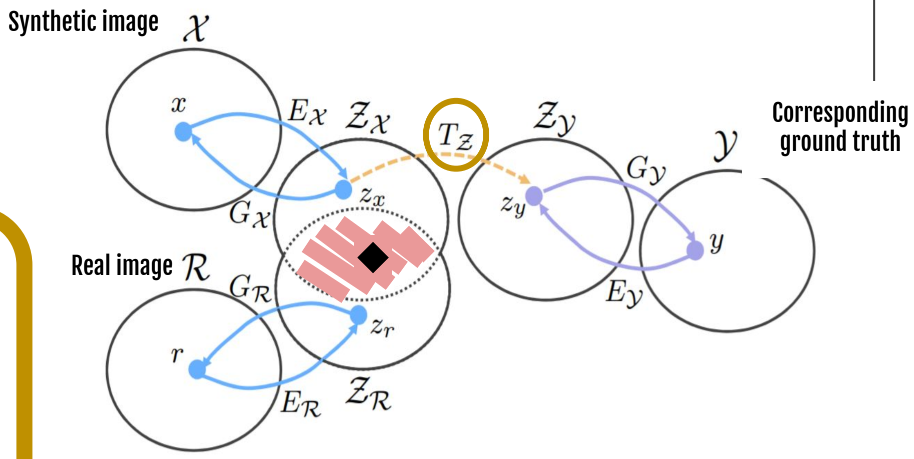
solutions

Restoration via latent space translation

Multiple degradation restoration

Restoration via latent space translation

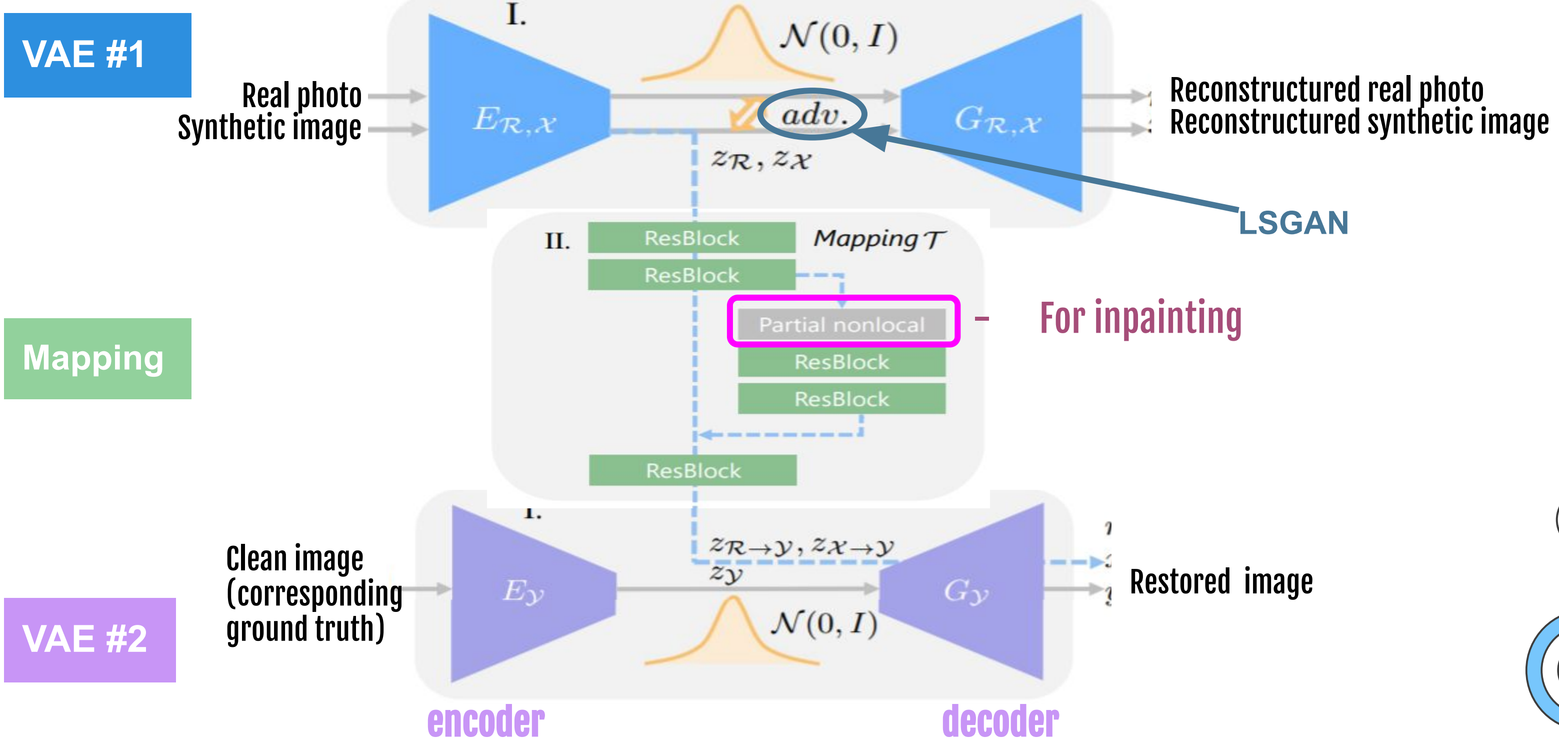




- Solves generalization issue by restoration via latent space(Z) translation among three domains: R,X,Y.
- To mitigate the domain gap, this project aligns latent spaces of synthetic images and real old photos into the shared domain, which encodes features for all the corrupted images, either synthetic or real ones by first VAE.
- Learn the translation from the latent space of corrupted images(Z_X) to the latent space of ground truth(Z_Y)

Architecture of network

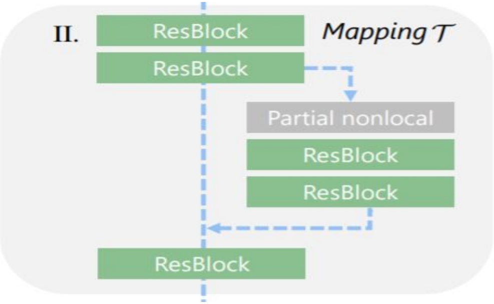
- Objective of Variational AutoEncoder(VAE): Create the most similar output as the input and find useful patterns in input datasets.

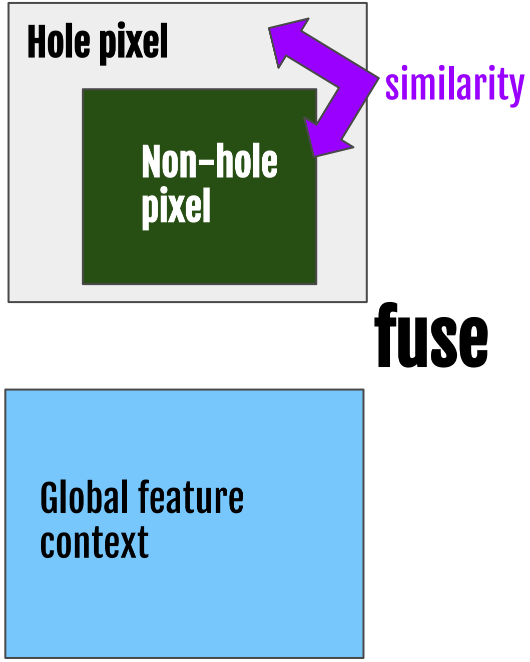


Multiple degradation restoration

For inpainting

Mapping





- Calculate similarity between non-hole pixel and hole pixel
- Fuse the non-hole pixel using global feature context
- Make a new feature map

Architecture in details

Pix2Pix model

Module	Layer	Kernel size / stride	Output size
Encoder E	Conv	$7 \times 7/1$	$256 \times 256 \times 64$
	Conv	$4 \times 4/2$	$128 \times 128 \times 64$
	Conv	$4 \times 4/2$	$64 \times 64 \times 64$
	ResBlock×4	$3 \times 3/1$	$64 \times 64 \times 64$
Generator G	ResBlock×4	$3 \times 3/1$	$64 \times 64 \times 64$
	Deconv	$4 \times 4/2$	$128 \times 128 \times 64$
	Deconv	$4 \times 4/2$	$256 \times 256 \times 64$
	Conv	$7 \times 7/1$	$256 \times 256 \times 3$
Mapping $\mathcal{T}$	Conv	$3 \times 3/1$	$64 \times 64 \times 128$
	Conv	$3 \times 3/1$	$64 \times 64 \times 256$
	Conv	$3 \times 3/1$	$64 \times 64 \times 512$
	Partial Non-local	$1 \times 1/1$	$64 \times 64 \times 512$
	Resblock×2	$3 \times 3/1$	$64 \times 64 \times 512$
	ResBlock×6	$3 \times 3/1$	$64 \times 64 \times 512$
	Conv	$3 \times 3/1$	$64 \times 64 \times 256$
	Conv	$3 \times 3/1$	$64 \times 64 \times 128$
	Conv	$3 \times 3/1$	$64 \times 64 \times 64$

03
Dataset

Enter a subtitle here if you need it

3 Datasets needed for training

None of the datasets were given in the project.

#1

VOC dataset
Popular dataset for image detection

Downloaded from website in .tar file

#2

Old grayscale image dataset

#3

Old RGB image dataset

Collected images by Image Scraping(Crawling)Preprocessed

BeautifulSoup

Demo file

04

Implementation in Code Demo

3 parts of training

Training dataset   Trained model

01
Train_domain_A.py

We train VAE1 to encode real old images and synthetic “old” images into a shared latent space

02
Train_domain_B.py

We train VAE2 to encode the clean images (ground truth)

03
Train_mapping.py

We learn the mapping that restores the corrupted images to the clean images in the latent space.

Training via transfer learning demo

https://colab.research.google.com/drive/1HO_8oQ2himYgib0z4RV6vH0OVZa0Vdz7?usp=sharing

Testing the model

Testing demo
https://colab.research.google.com/drive/1jS4faSuP3XHausSADuYA4wcC4-hmhjgU?usp=sharing

Tested image

05

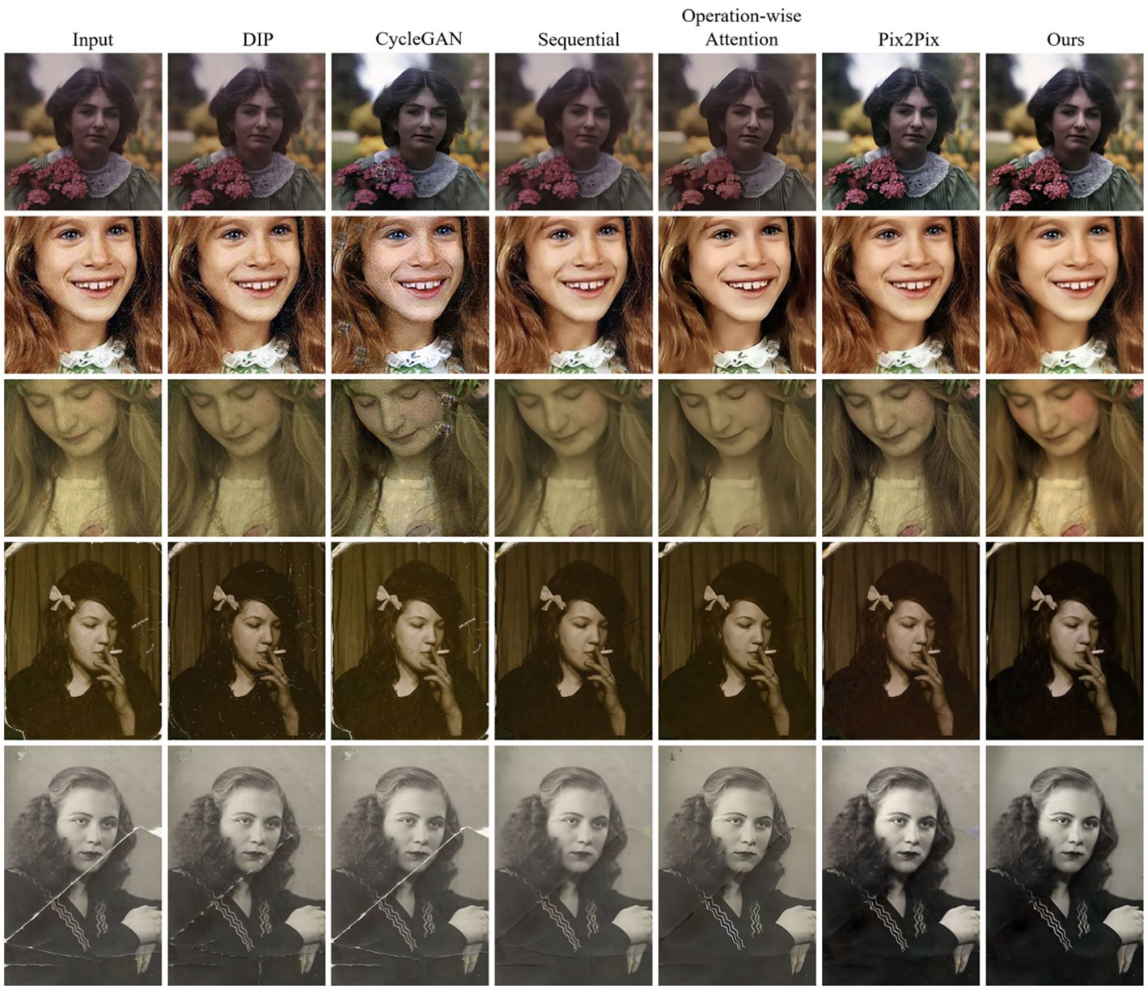
Evaluation

Enter a subtitle here if you need it

Quantitative analysis

	Metrics			
Method	PSNR	SSIM	LPIPS	FID
Input	12.92	0.49	0.59	306.80
Attention	24.12	0.70	0.33	208.11
DIP	22.59	0.57	0.54	194.55
Pix2Pix	22.18	0.62	0.23	135.14
Sequential	22.71	0.60	0.49	191.98
New model w/o PN	23.14	0.68	0.26	143.62
New model w PN	23.33	0.69	0.25	134.35

Qualitative analysis



User study

	User selection (percentage)				
Method	Top 1	Top 2	Top 3	Top 4	Top 5
DIP	2.75	6.99	12.92	32.63	69.70
Cycle GAN	3.39	8.26	15.68	24.79	52.12
Sequential	3.60	20.97	51.48	83.47	93.64
Attention	11.22	28.18	56.99	75.85	89.19
Pix2Pix	14.19	54.24	72.25	86.86	96.61
New model	**64.83**	**81.35**	**90.68**	**96.40**	**98.72**

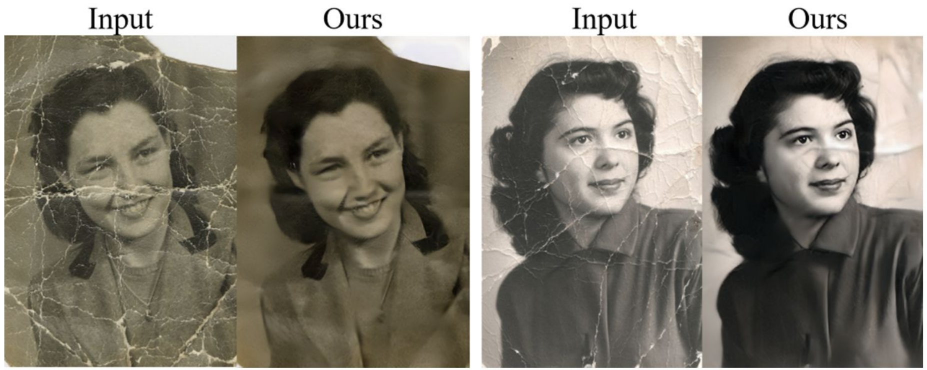
06

Conclusion and Project Challenges

Conclusion

- This project developed a triplet domain translation network to restore both structured and unstructured defects in old photos.
- The domain gap is reduced between old photos and synthetic images.
- The translation to clean images is learned in latent space.
- This method suffers less from generalization compared to other methods.
- The writers propose a partial nonlocal block which restores the latent features by taking the global context into account. This way the scratches can be restored consistently.
- This method demonstrates good performance even in severely degraded old photos.

Project challenges



1. This method cannot handle complex shading.

   The authors believe this is because the training dataset contained few photos with this defect.

Project challenges

2. This model does not inpaint large holes in corners / borders

07

Challenges

Challenges

Colab crashing often The colab crashed often, for not specific reasons. It was not due to the GPU, RAM, or run-time error. But we fixed it by decreasing the size of the image datasets, and keep on trying to make it work.	**Due to errors, code modifications were needed** There were a lot of errors, but we solved it through searching the errors in google, referring to the 'issue' page in the github, and trying new different things like changing hyper parameters.	**Heavy memory space** We were not able to run it locally in visual studio code, jupyter notebook or virtual machine, but could only run it through colab remotely because the size of the project was too large
Datasets were not given The project required 3 datasets, but none of them was given. So it was not only hard to image crawl, but defer which image is apt to train for this project.	**Image crawling** It was our first time doing image scraping, so we were not accustomed to the tools for image crawling. Also, finding old RGB images was hard because there weren't sufficient number of images per each website, so we visited about 30 websites to prepare the dataset for RGB.	**Environment not matching** We initially wanted to run each file independently, without the given training command.However, it was not possible in the windows environment. So, we had to change the bash commands or commands for linux to work in colab or Windows environment.
Many independent files As there were many independent .py files in the project, it was hard to figure out how each files relate, and in what order they were running. Therefore, we inserted codes to print out the name of the function when the function was called	**math equations and symbols** Understanding the objective function or loss function was hard because there were many kinds of models combined in this one project. Therefore, without beginning with the codes, we took an effort to understand the math part first by studying VAE and GAN in Notion	**Taking a long time to train** Whenever we got an error, and the training stopped due to that error, and we had to restart training from the very beginning. And the training took very long time, so we were afraid we might not be able to finish training the whole model in time. But, we solved this problem by reducing the size of image datasets and solving errors.

Thank you