Computer Vision Term Project

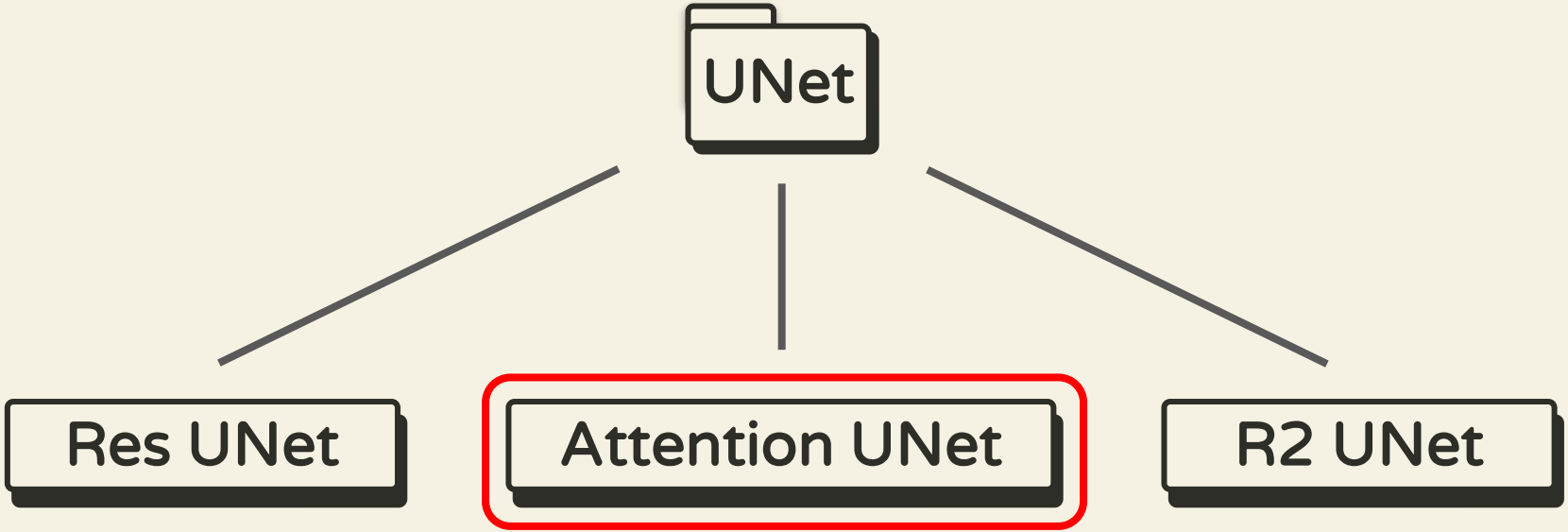
Team 6 - 방근호, 심우석, 조민혁, 이소정, 박윤재

Table of contents

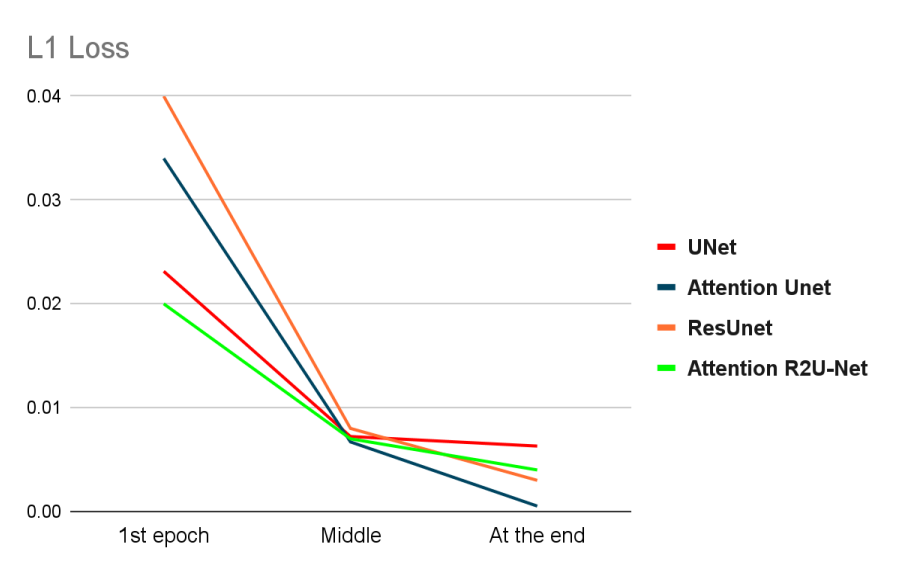
01

Introduction

1. Introduction



1. Introduction



Attention Unet is Best!

1. Introduction

Problem

- Cuda Memory out (In colab)
- Abnormally high loss value
- Evaporate learning model upon error

02

Related Work

2. Related Work

We looked for models mainly on the Unet

Unet

https://arxiv.org/pdf/1505.04597.pdf

Res Unet

https://arxiv.org/pdf/1904.00592.pdf

Attention R2U-Net

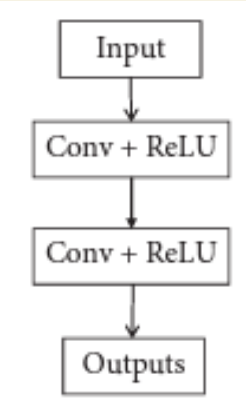
https://downloads.hindawi.com/journals/scn/2021/6625688.pdf

Attention Unet ✓

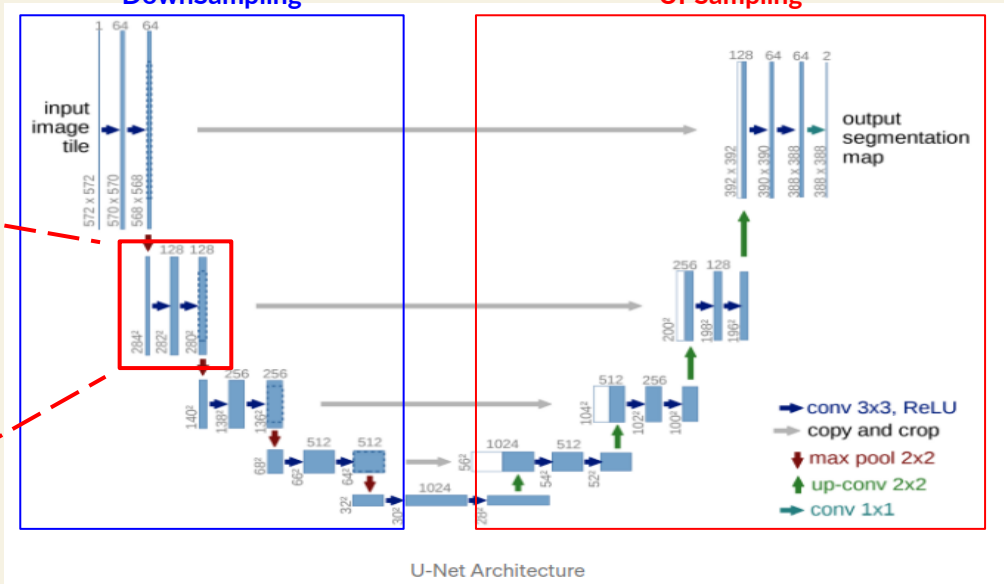
https://arxiv.org/pdf/1804.03999.pdf

2. Related Work

Unet



Convolutional Block

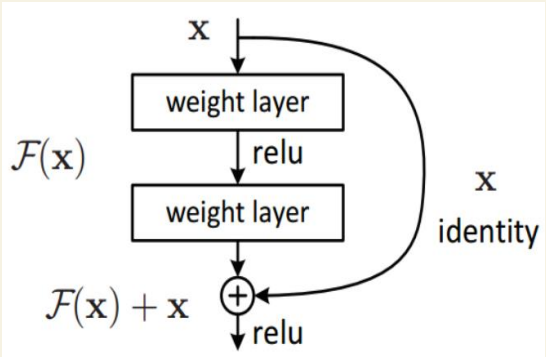


U-Net Architecture

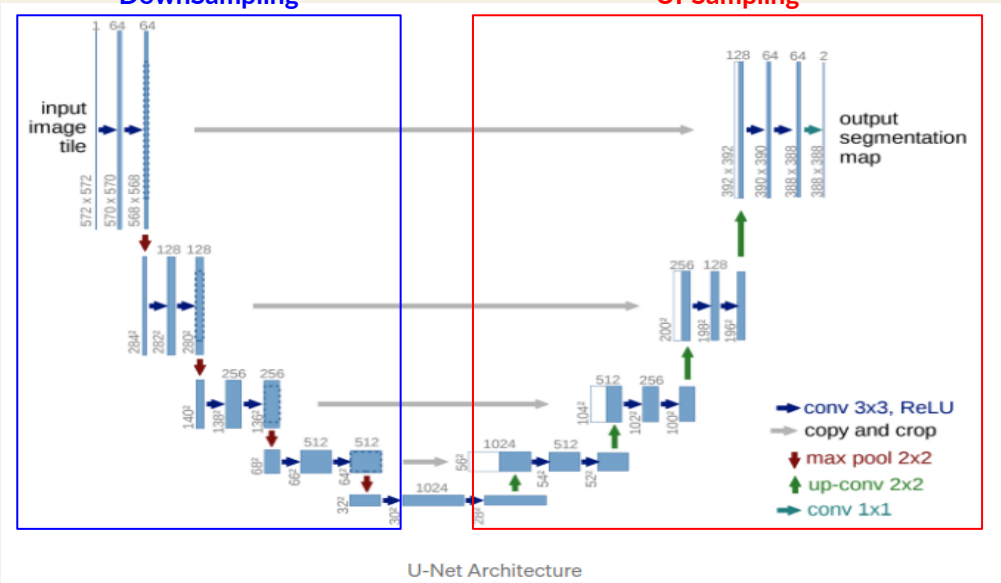
UNET is a U-shaped encoder -decoder network architecture, which consists of four encoder blocks and four decoder blocks that are connected via a bridge.

2. Related Work

Res Unet



Residual Block

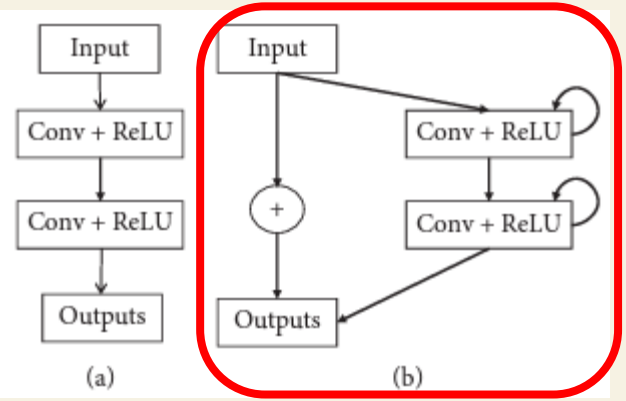


U-Net Architecture

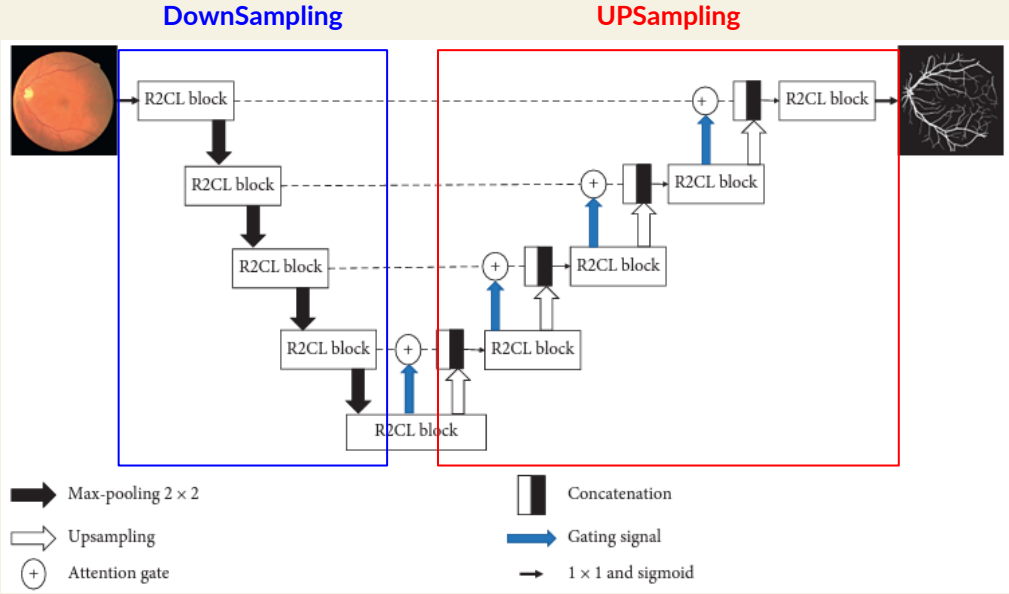
The structure of Res Unet and Unet is the same.
But there is a difference that the convolutional block is different.

2. Related Work

Attention R2U-Net



A - basic unit of the U-net
B - R2 Convolutional layer block



Convolutional encoding and decoding unit based on the recurrent residual convolutional layer and R2AU-Net structure with AG connection

2. Related Work

Attention Unet

Attention Gate

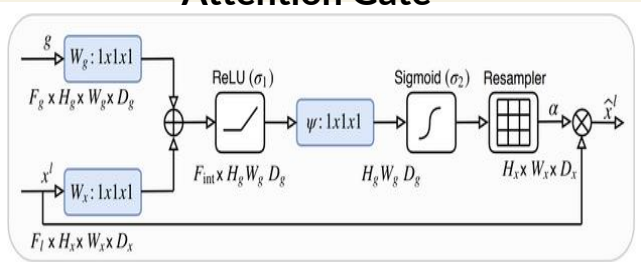
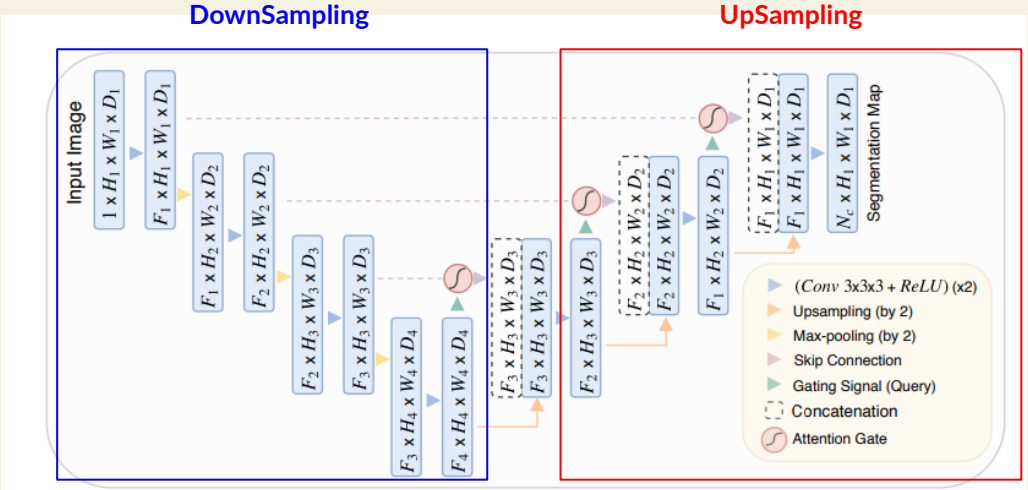


Figure 2: Schematic of the proposed additive attention gate (AG). Input features (x^l) are scaled with attention coefficients (α) computed in AG. Spatial regions are selected by analysing both the activations and contextual information provided by the gating signal (g) which is collected from a coarser scale. Grid resampling of attention coefficients is done using trilinear interpolation.



Learning Where to Look for the Pancreas (췌장)

AG is to connect the featuremap of the encoding layer with the featuremap of the previous decoding layer through skip connection.

03

Methods

3. Methods

Problem

- Cuda Memory out (In colab) -> batch size setting
- Abnormally high loss value -> Change Loss function
- Evaporate learning model upon error
  -> Save and load learning models per epoch

3. Methods

Data augmentation

Flip the image using opencv -> training dataset (10,000 -> 40, 000)

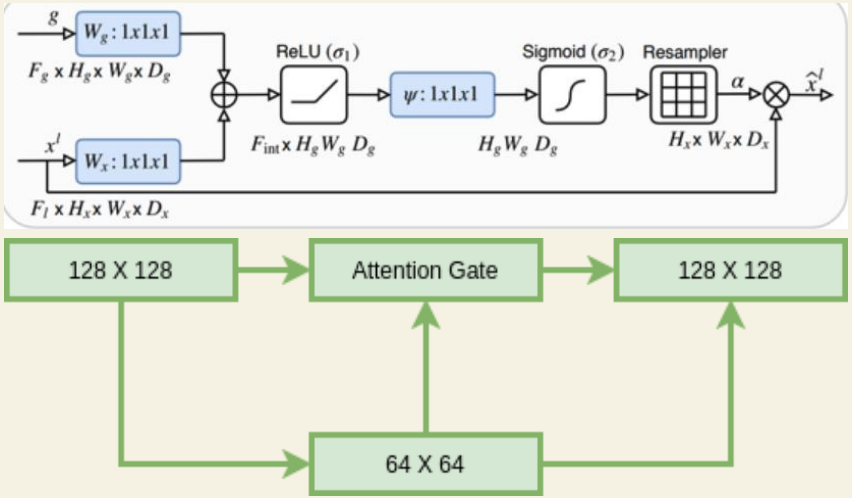
3. Methods

Hyperparameter

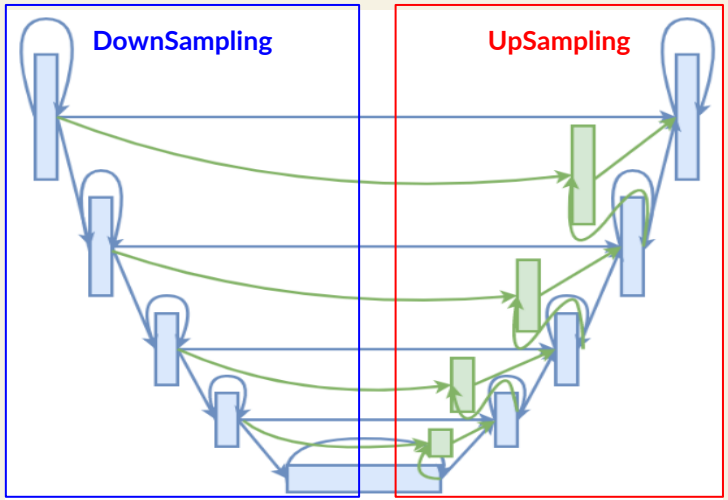
hyperparameter	
Batch size	6
Learning rate	0.0001
Lambda learning rate	Lambda epoch - 0.95
Loss function	L1 Loss
Optimizer	Adam

3. Methods

Our Model Architecture



Attention Gate



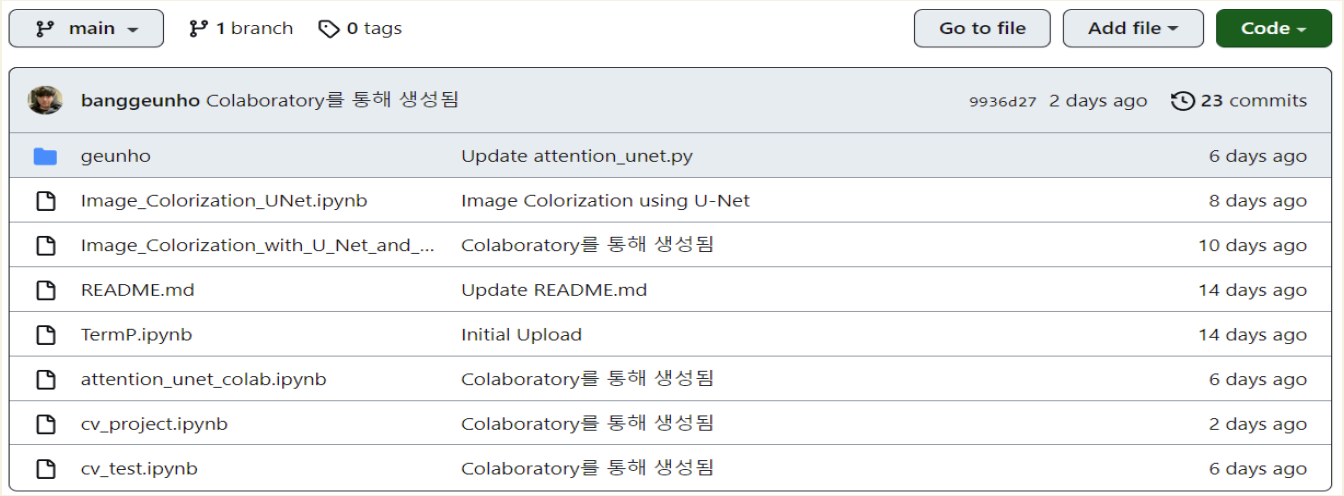
Our Network

04

Experiments

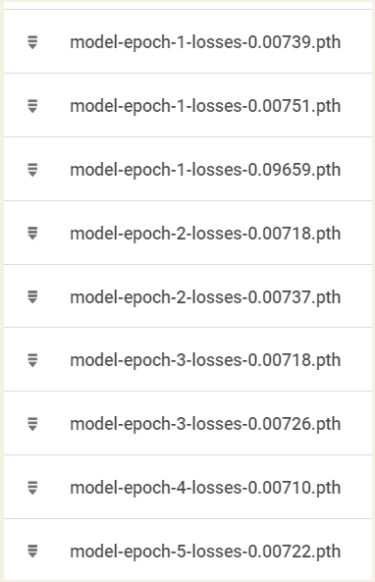
4. Experiments

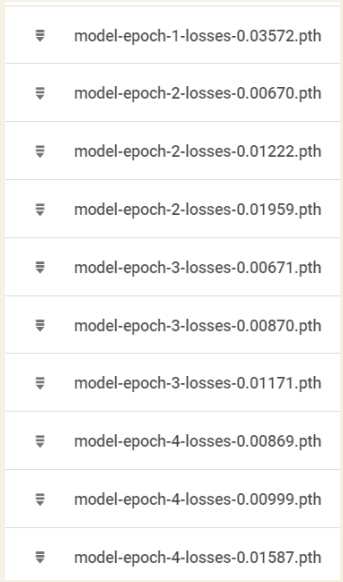
We used Github to share the experimental situation and manage the version -
https://github.com/dntjr41/CV_TermP

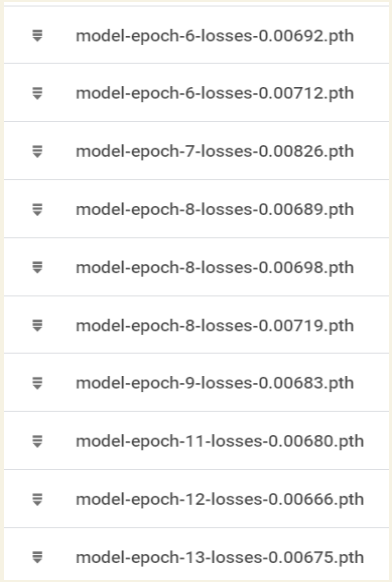


4. Experiments

Model - Checkpoint per 1-epoch







We were all conducted in colab, we had to save the experimental results in the middle.

To do this, we saved the model when performance improved after 1-epoch

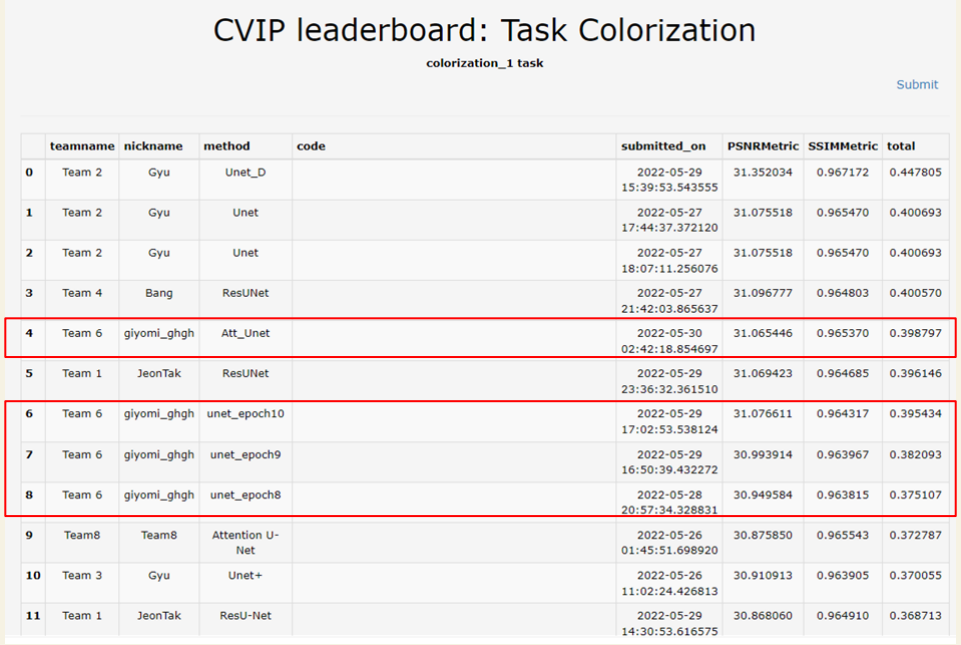
4. Experiments

epoch : 1

epoch : 5

epoch : 30

4. Experiments



CVIP leaderboard: Task Colorization

colorization_1 task

Submit

	teamname	nickname	method	code	submitted_on	PSNRMetric	SSIMMetric	total
0	Team 2	Gyu	Unet_D		2022-05-29 15:39:53.543555	31.352034	0.967172	0.447805
1	Team 2	Gyu	Unet		2022-05-27 17:44:37.372120	31.075518	0.965470	0.400693
2	Team 2	Gyu	Unet		2022-05-27 18:07:11.256076	31.075518	0.965470	0.400693
3	Team 4	Bang	ResUNet		2022-05-27 21:42:03.865637	31.096777	0.964803	0.400570
4	Team 6	giyomi_ghgh	Att_Unet		2022-05-30 02:42:18.854697	31.065446	0.965370	0.398797
5	Team 1	JeonTak	ResUNet		2022-05-29 23:36:32.361510	31.069423	0.964685	0.396146
6	Team 6	giyomi_ghgh	unet_epoch10		2022-05-29 17:02:53.538124	31.076611	0.964317	0.395434
7	Team 6	giyomi_ghgh	unet_epoch9		2022-05-29 16:50:39.432272	30.993914	0.963967	0.382093
8	Team 6	giyomi_ghgh	unet_epoch8		2022-05-28 20:57:34.328831	30.949584	0.963815	0.375107
9	Team8	Team8	Attention U-Net		2022-05-26 01:45:51.698920	30.875850	0.965543	0.372787
10	Team 3	Gyu	Unet+		2022-05-26 11:02:24.426813	30.910913	0.963905	0.370055
11	Team 1	JeonTak	ResU-Net		2022-05-29 14:30:53.616575	30.868060	0.964910	0.368713

Check the total score per epoch

05

Conclusion

5. Conclusion

What we learned

- There are various models just for colorization, and learning time and results vary widely depending on which model we choose.
- When doing deep learning, better performance hardware (mainly graphic cards) is needed for better results.
- It takes a very long time for an accurate result.
- Methods of various libraries held by PyTorch.

5. Conclusion

Future Extensions

Multi ResUNet

UNet++

No-New Net...

etc.

Peak Signal-to-Noise Ratio

Structural Similarity Index Map

Team 6 Info.

방근호

- 201635816
- 소프트웨어학과
- panggeunho@gmail.com

심우석

- 201636417
- 소프트웨어학과
- qkqh8639@gmail.com

조민혁

- 201735880
- 소프트웨어학과
- mhcho98@gmail.com

이소정

- 201935098
- 소프트웨어학과
- sjthwjd27@gachon.ac.kr

박윤재

- 201835455
- 소프트웨어학과
- rnjsxorl3075@gmail.com

Github

Team 6 - Github Repository

https://github.com/dntjr41/CV_TermP

Resources

Papers

- https://arxiv.org/pdf/1804.03999.pdf
- https://arxiv.org/pdf/1904.00592.pdf
- https://arxiv.org/pdf/1505.04597.pdf
- https://downloads.hindawi.com/journals/scn/2021/6625688.pdf
- https://www.koreascience.or.kr/article/JAKO201810263413596.pdf

Websites

- https://github.com/LeeJunHyun/Image_Segmentation
- https://github.com/sungyoonahn/Hint-based-image-colorization-using-Attention-Unet
- https://github.com/richzhang/colorization-pytorch
- https://hoya012.github.io/blog/Image-Data-Augmentation-Overview/

Thanks!