VITMST++

Efficient Hyperspectral Skin Reconstruction

Ana Clara Caznok Silveira
Diedre Do Carmo
Lucas Ueda
Guilherme Gelmi
Felipe Hiroshi Inazumi
Eduardo Rittner Coelho
Tárik Ponte e Sá
Paula Costa
Denis Fantinato
Letícia Rittner

RainbowAI

ICASSP 2024 KOREA

UNICAMP

FEEC

MICLab

Table of contents

01 Hyperskin

- Dataset
- Metrics

02 Challenges

03 Strategies

- VITMST++ architecture
- Custom Loss

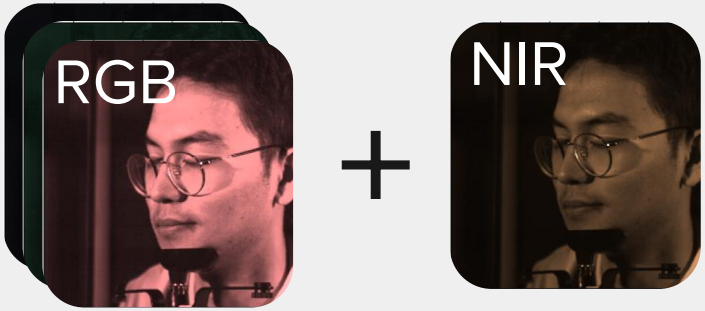
04 Final Results

05 Future improvements

01 HyperSkin Dataset

MSI

RGB + NIR



4 channels

264 images of 44 subjects

algorithm

baseline:
MST++ [1]

HSI

61 channels

[1] Cai et. al, 2021

01 HyperSkin Metrics

- SSIM: Structural Similarity Index
  - Brightness, contrast and channel structure
- SAM: Spectral Angle Mapper
  - Angles in the Spectral dimension

01 HyperSkin Metrics

- SSIM: Structural Similarity Index
  - Brightness, contrast and channel structure

MAXIMIZE — The higher, the better

- SAM: Spectral Angle Mapper
  - Angles in the Spectral dimension

MINIMIZE — The lower, the better

02 Challenges

- Data volume.
  - 1024x1024x61 images were too large to fit on the GPU
- Which architecture should we use?
- Metrics will be evaluated mainly on the face
  - SAM is more sensitive to background variations
  - How to distinguish the face from the background and improve the model's results on the face?
- Differences between channels.
  - Some channels are harder to get right than others

02 Challenges

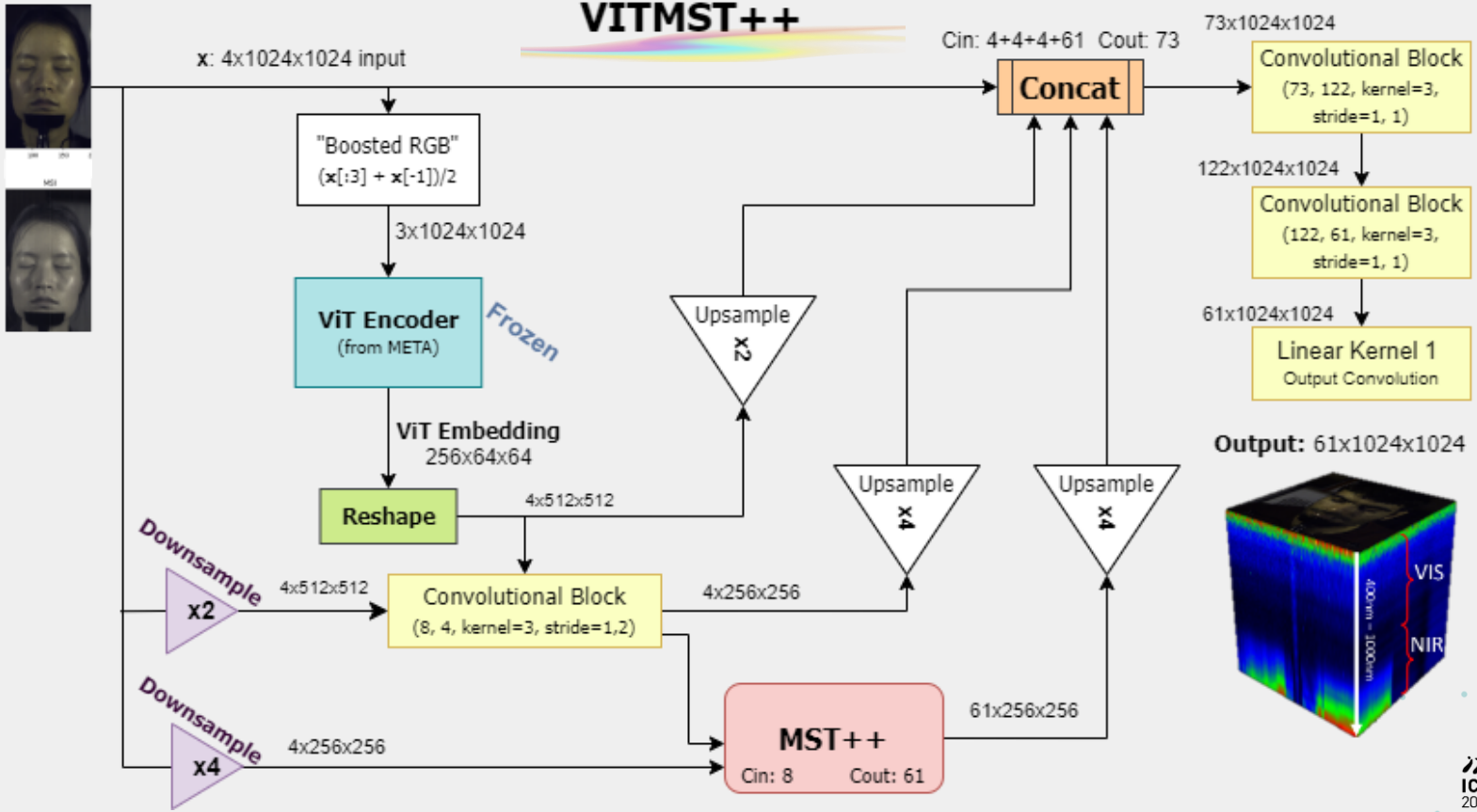
- Data volume.
  - 1024x1024x61 images were too large to fit on the GPU
- Which architecture should we use?
- Metrics will be evaluated mainly on the face
  - SAM is more sensitive to background variations
  - How to distinguish the face from the background and improve the model's results on the face?
- Differences between channels.
  - Some channels are harder to get right than others

02 Challenges

- Data volume.
  - 1024x1024x61 images were too large to fit on the GPU.
- Which architecture should we use?
- Metrics will be evaluated mainly on the face
  - SAM is more sensitive to background variations
  - How to distinguish the face from the background and improve the model's results on the face?
- Differences between channels.
  - Some channels are harder to get right than others

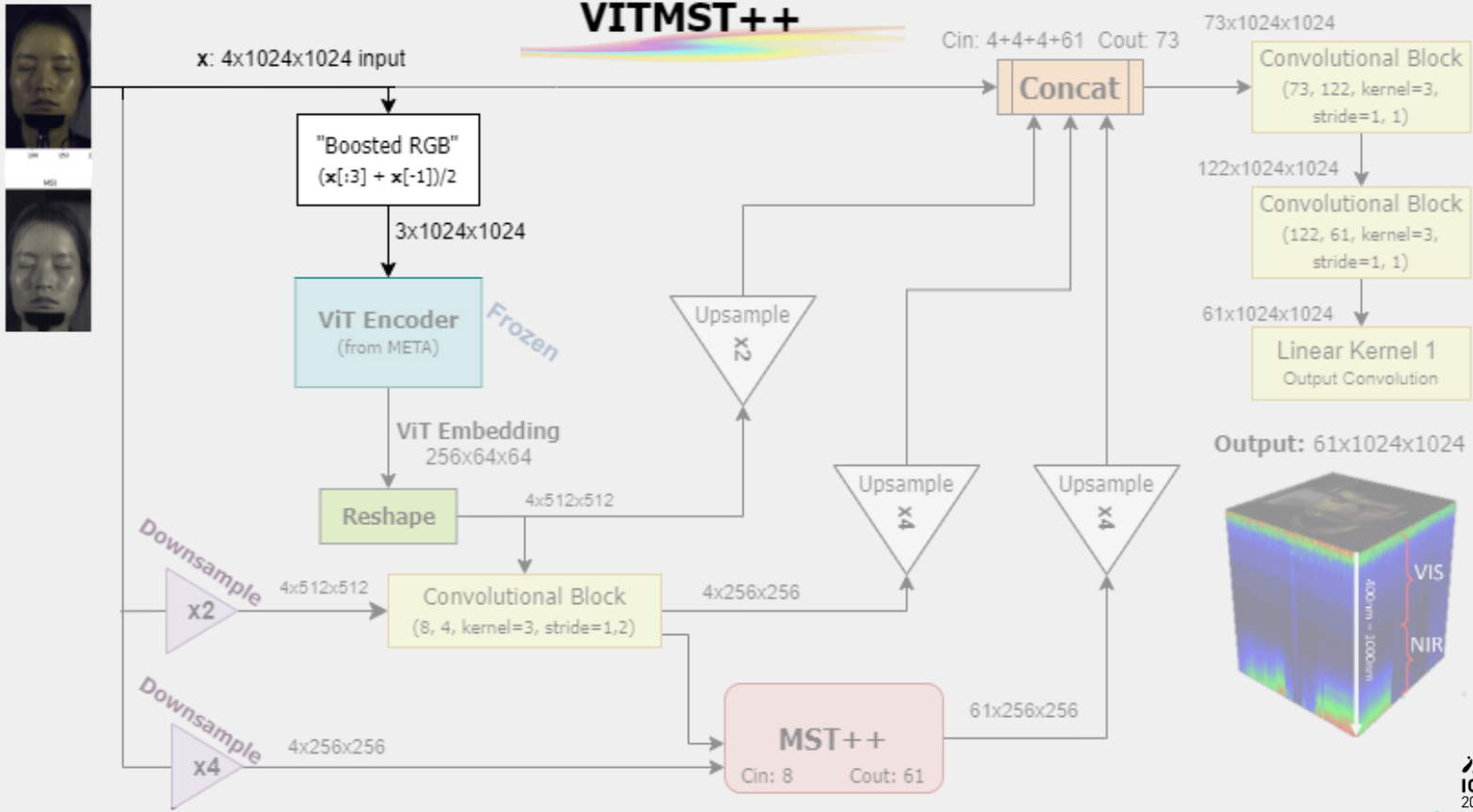
03 Strategies

Data volume



03 Strategies

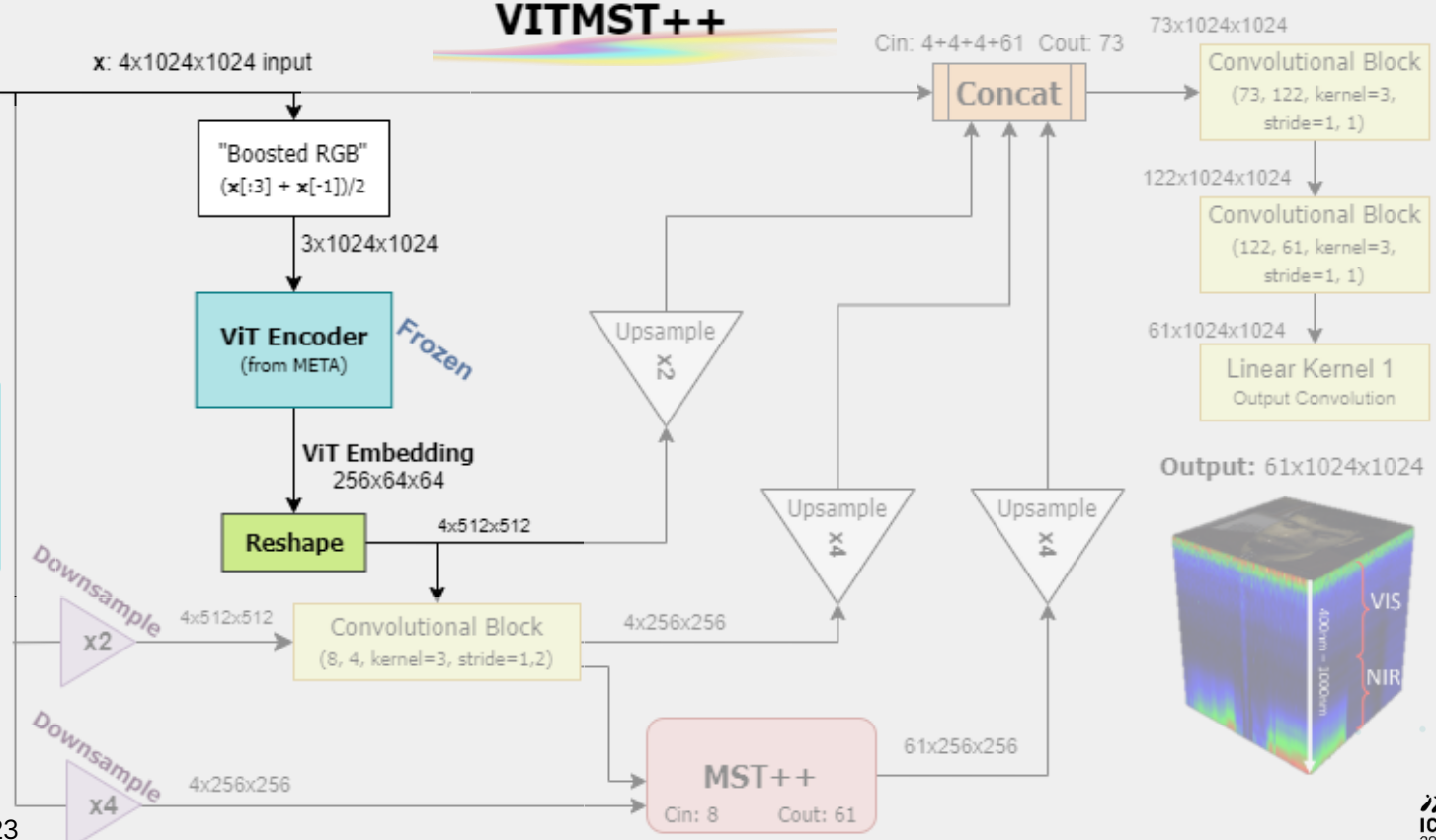
Data volume



03 Strategies

Data volume

Segment Anything Model [2]



[2] Kirillov et. al, 2023

03 Strategies

Data volume

VITMST++

x: 4x1024x1024 input

"Boosted RGB"
(x[:3] + x[-1])/2

3x1024x1024

ViT Encoder
(from META)

Frozen

ViT Embedding
256x64x64

Reshape

4x512x512

Upsample x2

Upsample x4

Upsample x4

Cin: 4+4+4+61 Cout: 73

Concat

73x1024x1024

Convolutional Block
(73, 122, kernel=3, stride=1, 1)

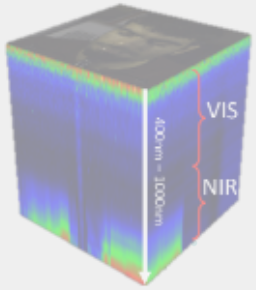
122x1024x1024

Convolutional Block
(122, 61, kernel=3, stride=1, 1)

61x1024x1024

Linear Kernel 1
Output Convolution

Output: 61x1024x1024



Downsample x2

4x512x512

Convolutional Block
(8, 4, kernel=3, stride=1,2)

4x256x256

Downsample x4

4x256x256

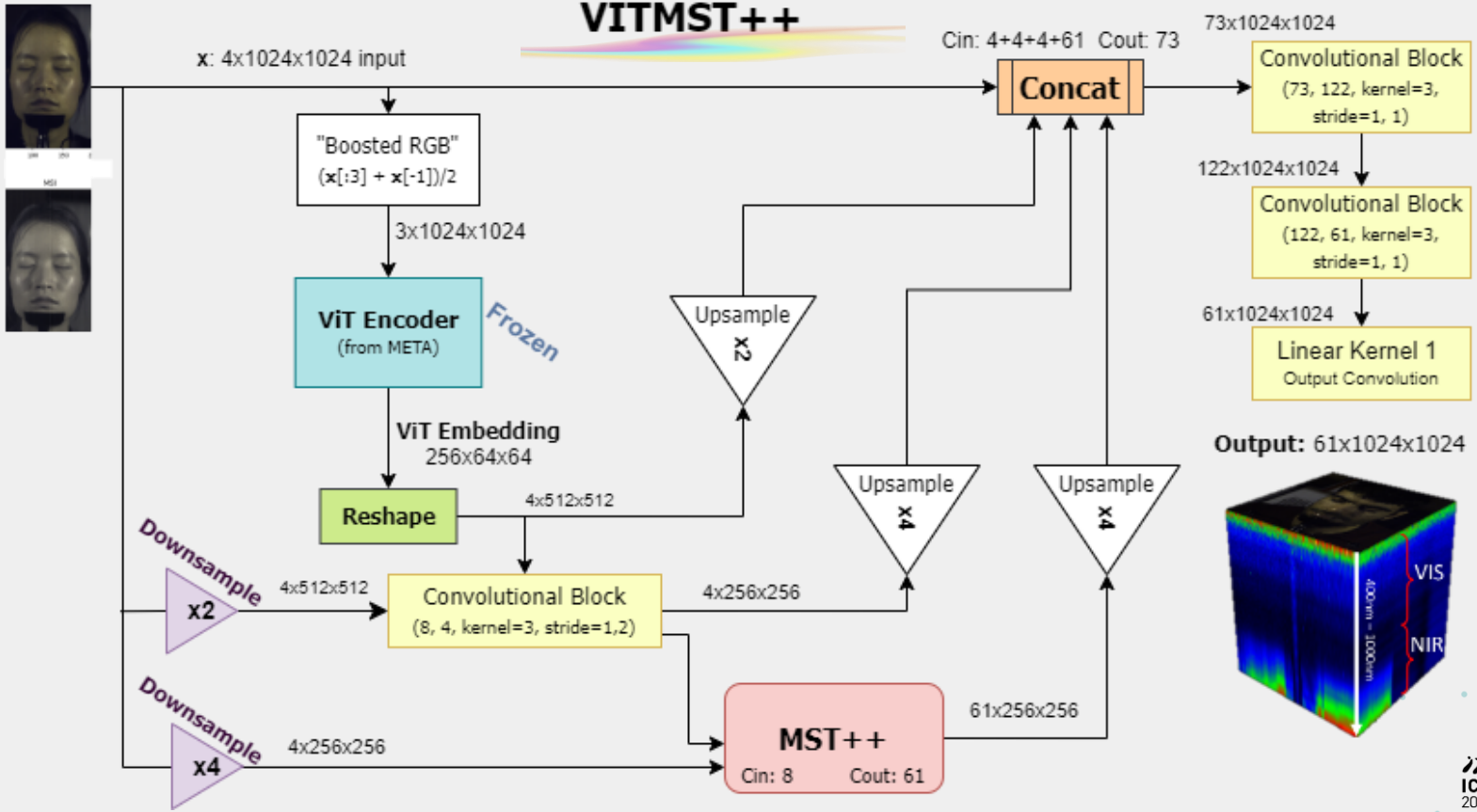
MST++
Cin: 8 Cout: 61

61x256x256

Multi resolution features

03 Strategies

Data volume



03 Strategies

Data volume and architecture

SAM: the lower, the better

Table 1. MST++ and VITMST++ quantitative results for reconstruction performance and training computational efficiency.

Model	Validation				Test	Training	
	Face SSIM	Image SSIM	Face SAM	Image SAM	ce SAM	VRAM	MACs
MST++	0.970	0.993	0.066	0.114	0. 2 ± 0.020	45 GB	576 G
VITMST++	0.974	0.921	0.057	0.101	± 0.031	8GB	130 G
VITMST++ w/ Custom Loss	0.976	0.925	0.053	0.088	0.074 ± 0.015	8GB	130 G

VITMST++ reduced SAM

03 Strategies

Data volume and architecture

SSIM: the higher, the better

Table 1. MST++ and VITMST++ quantitative results for reconstruction performance and training computational efficiency.

Model	Validation				Test	Training	
	Face SSIM	Image SSIM	e SAM	Image SAM	Face SAM	VRAM	MACs
MST++	0.970	0.993	066	0.114	0.118 ± 0.020	45 GB	576 G
VITMST++	0.974	0.921	057	0.101	0.091 ± 0.031	8GB	130 G
VITMST++ w/ Custom Loss	0.976	0.925	0.053	0.088	0.074 ± 0.015	8GB	130 G

VITMST++ increased face SSIM but decreased image SSIM

03 Strategies

Data volume and architecture

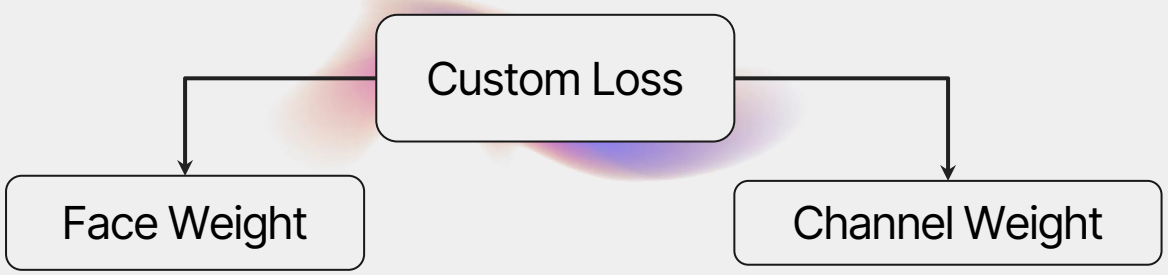
Table 1. MST++ and VITMST++ quantitative results for reconstruction performance and training computational efficiency.

Model	Validation				Test	Training	
	Face SSIM	Image SSIM	Face SAM	Image SAM	Face SAM	VRAM	MACs
MST++	0.970	0.993	0.066	0.114	0.118 ± 0.0	45 GB	576 G
VITMST++	0.974	0.921	0.057	0.101	0.091 ± 0.0	8GB	130 G
VITMST++ w/ Custom Loss	0.976	0.925	0.053	0.088	0.074 ± 0.01	8GB	130 G

VITMST++ reduced GPU memory use

03 Strategies

Custom loss



Focus on improving metrics on the face

Even out the channel interpolations

03 Strategies

Custom loss

Custom Loss

Face Weight

Channel Weight

Kirillov et.al, 2023

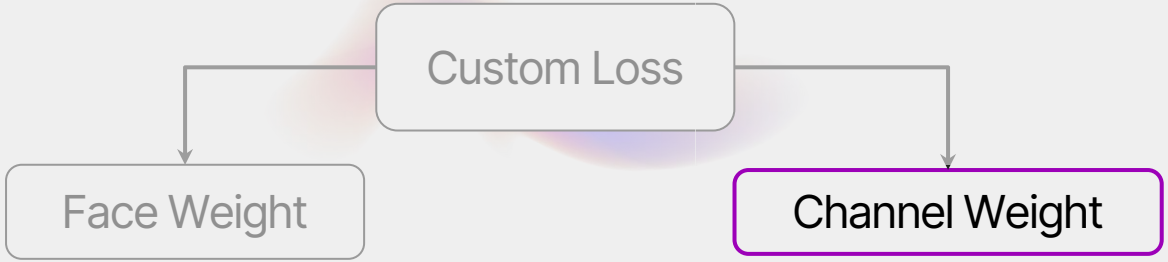
Segment Anything Model

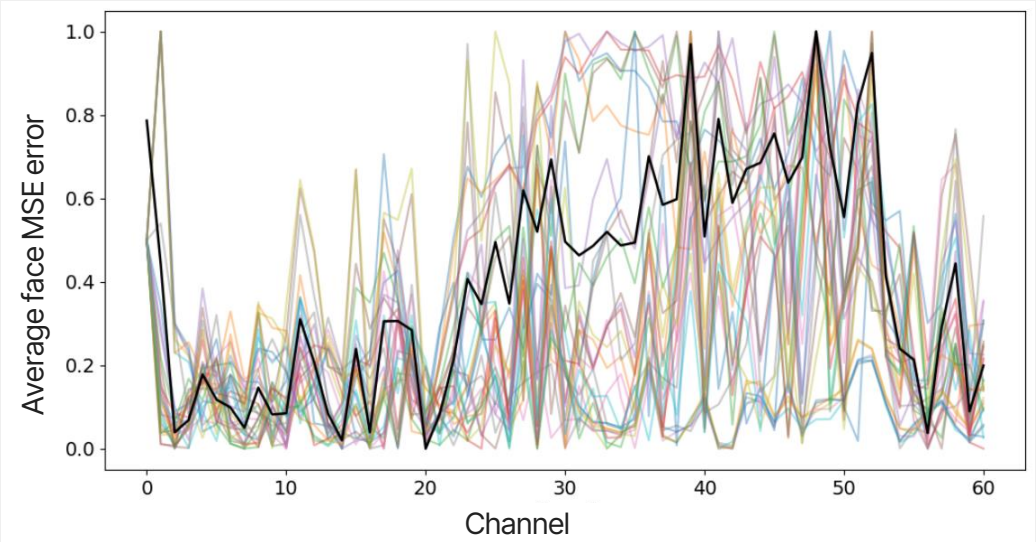
Segments any object given an RGB image and a prompt

Prompt: image center point

03 Strategies

Custom loss





In each validation step we calculate the average face error for each channel

Errors are normalized]0,1]

Metrics are calculated per channels and weighted by their respective normalized errors

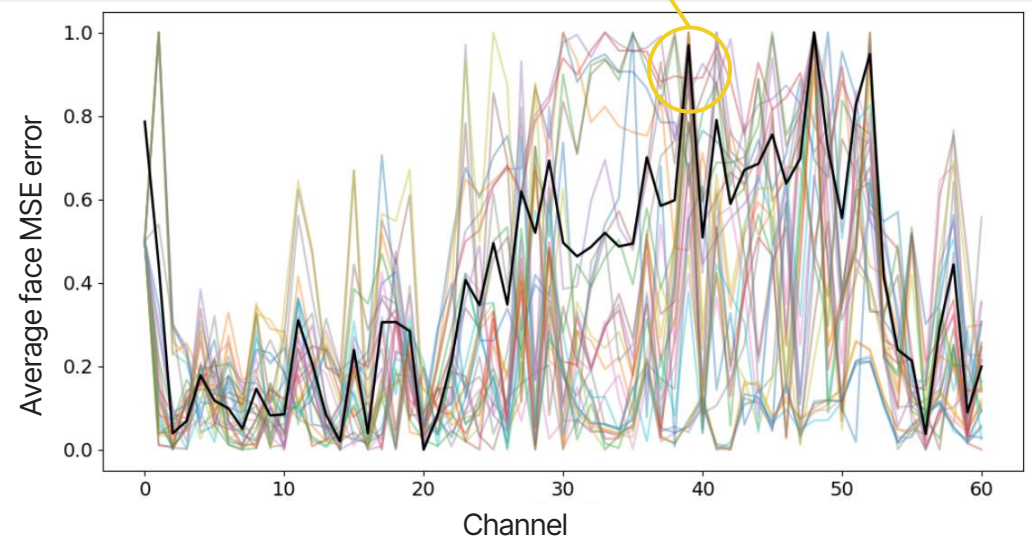
03 Strategies

Custom loss

Custom Loss

Channels with greatest errors will have greater influence in the loss

Channel Weight



In each validation step we calculate the average face error for each channel

Errors are normalized]0,1]

Metrics are calculated per channels and weighted by their respective normalized errors

03 Strategies

Custom loss

Table 1. MST++ and VITMST++ quantitative results for reconstruction performance and training computational efficiency.

Model	Validation				Test	Training	
	Face SSIM	Image SSIM	Face SAM	Image SAM	Face SAM	VRAM	MACs
MST++	0.970	0.993	0.066	0.114	0 ± 0.020	45 GB	576 G
VITMST++	0.974	0.921	0.057	0.101	0 ± 0.031	8GB	130 G
VITMST++ w/ Custom Loss	0.976	0.925	0.053	0.088	± 0.015	8GB	130 G

Custom Loss reduced SAM

03 Strategies

Custom loss

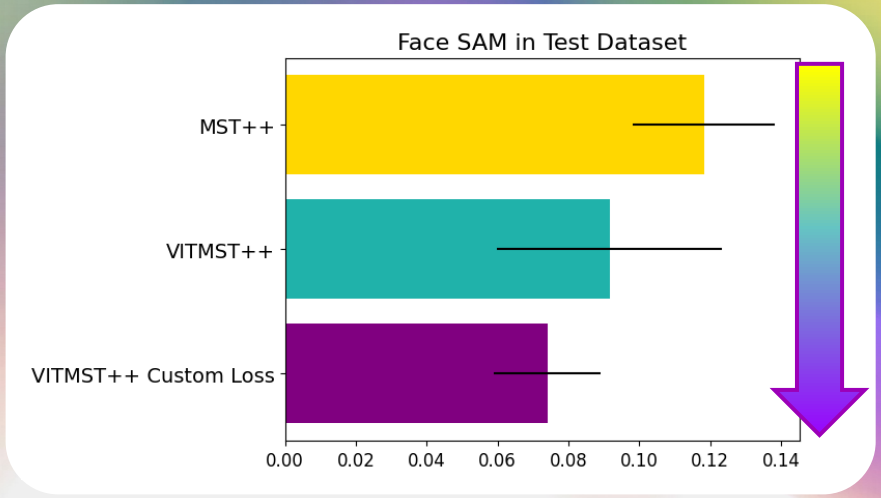
Table 1. MST++ and VITMST++ quantitative results for reconstruction performance and training computational efficiency.

Model	Validation				Test	Training	
	Face SSIM	Image SSIM	Face SAM	Image SAM	Face SAM	VRAM	MACs
MST++	0.970	0.993	0.066	0.114	0.118 ± 0.020	45 GB	576 G
VITMST++	0.974	0.921	0.057	0.101	0.091 ± 0.031	8GB	130 G
VITMST++ w/ Custom Loss	0.976	0.925	0.053	0.088	0.074 ± 0.015	8GB	130 G

Custom Loss improved SSIM

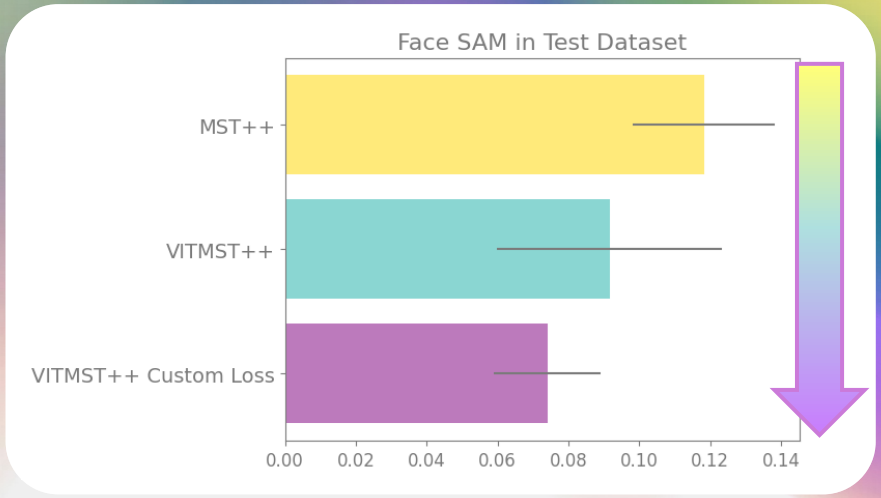
04 Final Results

Model	Test Face SAM
MST++	0.118 ± 0.020
VITMST++	0.091 ± 0.031
VITMST++ w/ Custom Loss	0.074 ± 0.015



04 Final Results

Model	Test Face SAM
MST++	0.118 ± 0.020
VITMST++	0.091 ± 0.031
VITMST++ w/ Custom Loss	0.074 ± 0.015



Custom Loss VITMST++ performs better than MST++ while consuming less than 12 GB of GPU memory in training

05 Future Improvements

- Domain adaptation strategies to improve results on different cameras
- Train for more epochs
- Work on SAM face/background contrastive loss

Thank you!

Funding support:

ELDORADO
INSPIRAÇÃO PELO NOVO

Lucas Hideki Ueda

Diedre Do Carmo

Tárik Ponte e Sá

Felipe Hiroshi Kano Inazumi

Eduardo Rittner Coelho

Guilherme Gelmi de Freitas Salvo

Paula Costa

Denis Gustavo Fantinato

Letícia Rittner

Any questions?

Check out our paper!

Ana Clara Caznok Silveira

ana caznok

ana.clara.caznok@gmail.com