Real-Time High-Resolution Background Matting

2022/06/08
Masatoshi Tateno
@Sato Lab M1

Background Replacement

Movie

https://www.famefocus.com/video/amazing-before-after-hollywood-vfx-fantastic-beasts-and-where-to-find-them/

https://www.studiobinder.com/blog/what-is-a-green-screen-video/

Video Conference

Ref. [1] S.Lin et al.

Ref. [1] S.Lin et al.

Background Matting

Basic Process

Source image

×

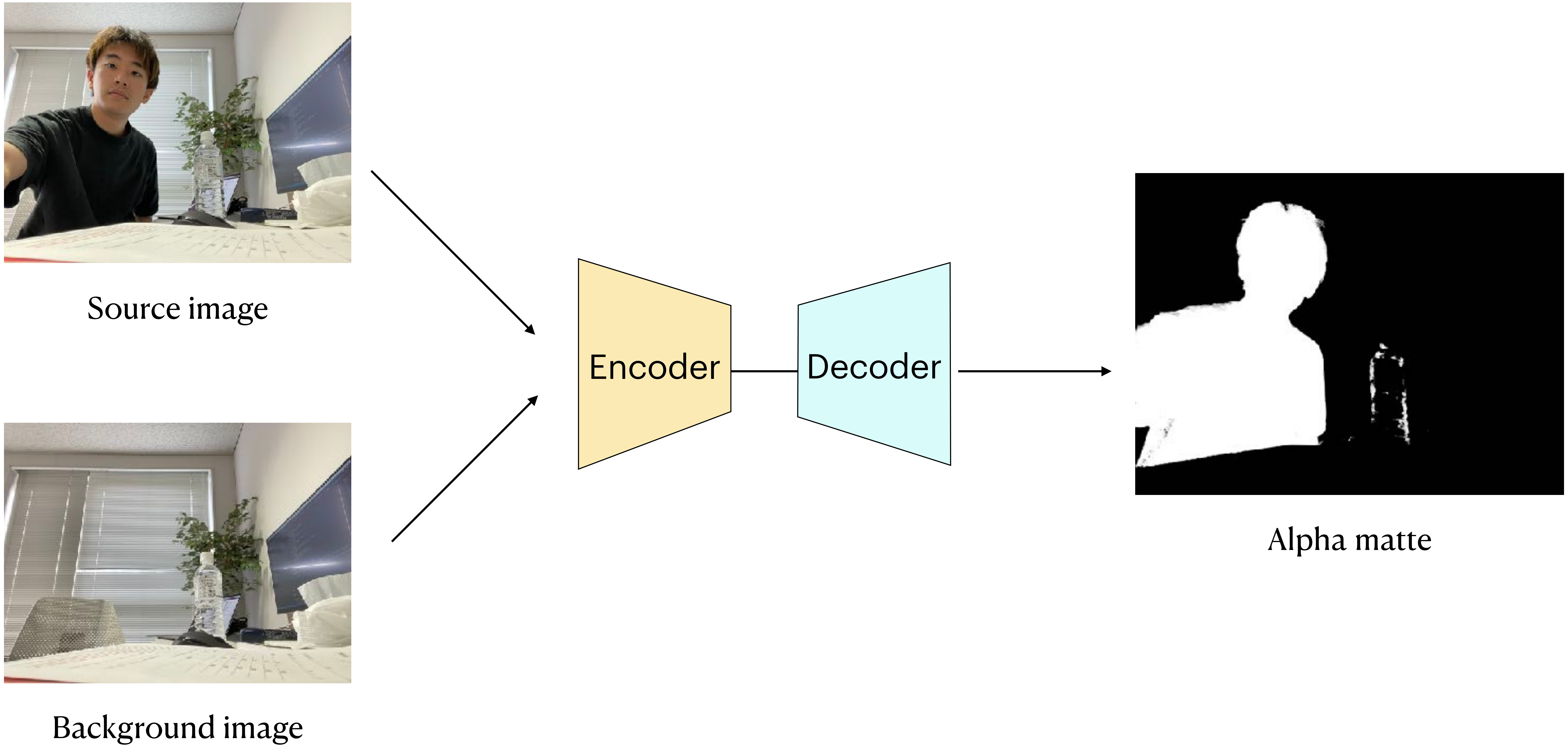
Alpha matte

=

Foreground image

Creating Alpha matte = **Alpha Matting**

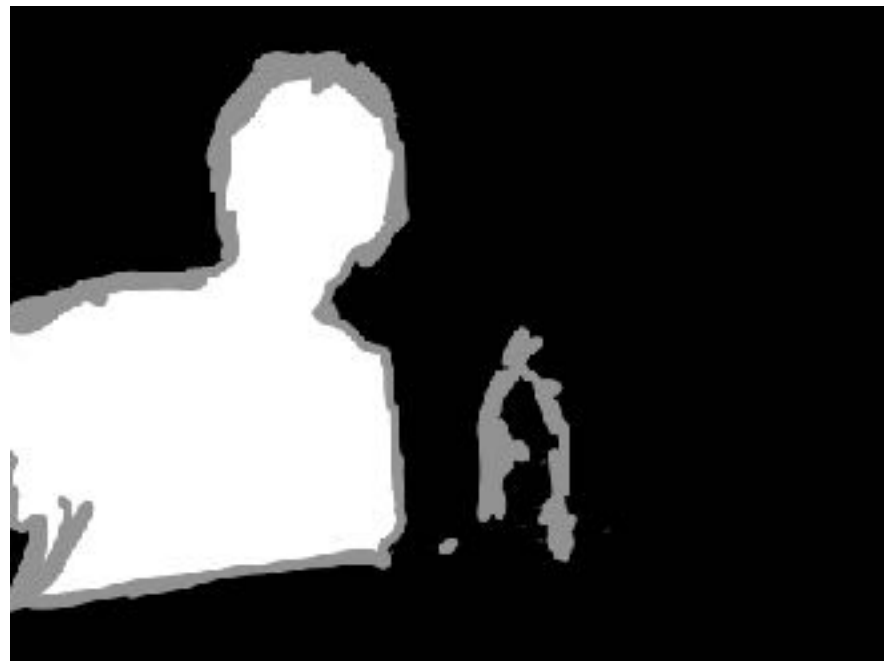
Matting with a known background



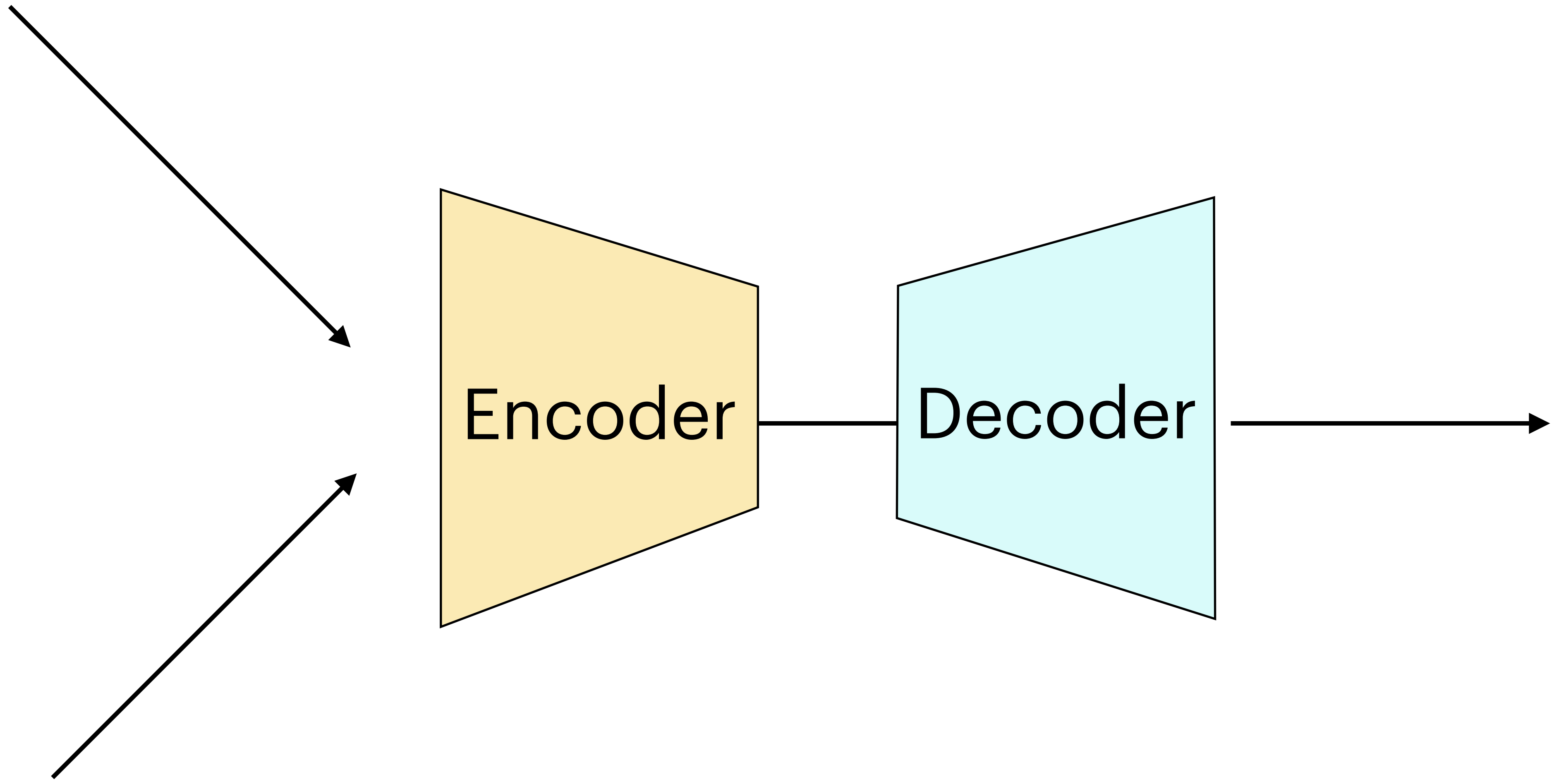
Related Work

Trimap-based matting

Source image



Trimap (manual annotation)



Alpha matte

❌ Manual annotation is needed

❌ Performance depends on the quality of Trimap

Related Work

Matting w/o any external input

Source image

Model that
separate fgr&bgr

Alpha matte

❌ often fail to generalize

Approach

Given

I : image

B : background image

composited image I' over new background B'

Predict

$$I' = \alpha F + (1 - \alpha)B'$$

α : alpha matte

F : foreground

⚠️ Actually, **not** predict foreground directly

But, predict foreground residual $F^R (= F - I)$

F can be recovered by

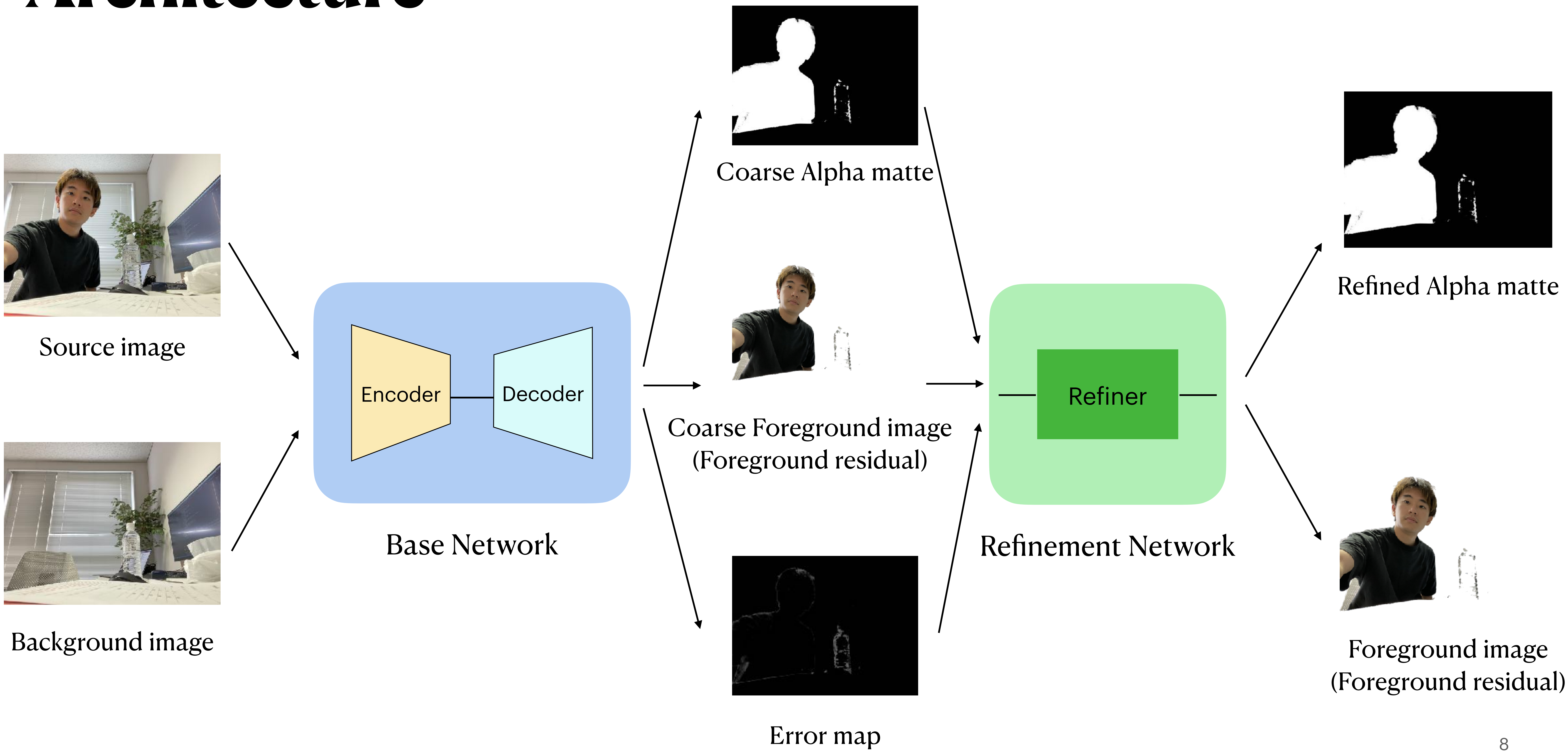
$$F = \max(\min(F^R + I), 0)$$

⇨

✅ Improves learning

✅ Allows us to apply low-res F^R on to high-res I

Architecture



Architecture

Ref. [1] S.Lin et al.

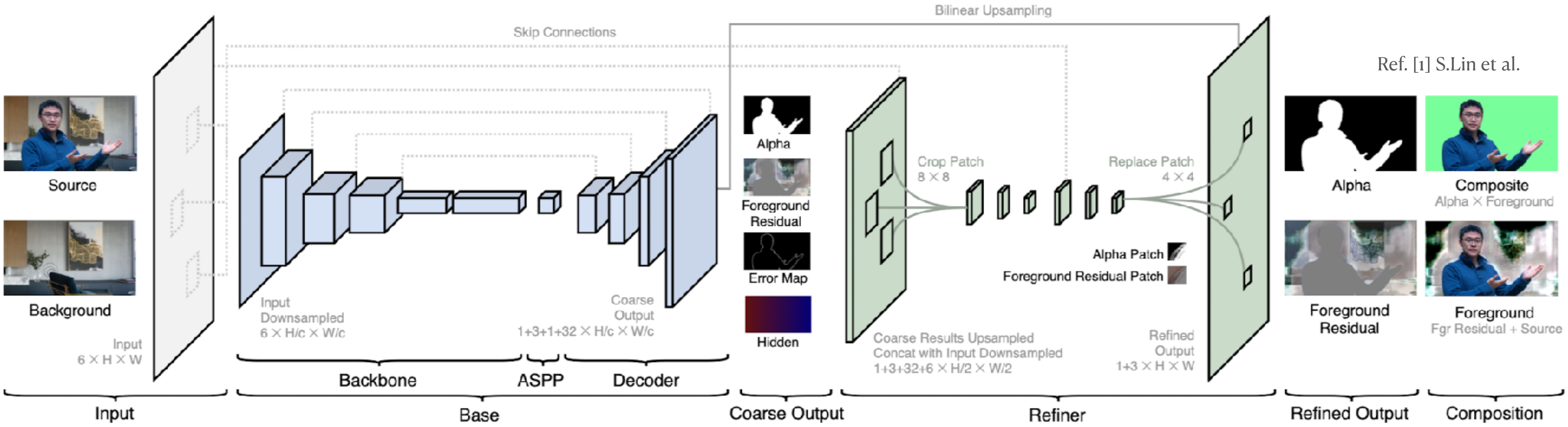
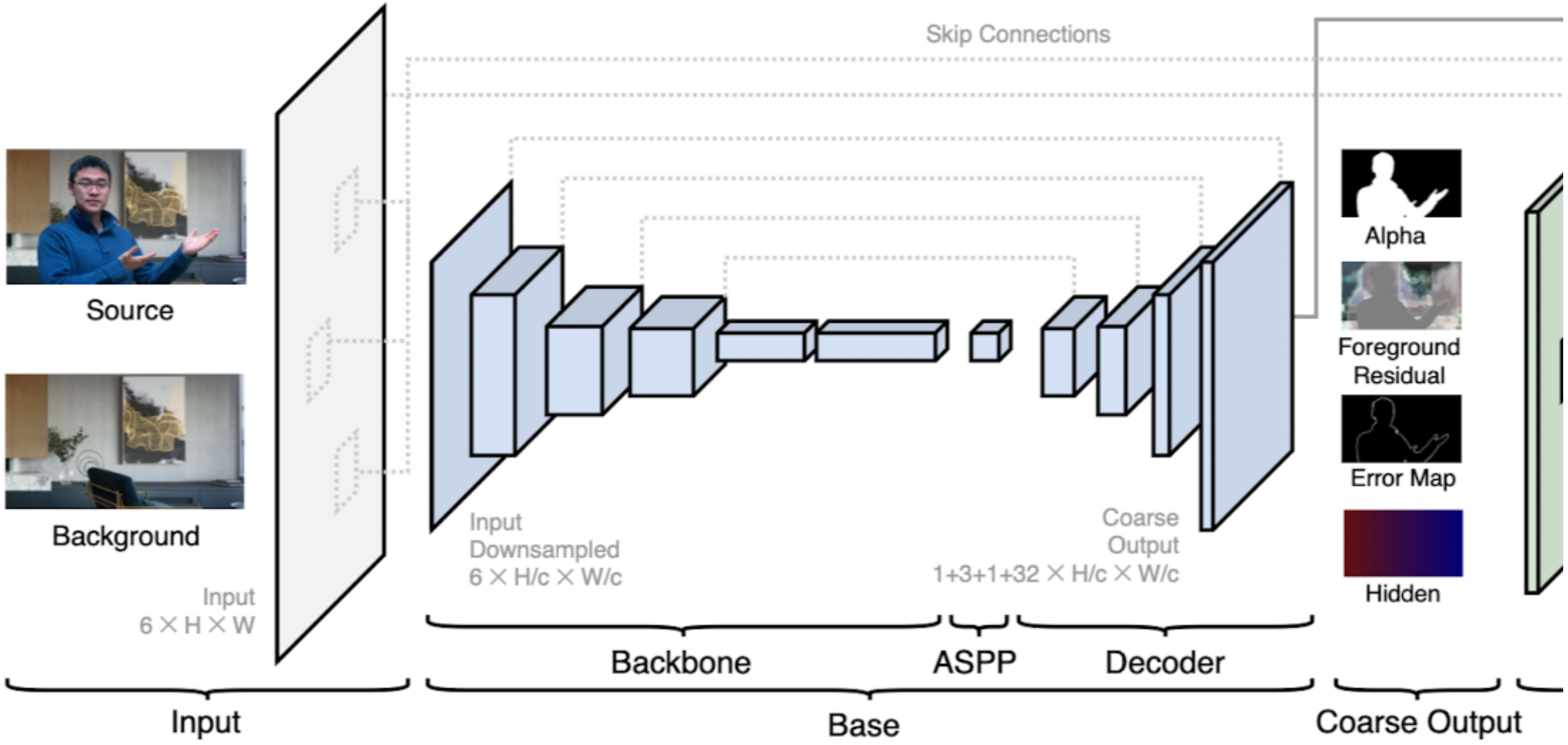


Figure 3: The base network G_{base} (blue) operates on the downsampled input to produce coarse-grained results and an error prediction map. The refinement network G_{refine} (green) selects error-prone patches and refines them to the full resolution.

Base Network

Ref. [1] S.Lin et al.

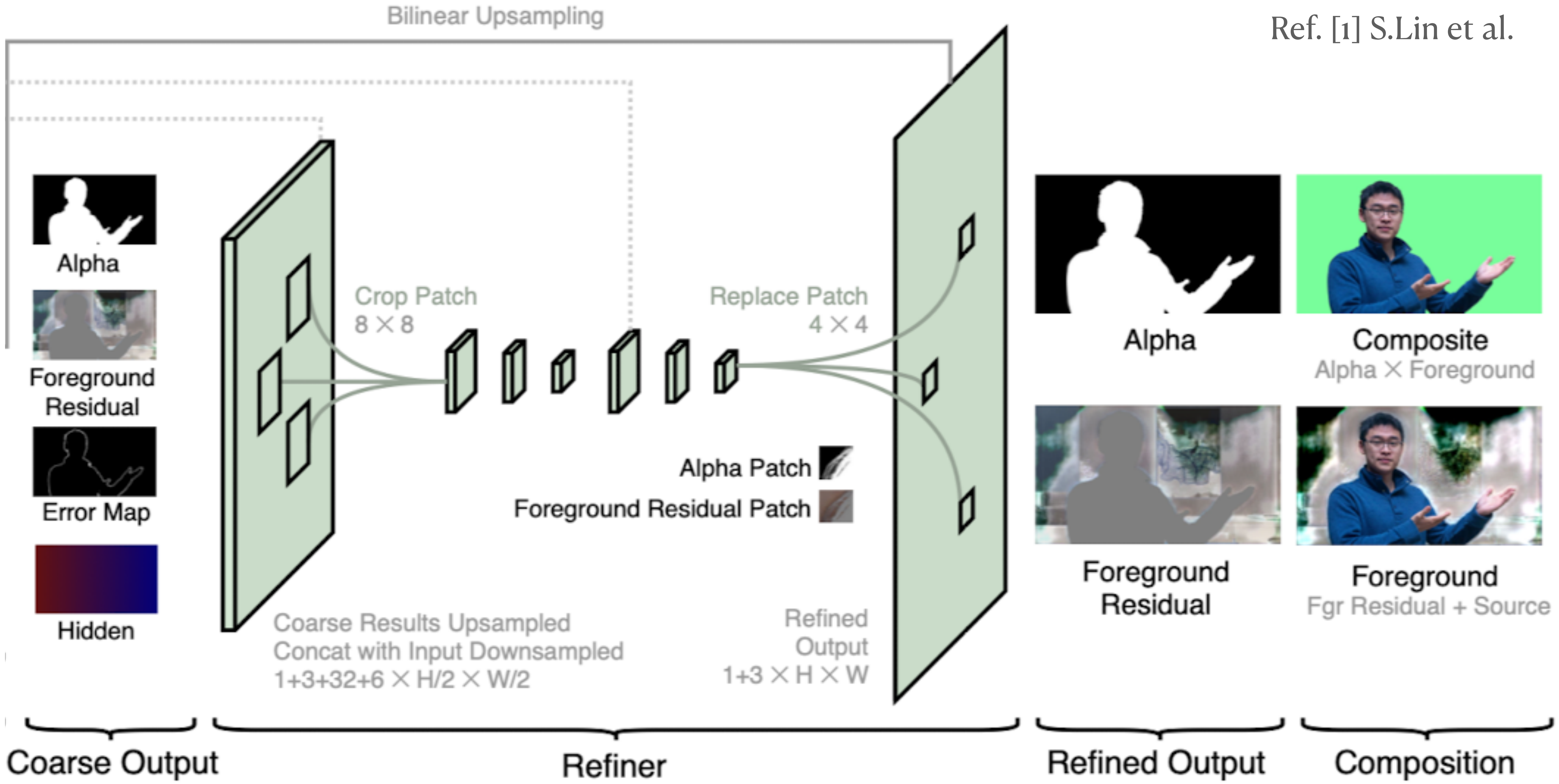


Backbone : **ResNet50** (ResNet101, MobileNetV2)

ASPP : **A**trous **S**patial **P**yramid **P**ooling module from DeepLabV3

Decoder : **skip connection, bilinear upsampling, 3*3 convolution, BN, ReLU**

Refinement Network



Ref. [1] S.Lin et al.

1. Choose k patches to refine using Error Map
2. Refine patches by upsampling, 3*3 convolution, BN, ReLU
3. Swap in the respective patches that have been refined

Datasets

Introducing 2 new datasets

VideoMatte240K

- 484 high-res videos
- total of 40,709 unique frames of alpha mattes and foregrounds
- 384 videos are 4K, 100 are in HD

PhotoMatte13K/85

- collection of 13,665 images
- averaging resolution is around 2000 * 2500
- For privacy and licensing issue, only 85 mattes of similar quality is publicly released

(a) VideoMatte240K

(b) PhotoMatte13K/85

Training

Augmentation

	Affine Trans	Horizontal Flipping	Brightness	Hue & Saturation	Blurring & Sharpening	Random Noise	Additional Noise & Jitter & Affine Trans	Shadow Effect
Foreground	○	○	○	○	○	○		
Background	○	○	○	○	○	○	○	○

Training

Loss

$$\mathcal{L}_{base} = \mathcal{L}_{\alpha} + \mathcal{L}_{F_c} + \mathcal{L}_{E_c}$$

$$\mathcal{L}_{refine} = \mathcal{L}_{\alpha} + \mathcal{L}_{F}$$

$$\mathcal{L}_{\alpha} = \|\alpha - \alpha^*\|_1 + \|\nabla\alpha - \nabla\alpha^*\|_1$$

(α^* : ground-truth.
L1 loss over the whole alpha matte and its Sobel gradient)

$$\mathcal{L}_{F} = \|(\alpha^* > 0) * (F - F^*)\|_1$$

(L1 loss only on the foreground pixels where $\alpha^* > 0$)

$$\mathcal{L}_{E} = \|E - E^*\|_2, \; E^* = |\alpha - \alpha^*|$$

(mean squared of error map)

Training

Training order

1. Train only the base network with VideoMatte240K
2. Train entire model jointly on VideoMatte240K
3. Train entire model jointly on PhotoMatte13K
4. Train entire model jointly on Distinctions-646 (dataset form Ref.[4] Y. Qiao)

I skipped 3,4 because of datasets' unavailability

Implementation

I implemented :

- Background Matting architecture from scratch with PyTorch
- Code for training (trained 1 epoch)
  - training base network
  - training refinement network
- Code for testing image background matting

I referred to :

- Official Implementation with PyTorch
- ResNet Implementation from scratch
- Pretrained weights of ASPP module from DeepLabV3

Quality Analysis

Speed Analysis

Image matting time : 0.0388 sec/image $\simeq$ 25.78 fps

- GPU : NVIDIA Tesla V100 32GB
- Image : 3840 * 2160 (4K)
- Average over 35 images

c.f. original speed measurement Ref. [1] S.Lin et al.

Method	Backbone	Resolution	FPS	GMac
FBA		HD	3.3	54.3
FBA_{auto}		HD	2.9	137.6
BGM		512^2	7.8	473.8
Ours	ResNet-50*	HD	60.0	34.3
	ResNet-101	HD	42.5	44.0
	MobileNetV2	HD	100.6	9.9
Ours	ResNet-50*	4K	33.2	41.5
	ResNet-101	4K	29.8	51.2
	MobileNetV2	4K	45.4	17.0

Table 3: Speed measured on Nvidia RTX 2080 TI as PyTorch model pass-through without data transferring at FP32 precision and with batch size 1. GMac does not account for interpolation and cropping operations. For the ease of measurement, BGM and FBA_{auto} use adapted PyTorch DeepLabV3+ implementation with ResNet101 backbone as segmentation.

Pull Requests

- Code for test video background matting
  - stock video footage
  - webcam
- Revise test_image.py with tricks to speed up inference time
- Create additional Backbone Network options (ResNet101, MobileNetV2)
- Homographic Alignment (see original implementation)

Reference

[1] S. Lin, A. Ryabtsev, S. Sengupta, B. Curless, S. Seitz, I. Kemelmacher-Shlizerman, “Real-time hith-resolution background matting,” CVPR, pp.8762-8771, 2021

[2] L.C. Chen, G. Papandreou, F. Schroff, H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arxiv:1706.05587, 2017

[3] L.C. Chen, G. Papandreou, I. Kokkinos, k. Murphy, A.L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE transactions on pattern analysis and machine intelligtnece, vol.40, no.4, pp834-848, Apr. 2018

[4] Y. Qiao, Y. Liu, X. Yang, D. Zhou, M. Xu, Q. Zhang, X. Wei, “Attention-guided hierarchical structure aggregation for image matting,” CVPR, pp.13676-13685, 2020

Appendix

ASPP : Atrous Spatial Pyramid Pooling

ASPP module consists of **multiple dilated convolution filters** with different dilation rates.

"it is effective to resample features at different scales for accurately and efficiently classifying regions of an arbitrary scale." Ref.[2] L.C.Chen

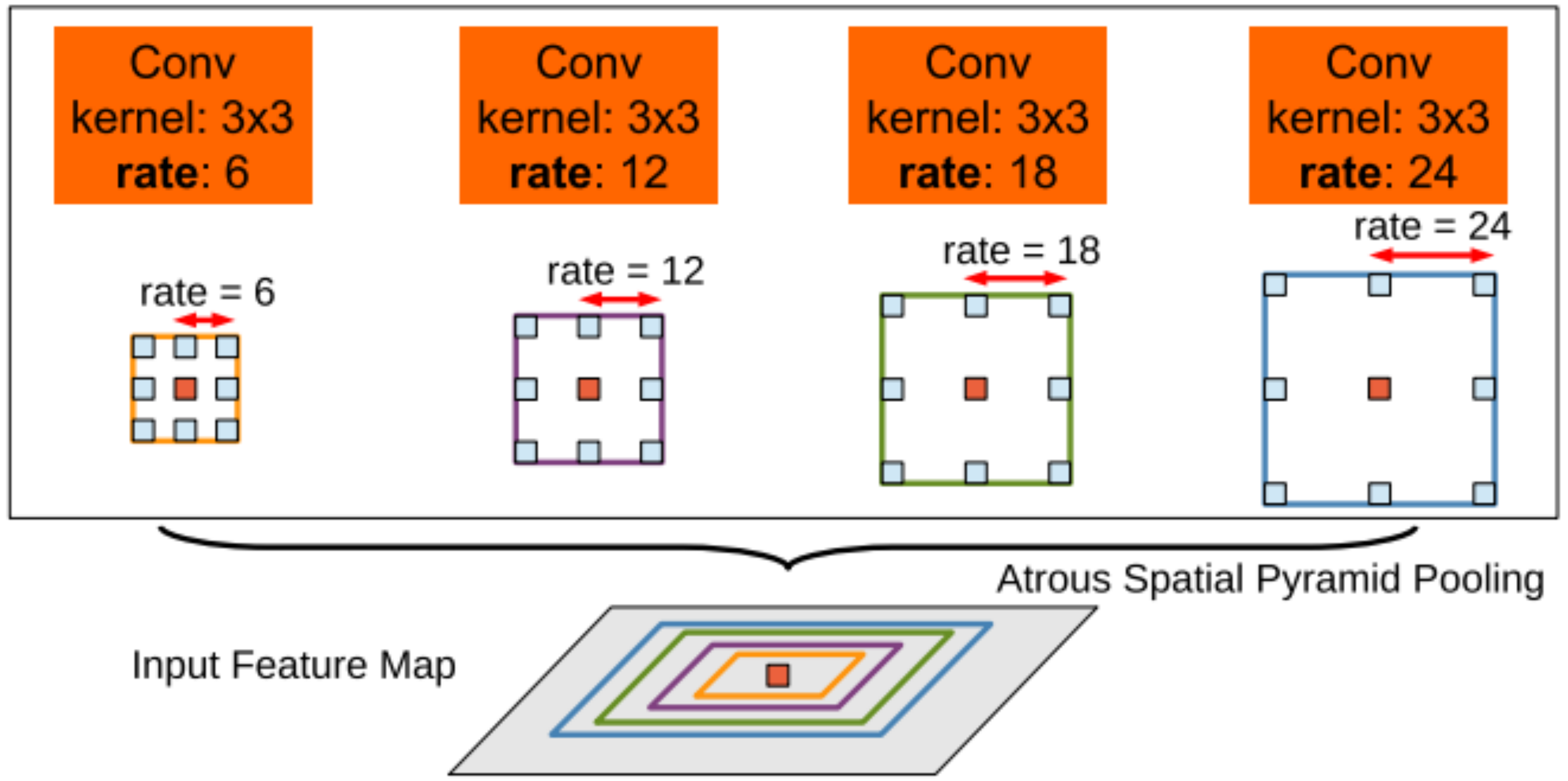


Fig. 4. Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

Ref.[3] L.C.Chen