Deep Image Matting

韩坤洋

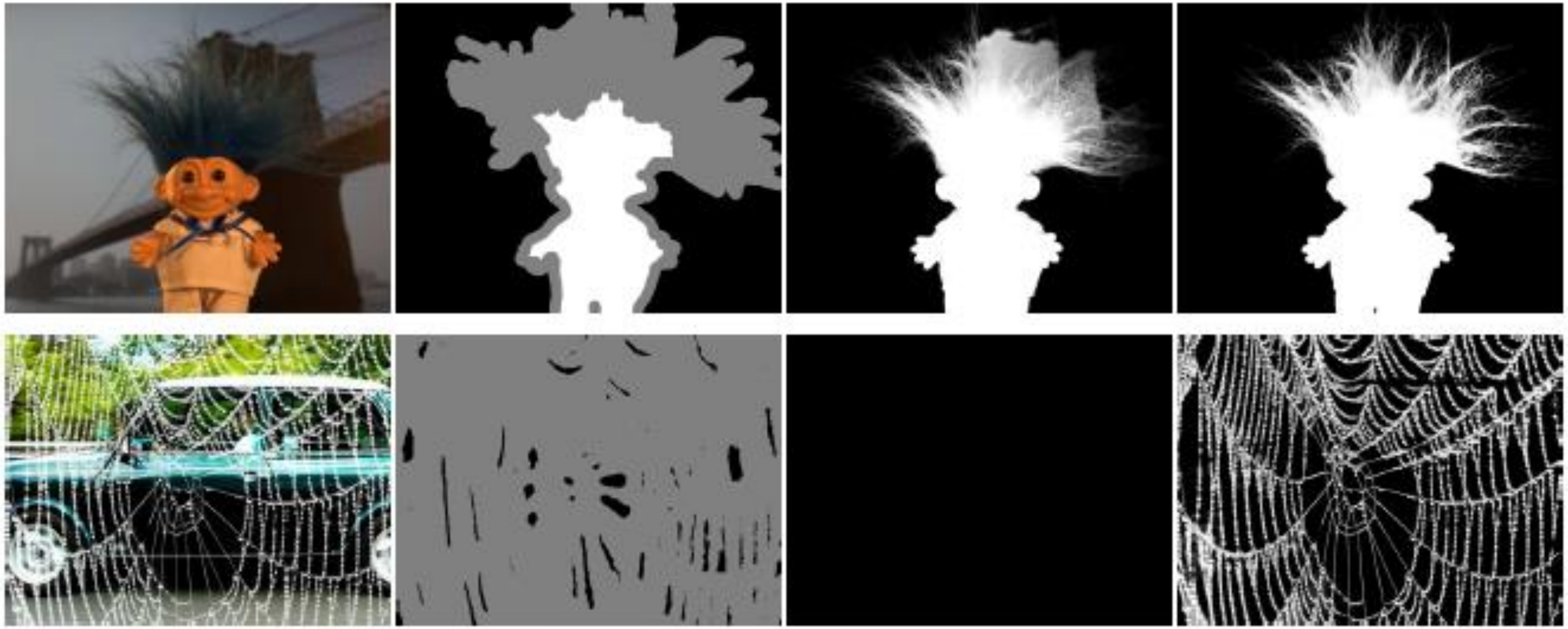


Image Trimap Closed-form Ours

1. Deep Image Matting CVPR 2017
2. Natural Image Matting via Guided Contextual Attention AAAI 2020
3. Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation ICCV 2019
4. F, B, Alpha Matting Arxiv
5. Background Matting: The World is Your Green Screen CVPR 2020
6. Real-Time High-Resolution Background Matting Arxiv

Deep Image Matting

University of Illinois at Urbana-Champaign
Adobe Research

CVPR 2017

Deep Image Matting

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad \alpha_i \in [0, 1].$$

- Current methods are designed to solve the matting equation
- Very small dataset
  - 27 training images and 8 test images

New matting dataset

- 1) Find images on simple or plain backgrounds, create alpha matte
- 2) Randomly sample N background images in MS COCO and Pascal VOC
- Training set,
  - 493 unique foreground objects and 49,300 images
- Testing set,
  - 50 unique objects and 1000 images
- Trimap
  - randomly dilated

Matting encoder-decoder stage

VGG-16

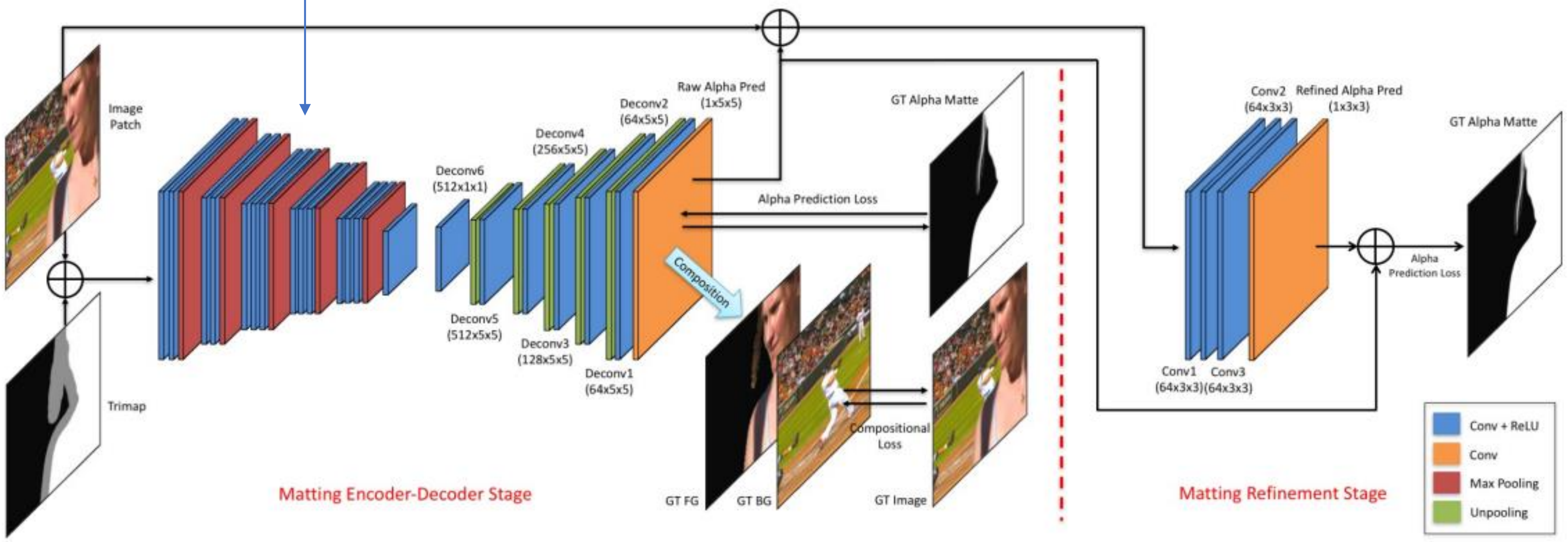
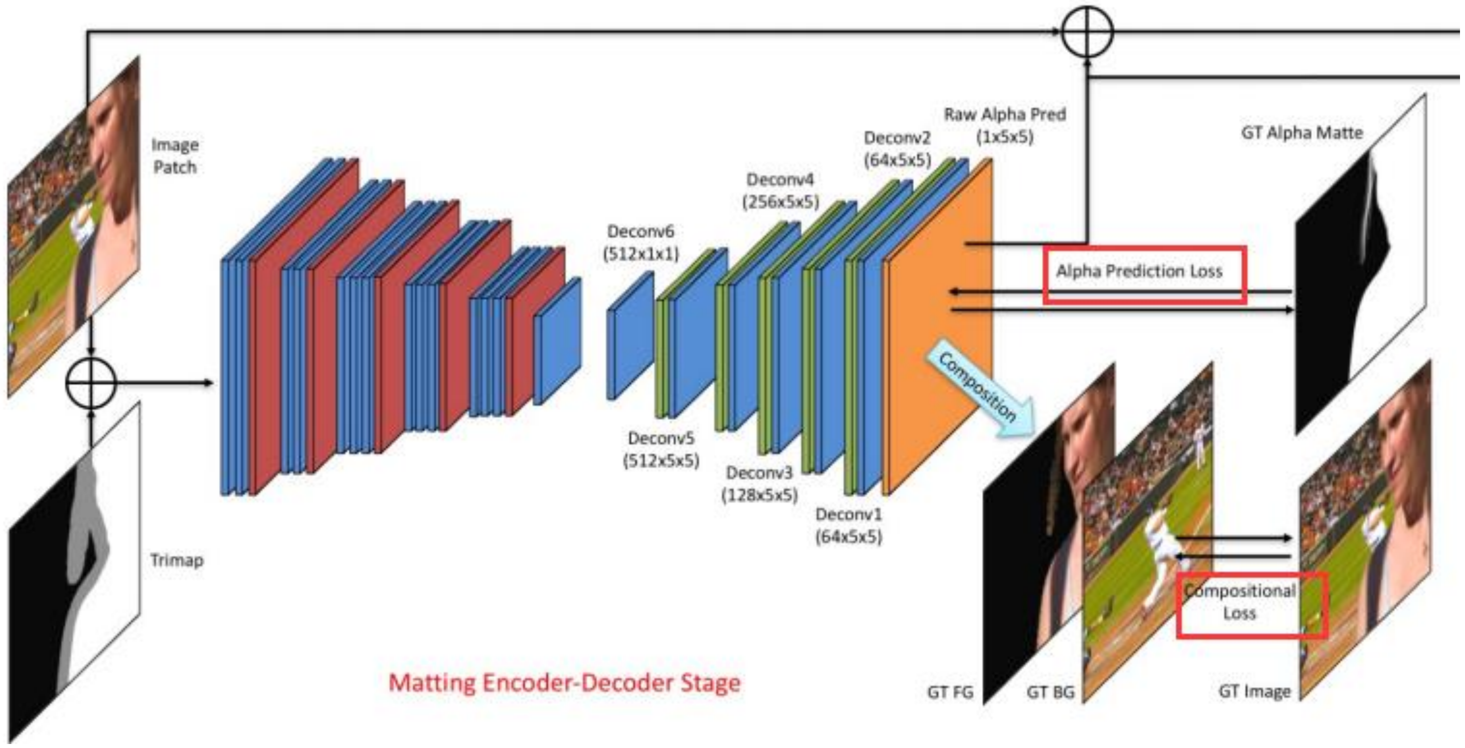


Figure 3. Our network consists of two stages, an encoder-decoder stage (Sec. 4.1) and a refinement stage (Sec. 4.2)

Matting encoder-decoder stage



Alpha Prediction Loss

$$\mathcal{L}_{\alpha}^{i} = \sqrt{(\alpha_p^i - \alpha_g^i)^2 + \epsilon^2}, \quad \alpha_p^i, \alpha_g^i \in [0, 1].$$

Compositional Loss

C_p = FG * α + BG * (1 - α)

$$\mathcal{L}_{c}^{i} = \sqrt{(c_p^i - c_g^i)^2 + \epsilon^2}.$$

Matting refinement stage

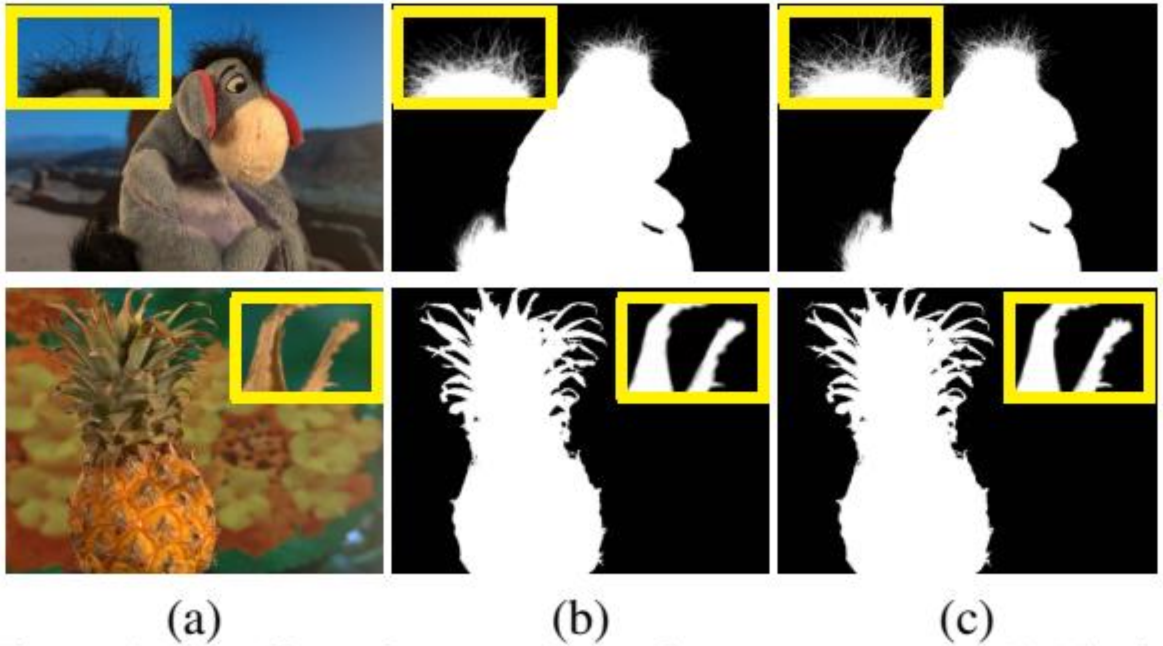
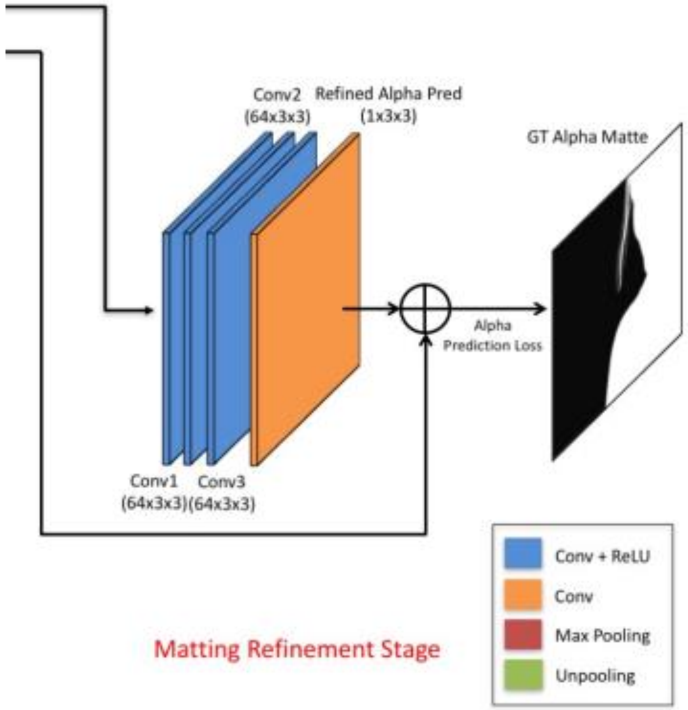


Figure 4. The effect of our matting refinement network. (a) The input images. (b) The results of our matting encoder-decoder stage. (c) The results of our matting refinement stage.

Input Image Concat Alpha Pred

Alpha Pred



Experimental results

Table 1. The quantitative results on the Composition-1k testing dataset. The variants of our approaches are emphasized in italic. The best results are emphasized in bold.

Methods	SAD	MSE	Gradient	Connectivity
Shared Matting [13]	128.9	0.091	126.5	135.3
Learning Based Matting [34]	113.9	0.048	91.6	122.2
Comprehensive Sampling [28]	143.8	0.071	102.2	142.7
Global Matting [16]	133.6	0.068	97.6	133.3
Closed-Form Matting [22]	168.1	0.091	126.9	167.9
KNN Matting [5]	175.4	0.103	124.1	176.4
DCNN Matting [8]	161.4	0.087	115.1	161.9
Encoder-Decoder network (single alpha prediction loss)	59.6	0.019	40.5	59.3
Encoder-Decoder network	54.6	0.017	36.7	55.3
Encoder-Decoder network + Guided filter[17]	52.2	0.016	**30.0**	52.6
Encoder-Decoder network + Refinement network	**50.4**	**0.014**	31.0	**50.8**

SAD(Sum of Absolute Differences)
MSE(Mean Squared Error)
Gradient

$$\sum_i (\nabla \alpha_i - \nabla \alpha_i^*)^q$$

Connectivity

$$\sum (\varphi(\alpha_i, \Omega) - \varphi(\alpha_i^*, \Omega))^p$$

$$\varphi(\alpha_i, \Omega) = 1 - (\lambda_i \cdot \delta(d_i \geq \theta) \cdot d_i)$$

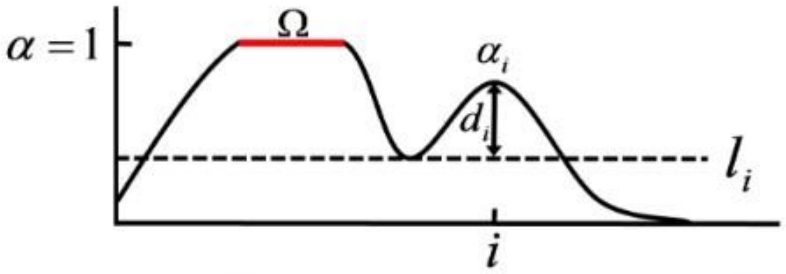


Figure 4. **Connectivity error.** See explanation in the text.

Natural Image Matting via Guided Contextual Attention

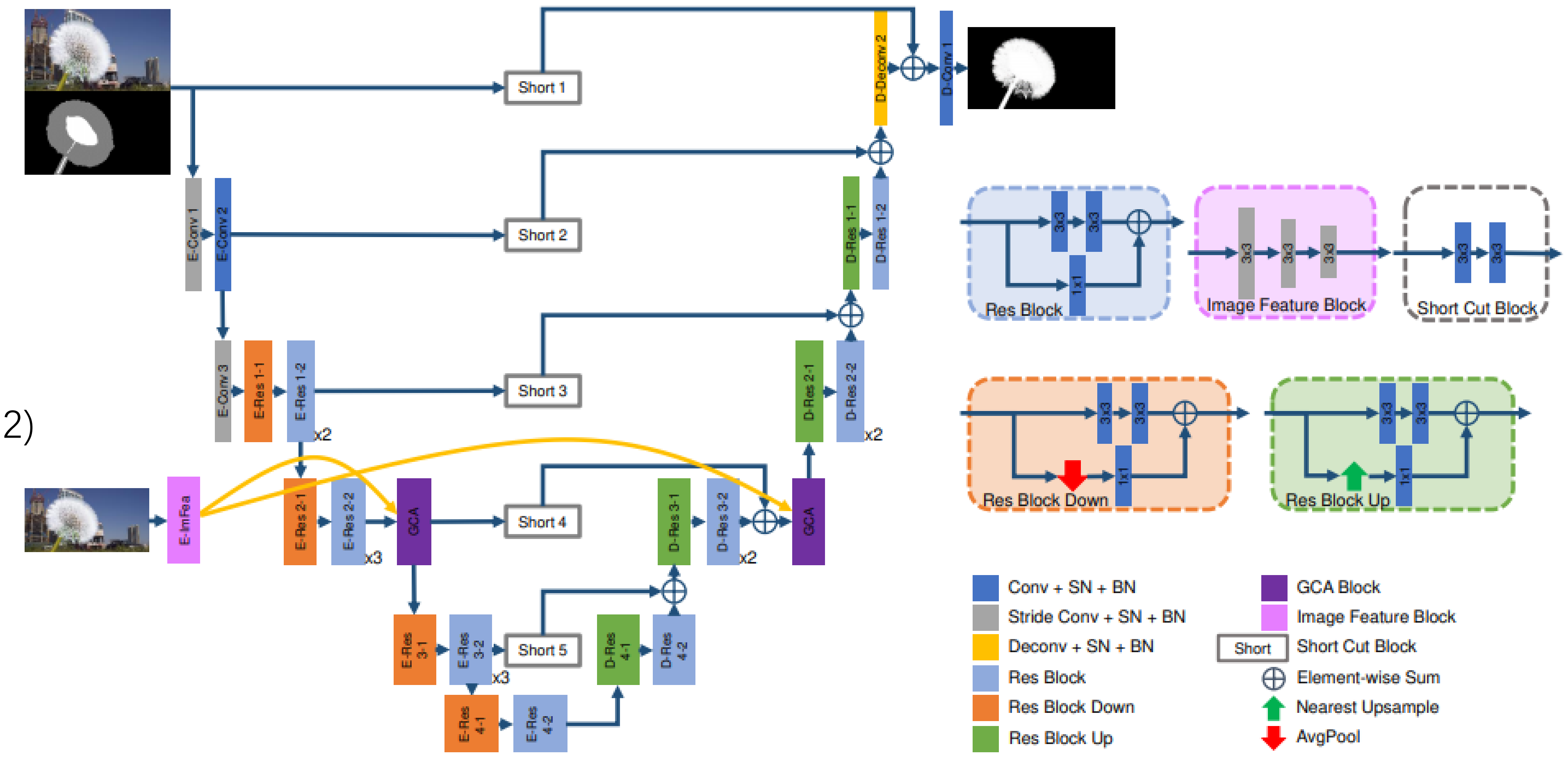
SJTU

AAAI 2020

Motivation

- Affinity-based and sampling-based algorithms
  - Need both FG and BG information to estimate the alpha matte
  - Only background and unknown areas in the trimap
- Learning-based image matting methods
  - SampleNet, deep inpainting methods, combination
- Propose a novel image matting method
  - based on the **opacity propagation** in a neural network
  - We devise a **guided contextual attention module**, mimic the affinity-based propagation

Baseline Structure

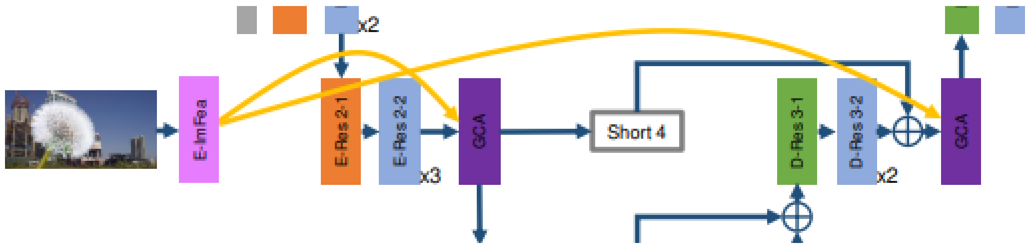


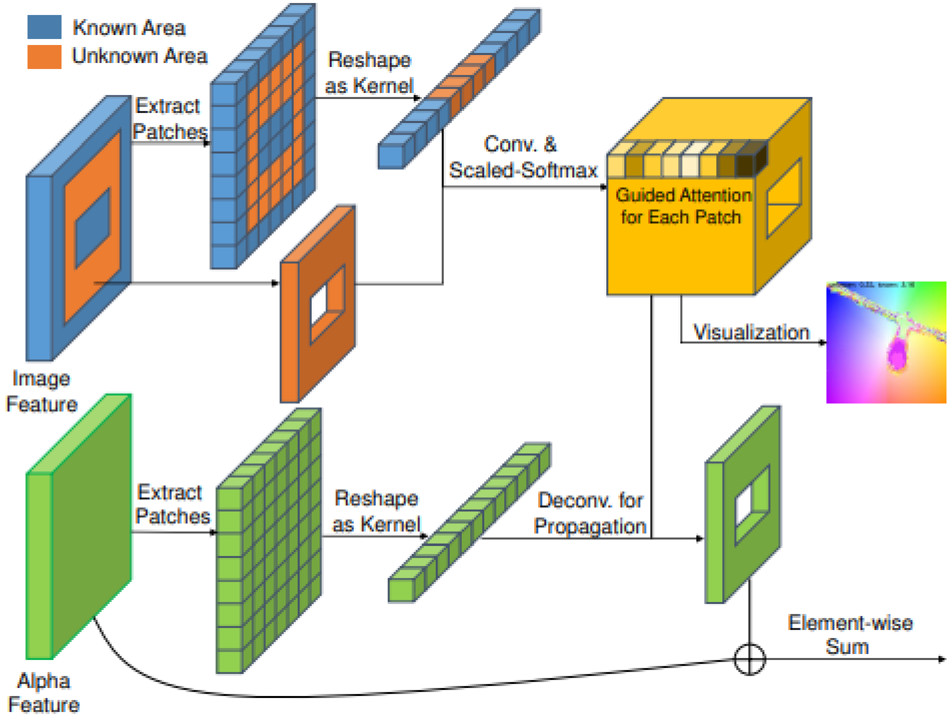
O(c(hw) 2)

Guided Contextual Attention Module

Two different feature flows

- High-level Alpha features
- Low-level image features



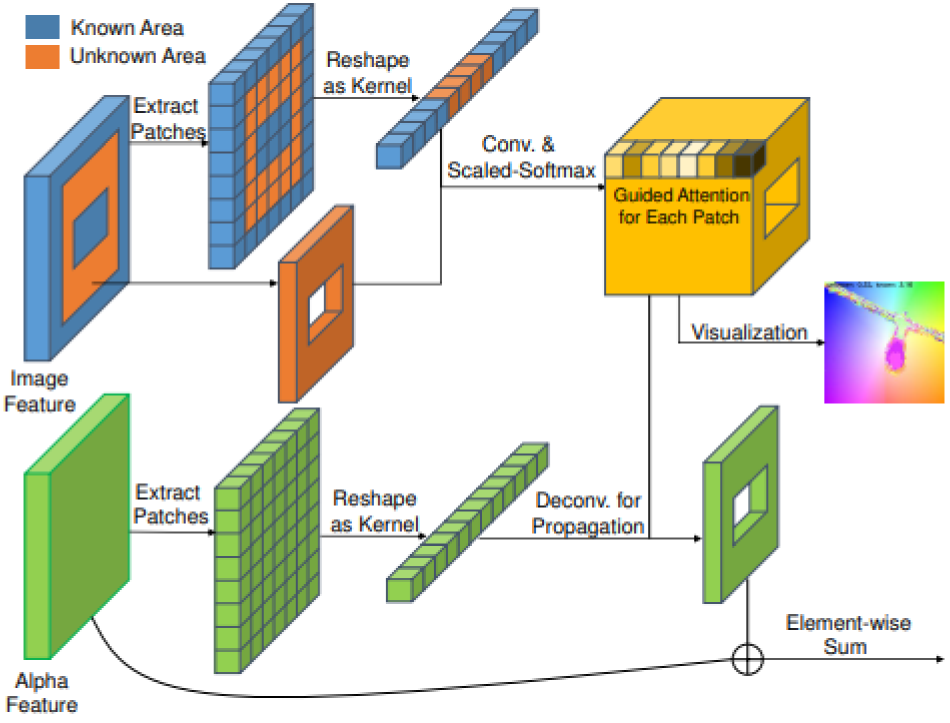


Low-level Image Feature

- Known part and unknown part
- Extract 3 × 3 patches (as conv kernels
- Correlation measure

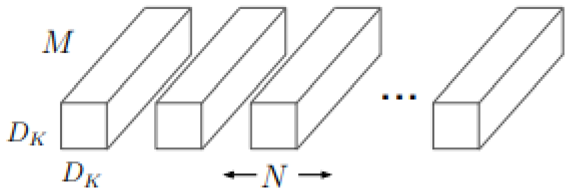
$$s_{(x,y),(x',y')} = \begin{cases} \lambda & (x,y) = (x',y'); \\ \langle \frac{U_{x,y}}{\|U_{x,y}\|}, \frac{I_{x',y'}}{\|I_{x',y'}\|} \rangle & \text{otherwise}, \end{cases}$$

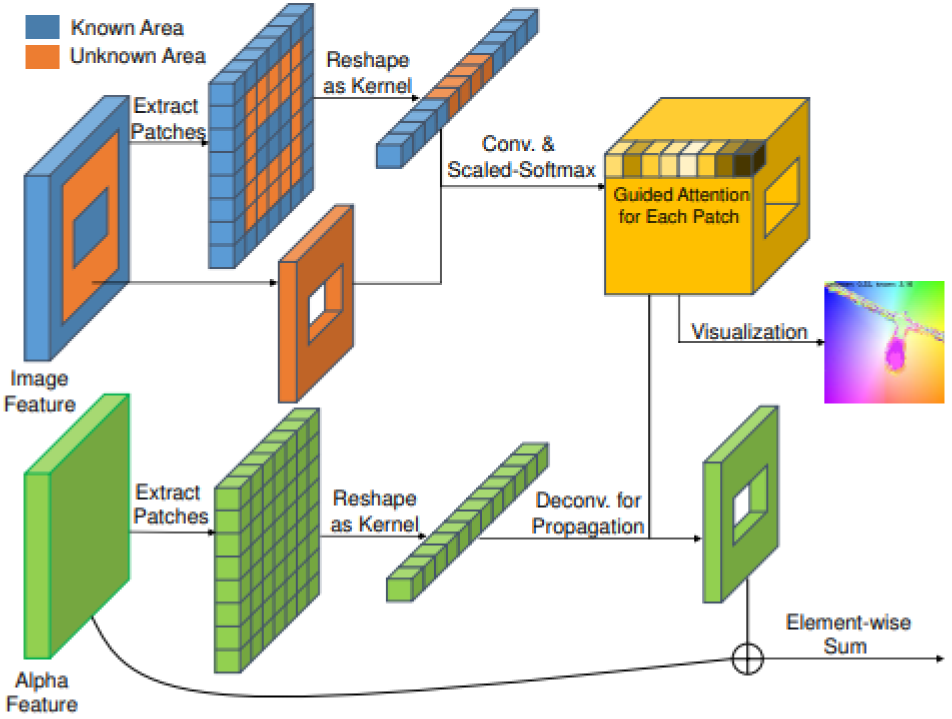
- Compute similarity by convolution



Kernel size： D_k = 3

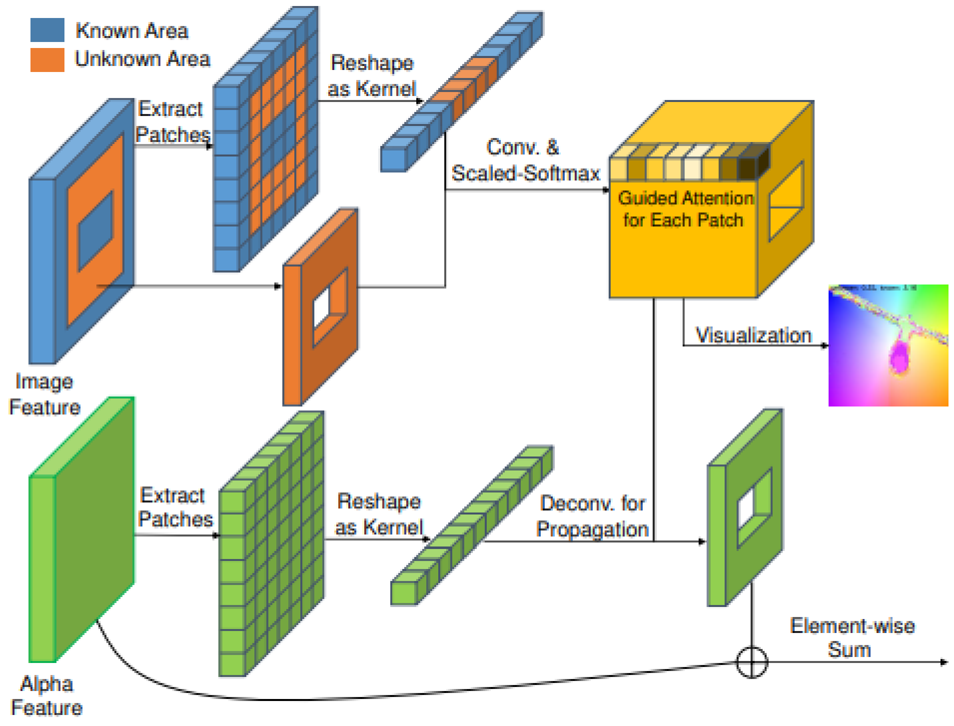
- Regular：
  - Input： M x H x W
  - Output： N x H x W
  - Param： 3 x 3 x M x N
- GCA：
  - Input： M x H1 x W1
  - Patch： N = H x W , 3 x 3 x M
  - Param： 3 x 3 x M x N
  - Output： N x H1 x W1





High-level Alpha features

- Extract 3 × 3 patches
- Reconstruct alpha features
- Element-wise summation, residual connection



Result

Methods	MSE	SAD	Grad	Conn
Learning Based Matting	0.048	113.9	91.6	122.2
Closed-Form Matting	0.091	168.1	126.9	167.9
KNN Matting	0.103	175.4	124.1	176.4
Deep Matting	0.014	50.4	31.0	50.8
IndexNet Matting	0.013	45.8	25.9	43.7
SampleNet Matting	0.0099	40.35	-	-
Baseline	0.0106	40.62	21.53	38.43
Ours	**0.0091**	**35.28**	**16.92**	**32.53**

Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation

Portland State University

ICCV 2019

Motivation

- Simultaneously estimate the alpha map and the foreground image
- Attribute
  - local visual features and global context information
  - combination of the Laplacian and feature loss
  - various effective data augmentation strategies

Context-Aware Image Matting

Xception 65 architecture

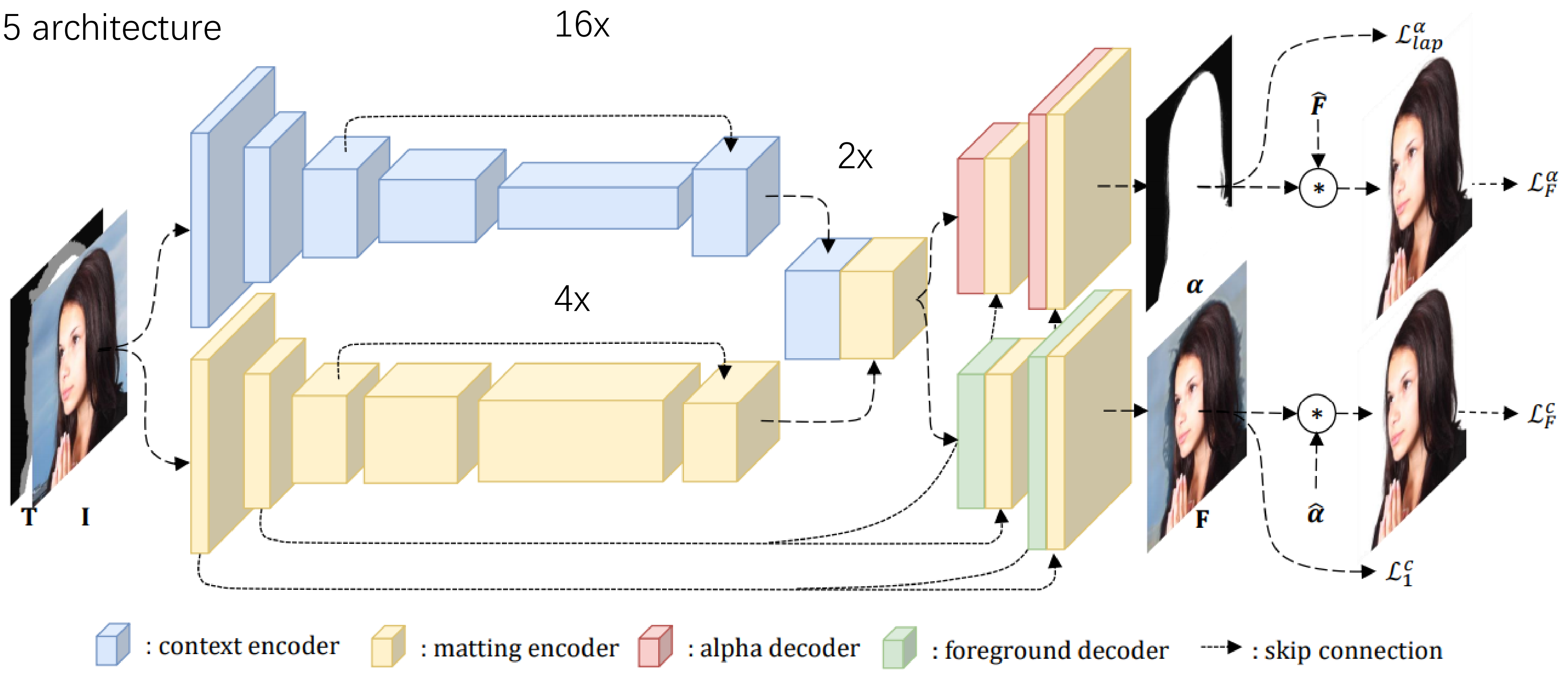
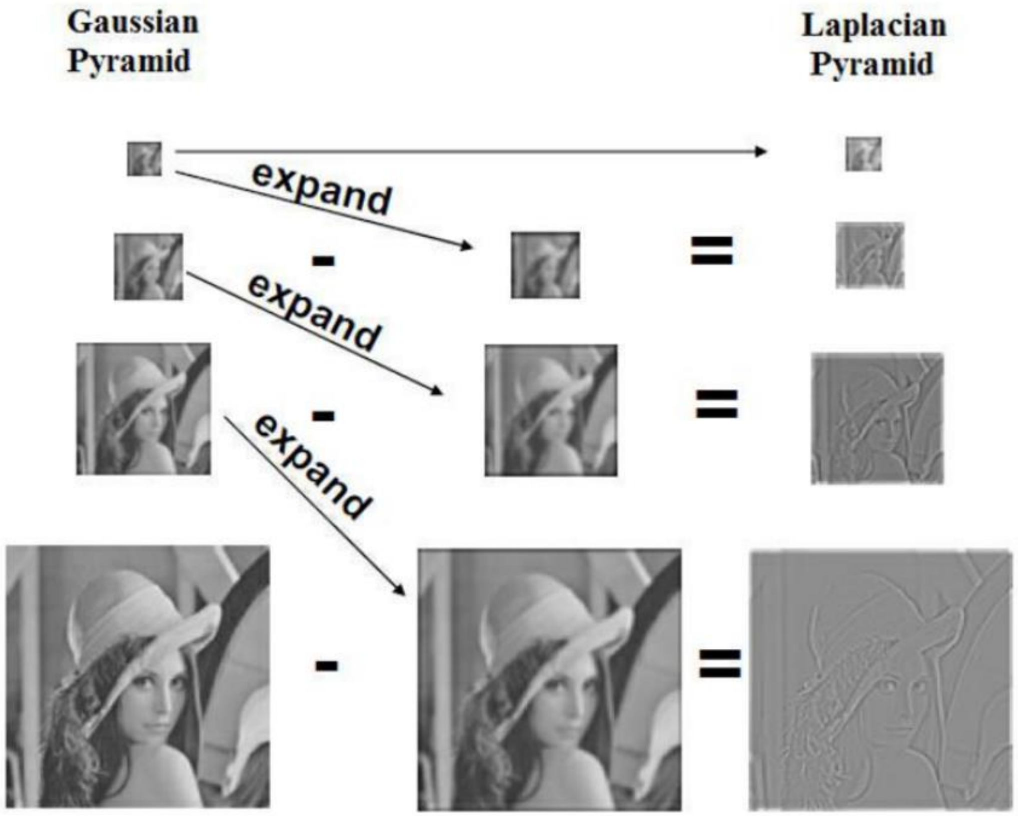
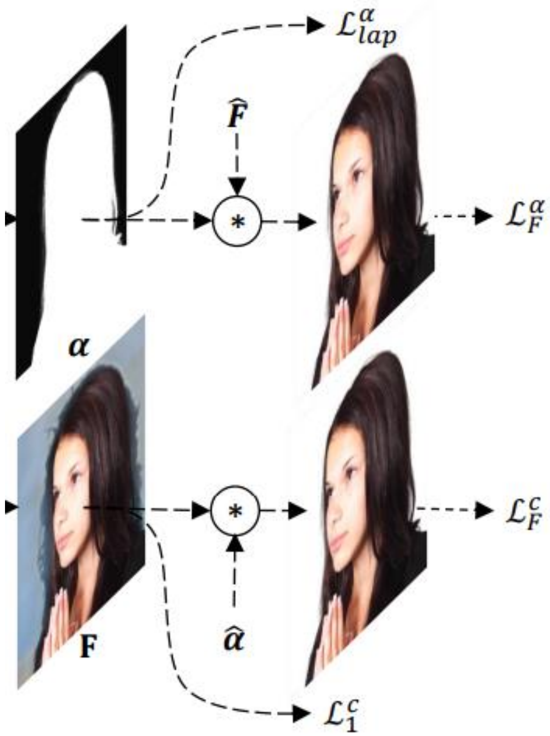


Figure 2. The architecture of our matting network. We design a two-encoder-two-decoder network. The matting encoder and the context encoder capture both visual features and more global context information. The features from these two encoders are concatenated and feed to the foreground and the alpha decoder to output the foreground image and the alpha map of the input image simultaneously.

Context-Aware Image Matting

Laplacian loss $$\mathcal{L}_{lap}^{\alpha} = \sum_{i=1}^{5} 2^{i-1} \| L^i(\hat{\boldsymbol{\alpha}}) - L^i(\boldsymbol{\alpha}) \|_1,$$





Context-Aware Image Matting

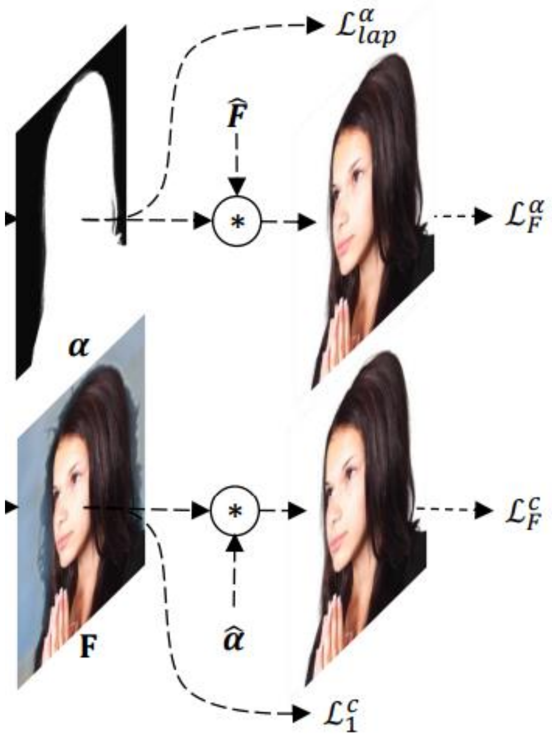
Laplacian loss

$$\mathcal{L}_{lap}^{\alpha} = \sum_{i=1}^{5} 2^{i-1} \| L^i(\hat{\boldsymbol{\alpha}}) - L^i(\boldsymbol{\alpha}) \|_1,$$

Feature loss

$$\mathcal{L}_{F}^{\alpha} = \sum_{layer} \| \phi_{layer}(\hat{\boldsymbol{\alpha}} * \hat{\mathbf{F}}) - \phi_{layer}(\boldsymbol{\alpha} * \hat{\mathbf{F}}) \|_2^2,$$

$$\mathcal{L}_{F}^{c} = \sum_{layer} \| \phi_{layer}(\hat{\boldsymbol{\alpha}} * \hat{\mathbf{F}}) - \phi_{layer}(\hat{\boldsymbol{\alpha}} * \mathbf{F}) \|_2^2,$$



- F^, ground truth foreground
- α^, ground truth Alpha matte
- φ_layer, features output by the layer in a pre-trained VGG16 network.
    - Our method uses [conv1 2, conv2 2, conv3 3, conv4 3]

Context-Aware Image Matting

Laplacian loss

$$\mathcal{L}^{\alpha}_{lap} = \sum_{i=1}^{5} 2^{i-1} \| L^i(\hat{\boldsymbol{\alpha}}) - L^i(\boldsymbol{\alpha}) \|_1,$$

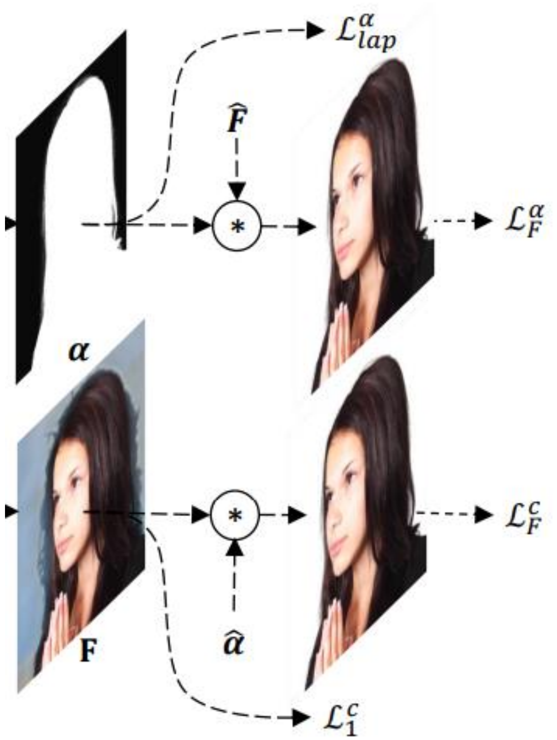
Feature loss

$$\mathcal{L}^{\alpha}_{F} = \sum_{layer} \| \phi_{layer}(\hat{\boldsymbol{\alpha}} * \hat{\mathbf{F}}) - \phi_{layer}(\boldsymbol{\alpha} * \hat{\mathbf{F}}) \|_2^2,$$

$$\mathcal{L}^{c}_{F} = \sum_{layer} \| \phi_{layer}(\hat{\boldsymbol{\alpha}} * \hat{\mathbf{F}}) - \phi_{layer}(\hat{\boldsymbol{\alpha}} * \mathbf{F}) \|_2^2,$$

L1 loss

$$\mathcal{L}^{c}_{1} = \| \mathbb{1}(\hat{\boldsymbol{\alpha}} > 0) * (\hat{\mathbf{F}} - \mathbf{F}) \|_1,$$



Data Augmentation

- Subtle artifacts
  - misaligned JPEG blocks, compression quantization artifacts, and resampling artifacts
- Augmentation
  - Resizing augmentation
  - Use re-JPEGing and Gaussian blur

Table 4. Comparison of visual quality on the real-world dataset.

Methods	Mean score	Std
ME + CE + $\mathcal{L}_{lap}$	4.64	0.42
ME + CE + $\mathcal{L}_{lap}$ + $\mathcal{L}_{F}$	4.69	0.40
ME + CE + $\mathcal{L}_{lap}$ + $\mathcal{L}_{F}$ + DA	5.03	0.25

Table 1. Alpha map results on the Composition-1K testing set.

Methods	SAD	MSE(10^3)	Grad	Conn
Shared Matting[16]	128.9	91	126.5	135.3
Learning Based Matting [54]	113.9	48	91.6	122.2
Comprehensive Sampling [42]	143.8	71	102.2	142.7
Global Matting [19]	133.6	68	97.6	133.3
Closed-Form Matting [27]	168.1	91	126.9	167.9
KNN Matting [6]	175.4	103	124.1	176.4
DCNN Matting [8]	161.4	87	115.1	161.9
Three-layer Graph [29]	106.4	66	70.0	-
Deep Matting [52]	50.4	14	31.0	50.8
Information-flow Matting [2]	75.4	66	63.0	-
AlphaGan-Best[1] [33]	52.4	30	38.0	-
(1) ME + $\mathcal{L}_{deepmatting}$	49.1	13.4	26.7	49.8
(2) ME + $\mathcal{L}_{lap}^{\alpha}$	43.9	11.8	20.6	41.6
(3) ME + CE + $\mathcal{L}_{lap}^{\alpha}$	**35.8**	**8.2**	17.3	**33.2**
(4) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$	38.8	9.0	19.0	36.0
(5) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$ + DA	71.3	23.6	38.8	72.0
(6) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$ + $\mathcal{L}_{1}^{c}$ + $\mathcal{L}_{F}^{c}$	38.0	8.8	**16.9**	35.4
(7) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$ + $\mathcal{L}_{1}^{c}$ + $\mathcal{L}_{F}^{c}$ + DA	84.1	29.1	39.2	-
(8) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$ + $\mathcal{L}_{1}^{c}$ + $\mathcal{L}_{F}^{c}$ + DA - ReJPEGing	55.1	15.5	24.6	54.7
(9) ME + CE + $\mathcal{L}_{lap}^{\alpha}$ + $\mathcal{L}_{F}^{\alpha}$ + $\mathcal{L}_{1}^{c}$ + $\mathcal{L}_{F}^{c}$ + DA - GaussianBlur	69.1	23.5	39.6	69.1

F, *B*, Alpha Matting

Trinity College Dublin

Motivation

- Recent two methods
  - Show improved results by also estimating the foreground colors,
  - Significant computational and memory cost
- This paper
  - low-cost modification to also predict the foreground and background colours
  - study variations of the training regime, loss functions

Contributions

1. a comparison of min-batch and stochastic gradient descent and the use of batchnorm vs. groupnorm
2. a study of the different **α-matte losses** (L1, gradient, laplacian pyramid, compositing loss).
3. a study of the potential benefit of **also predicting F and B** alongside α and the possible losses associated with this (L1 loss and exclusion loss).

Network Arch

- Encoder-decoder with Unet style architecture
- Main difference, also predicts F and B from single encoder-decoder
- Extending output channels from one to seven (1 for α, 3 for F and 3 for B)

Encoder

- ResNet-50
- increase the number of input channels from 3 to 9
- encode the trimap
  - using Gaussian blurs
  - of the definite foreground and background masks
  - at three different scales
- remove striding, add dilation, [‘layer 3’, ‘layer 4’]

Batch Normalisation vs. Group Normalisation

- A mini-batch size of one can greatly increase the network accuracy
- Use Group Normalisation (32 channels per group)

F, B, α Losses

Table 1. Training Loss Functions.

$\boldsymbol{\alpha}$ Losses	$\mathbf{F}, \mathbf{B}$ Losses
$\mathcal{L}_1^{\alpha} = \sum_i \lVert \hat{\boldsymbol{\alpha}}_i - \boldsymbol{\alpha}_i \rVert_1$	$\mathcal{L}_1^{\mathrm{FB}} = \sum_i \left\lVert \hat{\mathbf{F}}_i - \mathbf{F}_i \right\rVert_1 + \left\lVert \hat{\mathbf{B}}_i - \mathbf{B}_i \right\rVert_1$
$\mathcal{L}_c^{\alpha} = \sum_i \lVert \mathbf{C}_i - \hat{\boldsymbol{\alpha}}_i \mathbf{F}_i - (1 - \hat{\boldsymbol{\alpha}}_i) \mathbf{B}_i \rVert_1$	$\mathcal{L}_{\mathrm{excl}}^{\mathrm{FB}} = \sum_i \lVert \nabla \mathbf{F}_i \rVert_1 \lVert \nabla \mathbf{B}_i \rVert_1$
$\mathcal{L}_{\mathrm{lap}}^{\alpha} = \sum_{s=1}^{5} 2^{s-1} \left\lVert L_{\mathrm{pyr}}^{s}(\boldsymbol{\alpha}) - L_{\mathrm{pyr}}^{s}(\hat{\boldsymbol{\alpha}}) \right\rVert_1$	$\mathcal{L}_c^{\mathrm{FB}} = \sum_i \left\lVert \mathbf{C}_i - \boldsymbol{\alpha}_i \hat{\mathbf{F}} - (1 - \boldsymbol{\alpha}_i) \hat{\mathbf{B}}) \right\rVert_1$
$\mathcal{L}_g^{\alpha} = \sum_i \lVert \nabla \hat{\boldsymbol{\alpha}}_i - \nabla \boldsymbol{\alpha}_i \rVert_1$	$\mathcal{L}_{\mathrm{lap}}^{\mathrm{FB}} = \mathcal{L}_{\mathrm{lap}}^{\mathbf{F}} + \mathcal{L}_{\mathrm{lap}}^{\mathbf{B}}$

F^, B^, α^ Fusion

- Predictions for α^, F^ and B^ are decoupled
- Equation 1 is not explicitly enforced

$$\mathbf{C}_i = \boldsymbol{\alpha}_i \mathbf{F}_i + (1 - \boldsymbol{\alpha}_i)\mathbf{B}_i$$

- Propose a fusion mechanism based on maximum likelihood estimate

F^, B^, α^ Fusion

- Assuming Gaussian distributions

$$p(\mathbf{F}|\hat{\mathbf{F}}) \propto \exp\left(-\frac{\|\mathbf{F}-\hat{\mathbf{F}}\|_2^2}{2\sigma_{FB}^2}\right) \qquad p(\mathbf{B}|\hat{\mathbf{B}}) \propto \exp\left(-\frac{\|\mathbf{B}-\hat{\mathbf{B}}\|_2^2}{2\sigma_{FB}^2}\right)$$

$$p(\boldsymbol{\alpha}|\hat{\boldsymbol{\alpha}}) \propto \exp\left(-\frac{(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}})^2}{2\sigma_{\alpha}^2}\right) \qquad p(\boldsymbol{\alpha},\mathbf{F},\mathbf{B}) \propto \exp\left(-\frac{\|\mathbf{C}-\boldsymbol{\alpha}\mathbf{F}-(1-\boldsymbol{\alpha})\mathbf{B}\|_2^2}{2\sigma_C^2}\right)$$

F^, B^, α^ Fusion

- Adopt an iterative block solver approach

$$\hat{\mathbf{F}}^{(n+1)} = \hat{\mathbf{F}} + \frac{\sigma_F^2}{\sigma_{\mathbf{C}}^2}\hat{\boldsymbol{\alpha}}^{(n)}\left(\mathbf{C} - \hat{\boldsymbol{\alpha}}^{(n)}\hat{\mathbf{F}}^{(n)} - (1-\hat{\boldsymbol{\alpha}}^{(n)})\hat{\mathbf{B}}^{(n)}\right)$$

$$\hat{\mathbf{B}}^{(n+1)} = \hat{\mathbf{B}} + \frac{\sigma_B^2}{\sigma_{\mathbf{C}}^2}(1-\hat{\boldsymbol{\alpha}}^{(n)})\left(\mathbf{C} - \hat{\boldsymbol{\alpha}}^{(n)}\hat{\mathbf{F}}^{(n)} - (1-\hat{\boldsymbol{\alpha}}^{(n)})\hat{\mathbf{B}}^{(n)}\right)$$

$$\hat{\boldsymbol{\alpha}}^{(n+1)} = \frac{\hat{\boldsymbol{\alpha}}^{(n)} + \frac{\sigma_\alpha^2}{\sigma_{\mathbf{C}}^2}(\mathbf{C} - \hat{\mathbf{B}}^{(n+1)})^\top(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})}{1 + \frac{\sigma_\alpha^2}{\sigma_{\mathbf{C}}^2}(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})^\top(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})}$$

Test Time Augmentation

- We use a comprehensive test-time augmentation, combining rotation, flipping and scaling

Batch-Size and BN vs. GN
Loss Function and Activation

Model	Norm.	Batch-Size	Loss	MSE	SAD	GRAD	CONN
Training at 20 epochs:							
(1)	BN	6	$\mathcal{L}_1^\alpha$	11.2	36.3	14.9	32.5
(2)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha$	9.1	34.5	15.0	31.3
(3)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha$	7.4	33.5	12.9	28.5
(4)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	8.1	36.3	13.8	32.0
(5)	GN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	10.3	36.2	15.1	32.0
(6)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	7.2	32.8	13.3	28.6
(7)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	6.9	31.2	12.9	27.1
Training at 45 epochs:							
Ours$_\alpha$	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	5.3	26.5	10.6	21.8

Evaluating the Impact of Jointly Estimating F, B, α

Table 3. Ablation study of foreground results on the Composition-1k dataset. Here $\mathcal{L}^{FB} = \mathcal{L}_1^{FB} + \mathcal{L}_{\text{lap}}^{FB} + \mathcal{L}_{\text{c}}^{FB}$. In column two the * indicates that the $\mathcal{L}_1^{FB}, \mathcal{L}_{\text{lap}}^{FB}$ are computed over the entire image as opposed to just the unknown region of the trimap.

Model	$+\mathcal{L}_{FB}$	$+\mathcal{L}_{\text{excl}}$	output	α**F**		α	
				SAD	MSE	SAD	MSE
Closed-form Matting [20]				251.67	22.96	161.3	85.3
Context-Aware Matting [13]				70.00	11.49	38.1	8.9
Training at 20 epochs:							
(6)	N	N	sigmoid	-	-	32.8	7.2
(8)	Y	N	sigmoid	53.64	9.04	32.7	9.0
(9)	Y	Y	sigmoid	52.87	8.88	31.8	8.9
(7)	N	N	clip	-	-	31.2	6.9
(10)	Y	Y	clip	50.69	8.64	31.3	8.6
(11)	Y*	Y	clip	50.29	8.48	32.1	8.5
Training at 45 epochs:							
(11)	Y*	Y	clip	42.19	6.50	26.5	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion	39.21	6.19	26.4	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion +TTA	38.81	5.98	25.8	5.2

Result

Table 4. Alpha map results on the Composition-1k test set [37].

Method	SAD	MSE x10^3	Gradient	Connectivity
Closed-Form Matting [20]	168.1	91.0	126.9	167.9
KNN-Matting [4]	175.4	103.0	124.1	176.4
DCNN Matting [5]	161.4	87.0	115.1	161.9
Information-flow Matting [1]	75.4	66.0	63.0	-
Deep Image Matting [37]	50.4	14.0	31.0	50.8
AlphaGan-Best [25]	52.4	30.0	38.0	-
IndexNet Matting [24]	45.8	13.0	25.9	43.7
VDRN Matting [33]	45.3	11.0	30.0	45.6
AdaMatting [3]	41.7	10.2	16.9	-
Learning Based Sampling [34]	40.4	9.9	-	-
Context Aware Matting [13]	35.8	8.2	17.3	33.2
GCA Matting [21]	35.3	9.1	16.9	32.5
Ours$_\alpha$	26.5	5.3	10.6	21.8
Ours$_{FB\alpha}$	26.4	5.4	10.6	21.5
Ours$_{FB\alpha}$ **TTA**	**25.8**	**5.2**	**10.6**	**20.8**

Background Matting: The World is Your Green Screen

University of Washington

CVPR 2020

Motivation

- To extracting (pulling) a good quality matte, require either a **green screen studio**, or the manual creation of a **trimap**

- Paper propose
- Take an additional photo of the (static) background

Supervised Training on the Adobe Dataset

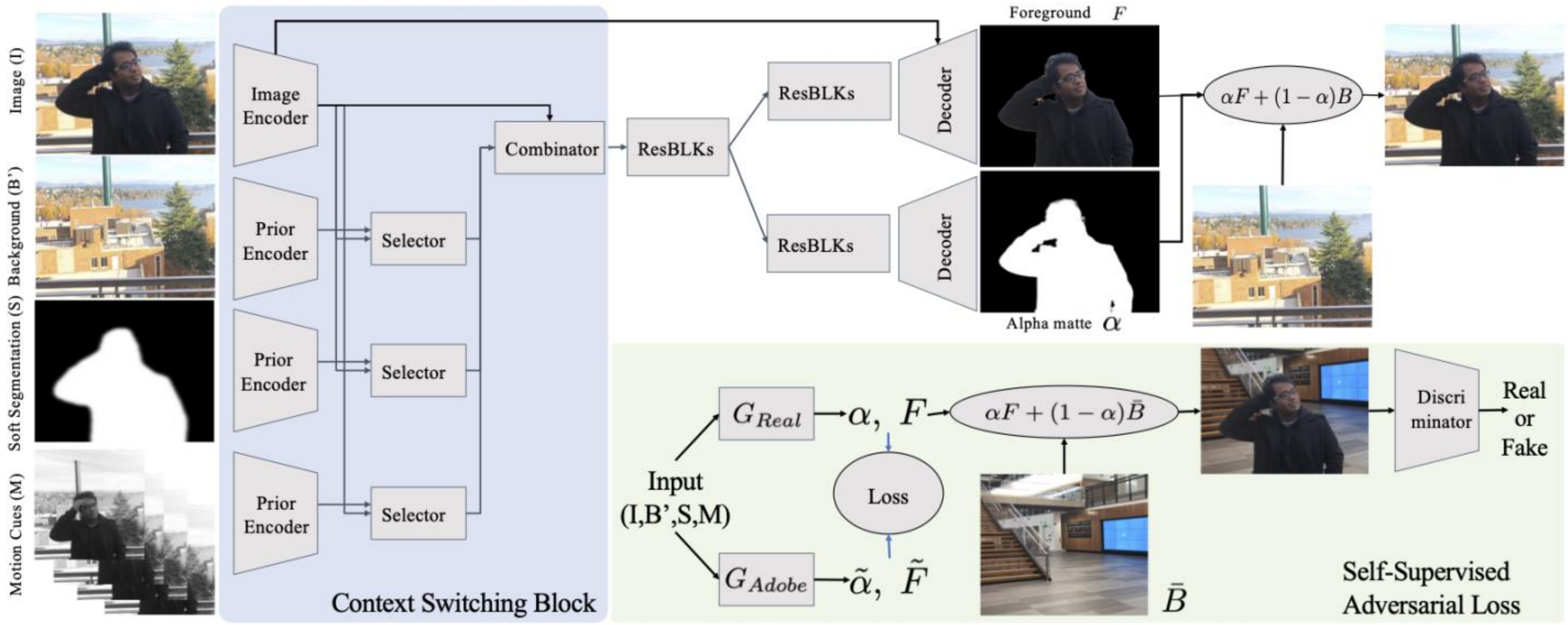


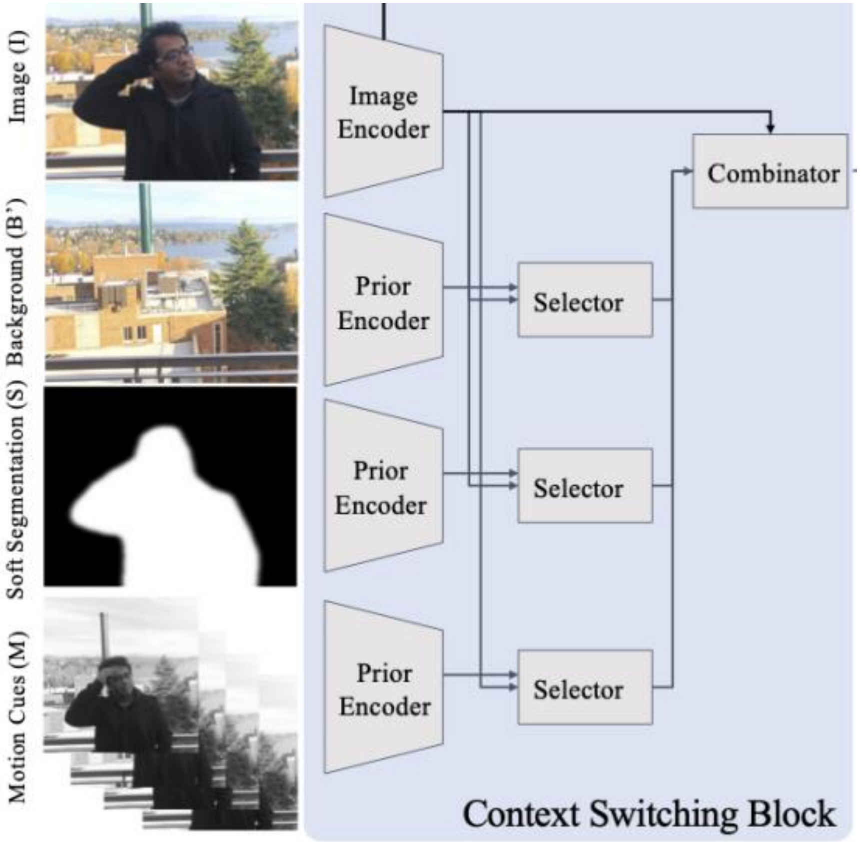
Figure 2: **Overview of our approach.** Given an input image I and background image B', we jointly estimate the alpha matte α and the foreground F using soft segmentation S and motion prior M (for video only). We propose a Context Switching Block that efficiently combines all different cues. We also introduce self-supervised training on unlabelled real data by compositing into novel backgrounds.

Supervised Training on the Adobe Dataset

- Input
  - An image I with a person in the foreground,
  - An image of the background B
  - A soft segmentation of the person S,
  - A stack of temporally nearby frames M, (optionally for video)
- Generate S
  - Apply person segmentation
  - Erode (5 steps), dilate (10 steps)
  - Apply a Gaussian blur ($\sigma = 5$)
- Set M to be the concatenation of the two frames before and after
  - converted to grayscale, focus more on motion cues

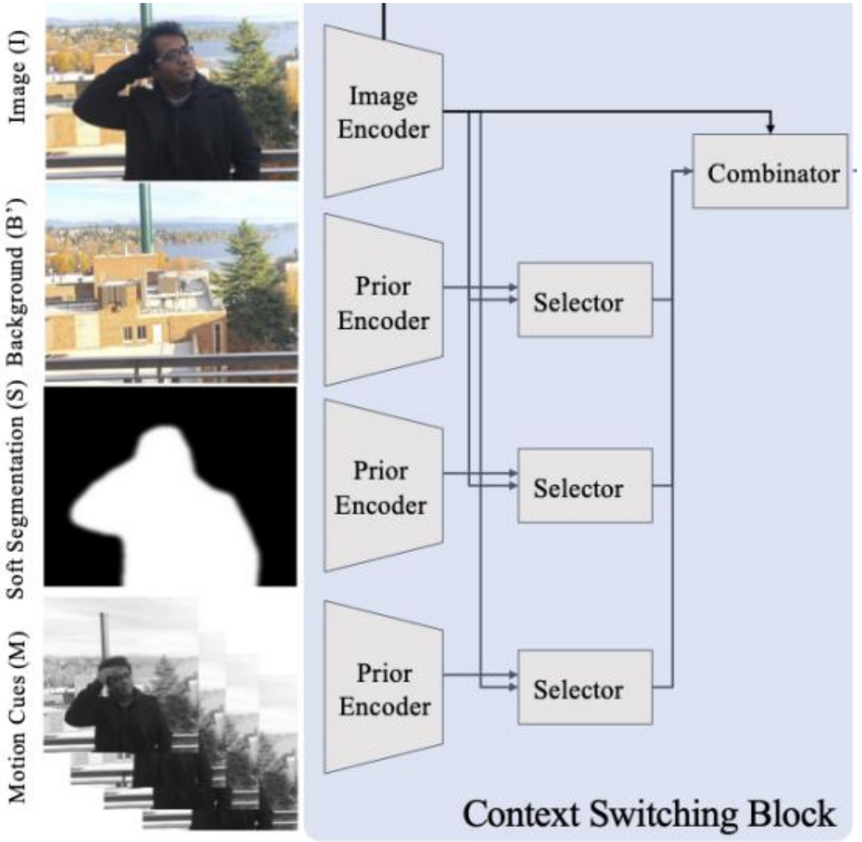
Supervised Training on the Adobe Dataset

- Residual-block-based encoder-decoder, doesn't work
- Reason, domain gap,
  - trusting the background B' too much and generating holes
- Instead, we propose a new Context Switching block

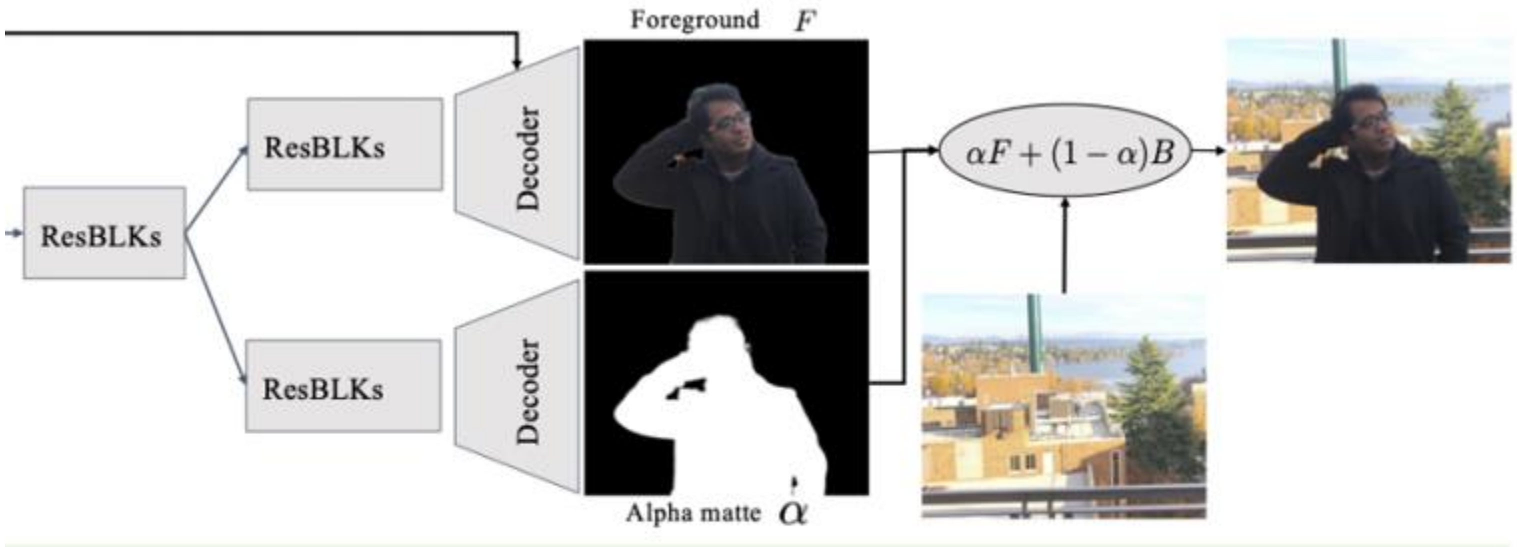


Supervised Training on the Adobe Dataset

- Separately produce 256 channels of feature maps
- Combines the image features from I, producing 64-channel features for each
- Combines 3*64 and 256 channel, 1x1 conv BN and ReLU



Supervised Training on the Adobe Dataset

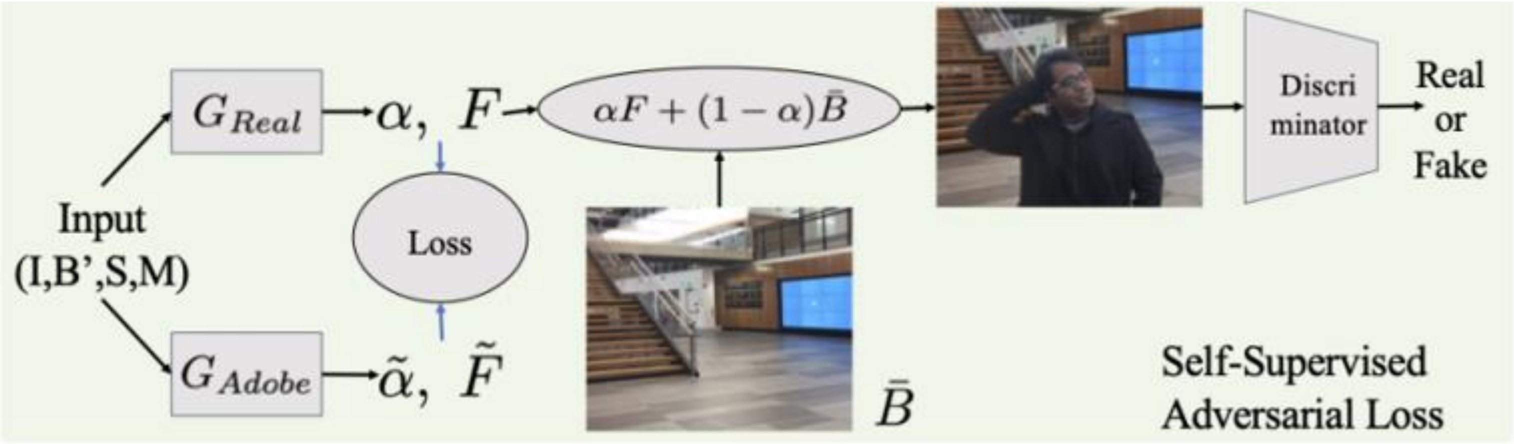


$$\min_{\theta_{\text{Adobe}}} E_{X \sim p_X}[\|\alpha - \alpha^*\|_1 + \|\nabla(\alpha) - \nabla(\alpha^*)\|_1$$
$$+ 2\|F - F^*\|_1 + \|I - \alpha F - (1-\alpha)B\|_1],$$

Adversarial Training on Unlabelled Real data

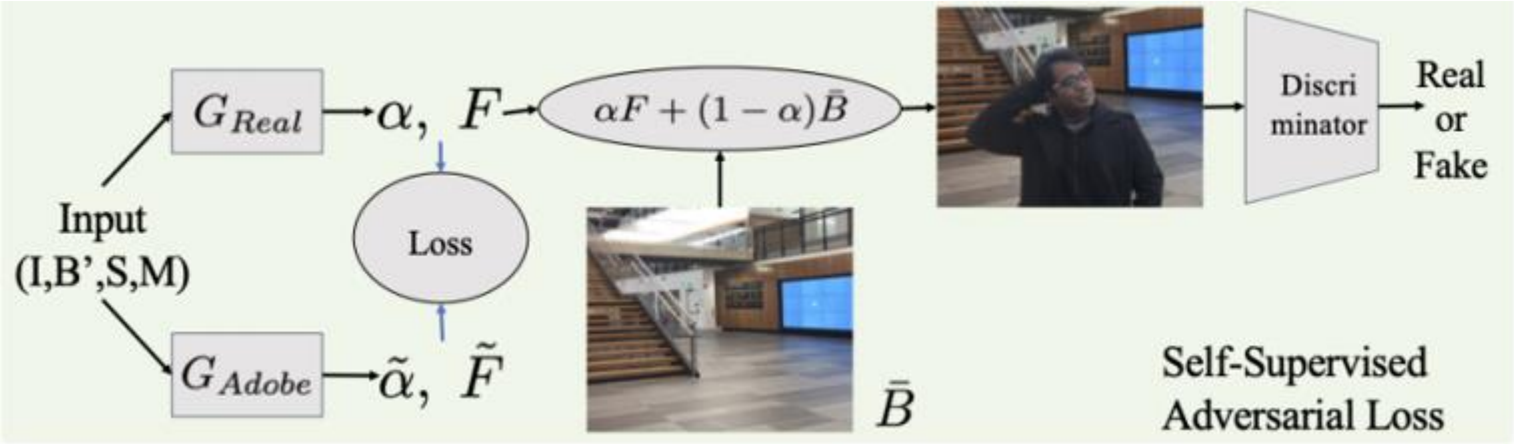
- Still fails to handle all difficulties present in real data

1. traces of background around fingers, arms, hairs copied into matte
2. segmentation failing
3. foreground color matching the background color
4. misalignment between the image and the background



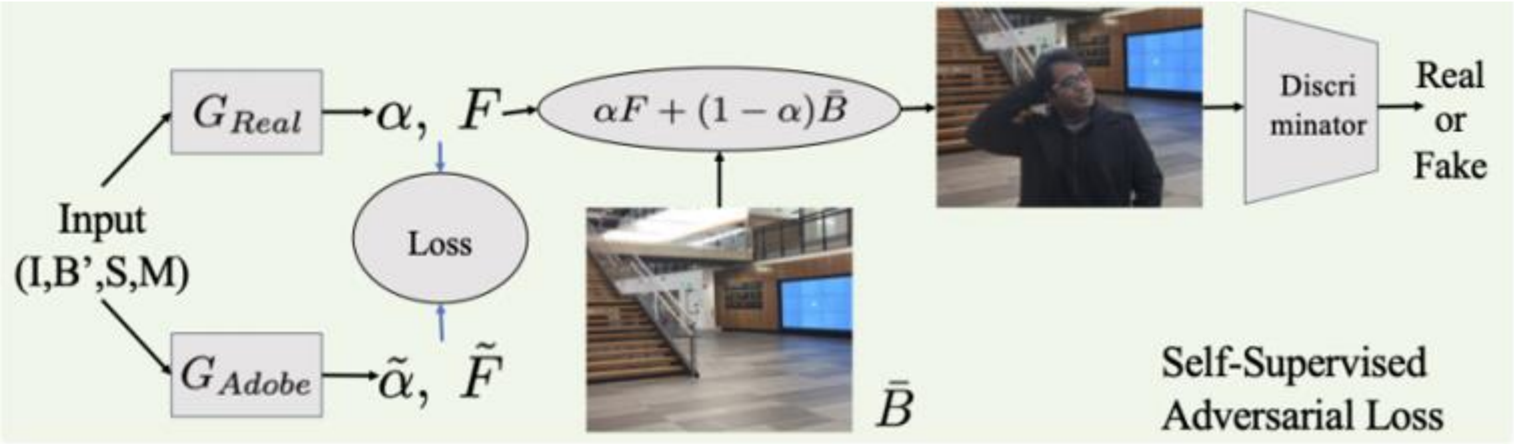
Adversarial Training on Unlabelled Real data

- Problem
  - 1. G_Real could settle on setting α = 1 everywhere
  - 2. Initializing with G_Adobe and fine-tuning with a low learning rate, not allow significant changes to generate good mattes on real data



Adversarial Training on Unlabelled Real data

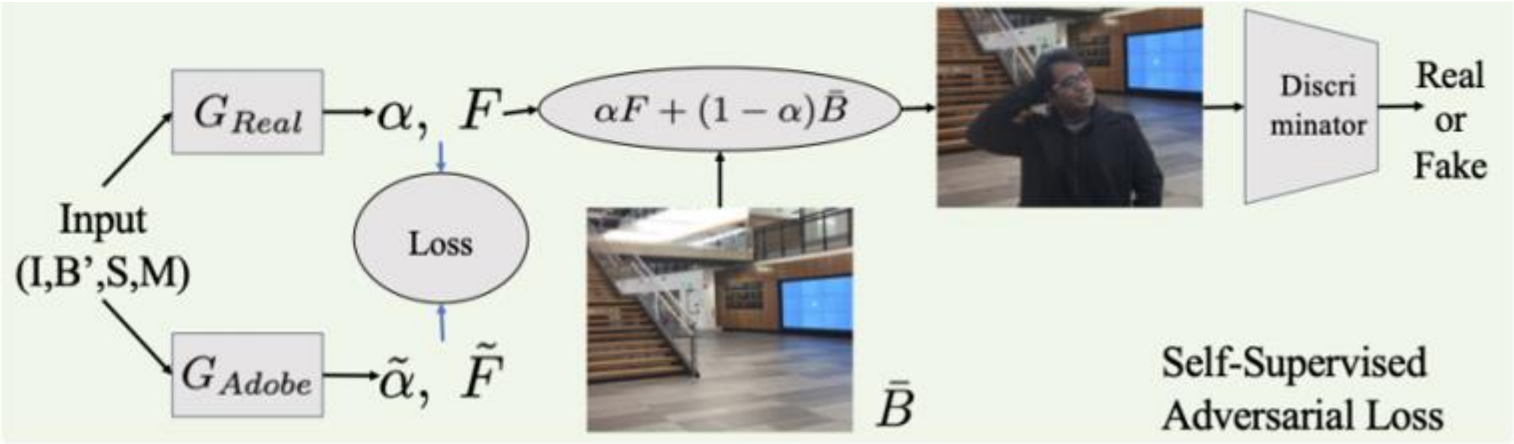
- Use G_Adobe for teacher-student learning.
- Obtain (F , ˜ α˜) = G(X; θ_Adobe) to serve as “pseudo ground-truth”



Adversarial Training on Unlabelled Real data

- Adversarial loss
- Loss on the output of G(X; θReal) compared to "pseudo ground-truth"

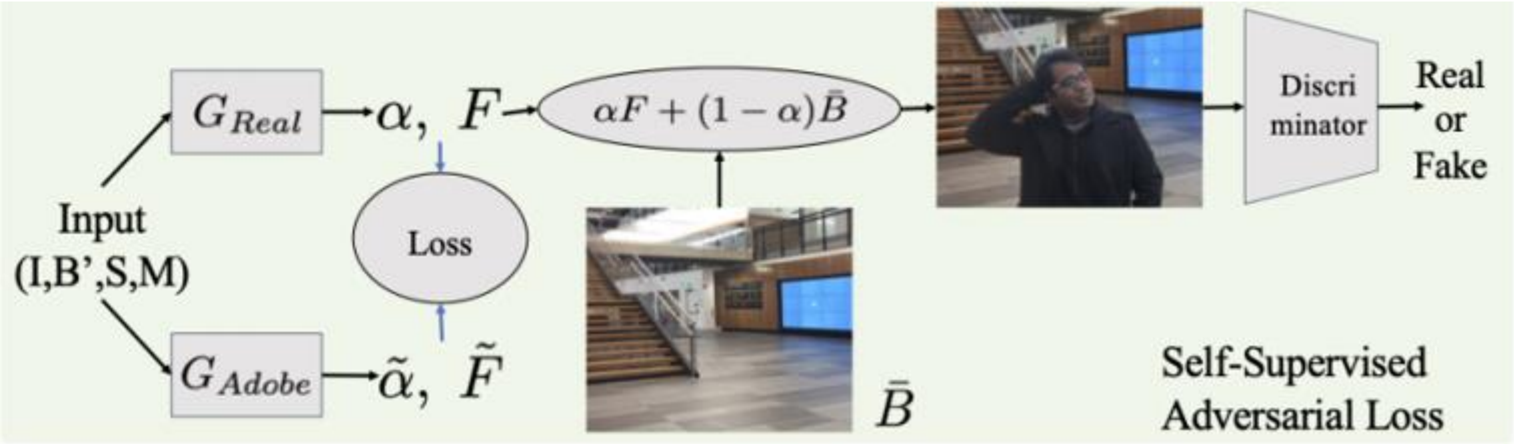
$$
\begin{aligned}
\min_{\theta_{\text{Real}}} \mathbb{E}_{X,\bar{B}\sim p_{X,\bar{B}}}[(D(\alpha F + (1-\alpha)\bar{B}) - 1)^2 \\
+ \lambda\{2\|\alpha - \tilde{\alpha}\|_1 + 4\|\nabla(\alpha) - \nabla(\tilde{\alpha})\|_1 \\
+ \|F - \tilde{F}\|_1 + \|I - \alpha F - (1-\alpha)B'\|_1\}],
\end{aligned}
$$



Adversarial Training on Unlabelled Real data

- For the discriminator, we minimize:

$$\min_{\theta_{\text{Disc}}} \mathbb{E}_{X,\bar{B}\sim p_{X,\bar{B}}}[(D(\alpha F + (1-\alpha)\bar{B}))^2] + \mathbb{E}_{I \in p_{data}}[(D(I)-1)^2],$$



Result

Algorithm	Additional Inputs	SAD	MSE(10^{-2})
BM	Trimap-10, B	2.53	1.33
BM	Trimap-20, B	2.86	1.13
BM	Trimap-20, B'	4.02	2.26
CAM	Trimap-10	3.67	4.50
CAM	Trimap-20	4.72	4.49
IM	Trimap-10	1.92	1.16
IM	Trimap-20	2.36	1.10
Ours-Adobe	B	**1.72**	**0.97**
Ours-Adobe	B'	**1.73**	**0.99**

Table 1: Alpha matte error on Adobe Dataset (lower is better).

Real-Time High-Resolution Background Matting

University of Washington

Motivation

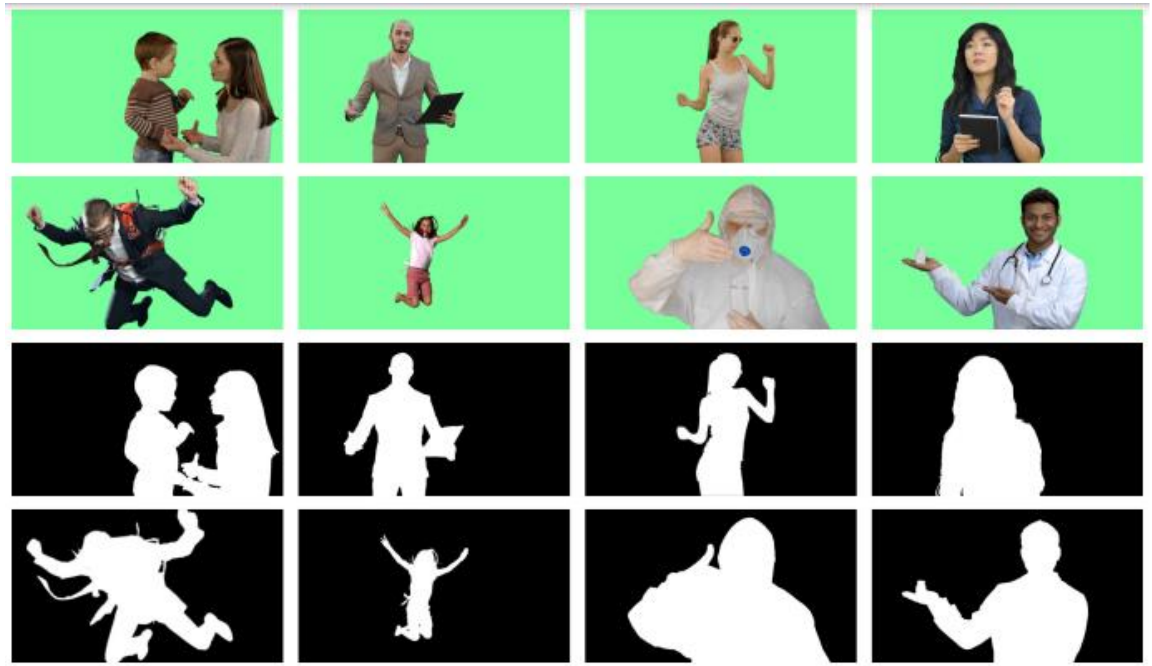
- While many tools now provide background replacement functionality
  - yield artifacts at boundaries
  - higher quality results, but do not run in real-time, at high resolution

- In this paper, we introduce the first fully-automated, real-time, high-resolution matting technique.

Dataset

- VideoMatte240K
  - 484 high-resolution **green screen**
  - generate a total of 240,709 unique frames
  - 384 videos are at 4K resolution and 100 are in HD
- PhotoMatte13K/85
  - 13,665 images shot with studio-quality lighting and cameras in front of a **green-screen**
  - narrow range of poses
  - high resolution, averaging around 2000×2500

Dataset



(a) VideoMatte240K

(b) PhotoMatte13K/85

Figure 2: We introduce two large-scale matting datasets containing 240k unique frames and 13k unique photos.

Network

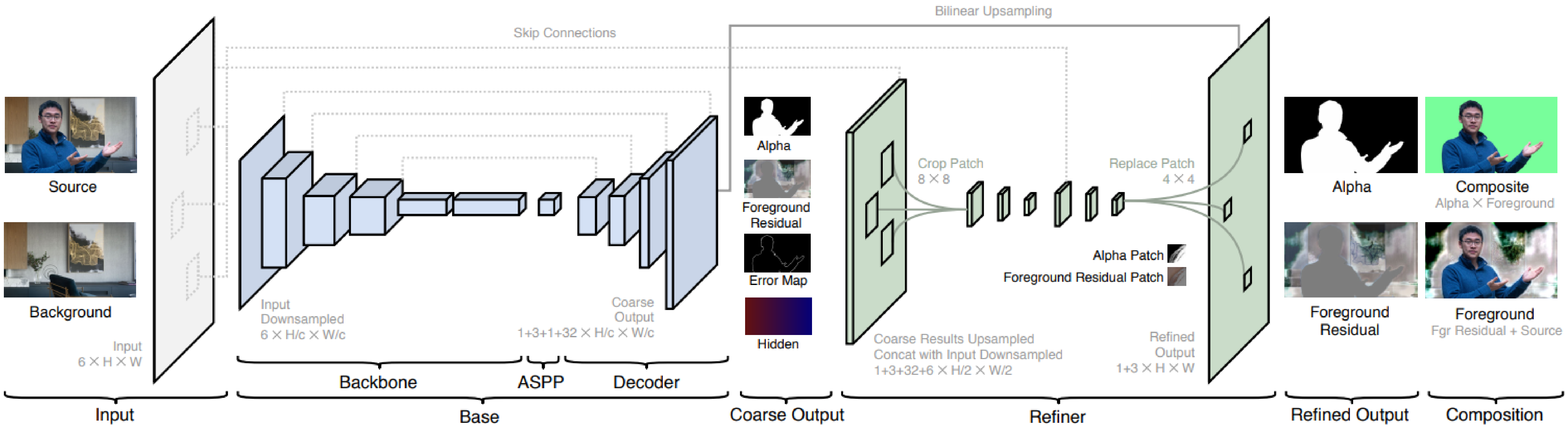
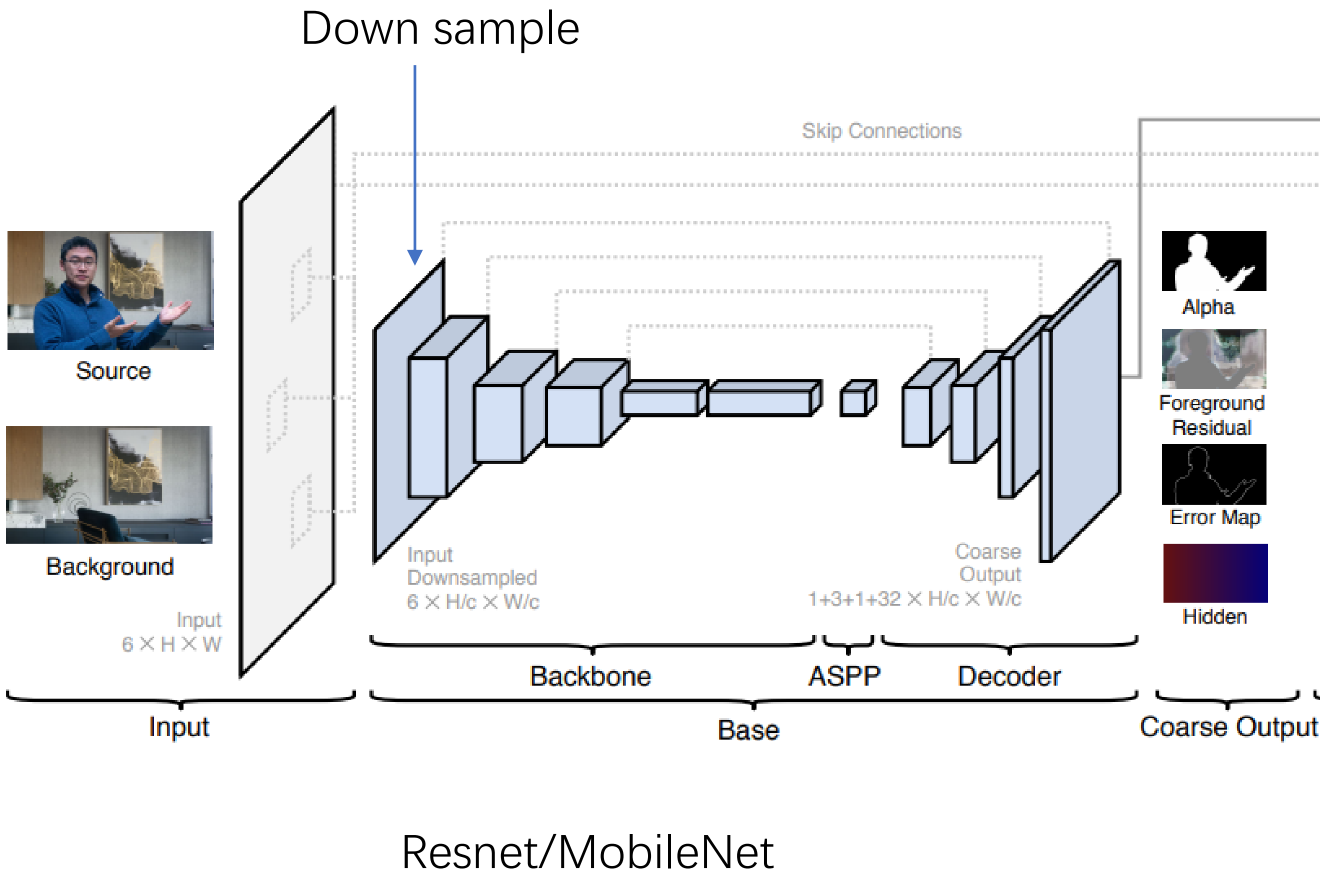


Figure 3: The base network G_{base} (blue) operates on the downsampled input to produce coarse-grained results and an error prediction map. The refinement network G_{refine} (green) selects error-prone patches and refines them to the full resolution.

Base net



- Alpha matte
- Foreground Residual

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

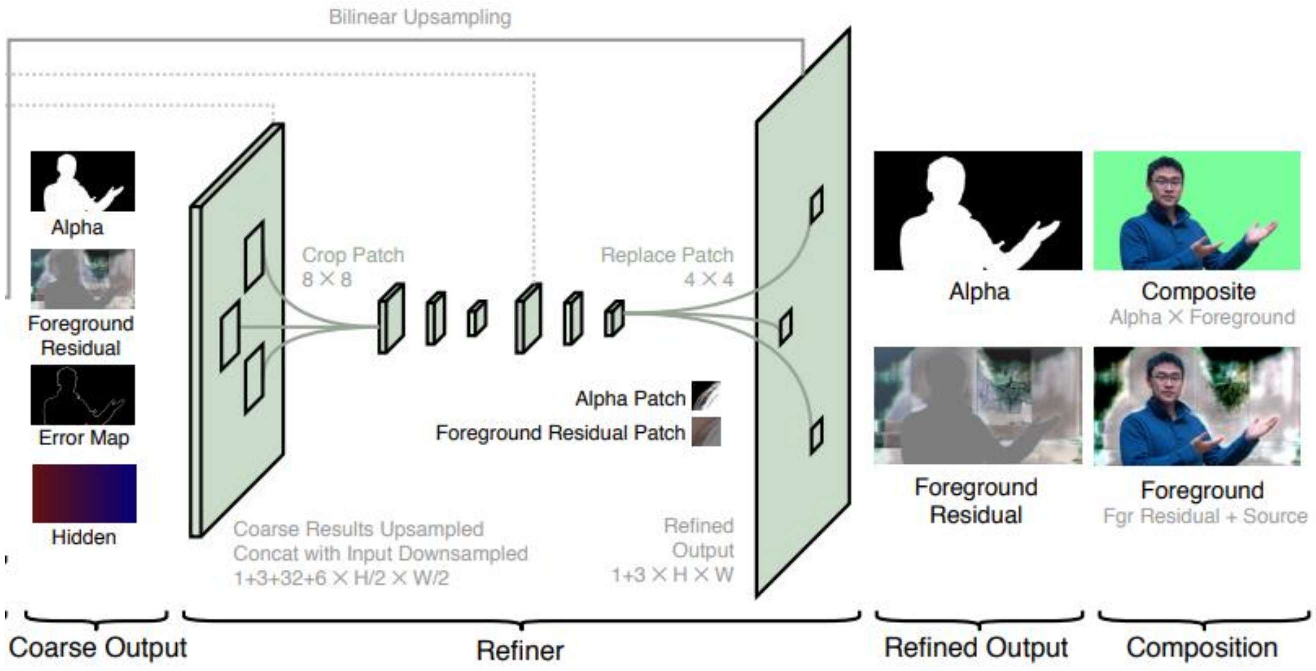
$$F^R = F - I.$$

$$F = \max(\min(F^R + I, 1), 0).$$

- Error Map
- Hidden Layer(32 channel)

Refinement Network

- select patch based on error map
- Input: [alpha, fgr, hid, src, bg] 1/2
- Crop 8 x 8 patch
- **3x3 conv, 3x3 conv, 0 pad**
- 4 x 4 patch, upsample 8 x 8
- Concat [src, bg] 1
- **3x3 conv, 3x3 conv, 0 pad**
- 4 x 4 patch, replace [alpha, fgr] 1



[32 + 1 + 3 + 6, 24, 16 + 6, 12, 4]

Loss

- Alpha loss

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

- Foreground Residual loss

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

$$F = \max(\min(F^R + I, 1), 0).$$

- Error map loss

$$\mathcal{L}_E = ||E - E^*||_2.$$

$$E^* = |\alpha - \alpha^*|.$$

- Loss function

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\alpha_c} + \mathcal{L}_{F_c} + \mathcal{L}_{E_c}.$$

$$\mathcal{L}_{\text{refine}} = \mathcal{L}_\alpha + \mathcal{L}_F.$$

Result

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.

		Alpha				FG
Dataset	Method	SAD	MSE	Grad	Conn	MSE
AIM	DIM†	37.94	80.67	32935	37861	-
	FBA†	**9.68**	**6.38**	**4265**	**7521**	**1.94**
	BGM	16.07	21.00	15371	14123	47.98
	BGM_a	19.28	29.31	19877	18083	42.84
	Ours	**12.86**	**12.01**	**8426**	**11116**	**5.31**
Distinctions	DIM†	43.70	86.22	49739	43914	-
	FBA†	**11.03**	**8.32**	**6894**	**9892**	**12.51**
	BGM	19.21	25.89	30443	18191	36.13
	BGM_a	16.02	20.18	24845	14900	43.00
	Ours	**9.19**	**7.08**	**6345**	**7216**	**6.10**
PhotoMatte85	DIM†	32.26	45.40	44658	30876	-
	FBA†	**7.37**	**4.79**	**7323**	**5206**	**7.03**
	BGM	17.32	21.21	27454	15397	14.25
	BGM_a	14.45	19.24	23314	13091	16.80
	Ours	**8.65**	**9.57**	**8736**	**6637**	**13.82**

Table 1: Quantitative evaluation on different datasets. † indicates methods that require a manual trimap.