Exploiting Aliasing for Manga Restoration (Supplemental Material)

Minshan Xie[1,2,*] Menghan Xia[1,*] Tien-Tsin Wong[1,2,†]
[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, CAS
{msxie, mhxia, ttwong}@cse.cuhk.edu.hk

1. Additional Results

For all the following results: the first column shows the degraded manga image. The second column shows the output restored by our method. The last column shows the binarized images of our restored results. All of the input images below are from the Manga109 dataset[1]. AisazuNihaIrarenai ©Yoshi Masako, AkkeraKanjinchou ©Kobayashi Yuki, Akuhamu ©Arai Satoshi, AosugiruHaru ©Okuda Momoko, HaruichibanNoFukukoro ©Yamada Uduki, HealingPlanet ©Sakurano Minene, MiraiSan ©Shintani Akihiro, MoeruOnisan ©Sato Tadashi.

2. Detail Network Architecture

2.1. Scale Estimation Network

The Scale Estimation Network (SE-Net) consists of four Convolutional Block Attention Module (CBAM) and an adaptive pooling layer where a global spatial pooling and a fully-connected layer are combined. Table 1 listed the detail network architecture of SE-Net and the CBAM is directly borrowed from [3].

2.2. Manga Restoration Network

In the paper, we have illustrated the overall architecture of Manga Restoration Network (MR-Net). Table 2 and Table 3 further illustrate the detail architecture of the Residual Attention Module (RAM)[2] and Convex Interpolation Module, respectively. The convex coefficient targets to refine the features based on the local features.

References

[1] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 1

[2] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 1

[3] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 1

*Equal contributions.
†Corresponding author.

Table 1: Detail network architecture of SE-Net.

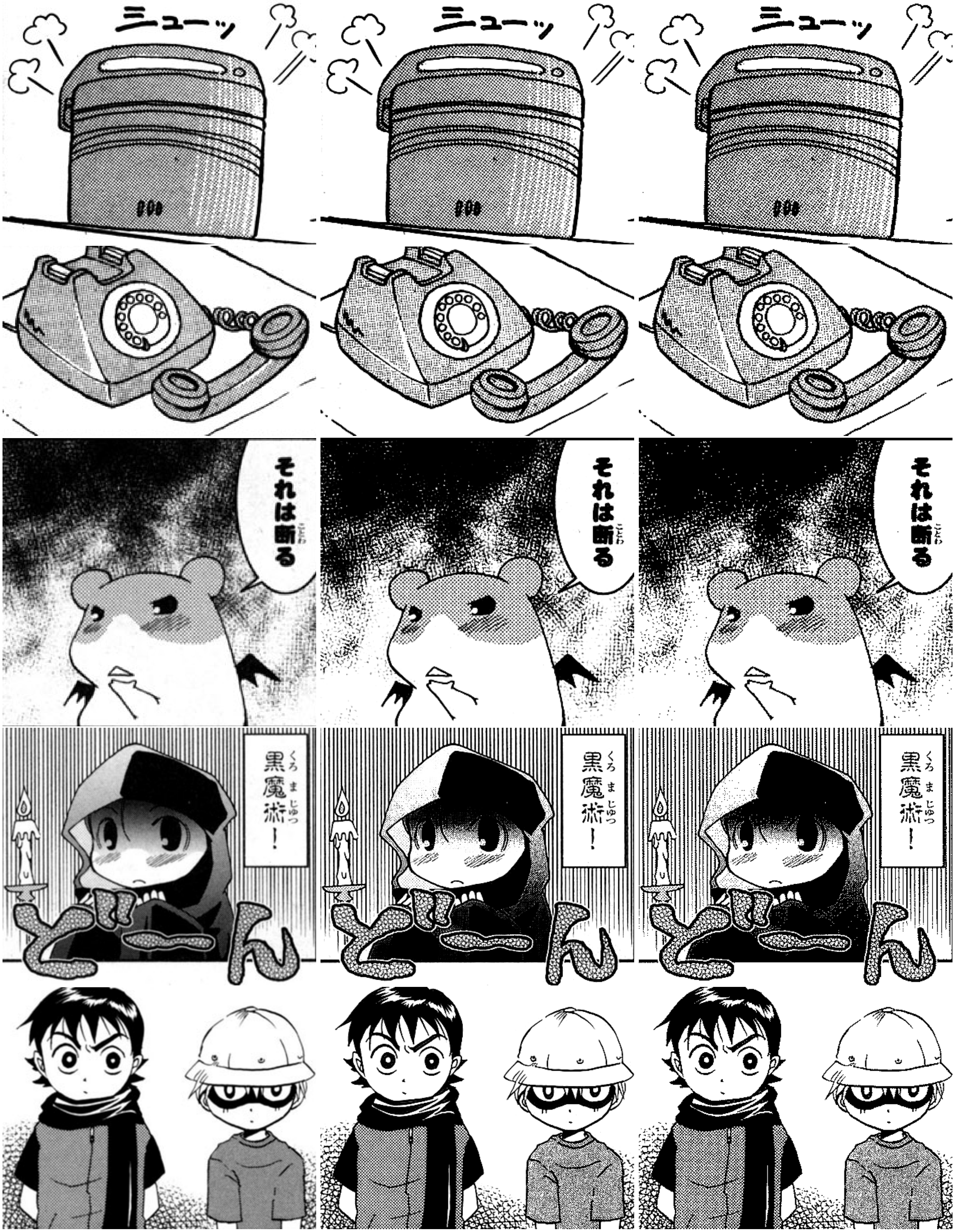
Layer Name			Layer Operations	Output Feature Map Size
Image Input Layer				$512 \times 512 \times 1$
Down-sampling	Conv		(N64,K3x3,S2,P1)	$512 \times 512 \times 64$
	ResCBAM	Conv	(N128,K3x3,S2,P1),LN,ReLU	$256 \times 256 \times 128$
		Conv	(N128,K3x3,S1,P1),LN,ReLU	
		CBAM	ChannelAttention,SpatialAttention	
		Conv(Shortcut)	(N128,K3x3,S2,P1),LN,ReLU	
	ResCBAM	Conv	N256,K3x3,S2,P1	$128 \times 128 \times 256$
		Conv	(N128,K3x3,S1,P1),LN,ReLU	
		CBAM	ChannelAttention,SpatialAttention	
		Conv(Shortcut)	(N128,K3x3,S2,P1),LN,ReLU	
	ResCBAM	Conv	N512,K3x3,S2,P1	$64 \times 64 \times 512$
		Conv	(N128,K3x3,S1,P1),LN,ReLU	
		CBAM	ChannelAttention,SpatialAttention	
		Conv(Shortcut)	(N128,K3x3,S2,P1),LN,ReLU	
	ResCBAM	Conv	N512,K3x3,S2,P1	$32 \times 32 \times 512$
		Conv	(N128,K3x3,S1,P1),LN,ReLU	
		CBAM	ChannelAttention,SpatialAttention	
		Conv(Shortcut)	(N128,K3x3,S2,P1),LN,ReLU	
	AVEPOOL		(1,1)	$1 \times 1 \times 512$
Output Layer	FC		N1	1

Table 2: Detail network architecture of Residual Attention Module.

Layer Name		Layer Operations	Output Feature Map Size
Image Input Layer			$256 \times 256 \times 32$
	Res1a	ResBlock(N32,K3x3,LN,LeakyReLU)	$256 \times 256 \times 32$
Branch1	Res1b	ResBlock(N32,K3x3,LN,LeakyReLU)	$256 \times 256 \times 32$
	Res1c	ResBlock(N32,K3x3,LN,LeakyReLU)	
Branch2	MAXPOOL	(K3x3,S2,P1)	$128 \times 128 \times 32$
	Res1d	ResBlock(N32,K3x3,LN,LeakyReLU)	
	MAXPOOL	(K3x3,S2,P1)	$64 \times 64 \times 32$
	Res1e	ResBlock(N32,K3x3,LN,LeakyReLU)	
	MAXPOOL	(K3x3,S2,P1)	$32 \times 32 \times 32$
	Res1f	ResBlock(N32,K3x3,LN,LeakyReLU)	
	UpsampleAdd	Upsample(Res1f)+Res1e	$64 \times 64 \times 32$
	Res1g	ResBlock(N32,K3x3,LN,LeakyReLU)	
	UpsampleAdd	Upsample(Res1g)+Res1d	$128 \times 128 \times 32$
	Res1h	ResBlock(N32,K3x3,LN,LeakyReLU)	
	UpsampleAdd	Upsample(Res1h)+Res1c	$256 \times 256 \times 32$
	Res1i	ResBlock(N32,K3x3,LN,LeakyReLU),Sigmoid	
Fusion		Branch1*Branch2+Branch2	$256 \times 256 \times 32$

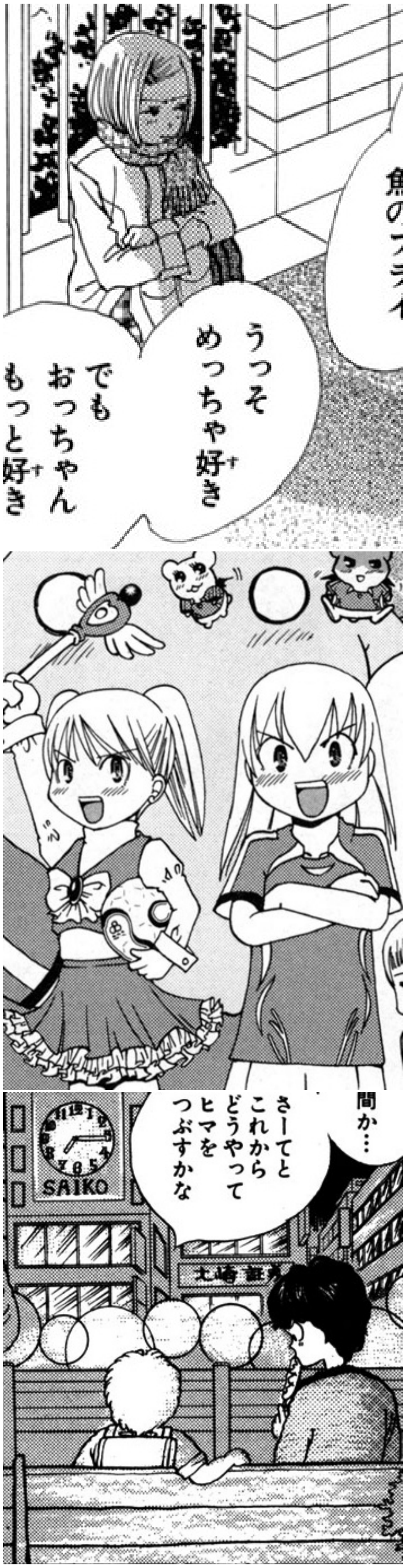
Table 3: Detail network architecture of Convex interpolation with scale s.

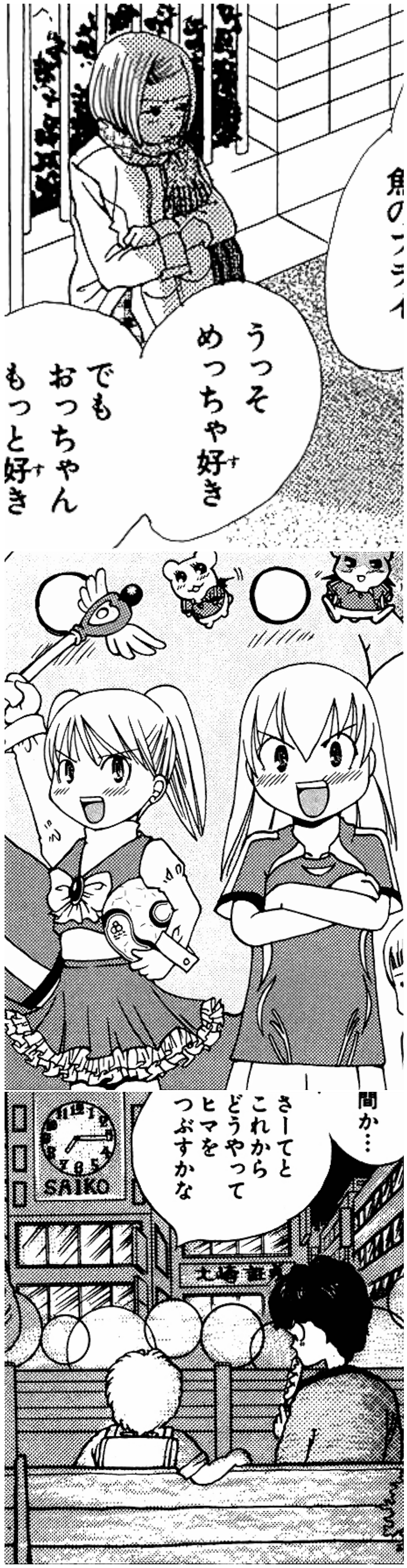
Layer Name		Layer Operations	Output Feature Map Size
Image Input Layer			$256 \times 256 \times 32$
Upsample		Upsample(s)	$(256 \times s) \times (256 \times s) \times 32$
Convexcoefficient	Conv	(N32,K3x3,S1,P1),ReLU	$(256 \times s) \times (256 \times s) \times 32$
	Conv	(N9,K1x1,S1,P0)	$(256 \times s) \times (256 \times s) \times 9$
	Softmax		
Fusion		Multiplication	$(256 \times s) \times (256 \times s) \times 32$

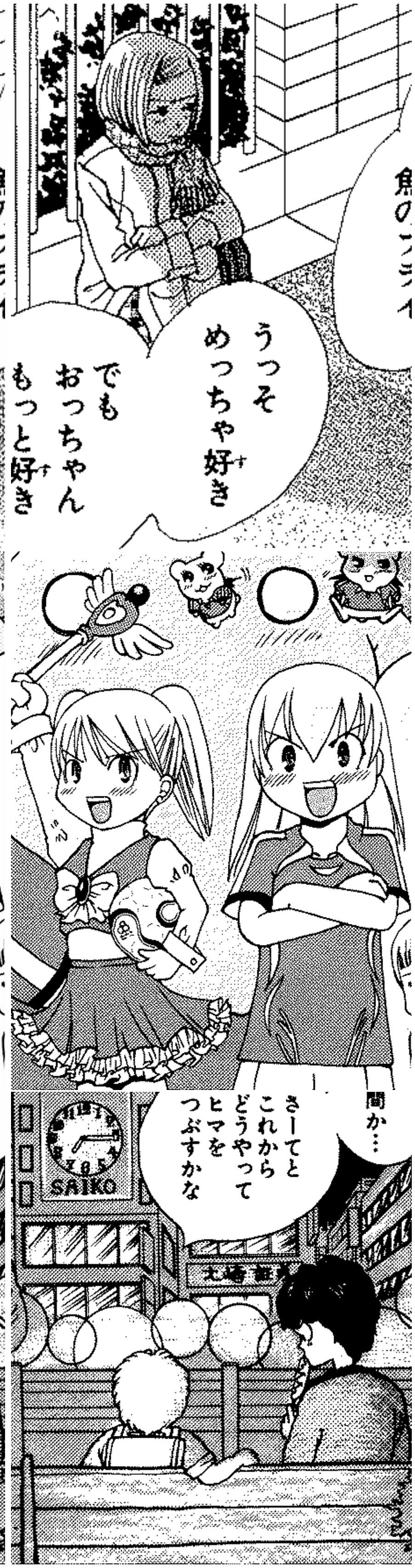

シューッ
それは断る
黒魔術！
どーん


さすが富士の樹海…
いいとも
ポリ ポリ
単純なヤツ
たんじゅん


あう!! とーちゃん とーちゃん! なんかして
なっ!! かえで!!
きのう やったから いい——っ
よかろう 我が力 お前らの為に 役立てよう
おぉーーー
ボス
ボス
ボス
チャ
カチャ
うるへーわ
鉄雄


魚のフライ
うっそ めっちゃ好き
でも おっちゃん もっと好き
間か…
さーてと これから どうやって ヒマを つぶすかな
SAIKO


魚のフライ
うっそ めっちゃ好き
でも おっちゃん もっと好き
間か…
さーてと これから どうやって ヒマを つぶすかな
SAIKO


魚のフライ
うっそ めっちゃ好き
でも おっちゃん もっと好き
間か…
さーてと これから どうやって ヒマを つぶすかな
SAIKO


それにしても
ここはなんじゃ――
グッ
それとも きいっちゃんと
同じ顔だから？