# High-Resolution Deep Image Matting

**Haichao Yu**[1], **Ning Xu**[2], **Zilong Huang**[1], **Yuqian Zhou**[1], **Humphrey Shi**[1,3]
[1]UIUC, [2]Adobe Research, [3]University of Oregon
{haichao3, yuqian2, hshi10}@illinois.edu, nxu@adobe.com, zilong.huang2020@gmail.com

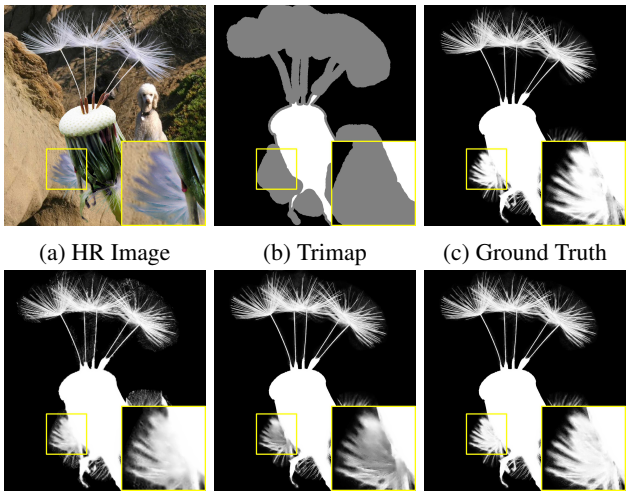
## Abstract

Image matting is a key technique for image and video editing and composition. Conventionally, deep learning approaches take the whole input image and an associated trimap to infer the alpha matte using convolutional neural networks. Such approaches set state-of-the-arts in image matting; however, they may fail in real-world matting applications due to hardware limitations, since real-world input images for matting are mostly of very high resolution. In this paper, we propose HDMatt, a first deep learning based image matting approach for high-resolution inputs. More concretely, HDMatt runs matting in a patch-based crop-and-stitch manner for high-resolution inputs with a novel module design to address the contextual dependency and consistency issues between different patches. Compared with vanilla patch-based inference which computes each patch independently, we explicitly model the cross-patch contextual dependency with a newly-proposed Cross-Patch Contextual module (CPC) guided by the given trimap. Extensive experiments demonstrate the effectiveness of the proposed method and its necessity for high-resolution inputs. Our HDMatt approach also sets new state-of-the-art performance on Adobe Image Matting and AlphaMatting benchmarks and produce impressive visual results on more real-world high-resolution images.

## Introduction

Image matting is a key technique in image and video editing and composition. Given an input image and a trimap indicating the background, forground and unknown regions, image matting is applied to estimate the alpha matte inside the unknown region to clearly separate the foreground from the background. Recently, many deep-learning-based methods (Xu et al. 2017; Lu et al. 2019; Hou and Liu 2019; Cai et al. 2019) have achieved significant improvements over traditional methods (Wang and Cohen 2007; Gastal and Oliveira 2010; Sun et al. 2004; Levin, Lischinski, and Weiss 2007; Grady et al. 2005). These deep learning methods (Xu et al. 2017; Lu et al. 2019; Hou and Liu 2019) mostly take *the whole images* and *the associated whole trimaps* as the inputs, and employ deep neural networks such as VGG (Simonyan and Zisserman 2014) and Xception (Chollet 2017) as their network backbones.





(a) HR Image (b) Trimap (c) Ground Truth
(d) ContextNet-DS (e) ContextNet-C (f) HDMatt (Ours)

Figure 1: Down-sampling (DS) and cropping (C) strategies applied to ContextNet (Hou and Liu 2019) on an HR image. DS results in blurry details, and trivial cropping causes cross-patch inconsistency. Our HDMatt resolves the above drawbacks. Best viewed when zoomed in with colors.

However, these methods may fail when dealing with high-resolution (HR) inputs. Image matting is frequently applied to HR images of size such as $5000 \times 5000$ or even higher in real-world applications. Due to hardware limitations like GPU memory, HR images cannot be directly handled by previous deep learning methods. Two common strategies of adapting those methods are down-sampling the inputs (He, Sun, and Tang 2010) or trivial patch-based inference. The former strategy results in losing most fine details, and the latter causes patch-wise inconsistency. Besides, HR images may have larger or even fully unknown regions within a patch. This further requires the models to understand contextual information from long-range patches for successful matting. A comparison run with ContextNet (Hou and Liu 2019) and our proposed method is shown in Fig. 1 to demonstrate these drawbacks.

In this paper, we propose HDMatt, a novel patch-based deep learning approach for high-resolution image matting. Specifically, we crop an input image into patches, and propose a Cross-Patch Contextual module (CPC) to explicitly

capture cross-patch long-range contextual dependency. For each given patch to be estimated (*i.e*. query patch), CPC samples other patches (*i.e*. reference patches) which are highly correlated with it within the image. Then CPC ensembles those correlated features towards a more faithful estimation.

To measure the correlation and ensemble the information effectively, inspired by traditional propagation-based methods (Levin, Lischinski, and Weiss 2007; Sun et al. 2004; Levin, Lischinski, and Weiss 2007), Trimap-Guided Non-Local operation (TGNL) is specifically designed for matting and embedded into the CPC. In particular, compared with the original non-local operation (Wang et al. 2018) applied to the whole patch, we leverage the pixel labels in the trimap to guide the correlation computing. Pixels in unknown regions in the query patch will be compared with three regions (*i.e*. foreground, background and unknown) in the reference patches separately, allowing an efficient information propagation across different pixel types.

The above mentioned designs are intended for cross-patch long-range dependency modeling. As a patch-based method, it is intrinsically indispensable for HR image matting. In summary, the contributions of this paper are three-folds:

- To our best knowledge, we are the first to propose a deep learning based approach to HR image matting, and makes high-quality HR matting practical in the real-world under hardware resources constraints.
- We propose a novel Cross-Patch Contextual module (CPC) to capture long-range contextual dependency between patches in our HDMatt approach. Inside the CPC, a newly-proposed Trimap-Guided Non-Local (TGNL) operation is designed to effectively propagate information from different regions in the reference patches.
- Both quantitatively and qualitatively, our method achieves new state-of-the-art performance in image matting on the Adobe Image Matting (AIM) (Xu et al. 2017), the AlphaMatting (Rhemann et al. 2009) benchmark, and our newly collected real-world HR image dataset.

## Related Work

### Image Matting

Before deep learning methods, there are two types of classic methods for matting task. One is sampling-based methods. Given an unknown pixel, these methods sample matched pixels from foreground and background regions and then find a proper combination of these pixels to predict alpha value of the unknown pixel. These methods include boundary sampling (Wang and Cohen 2007), ray casting sampling (Gastal and Oliveira 2010), etc. Another interesting sampling-based method is Divide-and-Conquer (Cao et al. 2016). In this paper, the authors proposed an adaptive patched-based method for HR image matting. To capture global information, they sample as context the pixels that are close to current pixel in RGBXY feature space in other patches. Although our method shares a similar sampling spirit with their method, ours is intrinsically different from theirs in many aspects. First, to our best knowledge, we are the first to use deep learning models to capture long-range contexts among patches for image matting. Second, we sample context patches in high-level feature space instead of pixels in RGBXY space. Thus, our method can better capture long-range context in semantic level.

Another type is propagation-based methods. These methods include Poisson equation based method (Sun et al. 2004), random walks for interactive matting (Grady et al. 2005) and closed-form matting(Levin, Lischinski, and Weiss 2007), which, based on local smoothness, formulates a cost function and then find the globally optimized alpha matte by solving linear equation system. Another popular propagation-based method is non-local image matting (Lee and Wu 2011; Chen, Li, and Tang 2013). For an unknown pixel to predict alpha value, this method sample pixels that match with current pixel in some feature space and make prediction with the the sampled pixels as context. Our method shares some spirit with this method in that our method make prediction by sampling context patches to capture long-range context.

Deep learning-based methods is another branch which has been widely explored. Cho *et al*. in (Cho, Tai, and Kweon 2016) proposed a novel deep learning method to combine alpha mattes from KNN matting (Chen, Li, and Tang 2013) and Closed-form matting. However, these methods are still restricted to specific type of images due to limited training set. The first large-scale image matting dataset is collected by Xu *et al*. (Xu et al. 2017). Building on this, they proposed a novel Deep Image Matting (DIM) model with refinement module. They achieved state-of-the-art performance on their collected test dataset. Since the availability of the large-scale dataset, deep learning methods for matting have been extensively explored. Lutz *et al*. proposed a generative adversarial network AlphaGan for image matting (Lutz, Amplianitis, and Smolic 2018). Hou *et al*. (Hou and Liu 2019) proposed ContextNet, which used dual-encoder structure to capture contextual and detail information and dual-decoder structure for foreground and alpha prediction. Among these methods, Unpooling is usually preferred to other upsampling methods like transposed convolution and bilinear upsampling. This is studied by Lu *et al*. (Lu et al. 2019). They further proposed IndexNet to dynamically determine the indices for unpooling operation. Recently, Li *et al*. (Li and Lu 2020) proposed GCAMatting, which utilizes pixel-wise contextual attention to capture long-range contexts. Though having impressive performance, these models will potentially fail on ultra-high-resolution image inference due to hardware limitation, thus not practical enough. Our proposed patch-based method works well on ultra-high-resolution images, and additional modeling of cross-patch dependency address the issues caused by crop-and-stitch manner.

### Non-local Operations

Non-local operations are widely used for various tasks such as video classification (Wang et al. 2018), object detection (Wang et al. 2018; Huang et al. 2020), semantic segmentation (Fu et al. 2019; Huang et al. 2020) and machine translation (Vaswani et al. 2017). Wang *et al*. (Wang et al. 2018) proposed a group of non-local operations to cap-

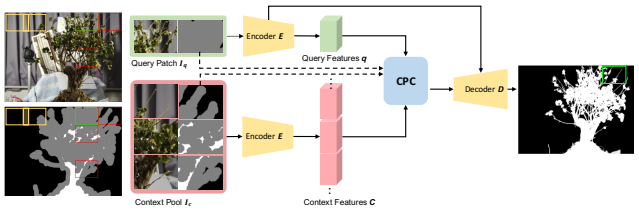



Figure 2: An overview of our proposed HDMatt approach. It works on patches and is basically an encode-decoder structure. Query patch concatenated with its associated trimap is fed into the encoder. The patches in the context pool and their trimaps are also fed into the encoder shared weights with $E$. The extracted features go through the Cross-Patch Context (CPC) module. Afterwards, the output feature of CPC is fed into the decoder for alpha estimation of the query patch. The green and red boxes are query and context patches during training. The yellow boxes are two consecutive query patches during test.

ture long-range context. Their method achieved impressive results on video classification task. Based on that, to reduce memory consumption inside the non-local operations, Huang *et al*. (Huang et al. 2020) used stacked criss-corss attention to mimic the non-local operations. DANet (Fu et al. 2019) used channel-wise and spatial attention to capture long-range dependency along both channel and spatial dimensions. In this paper, we are aware that long-range context dependency is potentially necessary for high-resolution images, especially those with large unknown areas. Therefore, we further develop the non-local module from (Wang et al. 2018) to make it adaptive to cross-patch modeling (*i.e*., CPC) and trimap guidance (*i.e*., TGNL). Our new state-of-the-art experimental results indicate the promising directions of adapting non-local operations to image matting.

## The Proposed HDMatt Approach

To handle high-resolution image matting, our method first crops an input image and trimap into patches (Sec. ) and then estimates the alpha values of each patch. Only using information from a single patch will cause information loss and prediction inconsistency between different patches. Therefore we propose a novel Cross-Patch Context Module (CPC) (Sec. ) to leverage cross-patch information for each query (current) patch effectively. Finally, the estimated alpha value of each patch is stitched together to output the final alpha matte of the whole image. The network structure and loss function are described in Sec. . Fig. 2 illustrates the framework of our method.

### Patch Cropping and Stitching

Given a training image and trimap, our method randomly samples image patches and their corresponding trimaps of different sizes (*e.g*. $320 \times 320, 480 \times 480, 640 \times 640$) at different locations and then they are resized to a fixed size $320 \times 320$. During inference, the whole test image $I$ and trimap $T$ are first cropped into overlapping patches (See the two yellow patches in Fig. 2 as an example). For those patches exceeding the image boundary, we utilize reflective

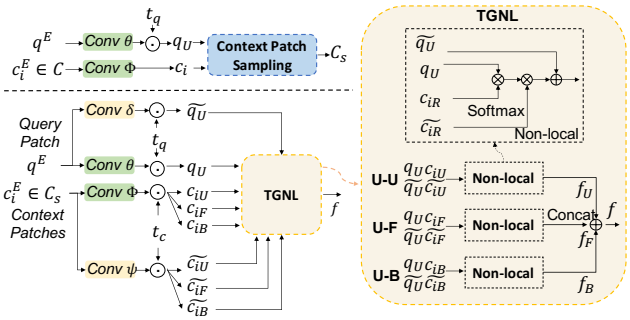



Figure 3: The workflow of the Cross-Patch Context (CPC) module. It consists of a context patch sampling, and a Trimap-guided Non-Local (TGNL) operation. $\otimes$: matrix multiplication, $\oplus$: feature map concatenation, $\odot$: element-wise multiplication.

padding to fill up the pixels. The small overlapping region is helpful to avoid boundary artifacts when stitching the alpha mattes of nearby patches together. In particular, we design a linear blending function to merge the estimated alphas of overlapping regions between nearby patches for a smooth transition. The blending weight of each pixel is proportional to its distance to the patch boundaries.

### Cross-Patch Context Module

Our method leverages cross-patch information for high-resolution image matting. For each query patch, Instead of using all the information from the other patches, we propose an effective sampling strategy to only sample top-$K$ patches which are most relevant and useful to the query patch, and thus save computation greatly without decreasing the accuracy. In addition, in contrast to most prior works that only concatenate the trimap with image as input, our method uses a more effective and explicit way to leverage trimap as guidance to propagate information from other regions.

**Context Patch Sampling** Given a query patch $I_q$, to select top-$K$ patches from $N$ context patches $I_{c_i}, c_i \leq N$, our method first computes the correlation between the *unknown regions* of $I_q$ and the *whole regions* of each $I_{c_i}$. Specifically, as shown in Fig. 2, both $I_q$ and $I_{c_i}$ along with their trimaps are fed into an encoder to extract higher-level feature maps (For simplicity, let $\boldsymbol{q}^E$ and $\boldsymbol{c}_i^E$ denote their corresponding feature maps). Then $\boldsymbol{q}^E$ and $\boldsymbol{c}_i^E$ are further embedded by two convolutional layers $\theta$ and $\phi$ into $\boldsymbol{q}$ and $\boldsymbol{c}_i$, as shown in Fig. 3. To get the unknown regions (U) of the new feature $\boldsymbol{q}$, we use the downsampled trimap to zero out the foreground (F) and background (B) regions of $\boldsymbol{q}$, *i.e*., $\boldsymbol{q}_U = \boldsymbol{q} \odot \mathbf{1}_{s \in U}$, where $s$ is the pixel index. Then the correlation between the two feature maps $\boldsymbol{q}_U$ and $\boldsymbol{c}_i$ can be computed by summing over the dot product of their features at each location, *i.e*.

$$h(\boldsymbol{q}_U, \boldsymbol{c}_i) = \sum_{s,s'} \boldsymbol{q}_{U,s} \cdot \boldsymbol{c}_{i,s'}, \tag{1}$$

where $s$ and $s'$ are the pixel positions in $\boldsymbol{q}_U$ and $\boldsymbol{c}_i$ respectively. The correlations between query patch and all $N$ context patches are normalized via the softmax operation, which

results in a similarity score for each context patch, *i.e*.

$$d_{c_i} = \frac{e^{h(\boldsymbol{q}_U, \boldsymbol{c}_i)}}{\sum_{\hat{i}} e^{h(\boldsymbol{q}_U, \boldsymbol{c}_{\hat{i}})}}, \tag{2}$$

A higher score indicates that the candidate context patch is more relevant to the unknown regions of the query patch, and thus should play a more important role in information propagation. During inference, we rank all the context patches according to their similarity scores $d_{c_i}$ and only select the top-$K$ context patches for feature propagation. Empirically we find that $K = 3$ can already achieve comparable accuracy compared to utilizing all N context patches.

**Trimap-Guided Non-Local (TGNL) Operation** To propagates the useful information of context patches to the query patch, we leverage non-local operation (Wang et al. 2018; Oh et al. 2019) which were proposed for different tasks. In addition, for the matting problem, trimap provides very useful information about the foreground, background and unknown regions. A unknown pixel which is similar to a foreground pixel is more likely to be foreground pixel than background pixel, and vice versa. Therefore, it is important to propagate the context information from different regions indicated by the trimaps. While recent deep-learning-based matting methods usually concatenate the trimaps as input, which makes it difficult for their methods to leverage such information precisely.

To remedy this issue, our method incorporates the trimap information into the non-local operation. Specifically, our method compares the unknown region (U) of the query patch with the unknown (U), foreground (F) and background (B) regions of the context patches separately. Then the correlation features from the three different relationships (*i.e*. U-U, U-F, and U-B) are concatenated together and used as the decoder input.

As shown in Fig. 3, the query feature $\boldsymbol{q}^E$ from the encoder output is further embedded by two convolutional layers $\theta$ and $\delta$ into a key feature map $\boldsymbol{q}$ and a value feature map $\widetilde{\boldsymbol{q}}$. Similarly, every sampled context patch feature $\boldsymbol{c}_i^E$ is embedded by two convolutional layers $\theta$ and $\phi$ into a key feature map $\boldsymbol{c}_i$ and a value feature map $\widetilde{\boldsymbol{c}_i}$. We then use the downsampled query trimap to extract the feature maps of the unknown region, *i.e*. $\boldsymbol{q}_U = \boldsymbol{q} \odot \mathbf{1}_{s \in U}$, and $\widetilde{\boldsymbol{q}_U} = \widetilde{\boldsymbol{q}} \odot \mathbf{1}_{s \in U}$. Similarly, we use the context trimap to extract the feature maps of the three regions separately, *i.e*. $\boldsymbol{c}_{i,R} = \boldsymbol{c}_i \odot \mathbf{1}_{s \in R}$ and $\widetilde{\boldsymbol{c}_{i,R}} = \widetilde{\boldsymbol{c}_i} \odot \mathbf{1}_{s \in R}$, where $R \in \{\mathrm{U}, \mathrm{F}, \mathrm{B}\}$. Then the propagated features by comparing the U region of the query patch with the R region of all sampled context patches can be computed as follows,

$$\boldsymbol{f}_{R,s} = \widetilde{\boldsymbol{q}_{U,s}} + \sum_{i,s'} \frac{e^{(\boldsymbol{q}_{U,s} \cdot \boldsymbol{c}_{i,R,s'})}}{\sum_{\hat{i},\hat{s'}} e^{(\boldsymbol{q}_{U,s} \cdot \boldsymbol{c}_{\hat{i},R,\hat{s'}})}} \widetilde{\boldsymbol{c}_{i,R,s'}}, \tag{3}$$

where $s$ is a pixel location of the aggregated feature map $\boldsymbol{f}_{R,s}$. Finally, the aggregated feature maps of all three regions $\boldsymbol{f}_U$, $\boldsymbol{f}_F$ and $\boldsymbol{f}_B$ are concatenated together as the module output and are used for the decoder. It is possible that some context trimaps may not contain all the three regions, but our Eqn. 3 still works, and empirically we find that our model have robust results for such cases.

| | Methods | SAD | MSE | Grad | Conn |
|---|---|---|---|---|---|
| Whole | AlphaGAN | 52.4 | 30 | 38 | 53 |
| | DIM | 50.4 | 14 | 31.0 | 50.8 |
| | IndexNet | 45.8 | 13 | 25.9 | 43.7 |
| | AdaMatting | 41.7 | 10 | 16.8 | - |
| | ContextNet | 35.8 | 8.2 | 17.3 | 33.2 |
| | GCAMatting | 35.3 | 9.1 | 16.9 | 32.5 |
| Patch | IndexNet | 54.5 | 16.8 | 31.8 | 54.0 |
| | ContextNet | 37.6 | 8.7 | 18.7 | 34.8 |
| | GCAMatting | 36.9 | 8.9 | 17.1 | 34.1 |
| | HDMatt (Ours) | **33.5** | **7.3** | **14.5** | **29.9** |

Table 1: The quantitative results on AIM testset (Xu et al. 2017). Methods in the section "Whole" take as input the whole image, which are also the proposed testing strategy for these methods. We also test several methods on overlapping patches with the same patch size as our method.

## Network Structure and Losses

Fig. 2 illustrates the overall framework of our method. The encoder $E$ consists of a backbone feature extractor ResNet-34 (He et al. 2016) and Atrous Spatial Pyramid Pooling (ASPP) (Chen et al. 2017). Pooling outputs from encoder blocks are skip-connected to the corresponding decoder layers. Following (Xu et al. 2017), we use unpooling operation for feature map upsampling in decoder, which is verified to be more effective (Lu et al. 2019) for matting-related tasks.

For fair comparison, we use the same loss function as in (Xu et al. 2017) to train the whole network end-to-end. It is an average of alpha loss $\mathcal{L}_\alpha$ and composite loss $\mathcal{L}_c$. Formally, for each pixel, the losses are defined as

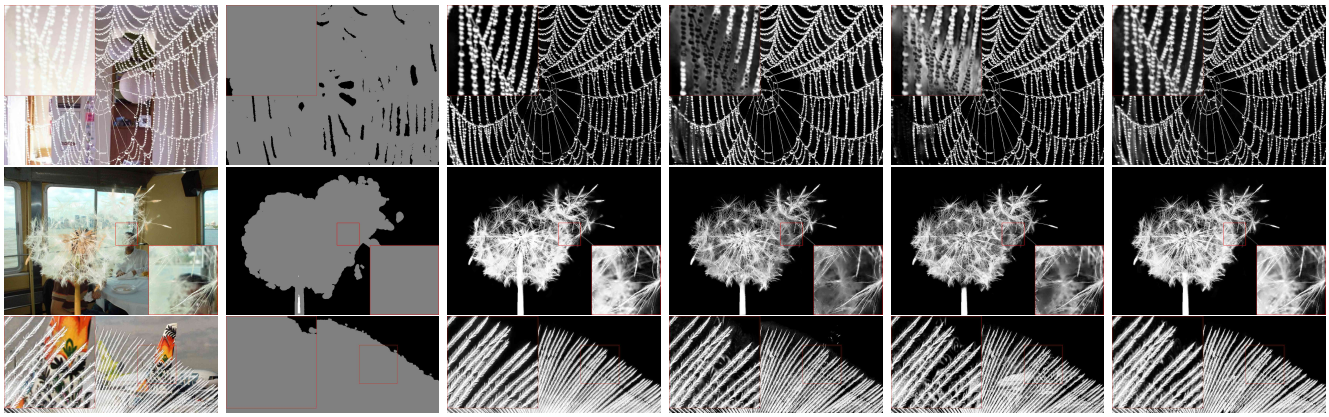
$$\begin{aligned} \mathcal{L}_{overall} &= 0.5\mathcal{L}_\alpha + 0.5\mathcal{L}_c, \\ \mathcal{L}_\alpha &= \sqrt{||\alpha_{gt} - \alpha_q||^2 + \epsilon}, \\ \mathcal{L}_c &= \sqrt{||I_q - (\alpha_q I_q^F + (1-\alpha_q) I_q^B)||^2 + \epsilon}, \end{aligned} \tag{4}$$

where $\alpha_{gt}$ is the ground truth alpha matte, $I_q^F$ and $I_q^B$ are the ground truth background and foreground images to composite $I_q$, and $\epsilon$ is the slacking factor to be set as $10^{-12}$. As mentioned earlier, smooth blending is employed on the overlapping region between neighboring patches during test, and thus pixels along the boundary regions of each training patch should be weighted accordingly. Therefore, we employ the same blending function as a weighted mask to the loss $\mathcal{L}_{overall}$.

# Experiments

## Dataset

We trained our models on Adobe Image Matting (AIM) dataset (Xu et al. 2017). AIM has 431 foreground images for training, each of which has a fine-annotated alpha matte. We augmented the data following Tang *et al*. (Tang et al. 2019). Specifically, we first augmented the ground truth alpha matte by compositing two foreground images. To generate the associated trimaps, we randomly dilated ground truth alpha mattes. The synthetic training images will be the compositions of a foreground images in augmented AIM training



(a) Image (b) Trimap (c) Ground Truth (d) ContexNet (e) IndexNet (f) HDMatt (Ours)

Figure 4: Visual results on AIM testset. Our method has an obvious advantage in large unknown regions.

set with a randomly sampled background image from COCO dataset (Lin et al. 2014). For each training image, we sampled a pixel from unknown area and crop a square patch centered at this pixel with side length in $\{640, 480, 320\}$. Then, these patches are resized to $320 \times 320$ and randomly flipped horizontally and rotated by an angle less than $15°$. We tested our models on AIM testset, AlphaMatting (Rhemann et al. 2009) and newly collected HR real-world images.

## Implementation

To optimize the entire framework, we used a two-stage training strategy. On the first stage, we pre-trained a Resnet-34 classification model on ImageNet (Deng et al. 2009). Then all layers from the model before the fully-connected layer were used as our matting encoder. We changed the last two convolution blocks with dilated convolution to keep HR feature maps as in (Chen et al. 2017). On the second stage, the encoder-decoder model together with CPC module was jointly trained end-to-end for image matting. During both training and testing, we set context patch number $K = 3$ (*i.e.* top-3 patches) by default in all the following experiments unless otherwise stated. We also make an ablation study on different choices of $K$. When input image is of ultra-high resolution, we set the candidate context patches number $N = 30$ to save computational resources. Adam (Kingma and Ba 2014) optimizer was used with initial learning rate $0.5 \times 10^{-3}$ and decayed by cosine scheduler. The model is trained for $200k$ steps with batch size 32 and weight decay $10^{-4}$.

## Adobe Image Matting Benchmark

We tested our methods on Adobe Image Matting testset. This dataset has 1000 test images with alpha matte, which are synthesized from 50 foreground images and 1000 background images from Pascal VOC (Everingham et al. 2015). We use four evaluation metrics, SAD, MSE, Gradient and Connectivity as in (Rhemann et al. 2009). Tab. 1 shows our results together with other state-of-the-arts. We achieve the best performance with other top-ranked methods in all the evaluation metrics. In addition to the results on whole images, we also test several methods in a naive patch-wise manner (Note that they use the same patch size and blending function as our method). It is clear that their results on patches are worse than those on whole images, indicating that a naive adaptation of previous methods under limited computation resources will have degraded performance.

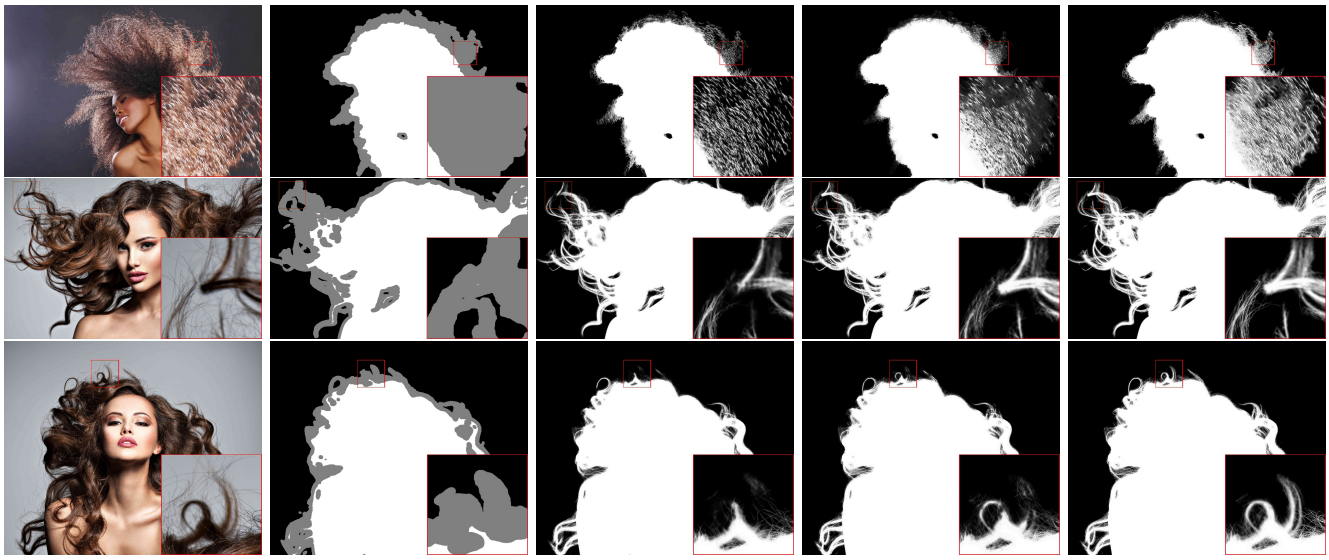
In Fig. 4, we qualitatively compare with recent state-of-the-art matting methods including IndexNet (Lu et al. 2019) and ContextNet (Hou and Liu 2019). For IndexNet, we used the official released code. For ContextNet, we used the publicly available test results provided by the authors. It demonstrates that our method works better especially in large unknown regions where little foreground or background information is available. Both IndexNet and ContextNet take whole image as input. For each pixel, they capture contextual dependency within a fixed receptive field formed by a stack of convolutional and pooling layers. Although this may capture local context, but it is not very effective to build a strong long-range contextual dependency. In contrast, our method globally samples context patches that contain useful background and foreground information and explicitly correlates them with the given patch.

## AlphaMatting Benchmark

AlphaMatting (Rhemann et al. 2009) is a popular image matting benchmark while all of its test images are around $800 \times 600$. Although our method is particularly effective for high-resolution images, our method still achieves the top-1 performance under SAD, MSE and Gradient metrics among all the published methods at the time of submission, which demonstrates that our method is general and effective for images of various resolutions. See Tab. 2 for details.

## Real-world Images

Although our method achieves the state-of-the-art results on existing benchmarks, the advantages of our method are not fully reflected given that the test images of existing benchmarks are not very high resolution. Therefore we collect dozens of online HR images with resolution up to $6000 \times 6000$. In Fig. 5, we test IndexNet (Lu et al. 2019) and ContextNet (Hou and Liu 2019) with the whole image as input. Since these images are too large to be fed into a



(a) HR Image (b) Trimap (c) ContexNet (d) IndexNet (e) HDMatt (Ours)

Figure 5: Visual comparison on real-world HR images. We test ContextNet and IndexNet on CPUs on the full images. Zoom in for details. Image sizes from top to bottom: $5616 \times 3744$, $5779 \times 3594$, $4724 \times 3929$.

| | Average | | | | Troll | | | Doll | | | Donkey | | | Elephant | | | Plant | | | Pineapple | | | Plastic bag | | | Net | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | All | S | L | U | S | L | U | S | L | U | S | L | U | S | L | U | S | L | U | S | L | U | S | L | U | S | L | U |
| *Ours* | **5** | 6.3 | **3.9** | **5** | 9.5 | 10 | 11 | 4.7 | **4.8** | 5.8 | 2.9 | 3 | **2.6** | 1.1 | 1.2 | **1.3** | 5.2 | **5.9** | **6.7** | 2.4 | **2.6** | **3.1** | **17** | **17** | 17 | 22 | 22 | 23 |
| AM | 6.9 | 5.9 | 6 | 8.9 | 10 | 11 | 11 | 4.9 | 5.4 | 6.6 | 3.6 | 3.4 | 3.4 | 0.9 | 0.9 | 1.8 | 4.7 | 6.8 | 9.3 | 2.2 | 2.6 | 3.3 | 19 | 20 | 19 | 18 | 19 | 19 |
| SN | 7.3 | 5.4 | 6.9 | 9.8 | 9.1 | 9.7 | 9.8 | 4.3 | 4.8 | 5.1 | 3.4 | 3.7 | 3.2 | 0.9 | 1.1 | 2 | 5.1 | 6.8 | 9.7 | 2.5 | 4 | 3.7 | 19 | 19 | 19 | 20 | 22 | 23 |
| GM | 8.4 | 9 | 5.8 | 10 | 8.8 | 9.5 | 11 | 4.9 | 4.8 | 5.8 | 3.4 | 3.7 | 3.2 | 1.1 | 1.2 | 1.3 | 5.7 | 6.9 | 7.6 | 2.8 | 3.1 | 4.5 | 18 | 19 | 19 | 21 | 22 | 25 |

Table 2: Top-4 methods on AlphaMatting benchmark (Rhemann et al. 2009). Our method achieves best overall performance in SAD. AM: AdaMatting (Cai et al. 2019). SN: SampleNet (Tang et al. 2019). GM: GCAMatting (Li and Lu 2020).

single GPU, we use CPU instead, which has a prohibited long inference time for each test image. From the results we can see that our method extracts finer and more accurate details than the other two state-of-the-art matting methods, while having a much faster inference speed. We also notice that our method also misses some finest details. A possible explanation for this issue is that the AIM training set lacks similar training examples with such small details.

In Fig. 6, we test the previous matting methods with more realistic settings. The first setting is to run inference on the downsampled images (*i.e.* $1024\times1024$) and then the predicted results are upsampled to the original resolution. The second setting is to run both methods in a crop-and-stitch manner with the same patch size (*i.e.* $320\times320$) and smooth blending function as our method. From the results of the prior works, we can clearly see that the downsampling strategy will lose a lot of details and produce blurry results while the naive-patch strategy will cause inconsistent results across patches due to the lack of cross-patch and long-range information. In contrast, our method is able to produce high-quality alpha mattes on the high-resolution images.

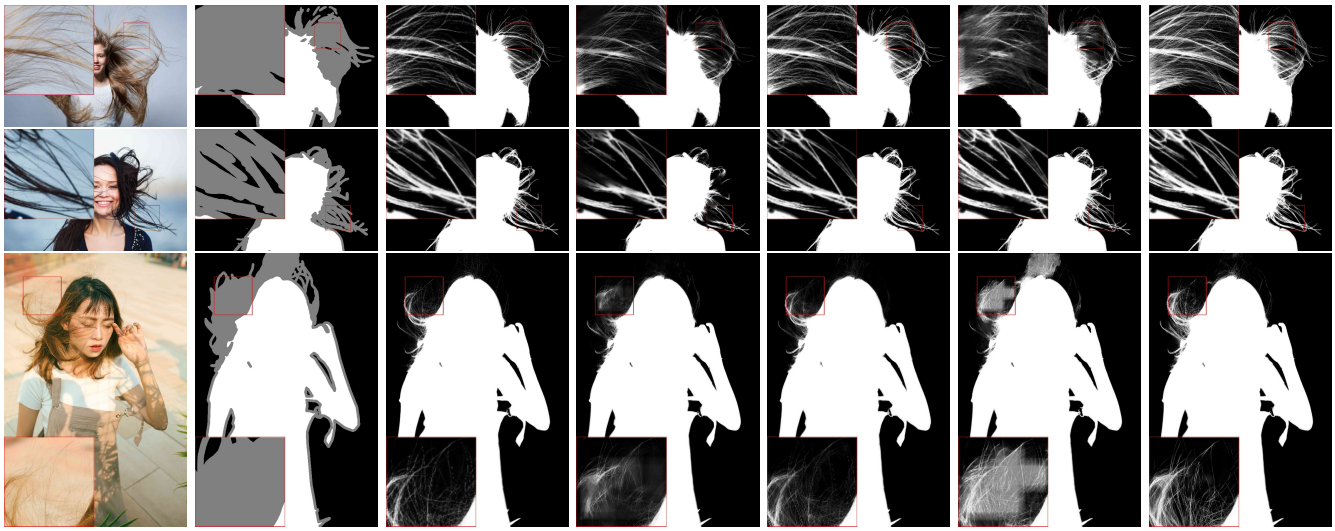
## Attention Visualization on Context Patches

In Fig. 7, we visualize the attention maps of the selected context patches for some given query patch. For each query patch in green box, we first select the top-3 context patches indicated by the red boxes. We then randomly sample a pixel in query patch marked by blue circle and show its attention/correlation maps on the context patches. Brighter color represents larger attention weights. It is worth noting that our method could select context patches which are far away from the query patch, which cannot be achieved with conventional CNNs with a fixed receptive field. Also, the attention weights indicate that our method can effectively leverage the information of similar pixels in the context patches.

## Ablation Study

**Module Selection** To investigate how each module contributes to the overall performance of our method, we make an ablation study on the proposed modules in AIM dataset. The results are summarized in Tab. 3. The baseline model is an encoder-decoder structure without CPC module (Model A), and it shares the same backbone with our proposed model. Compared with the baseline, CPC helps gain a large performance improvement. Substituting normal Non-local operation (Model B) with TGNL (Model C) inside the CPC further boosts the performance of the overall model.

**Patch Size** In this section, we make an ablation study to show the dependency of CPC on patch size of query and context patches. We test our CPC (TGNL) model with patch sizes 320, 480, 640 and $\{320, 480, 640\}$ as shown in the sec-



(a) HR Image (b) Trimap (c) ContexNet-DS (d) ContexNet-C (e) IndexNet-DS (f) IndexNet-C (g) HDMatt (Ours)

Figure 6: Visual comparison on real-world HR images. Zoom in for details. Image sizes from top to bottom: $4601 \times 3069$, $5760 \times 3840$, $3840 \times 5760$. DS: Down-sampling. C: Patch-based cropping.

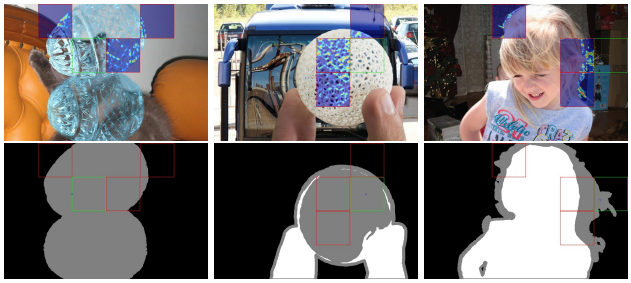


Figure 7: CPC attention visualization. Top row: whole image. Bottom row: whole trimap. Green box: query patch with the sampled pixel in blue circle. Red boxes: context patches. Zoom in for more details.

ond section of Tab. 3. The maximum SAD difference among various patch size settings is only 0.7. This implies a nice property that CPC module is agnostic to patch size to some extent. This is because our model is already designed to capture long-range cross-patch context by CPC and larger patch size does not make the model capture much extra context. This property is useful in real applications since given limited computation resources, our method can run on smaller patches without sacrificing the performance.

**Context Patch Number** During testing, to predict alpha matte for the query patch, we sample $K$ patches in the context pool. In this section, we explore how $K$ impacts test performance. We trained and tested the CPC (TGNL) model with $K = 1, 3, 5, 7$ and all the context patches. As shown in Tab. 3, even with a single context patch, our method already achieves significantly improvement over the baseline model (no CPC), showing the effectiveness of our CPC module. When all patches in the original image are used as context patches, the model yields the best performance of SAD 32.2. Since when $K \geq 3$ the model performs stably, we choose $K = 3$ in our experiments considering the trade-off of computational cost and performance.

| Models | SAD | MSE ($10^{-3}$) | Grad | Conn |
|---|---|---|---|---|
| Model A (3, 320) | 41.1 | 10.1 | 19.1 | 38.7 |
| Model B (3, 320) | 35.6 | 8.0 | 17.9 | 33.1 |
| Model C (3, 320) | 33.5 | 7.3 | 14.5 | 29.9 |
| Model C (3, 480) | 33.0 | 7.0 | 14.2 | 29.3 |
| Model C (3, 640) | 33.1 | 7.0 | 14.4 | 29.6 |
| Model C (3, MS) | 32.8 | 6.9 | 14.2 | 29.1 |
| Model C (1, 320) | 34.9 | 7.4 | 14.8 | 30.6 |
| Model C (5, 320) | 33.3 | 7.2 | 14.3 | 29.7 |
| Model C (7, 320) | 33.2 | 7.2 | 14.3 | 29.6 |
| Model C (All, 320) | 32.2 | 7.2 | 14.2 | 29.5 |

Table 3: The ablation study on module, patch size and sampled patch number. Model A: without CPC. Model B: with CPC (Non-local). Model C: with CPC (TGNL). Each model name is followed by (sampled patch number $K$, patch size). MS: $\{320, 480, 640\}$.

## Conclusion

In this paper, we propose HDMatt, a first deep learning based model for HR image matting. Instead of taking the whole image as input for inference, we apply a patch-based training and inference strategy to overcome hardware limitations for HR inference. To maintain a high-quality alpha matte, we explicitly model the cross-patch long-range context dependency using a Cross-Patch Context module. Our method achieves new state-of-the-arts on AIM, AlphaMatting benchmarks and produce impressive visual results on real-world high-resolution images.

## References


Cai, S.; Zhang, X.; Fan, H.; Huang, H.; Liu, J.; Liu, J.; Liu, J.; Wang, J.; and Sun, J. 2019. Disentangled Image Matting. *arXiv preprint arXiv:1909.04686* .

Cao, G.; Li, J.; He, Z.; and Chen, X. 2016. Divide and conquer: a self-adaptive approach for high-resolution image matting. In *2016 International Conference on Virtual Reality and Visualization (ICVRV)*, 24–30. IEEE.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* .

Chen, Q.; Li, D.; and Tang, C.-K. 2013. KNN matting. *IEEE transactions on pattern analysis and machine intelligence* 35(9): 2175–2188.

Cho, D.; Tai, Y.-W.; and Kweon, I. 2016. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision*, 626–643. Springer.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111(1): 98–136.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3146–3154.

Gastal, E. S.; and Oliveira, M. M. 2010. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, 575–584. Wiley Online Library.

Grady, L.; Schiwietz, T.; Aharon, S.; and Westermann, R. 2005. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, 423–429.

He, K.; Sun, J.; and Tang, X. 2010. Guided image filtering. In *European conference on computer vision*, 1–14. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, Q.; and Liu, F. 2019. Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation. In *IEEE International Conference on Computer Vision*.

Huang, Z.; Wang, X.; Wei, Y.; Huang, L.; Shi, H.; Liu, W.; and Huang, T. S. 2020. CCNet: Criss-Cross Attention for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* ISSN 1939-3539. doi: 10.1109/TPAMI.2020.3007032.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lee, P.; and Wu, Y. 2011. Nonlocal matting. In *CVPR 2011*, 2193–2200. IEEE.

Levin, A.; Lischinski, D.; and Weiss, Y. 2007. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence* 30(2): 228–242.

Li, Y.; and Lu, H. 2020. Natural Image Matting via Guided Contextual Attention. *arXiv preprint arXiv:2001.04069* .

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lu, H.; Dai, Y.; Shen, C.; and Xu, S. 2019. Indices Matter: Learning to Index for Deep Image Matting. *arXiv preprint arXiv:1908.00672* .

Lutz, S.; Amplianitis, K.; and Smolic, A. 2018. AlphaGAN: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088* .

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. *arXiv preprint arXiv:1904.00607* .

Rhemann, C.; Rother, C.; Wang, J.; Gelautz, M.; Kohli, P.; and Rott, P. 2009. A perceptually motivated online benchmark for image matting. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1826–1833. IEEE.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Sun, J.; Jia, J.; Tang, C.-K.; and Shum, H.-Y. 2004. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, 315–321. ACM.

Tang, J.; Aksoy, Y.; Oztireli, C.; Gross, M.; and Aydin, T. O. 2019. Learning-based sampling for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3055–3063.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, J.; and Cohen, M. F. 2007. Optimized color sampling for robust matting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Xu, N.; Price, B.; Cohen, S.; and Huang, T. 2017. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2970–2979.