Hao Lu, Yutong Dai, Chunhua Shen, Songcen Xu
Indices matter: Learning to index for deep image matting
Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019 / vol.2019-October, pp.3265-3274

Published version at: http://dx.doi.org/10.1109/ICCV.2019.00336

Indices Matter: Learning to Index for Deep Image Matting

Hao Lu[†] Yutong Dai[†] Chunhua Shen[†*] Songcen Xu[‡]
[†]The University of Adelaide, Australia [‡]Noah's Ark Lab, Huawei Technologies
e-mail: {hao.lu, yutong.dai, chunhua.shen}@adelaide.edu.au

Abstract

We show that existing upsampling operators can be unified with the notion of the index function. This notion is inspired by an observation in the decoding process of deep image matting where indices-guided unpooling can recover boundary details much better than other upsampling operators such as bilinear interpolation. By looking at the indices as a function of the feature map, we introduce the concept of learning to index, and present a novel index-guided encoder-decoder framework where indices are self-learned adaptively from data and are used to guide the pooling and upsampling operators, without the need of supervision. At the core of this framework is a flexible network module, termed IndexNet, which dynamically predicts indices given an input. Due to its flexibility, IndexNet can be used as a plug-in applying to any off-the-shelf convolutional networks that have coupled downsampling and upsampling stages.

We demonstrate the effectiveness of IndexNet on the task of natural image matting where the quality of learned indices can be visually observed from predicted alpha mattes. Results on the Composition-1k matting dataset show that our model built on MobileNetv2 exhibits at least 16.1% *improvement over the seminal VGG-16 based deep matting baseline, with less training data and lower model capacity. Code and models has been made available at:* https://tinyurl.com/IndexNetV1.

1. Introduction

Upsampling is an essential stage for most dense prediction tasks using deep convolutional neural networks (CNNs). The frequently used upsampling operators include transposed convolution [50, 32], unpooling [2], *periodic shuffling* [41] (also known as depth-to-space), and naive interpolation [30, 4] followed by convolution. These operators, however, are not general-purpose designs and often have different behaviors in different tasks.

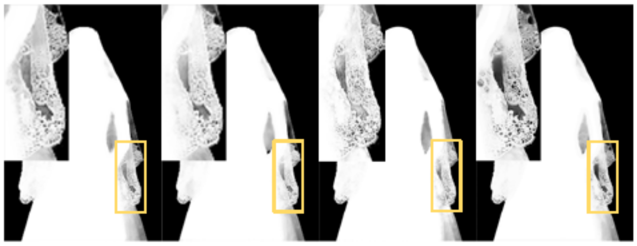

Figure 1: Alpha mattes of different models. From left to right, Deeplabv3+ [4], RefineNet [30], Deep Matting [49] and Ours. Bilinear upsampling fails to recover subtle details, but unpooling and our learned upsampling operator can produce much clear mattes with good local contrast.

The widely-adopted operator in semantic segmentation or depth estimation is bilinear interpolation, rather than unpooling. A reason is that the feature map generated by unpooling is too sparse, while bilinear interpolation is likely to generate the feature map that depicts semantically-consistent regions. This is particularly true for semantic segmentation and depth estimation where pixels in a region often share the same class label or have similar depth. However, bilinear interpolation performs much worse than unpooling in boundary-sensitive tasks such as image matting. A fact is that the leading deep image matting model [49] largely borrows the design from the SegNet [2], where unpooling is introduced. When adapting other state-of-the-art segmentation models, such as DeepLabv3+ [4] and RefineNet [30], to this task, unfortunately, we observe both DeepLabv3+ and RefineNet fail to recover boundary details (Fig. 1), compared to SegNet. This makes us to ponder over *what is missing* in these encoder-decoder models. After making a thorough comparison between different architectures and conducting ablative studies (Section 5.2), the answer is finally made clear—*indices matter*.

Compared to the bilinearly upsampled feature map, unpooling uses max-pooling indices to guide upsampling. Since boundaries in the shallow layers usually have the maximum responses, indices extracted from these responses record the boundary locations. The feature map projected by the indices thus shows improved boundary de-

[*] Corresponding author.

lineation. Above analyses reveal a fact that, different upsampling operators have different characteristics, and we expect a specific behavior of the upsampling operator when dealing with specific image content in a certain visual task.

It would be interesting to pose the question: *Can we design a generic operator to upsample feature maps that better predict boundaries and regions simultaneously?* A key observation of this work is that max unpooling, bilinear interpolation or other *upsampling operators are some forms of index functions*. For example, the nearest neighbor interpolation of a point is equivalent to allocating indices of one to its neighbor and then map the value of the point. In this sense, indices are models [24], therefore indices can be modeled and learned. In this work, we model indices as a function of the local feature map and learn an index function to perform upsampling within deep CNNs. In particular, we present a novel index-guided encoder-decoder framework, which naturally generalizes SegNet. Instead of using max-pooling and unpooling, we introduce indexed pooling and indexed upsampling operators where downsampling and upsampling are guided by learned indices. The indices are generated dynamically conditioned on the feature map and are learned using a fully convolutional network, termed IndexNet, without supervision. IndexNet is a highly flexible module, which can be used as a plug-in applying to any off-the-shelf convolutional networks that have coupled downsampling and upsampling stages. Compared to the fixed $\max$ function, learned index functions show potentials for simultaneous boundary and region delineation.

We demonstrate the effectiveness of IndexNet on natural image matting as well as other visual tasks. In image matting, the quality of learned indices can be visually observed from predicted alpha mattes. By visualizing learned indices, we show that the indices automatically learn to capture the boundaries and textural patterns. We further investigate alternative ways to design IndexNet, and show through extensive experiments that IndexNet can effectively improve deep image matting both qualitatively and quantitatively. In particular, we observe that our best MobileNetv2-based [39] model exhibits at least 16.1% improvement against the previous best deep model, i.e., the VGG-16-based model in [49], on the Composition-1k matting dataset. We achieve this with using less training data, and a much more compact model, therefore significantly faster inference speed.

2. Related Work

We review existing widely-used upsampling operators and the main application of IndexNet—deep image matting.

Upsampling in Deep Networks Upsampling is an essential stage for almost all dense prediction tasks. It has been intensively studied about *what is the principal way to recover the resolution of the downsampled feature map* (decoding). The *deconvolution* operator, also known as transposed convolution, was initially used in [50] to visualize convolutional activations and latter introduced to semantic segmentation [32]. To avoid checkerboard artifacts, a follow-up suggestion is the "resize+convolution" paradigm, which has currently become the standard configuration in state-of-the-art semantic segmentation models [4, 30]. Aside from these, *perforate* [35] and *unpooling* [2] are also two operators that generate sparse indices to guide upsampling. The indices are able to capture and keep boundary information, but the problem is that two operators induce sparsity after upsampling. Convolutional layers with large filter sizes must follow for densification. In addition, *periodic shuffling* ($\mathcal{PS}$) was introduced in [41] as a fast and memory-efficient upsampling operator for image super-resolution. $\mathcal{PS}$ recovers resolution by rearranging the feature map of size $H \times W \times Cr^2$ to $rH \times rW \times C$.

Our work is primarily inspired by the unpooling operator [2]. We remark that, it is important to keep the spatial information before loss of such information occurred in feature map downsampling, and more importantly, to use stored information during upsampling. Unpooling shows a simple and effective case of doing this, but we argue there is much room to improve. In this paper, we illustrate that the unpooling operator is a special form of index function, and we can learn an index function beyond unpooling.

Deep Image Matting In the past decades, image matting methods have been extensively studied from a low-level view [1, 6, 7, 9, 14, 15, 28, 29, 45]; and particularly, they have been designed to solve the matting equation. Despite being theoretically elegant, these methods heavily rely on the color cues, rendering failures of matting in general natural scenes where colors cannot be used as reliable cues.

With the tremendous success of deep CNNs in high-level vision tasks [13, 26, 32], deep matting methods are emerging. Some initial attempts appeared in [8] and [40], where classic matting approaches, such as closed-form matting [29] and KNN matting [6], are still used as the backends in deep networks. Although the networks are trained end-to-end and can extract powerful features, the final performance is limited by the conventional backends. These attempts may be thought as semi-deep matting. Recently fully-deep image matting was proposed [49]. In [49] the authors presented the first deep image matting approach based on SegNet [2] and significantly outperformed other competitors. Interestingly, this SegNet-based architecture becomes the standard configuration in many recent deep matting methods [3, 5, 47].

SegNet is effective in matting but also computation-expensive and memory-inefficient. For instance, the inference can only be executed on CPU when testing high-resolution images, which is practically unattractive. We

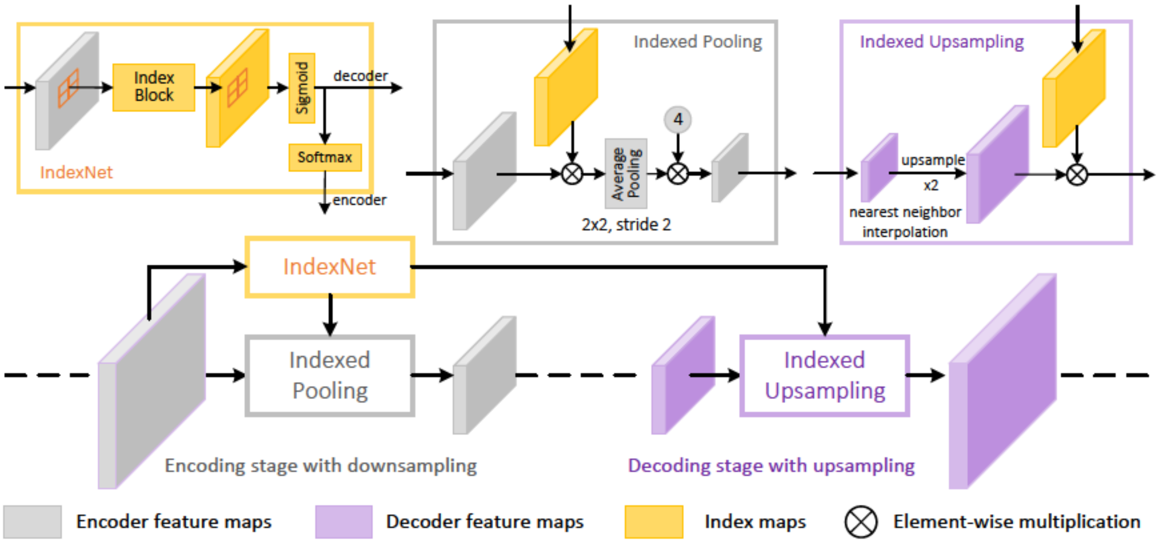

Figure 2: Index-guided encoder-decoder framework. The proposed IndexNet dynamically predicts indices for individual local regions, conditional on the input local feature map itself. The predicted indices are further utilized to guide the downsampling in the encoding stage and the upsampling in corresponding decoding stage.

show that, with our proposed IndexNet, even a lightweight backbone such as MobileNetv2-based model can surpass the VGG-16 based method in [49].

3. An Indexing Perspective of Upsampling

With the argument that upsampling operators are index functions, here we offer an unified index perspective of upsampling operators. The unpooling operator is straightforward. We can define its index function in a $k \times k$ local region as an indicator function

$$I_{max}(x) = \mathbb{1}(x = \max(\boldsymbol{X}))\,, x \in \boldsymbol{X}\,, \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{k \times k}$. Similarly, if one extracts indices from the average pooling operator, the index function takes the form

$$I_{avg}(x) = \mathbb{1}(x \in \boldsymbol{X})\,. \tag{2}$$

If further using $I_{avg}(x)$ during upsampling, it is equivalent to the nearest neighbor interpolation. Regarding the bilinear interpolation and deconvolution operators, their index functions have an identical form

$$I_{bilinear/dconv}(x) = \boldsymbol{W} \otimes \mathbb{1}(x \in \boldsymbol{X})\,, \tag{3}$$

where $\boldsymbol{W}$ is the weight/filter of the same size as $\boldsymbol{X}$, and $\otimes$ denotes the element-wise multiplication. The difference is that, $\boldsymbol{W}$ in deconvolution is learned, while $\boldsymbol{W}$ in bilinear interpolation stays fixed. Indeed, bilinear upsampling has been shown to be a special case of deconvolution [32]. Notice that, in this case, the index function generates soft indices. The sense of index for the $\mathcal{PS}$ operator [41] is even much clear, because the rearrangement of the feature map per se is an indexing process. Considering $\mathcal{PS}$ a tensor $\mathcal{Z}$ of size $1 \times 1 \times r^2$ to a matrix $\boldsymbol{Z}$ of size $r \times r$, the index function can be expressed by the one-hot encoding

$$I^l_{ps}(x) = \mathbb{1}(x = \mathcal{Z}_l)\,, l = 1, ..., r^2\,, \tag{4}$$

such that $\boldsymbol{Z}_{m,n} = \mathcal{Z}[I^l_{ps}(x)]$, where $m = 1, ..., r$, $n = 1, ..., r$, and $l = (r-1) * m + n$. $\mathcal{Z}_l$ denotes the l-th element of $\mathcal{Z}$. A similar notation applies to $\boldsymbol{Z}_{m,n}$.

Since upsampling operators can be unified by the notion of index function, in theory it is possible to learn an index function that adaptively captures local spatial patterns.

4. Index-Guided Encoder-Decoder Framework

Our framework is a natural generalization of SegNet, as schematically illustrated in Fig. 2. For ease of exposition, we assume the downsampling and upsampling rates are 2, and the pooling operator has a kernel size of 2×2. At the core of our framework is the IndexNet module that dynamically generates indices given the feature map. The proposed indexed pooling and indexed upsampling operators further receive generated indices to guide the downsampling and upsampling, respectively. In practice, multiple such modules can be combined and used analogues to the max pooling layers. We provide details as follows.

4.1. Learning to Index, to Pool, and to Upsample

IndexNet models the index as a function of the feature map $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$. It generates two index maps for downsampling and upsampling given the input $\mathcal{X}$. An important concept for the index is that an index can either be represented in a natural order, e.g., 1, 2, 3, ..., or be represented in a logical form, i.e., 0, 1, 0, ..., which means an index map can be used as a mask. In fact, this is how we use the index map in downsampling and upsampling. The predicted index shares the same physical notation of the index in computer science, except that we generate *soft* indices for smooth optimization, i.e., for any index i, $i \in [0, 1]$.

IndexNet consists of a predefined index block and two index normalization layers. An index block can simply be a heuristically defined function, e.g., a $\max$ function, or more generally, a neural network. In this work, the index block is designed to use a fully convolutional network. According to the shape of the output index map, we investigate two families of index networks: *holistic index networks (HINs)* and *depthwise (separable) index networks (DINs)*. Their conceptual differences are shown in Fig. 3. HINs learn an index function $I(\mathcal{X}) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times 1}$. In this case, all channels of the feature map share a holistic index map. In contrast, DINs learn an index function $I(\mathcal{X}) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$, where the index map is of the same size as the feature map. We will discuss concrete design of index networks in Sections 4.2 and 4.3.

Note that the index map sent to the encoder and decoder are normalized differently. The decoder index map only goes through a *sigmoid* function such that for any predicted index $i \in (0, 1)$. As for the encoder index map, indices of a local region L are further normalized by a *softmax* function such that $\sum_{i\in L} i = 1$. The reason behind the second normalization is to guarantee the magnitude consistency of the feature map after downsampling.

Indexed Pooling ($\mathcal{IP}$) executes downsampling using generated indices. Given a local region $E \in \mathbb{R}^{k\times k}$, $\mathcal{IP}$ calculates a weighted sum of activations and corresponding indices over E as $\mathcal{IP}(E) = \sum_{x\in E} I(x)x$, where $I(x)$ is the index of x. It is easy to infer that max pooling and average pooling are both special cases of $\mathcal{IP}$. In practice, this operator can be easily implemented with an element-wise multiplication between the feature map and the index map, an average pooling layer, and a multiplication of a constant, as instantiated in Fig. 2.

Indexed Upsampling ($\mathcal{IU}$) is the inverse operator of $\mathcal{IP}$. $\mathcal{IU}$ upsamples $d \in \mathbb{R}^{1\times 1}$ that spatially corresponds to E taking the same indices into account. Let $I \in \mathbb{R}^{k\times k}$ be the local index map formed by $I(x)$s, $\mathcal{IU}$ upsamples d as $\mathcal{IU}(d) = I \otimes D$, where $\otimes$ denotes the element-wise multiplication, and D is of the same size as I and is upsampled from d with the nearest neighbor interpolation. An important difference between deconvolution and $\mathcal{IU}$ is that, deconvolution applies a fixed kernel to all local regions, even if the kernel is learned, while $\mathcal{IU}$ upsamples different regions with different kernels (indices).

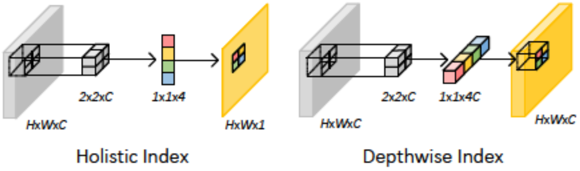

Figure 3: Conceptual differences between holistic index and depthwise index.

4.2. Holistic Index Networks

Here we instantiate two types of HINs. Recall that HINs learn an index function $I(\mathcal{X}) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times 1}$. A naive design choice is to assume a linear relationship between the feature map and the index map.

Linear Holistic Index Networks. An example is shown in Fig. 4(a). The network is implemented in a fully convolutional way. It first applies 2-stride 2×2 convolution to the feature map of size $H \times W \times C$, generating a concatenated index map of size $H/2 \times W/2 \times 4$. Each slice of the index map ($H/2 \times W/2 \times 1$) is designed to correspond to the indices of a certain position of all local regions, e.g., the top-left corner of all 2×2 regions. The network finally applies a $\mathcal{PS}$-like shuffling operator to rearrange the index map to the size of $H \times W \times 1$.

In many situations, assuming a linear relationship is not sufficient. An obvious fact is that a linear function even cannot fit the $\max$ function. Naturally the second design choice is to add nonlinearity into the network.

Nonlinear Holistic Index Networks. Fig. 4(b) illustrates a nonlinear HIN where the feature map is first projected to a map of size $H/2 \times W/2 \times 2C$, followed by a batch normalization layer and a ReLU function for nonlinear mappings. We then use point-wise convolution to reduce the channel dimension to an indices-compatible size. The rest transformations follow its linear counterpart.

Remark 1. Note that, the holistic index map is shared by all channels of the feature map, which means the index map should be expanded to the size of $H \times W \times C$ when feeding into $\mathcal{IP}$ and $\mathcal{IU}$. Fortunately, many existing packages support implicit expansion over the singleton dimension. This index map could be thought as a collection of local attention maps [34] applied to individual local spatial regions. In this case, the $\mathcal{IP}$ and $\mathcal{IU}$ operators can also be referred to "attentional pooling" and "attentional upsampling".

4.3. Depthwise Index Networks

In DINs, we find $I(\mathcal{X}) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$, i.e., each spatial index corresponds to each spatial activation. This family of networks further has two high-level design strategies that correspond to two different assumptions.

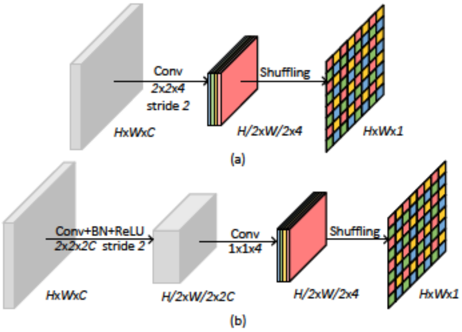

Figure 4: Holistic index networks. (a) a linear index network; (b) a nonlinear index network.

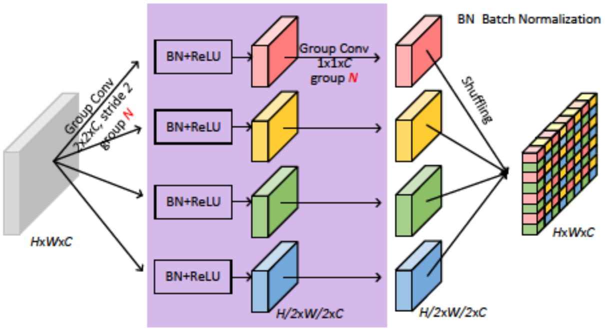

Figure 5: Depthwise index networks. $N = C$ for the O2O assumption, and $N = 1$ for the M2O. The masked modules are invisible to linear networks.

One-to-One (O2O) Assumption assumes that each slice of the index map only relates to its corresponding slice of the feature map. It can be denoted by a local index function $l(\mathcal{X}) : \mathbb{R}^{k\times k\times 1} \rightarrow \mathbb{R}^{k\times k\times 1}$, where k denotes the size of local region. Similar to HINs, DINs can also be designed to have linear/nonlinear modeling ability. Fig. 5 shows an example when $k = 2$. Note that, different from HINs, DINs follow a multi-column architecture. Each column predicts indices specific to a certain spatial location of all local regions. The O2O assumption can be easily satisfied in DINs with grouped convolution.

Linear Depthwise Index Networks. As per Fig. 5, a feature map goes through four parallel convolutional layers with the same kernel size of $2 \times 2 \times C$, a stride of 2, and C groups, leading to four downsampled feature maps of size $H/2\times W/2\times C$. The final index map is composed from the four feature maps by shuffling and rearrangement. Note that the parameters of four convolutional layers are not shared.

Nonlinear Depthwise Index Networks. Nonlinear DINs can be easily modified from linear DINs by inserting four extra convolutional layers. Each of them is followed by a BN layer and a ReLU unit, as shown in Fig. 5. The rest remains the same as the linear DINs.

Many-to-One (M2O) Assumption assumes that each slice of the index map relates with all channels of the feature map. The local index function is defined as $l(\mathcal{X}) : \mathbb{R}^{k\times k\times C} \rightarrow \mathbb{R}^{k\times k\times 1}$. Compared to O2O DINs, the only difference in implementation is the use of standard convolution instead of group convolution, i.e., $N = 1$ in Fig. 5.

Learning with Weak Context. A desirable property of IndexNet is that it can predict indices even from a large local feature map, e.g., $l(\mathcal{X}) : \mathbb{R}^{2k\times 2k\times C} \rightarrow \mathbb{R}^{k\times k\times 1}$. An intuition behind this idea is that, if one identifies a local maximum point from a $k \times k$ region, its surrounding $2k \times 2k$ region can further support whether this point is a part of a boundary or just an isolated noise point. This idea can be easily implemented by enlarging the convolutional kernel and is also applicable to HINs.

Remark 2. Both HINs and DINs have merits and drawbacks. It is clear that DINs have higher capacity than HINs, so DINs may capture more complex local patterns but also be at a risk of overfitting. By contrast, the index map generated by HINs is shared by all channels of the feature map, so the decoder feature map can reserve its expressibility without forcibly reducing its dimensionality to fit the shape of the index map during upsampling. This gives much flexibility for decoder design, while it is not the case for DINs.

4.4. Relation to Other Networks

If considering the dynamic property of IndexNet, IndexNet shares a similar spirit with some recent networks.

Spatial Transformer Networks (STNs) [21]. The STN learns dynamic spatial transformation by regressing desired transformation parameters θ with a localized network. A spatially-transformed output is then produced by a sampler parameterized by θ. Such a transformation is holistic for the feature map, which is similar to HINs. The differences between STN and IndexNet are that their learning targets have different physical definitions (spatial transformations vs. spatial indices), and that, STN is designed for global transformation, while IndexNet predicts local indices.

Dynamic Filter Networks (DFNs) [22]. The DFN dynamically generates filter parameters on-the-fly with a so-called filter generating network. Compared to conventional filter parameters that are initialized, learned, and stayed fixed during inference, filter parameters in DFN are dynamic and sample-specific. The main difference between DFN and IndexNet lies in the motivation of the design. Dynamic filters are learned for adaptive feature extraction, but learned indices are used for dynamic downsampling and upsampling.

Deformable Convolutional Networks (DCNs) [10]. The DCN introduces deformable convolution and deformable RoI pooling. The key idea is to predict offsets for convolutional and pooling kernels, so DCN is also a dynamic network. While these convolution and pooling operators concern spatial transformations, they are still built upon standard max pooling and are not designed for upsampling purposes. By contrast, index-guided $\mathcal{IP}$ and $\mathcal{IU}$ are fundamental operators and may be integrated into RoI pooling.

Attention Networks [34]. Attention networks are a broad family of networks that adopt attention mechanisms. The mechanisms introduce multiplicative interactions between inferred attention maps and feature maps. In Computer Vision, these mechanisms often refer to spatial attention [46], channel attention [20] or both [48]. As aforementioned, $\mathcal{IP}$ and $\mathcal{IU}$ in HINs can be viewed as attentional operators to some extent, which means indices are attention. In a reverse sense, attention is also indices. For example, max-pooling indices are a form of hard attention. Indices offer a new perspective to understand attention. It is worth noting that, despite IndexNet in its current implementation closely relates to attention, it has a distinct physical definition and specializes in upsampling rather than refining feature maps.

5. Results and Discussions

We evaluate our framework and IndexNet on the task of image matting. This task is particularly suitable for visualizing the quality of learned indices. We mainly conduct experiments on the Adobe Image Matting dataset [49]. This is so far the largest publicly available matting dataset. The training set has 431 foreground objects and ground-truth alpha mattes.[1] Each foreground is composited with 100 background images randomly chosen from MS COCO [31]. The test set termed Composition-1k includes 100 unique objects. Each of them is composited with 10 background images chosen from Pascal VOC [12]. Overall, we have 43100 training images and 1000 testing images. We evaluate the results using widely-used Sum of Absolute Differences (SAD), Mean Squared Error (MSE), and perceptually-motivated Gradient (Grad) and Connectivity (Conn) errors [37]. The evaluation code implemented by [49] is used. In what follows, we first describe our modified MobileNetv2-based architecture and training details. We then perform extensive ablation studies to justify choices of model design, make comparisons of different index networks, and visualize learned indices. We also report performance on the `alphamatting.com` online benchmark [37] and extend IndexNet to other visual tasks.

5.1. Implementation Details

Our implementation is based on PyTorch [36]. Here we describe the network architecture used and some essential training details.

Network Architecture. We build our model based on MobileNetv2 [39] with only slight modifications to the

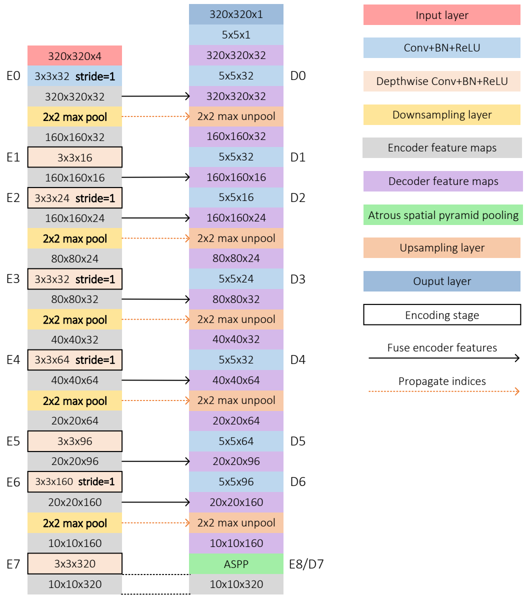

Figure 6: Customized MobileNetv2-based encoder-decoder network architecture. Our modifications are boldfaced.

backbone. An important reason why we choose MobileNetv2 is that this lightweight model allows us to infer high-resolution images on a GPU, while other high-capacity backbones cannot. The basic network configuration is shown in Fig. 6. It also follows the encoder-decoder paradigm same as SegNet. We simply change all 2-stride convolution to be 1-stride and attach 2-stride 2×2 max pooling after each encoding stage for downsampling, which allows us to extract indices. If applying the IndexNet idea, max pooling and unpooling layers can be replaced with $\mathcal{IP}$ and $\mathcal{IU}$, respectively. We also investigate alternative ways for low-level feature fusion and whether encoding context (Section 5.2). Notice that, the matting refinement stage [49] is not considered in this paper.

Training Details. To enable a direct comparison with deep matting [49], we follow the same training configurations used in [49]. The 4-channel input concatenates the RGB image and its trimap. We follow exactly the same data augmentation strategies, including 320×320 random cropping, random flipping, random scaling, and random trimap dilation. All training samples are created on-the-fly. We use a combination of the alpha prediction loss and the composition loss during training as in [49]. Only losses from the unknown region of the trimap are calculated. Encoder parameters are pretrained on ImageNet [11]. Note that, the parameters of the 4-th input channel are initialized with zeros. All other parameters are initialized with the improved Xavier [16]. The Adam optimizer [23] is used. We update parameters with 30 epochs (around $90,000$ iterations). The learning rate is initially set to 0.01 and reduced by $10\times$ at

[1] The original paper reported that there were 491 images, but the released dataset only includes 431 images. As a result, we use fewer training data than the original paper.

No.	Architecture	Backbone	Fusion	Indices	Context	OS	SAD	MSE	Grad	Conn
B1	DeepLabv3+ [4]	MobileNetv2	Concat	No	ASPP	16	60.0	0.020	39.9	61.3
B2	RefineNet [30]	MobileNetv2	Skip	No	CRP	32	60.2	0.020	41.6	61.4
B3	SegNet [49]	VGG16	No	Yes	No	32	**54.6**	**0.017**	36.7	55.3
B4	SegNet	VGG16	No	No	No	32	122.4	0.100	161.2	130.1
B5	SegNet	MobileNetv2	No	Yes	No	32	60.7	0.021	40.0	61.9
B6	SegNet	MobileNetv2	No	No	No	32	78.6	0.031	101.6	82.5
B7	SegNet	MobileNetv2	No	Yes	ASPP	32	58.0	0.021	39.0	59.5
B8	SegNet	MobileNetv2	Skip	Yes	No	32	57.1	0.019	36.7	57.0
B9	SegNet	MobileNetv2	Skip	Yes	ASPP	32	56.0	**0.017**	38.9	55.9
B10	UNet	MobileNetv2	Concat	Yes	No	32	54.7	**0.017**	34.3	**54.7**
B11	UNet	MobileNetv2	Concat	Yes	ASPP	32	54.9	**0.017**	**33.8**	55.2

Table 1: Ablation study of design choices. Fusion: fuse encoder features; Indices: max-pooling indices (when Indices is 'No', bilinear interpolation is used for upsampling); CRP: chained residual pooling [30]; ASPP: atrous spatial pyramid pooling [4]; OS: output stride. The lowest errors are boldfaced.

the 20-th and 26-th epoch respectively. We use a batch size of 16 and fix the BN layers of the backbone.

5.2. Adobe Image Matting Dataset

Ablation Study on Model Design. Here we investigate strategies for fusing low-level features (no fusion, skip fusion as in ResNet [17] or concatenation as in UNet [38]) and whether encoding context for image matting. 11 baselines are consequently built to justify model design. Results on the Composition-1k testing set are reported in Table 1. B3 is cited from [49]. We can make the following observations: **i**) Indices are of great importance. Matting can significantly benefit from only indices (B3 vs. B4, B5 vs. B6); **ii**) State-of-the-art semantic segmentation models cannot be directly applied to image matting (B1/B2 vs. B3); **iii**) Fusing low-level features help, and concatenation works better than the skip connection but at a cost of increased computation (B5 vs. B8 vs. B10 or B7 vs. B9 vs. B11); **iv**) Our intuition tells that the context may not help a low-level task like matting, while results show that encoding context is generally encouraged (B5 vs. B7 or B8 vs. B9 or B10 vs. B11). Indeed, we observe that the context sometimes can help to improve the quality of the background; **v**) A MobileNetv2-based model can work as well as a VGG-16-based one with appropriate design choices (B3 vs. B11).

For the following experiments, we now mainly use B11.

Ablation Study on Index Networks. Here we compare different index networks and justify their effectiveness. The configurations of index networks used in the experiments follow Figs. 4 and 5. We primarily investigate the 2×2 kernel with a stride of 2. Whenever the weak context is considered, we use a 4×4 kernel in the first convolutional layer of index networks. To highlight the effectiveness of HINs, we further build a baseline called *holistic max index (HMI)* where max-pooling indices are extracted from a squeezed feature map $\mathcal{X}' \in \mathbb{R}^{H \times W \times 1}$. $\mathcal{X}'$ is generated by applying the max function along the channel dimension of $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$. We also report the performance when setting the width multiplier of MobileNetV2 used in B11 to be 1.4 (B11-1.4). This allows us to justify whether the improved performance is due to increased model capacity. Results on the Composition-1k testing dataset are listed in Table 2. We observe that, except the most naive linear HIN, all index networks consistently reduce the errors. In particular, nonlinearity and the context generally have a positive effect on deep image matting. Compared to HMI, the direct baseline of HINs, the best HIN ("Nonlinear+Context") has at least 12.3% relative improvement. Compared to B11, the baseline of DINs, M2O DIN with "Nonlinear+Context" exhibits at least 16.5% relative improvement. Notice that, our best model even outperforms the state-of-the-art DeepMatting [49] that has the refinement stage, and is also computationally efficient with less memory consumption—the inference can be performed on the GTX 1070 over 1920×1080 high-resolution images. Some qualitative results are shown in Fig. 7. Our predicted mattes show improved delineation for edges and textures like hair and water drops.

Index Map Visualization. It is interesting to see what indices are learned by IndexNet. For the holistic index, the index map itself is a 2D matrix and is easily to be visualized. Regarding the depthwise index, we squeeze the index map along the channel dimension and calculate the average responses. Two examples of learned index maps are visualized in Fig. 8. We observe that, initial random indices have poor delineation for edges, while learned indices automatically capture the complex structural and textual patterns, e.g., the fur of the dog, and even air bubbles in the water.

5.3. alphamatting.com Online Benchmark

We also report results on the `alphamatting.com` online benchmark [37]. We directly test our best model trained on the Adobe Image Dataset, without fine-tuning. Our approach (IndexNet Matting) ranks the first in terms of the gradient error among published methods, as shown in Ta-

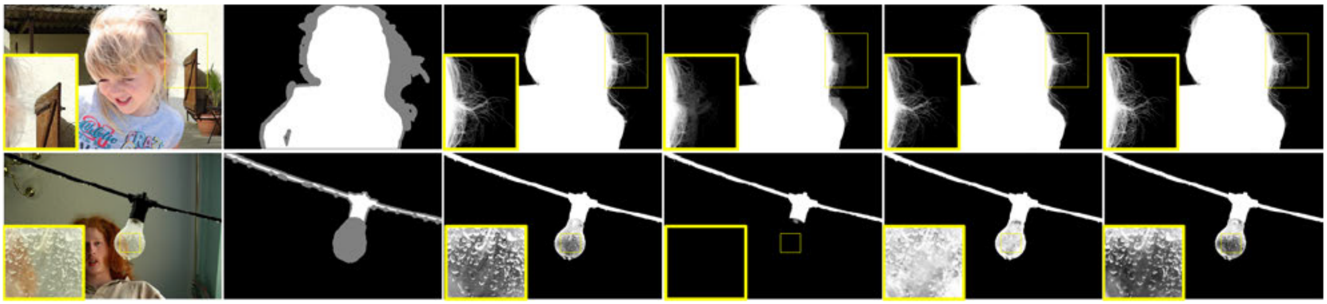

Figure 7: Qualitative results on the Composition-1k testing set. From left to right, the original image, trimap, ground-truth alpha matte, closed-form matting [29], deep image image [29], and ours (M2O DIN with "nonlinear + context").

Method		#Param.	GFLOPs	SAD	MSE	Grad	Conn
B3 [49]		130.55M	32.34	54.6	0.017	36.7	55.3
B11		3.75M	4.08	54.9	0.017	33.8	55.2
B11-1.4		8.86M	7.61	55.6	0.016	36.4	55.7
HMI		3.75M	4.08	56.5	0.021	33.0	56.4
NL	C	Δ					
				HINs			
		+4.99K	4.09	55.1	0.018	32.1	55.2
	✓	+19.97K	4.11	53.5	0.018	31.0	53.5
✓		+0.26M	4.22	50.6	0.015	27.9	49.4
✓	✓	+1.04M	4.61	49.5	0.015	**25.6**	49.2
				O2O DINs			
		+4.99K	4.09	50.3	0.015	33.7	50.0
	✓	+19.97K	4.11	47.8	0.015	26.9	45.6
✓		+17.47K	4.10	50.6	0.016	26.5	50.3
✓	✓	+47.42K	4.15	50.2	0.016	26.8	49.3
				M2O DINs			
		+0.52M	4.34	51.0	0.015	33.7	50.5
	✓	+2.07M	5.12	50.6	0.016	31.9	50.2
✓		+1.30M	4.73	48.9	0.015	32.1	47.9
✓	✓	+4.40M	6.30	**45.8**	**0.013**	25.9	**43.7**
Closed-Form [29]				168.1	0.091	126.9	167.9
DeepMatting w. Refinement [49]				50.4	0.014	31.0	50.8

Table 2: Results on the Composition-1k testing set. GFLOPs are measured on a 224 × 224 × 4 input. NL: Non-Linearity; C: Context. The lowest errors are boldfaced.

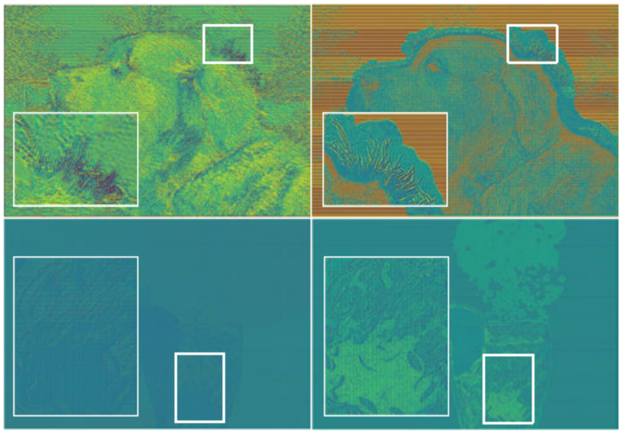

Figure 8: Visualization of the randomly initialized index map (left) and the learned index map (right) of HINs (top) and DINs (bottom). Best viewed by zooming in.

ble 3. According to the qualitative results in Fig. 9, our approach produces significantly better mattes on hair.

5.4. Extensions to Other Visual Tasks

We further evaluate IndexNet on other three visual tasks. For image classification, we compare three classification networks (LeNet [27], MobileNet [18] and VGG-16 [43]) on the CIFAR-10 and CIFAR-100 datasets [25] with/without IndexNet. For monocular depth estimation, we attach IndexNet upon a recent ResNet-50 based baseline [19] and report the performance on the NYUDv2 dataset [42]. On the task of scene understanding, we evaluate SegNet [2] with/without IndexNet on the SUN-RGBD dataset [44]. Results show that IndexNet consistently improves the performance in all three tasks. We refer readers to the Supplement for quantitative and qualitative results.

6. Conclusion

Inspired by an observation in image matting, we delve deep into the role of indices and present an unified perspective of upsampling operators using the notion of index function. We show that an index function can be learned within a proposed index-guided encoder-decoder framework. In this framework, indices are learned with a flexible network module termed IndexNet, and are used to guide downsampling and upsampling using two operators called $\mathcal{IP}$ and $\mathcal{IU}$. IndexNet itself is also a sub-framework that can be designed depending on the task at hand. We instantiated, investigated three index networks, compared their conceptual differences, discussed their properties, and demonstrated their effectiveness on the task of image matting, image classification, depth prediction and scene understanding. We report state-of-the-art performance on image matting with a modified MobileNetv2-based model on the Composition-1k dataset. We believe that IndexNet is an important step towards the design of generic upsampling operators.

Our model is simple with much room for improvement. It may be used as a strong baseline for future research. We plan to explore the applicability of IndexNet to other dense prediction tasks.

Gradient Error	Average Rank				Troll			Doll			Donkey			Elephant			Plant			Pineapple			Plastic Bag			Net		
	Overall	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
IndexNet Matting	**9**	**7.3**	**7.6**	**12.3**	**0.2**	**0.2**	**0.2**	**0.1**	**0.1**	0.3	0.2	0.2	**0.2**	**0.2**	**0.2**	**0.4**	1.7	1.9	2.5	1	1.1	**1.3**	1.1	1.2	1.2	0.4	0.5	**0.5**
AlphaGAN [33]	13.2	12	10.8	16.8	0.2	0.2	0.2	0.2	0.2	0.3	0.2	0.3	0.3	0.2	0.2	0.4	1.8	2.4	2.7	1.1	1.4	1.5	0.9	1.1	1	0.5	0.5	0.6
Deep Matting [49]	14.3	10.8	11	21	0.4	0.4	0.5	0.2	0.2	**0.2**	**0.1**	**0.1**	0.2	0.2	0.2	0.6	**1.3**	**1.5**	**2.4**	**0.8**	**0.9**	1.3	**0.7**	**0.8**	**1.1**	**0.4**	**0.5**	0.5

Table 3: Gradient errors (top 3) on the `alphamatting.com` online benchmark. The lowest errors are boldfaced.

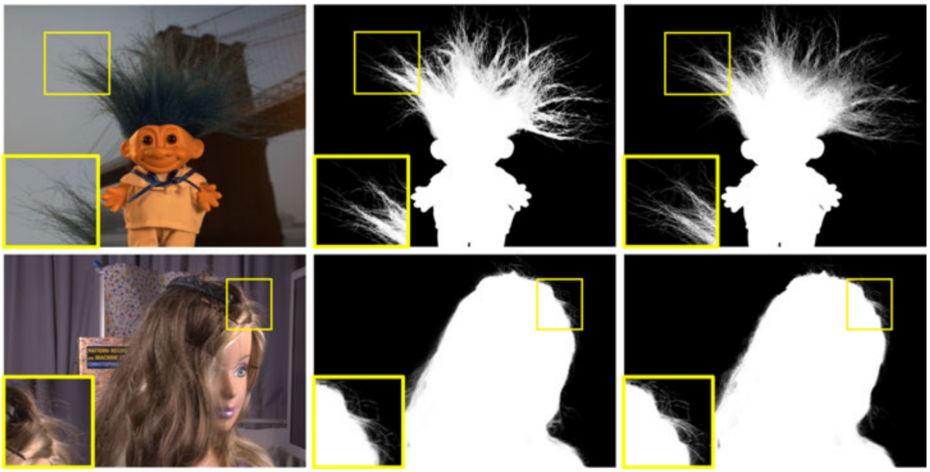

Figure 9: Qualitative results on the `alphamatting.com` dataset. From left to right, the original image, deep image matting, ours.

Acknowledgments We would like to thank Huawei Technologies for the donation of GPU cloud computing resources.

References

[1] Y. Aksoy, T. Ozan Aydin, and M. Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29–37, 2017. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 1, 2, 8

[3] G. Chen, K. Han, and K.-Y. K. Wong. TOM-Net: Learning transparent object matting from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9233–9241, 2018. 2

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 7

[5] Q. Chen, T. Ge, Y. Xu, Z. Zhang, X. Yang, and K. Gai. Semantic human matting. In *Proc. ACM Multimedia*, pages 618–626, 2018. 2

[6] Q. Chen, D. Li, and C.-K. Tang. KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188, 2013. 2

[7] X. Chen, D. Zou, S. Zhiying Zhou, Q. Zhao, and P. Tan. Image matting with local and nonlocal smooth priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1902–1907, 2013. 2

[8] D. Cho, Y.-W. Tai, and I. Kweon. Natural image matting using deep convolutional neural networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 626–643. Springer, 2016. 2

[9] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 264. IEEE, 2001. 2

[10] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 5

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 6

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 2

[14] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting-a stroke based approach for continuous image matting. *Computer Graphics Forum*, 25(3):567–576, 2006. 2

[15] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056. IEEE, 2011. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 6

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017. 8

[19] J. Hu, M. Ozay, Y. Zhang, and T. Okatani. Revisiting single image depth estimation: toward higher resolution maps with accurate object boundaries. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 8

[20] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 6

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 5

[22] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 667–675, 2016. 5

[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 6

[24] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In *Proc. International Conference on Management of Data*, pages 489–504. ACM, 2018. 2

[25] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 8

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 2

[27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

[28] P. Lee and Y. Wu. Nonlocal matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2193–2200. IEEE, 2011. 2

[29] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. 2, 8

[30] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1925–1934, 2017. 1, 2, 7

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 6

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1, 2, 3

[33] S. Lutz, K. Amplianitis, and A. Smolic. AlphaGAN: Generative adversarial networks for natural image matting. In *Proc. British Machine Vision Conference (BMVC)*, 2018. 9

[34] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2204–2212, 2014. 4, 6

[35] C. Osendorfer, H. Soyer, and P. Van Der Smagt. Image super-resolution with fast approximate convolutional sparse coding. In *Proc. International Conference on Neural Information Processing (ICONIP)*, pages 250–257. Springer, 2014. 2

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshops (NIPSW)*, 2017. 6

[37] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1826–1833. IEEE, 2009. 6, 7

[38] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 7

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 2, 6

[40] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia. Deep automatic portrait matting. In *Proc. European Conference on Computer Vision (ECCV)*, pages 92–107. Springer, 2016. 2

[41] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 1, 2, 3

[42] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 8

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. International Conference on Learning Representations (ICLR)*, 2014. 8

[44] S. Song, S. P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015. 8

[45] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004. 2

[46] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2017. 6

[47] Y. Wang, Y. Niu, P. Duan, J. Lin, and Y. Zheng. Deep propagation based image matting. In *Proc. International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 999–1066, 2018. 2

[48] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. CBAM: Convolutional block attention module. In *Proc. European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 6

[49] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2970–2979, 2017. 1, 2, 3, 6, 7, 8, 9

[50] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014. 1, 2