ATM: Attentional Text Matting

Peng Kang*
Northwestern University
pengkang2022@u.northwestern.edu

Jianping Zhang*
Northwestern University
jianpingzhang2018@u.northwestern.edu

Chen Ma
McGill University
chen.ma2@mail.mcgill.ca

Guiling Sun
Nankai University
sungl@nankai.edu.cn

Abstract

Image matting is a fundamental computer vision problem and has many applications. Previous image matting methods always focus on extracting a general object or portrait from the background in an image. In this paper, we try to solve the text matting problem, which extracts characters (usually WordArts) from the background in an image. Different from traditional image matting problems, text matting is much harder because of its foreground's three properties: smallness, multi-objectness, and complicated structures and boundaries. We propose a two-stage attentional text matting pipeline to solve the text matting problem. In the first stage, we utilize text detection methods to serve as the attention mechanism. In the second stage, we employ the attentional text regions and matting system to obtain mattes of these text regions. Finally, we post-process the mattes and obtain the final matte of the input image. We also construct a large-scale dataset with high-quality annotations consisting of 46,289 unique foregrounds to facilitate the learning and evaluation of text matting. Extensive experiments on this dataset and real images clearly demonstrate the superiority of our proposed pipeline over previous image matting methods on the task of text matting.

1. Introduction

Image matting is a fundamental problem in computer vision. Text matting, which aims at extracting various disconnected foreground characters from an image and estimating the foreground opacity, is an unexplored sub-domain of image matting. Like image matting can be applied to many areas, text matting also has a wide variety of applications in the real world, such as the smart creative composition, film production, mixed reality, WordArt copyright protection, *etc*. For example, in an online media website (e.g., Netflix), the smart creative composition provides the personalized creative image to customers (e.g., generating personalized posters to attract customers). This requires extracting the text from huge amounts of original images and re-compositing them with new creative backgrounds. In this scenario, due to the huge volume of images to be processed and in pursuit of a better customer experience, it is critical to have an automatic high-quality extraction method. Figure 1 presents an example of a poster generation system with the automatic text matting system.

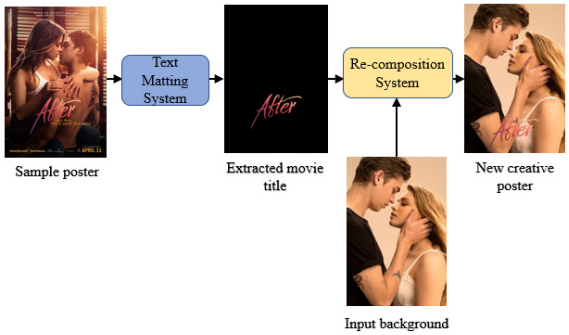

Figure 1. An example of a poster generation system

Previous methods always focus on extracting the objects from an image. For example, portrait matting methods extract people from an image. And these previous matting algorithms tend to have bad performance on the task of text matting. The reason is three-fold. First, text tends to be smaller than general objects in images. From Figure 1, it is clear that characters occupy a small region in the image. Second, foreground objects are always disconnected (like "A", "f", "t", "e", "r" in Figure 1, some of them are disconnected). To some extent, text matting is a multi-object matting problem and most image matting problems focus on extracting a single object from images, thus text matting should be a challenge to them. Third, the structures of characters in an image are always non-convex and text boundaries are complex because of the use of WordArt. From

*Equal contributions

Figure 1, it is clear that the complex structures and boundaries of these WordArts pose a great challenge to the task of text matting.

In this paper, we focus on building a two-stage attentional text matting pipeline to solve the text matting problem. In the first stage, we utilize text detection methods to serve as the attention mechanism. Specifically, the text detection part localizes text regions from an image. It then crops the text regions based on their locations and sends them to the second stage. In the second stage, we employ the attentional text regions and matting system to obtain mattes of text regions in the image. Finally, we design post-processing methods to refine matte results from the second stage. Moreover, since no public text matting dataset is available, we create a text matting image synthesis engine, which generates 46,289 text images with their corresponding file names, text, text locations, foregrounds, backgrounds, trimaps, and alpha mattes. Extensive experiments are conducted on this dataset to empirically evaluate the effectiveness of our pipeline. Under the commonly used metrics of matting performance, our pipeline clearly demonstrates its superiority over previous state-of-the-art image matting methods on the task of text matting. Moreover, we demonstrate that our learned model generalizes well to real images crawled from the Internet. To summarize, the contributions of our work are three-fold:

- We propose a two-stage attentional text matting pipeline to solve the text matting problem. To the best of our knowledge, this is the first pipeline for text matting.
- We build a text matting image synthesis engine and synthesize a large scale high-quality text matting dataset. This dataset contributes with its diversity to the text matting research.
- We conduct extensive quantitative and qualitative experiments on the synthetic dataset and real images to demonstrate the superiority of our pipeline over the state-of-the-art image matting methods on the task of text matting. Codes and datasets are available here.[1]

2. Related Work

In this section, we will review text detection and image matting methods that are related to our work.

Text detection. Recently, with the prevalence of deep learning, more and more text detectors have been proposed by adopting popular object detection/segmentation methods. Basically, there are three common ways of constructing deep learning based text detectors. The first way follows region proposal methods in object detection like Faster R-CNN [26]. Rotation Region Proposal Networks [24] follows and adapts the standard Faster R-CNN framework. To fit into text of arbitrary orientations, rotating region proposals are generated instead of the standard axis-aligned rectangles. Similarly, R2CNN [17] modifies the standard region proposal based object detection methods. To adapt to the varying aspects ratios, three Region of Interests Poolings of different sizes are used and concatenated for further prediction and regression. The second way follows anchor-based methods like SSD [21]. TextBoxes [19] adapts the SSD network specially to fit the varying orientations and aspect-ratios of the text line. The third way is based on image segmentation, which aims to seek text regions at the pixel level [10], [12], [15], [33], [35]. PixelLink [10] learns to predict whether two adjacent pixels belong to the same text instance by adding link prediction to each pixel. Our pipeline adopts text detection models to serve as the attention mechanism and localize text regions in images.

Image matting. In the past decades, researchers have put forward various image matting methods for natural images. Basically, most methods utilize sampling or propagating ways to predict alpha mattes.

In the sampling field [9], [32], [11], [13], [28], researchers sample the known foreground and background regions to find candidate colors for a given pixel's foreground and background, then they use a metric to determine the best foreground/background combination. Chuang et al. [9] use Gaussian mixtures to model background and foreground color samples. Shahrian et al. [28] improve the performance of sampling-based matting methods by establishing more comprehensive sampling sets.

In the propagating field [30], [18], [14], [7], [1], the aim is to propagate the known information (like the user-drawn information) to unknown pixels according to pixel affinities. Sun et al. [30] formulate the problem of natural image matting as solving Poisson equations with the matte gradient field. The information-flow matting method [1] shows that high-quality mattes can be produced by combining local and non-local affinities.

Recently, several deep learning based methods are proposed. Shen et al. [29] utilize CNN-based models to create a trimap of a person in a portrait image. And then, closed-form matting [18] is used for matting results. Cho et al. [8] take the matting results of [18] and [7] and normalized RGB colors as inputs and learn an end-to-end deep network to predict a new alpha matte. Xu et al. [34] design a deep convolutional encoder-decoder network with a refinement part to improve the image matting performance. Lutz et al. [23] present the first generative adversarial network (GAN) for natural image matting and obtain state-of-the-art performance. Chen et al. [6] propose semantic human matting, which is the first automatic matting algorithm. It

[1] https://github.com/TextMatting/TextMatting

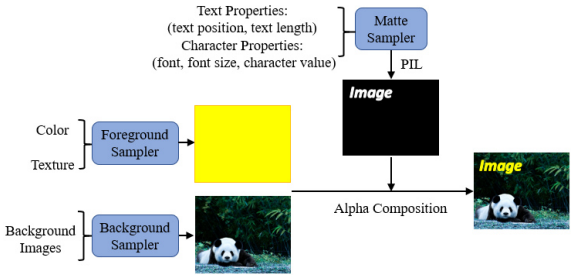

Figure 2. Text matting image synthesis engine

learns to jointly fit both trimap information and high-quality matting details with deep networks.

Different from the methods mentioned above, our pipeline focuses on solving the task of text matting. In addition, text matting is hard for the previous image matting methods because of text matting foreground's three properties: smallness, multi-objectness, and complicated structures and boundaries.

3. Text Matting Image Synthesis Engine

Generally, it requires massive amounts of labeled data to train large models such as deep neural networks. However, precisely labeled training data for text matting is expensive to obtain manually. Furthermore, existing image matting datasets are about portraits [29] [6] and objects [35] [34], while no public text matting dataset is available. Hence, we develop a text matting image synthesis engine to build a large annotated dataset for text matting. Such dataset is required to be not only large enough to train a deep neural network model but adequate to represent the possible text variations in real images, such as fonts, font sizes, colors, textures, and positions.

The text matting image synthesis engine is illustrated in Figure 2. The text matting image synthesis engine contains three parts: (1) foreground sampler, (2) background sampler, and (3) matte sampler. Foreground sampler samples the foreground from various colors and textures. Background sampler randomly selects a background image without text. Matte sampler generates the text matte value with Python Image Library (PIL)[2]. After acquiring the foreground F, background B, and text matte value α, synthetic text image I is composed of image matting equation 1.

$$I = \alpha F + (1 - \alpha)B \quad \alpha \in [0, 1] \tag{1}$$

Our text matting image synthesis engine has three characteristics: (1) it produces **realistic** images so that our trained model can generalize to real images; (2) it is fully **automated**; (3) it is **fast**. The characteristics enable the generation of large quantities of high-quality and fully annotated data without supervision.

[2]https://www.pythonware.com/products/pil/

3.1. Foreground Sampler

To generate a foreground image, we first randomly choose a pure color from RGB color space. Then, we render the canvas with the selected pure color. Besides, a random variable δ determines whether to add textures on the pure color rendered canvas. If δ is higher than the predefined threshold, we add texture on the canvas. The added texture is uniformly sampled from the texture library. We blend the pure color canvas with the sampled texture by Poisson Image Editing [25].

3.2. Background Sampler

To favor variety, a large number of background images are collected from the Internet. Because some background images may contain text, we first use a text detection model i.e., EAST [38] to detect text in all the background images and remove the background images with high text confidence. Then we manually check the rest of the background images to make sure no background image contains the text. After that, we choose a sample from the remaining background images following the uniform distribution.

3.3. Matte Sampler

Matte sampler aims to generate a text matte for composing the foreground and background together. To get the text matte, we first determine text properties and character properties to sample text characters. Then we get the corresponding matte value for sampled text characters by PIL. Specifically, text properties include the text position and text length. Character properties include the character font, font size, and character values. Then, each character is randomly selected from a character dictionary. Note that PIL renders text on background images by masks of the text. Therefore, we obtain each character matte value through the function *getmask2* in PIL with the input of the character properties. Finally, based on text properties, we draw the mask of each text character on a black canvas as the matte value for the foreground text.

4. Our Method

Our two-stage attentional text matting pipeline is targeted to automatically extract the alpha mattes of text from images. Figure 3 shows its structure. It takes a colorful image as the input and outputs a single-channel alpha matte with the same size as the input. Note that no auxiliary information (e.g. trimap or scribbles) is required in our pipeline. Creating trimaps or scribble paintings by users themselves is always time-consuming and needed in the traditional image matting methods.

From Figure 3, it is clear that the pipeline has two main stages. The first stage is served as the attention mechanism. It localizes the text in the input image and sends

these attentional text regions to the second stage. Then, for each attentional text region, the second stage simultaneously captures both coarse semantic segmentation information (trimap) and fine matting details. After going through the second stage, we use post-processing methods to process the fine matting details and obtain a final matte for the input image.

4.1. First Stage: Attentional Text Region Detector

As we mentioned in the section 1, three reasons (i.e., text's smallness, multi-objectness, and complicated structures and boundaries) make traditional image matting methods fail on the task of text matting. To reduce these text properties' influence on the task of text matting, we adopt the idea of text detection in the field of Optical Character Recognition (OCR). In OCR systems, text detection aims to localize the text from images and send these text regions to the text recognition part. Essentially, text detection provides attention guidance to the text recognition part. For the task of text matting, the text always occupies a small region in images. Therefore, we utilize a text detection model to extract the text regions from images. The text detection model is served as the attention mechanism which tells the later stage where it should look. Specifically, the text detection model takes the colorful image as the input and outputs attentional text regions. In general, this attentional text region detector can be implemented as any of the state-of-the-art text detectors [31], [38], [19], [17], [10], [2]. The selection of the attentional text region detector depends on the practical application. In this paper, we want to apply our two-stage attentional text matting pipeline to the task of poster matting and recreation, and most movie titles in posters are horizontal. Because connectionist text proposal network (CTPN) is robust and efficient for the horizontal text [31], we choose it as the attentional text region detector in our pipeline.

4.2. Second Stage: Trimap and Matte Generation

The aim of the second stage is to simultaneously capture both coarse semantic segmentation information (trimap) and fine matting details. Specifically, we utilize the T-Net and M-Net structures [6] to achieve this goal.

Following the traditional trimap definition, T-Net conducts a 3-class segmentation separating the foreground, background, and unknown region. More specifically, T-Net accepts the attentional text regions from the first stage. And for each attentional text region, T-Net outputs a 3-channel attentional text segmentation result indicating the possibility that each pixel belongs to each of the 3 classes. In general, T-Net can be implemented as any of the state-of-the-art semantic segmentation networks [22], [36], [3], [4], [5]. In this paper, we choose PSPNet-50 [36] for its efficacy and efficiency.

Following the work [6], the purpose of M-Net is to capture the detail information and generate alpha mattes. In general, M-Net is a deep convolutional encoder-decoder network. More specifically, M-Net takes the concatenation of the 3-channel attentional text regions and the 3-channel attentional text segmentation results from T-Net as 6-channel input. And M-Net generates an attentional alpha matte for each attentional text region.

Moreover, the fusion module is utilized to alleviate the problem that M-Net cannot retain the foreground and background information well [6]. For each attentional text region, we can obtain its corresponding attentional fusion alpha matte α_{af} following the below equation:

$$\alpha_{af} = F_a + U_a \alpha_{am} \tag{2}$$

where α_{am} is the attentional alpha matte generated by M-Net, F_a is the probability of each pixel in the attentional text region belonging to the foreground, and U_a is the probability of each pixel in the attentional text region belonging to the unknown region. Note that F_a and U_a come from the 3-channel attentional text segmentation result of T-Net.

4.3. Post-processing

In general, text detection models generate several attentional text regions based on the confidence scores of detected regions and predefined confidence threshold. Among these attentional text regions, some of them contain the text we are interested in, while others are false detections and contain the background information. To appropriately process the attentional fusion alpha matte α_{af} of these 'false' attentional text regions and efficiently obtain the final alpha matte for an input image α_f, we propose two post-processing methods. The choice of post-processing methods depends on practical applications.

Matte Ranking. Based on each attentional text region's location in the input image, we place each attentional fusion alpha matte α_{af} into an all-zero (background) map with the same size as the input image. This gives us the temporary final matte α'_{af} for each attentional text region. Then, we rank all the α'_{af}s following the below score:

$$Score_{\alpha'_{af}} = \frac{\alpha'_{af} == 1}{W * H} \tag{3}$$

where W and H are the width and height of the input image, respectively. Finally, we select the α'_{af} with the highest score as the final alpha matte α_f. This post-processing method tends to be good for the situation that there is only one text region we are interested in the input image and the interested text region is generally larger than any other text areas. For example, in the movie poster matting, there is only one text region - the movie title region we are interested in. Then, we use Matte Ranking to obtain the final matte.

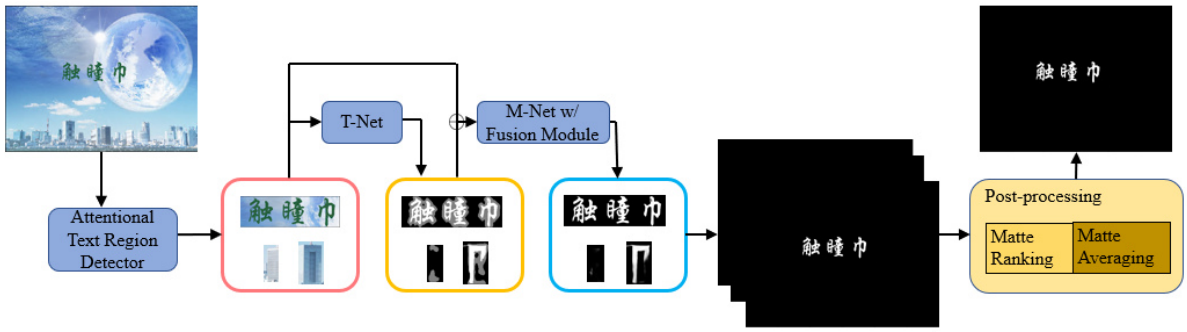

Figure 3. Network Structure of Two-Stage Attentional Text Matting Pipeline. In the first stage, the attentional text region detector detects and crops text regions then sends them to the second stage. In the second stage, a matting system including M-Net and T-Net obtains mattes of text regions in the image. Finally, post-processing methods refine matte results from the second stage.

Matte Averaging. We utilize the same way to obtain the temporary final matte α'_{af} for each attentional text region. Then, we post-process these α'_{af}s with the following equation:

$$\alpha_f = \frac{\sum_{i \in I} \alpha'_{afi}}{\sum_{i \in I} A_i} \tag{4}$$

where $i \in I$ is the index of attentional text regions and A is the 0-1 map (1s mean the attentional text region) with the same size as the input. We take the average of matte values in attentional text regions. Finally, we have the final alpha matte α_f. This post-processing method tends to be good for the situation that there are several text regions we are interested in the input image.

4.4. Training and Implementation Details

To make the two-stage pipeline training converge efficiently, we train the first stage and second stage separately.

Attentional text region detector training. We use 45,384 images from the synthetic text matting dataset to train the attentional text region detector. Each image is labelled with a text line bounding box. Moreover, because the detector (following the structure of CTPN) is an anchor-based text detection method and the width of anchor is 16-pixel, we divide the text line bounding box equally into 16-pixel width text proposals. We follow the training process and multi-task loss described in [31] minimize the errors of text/non text score and coordinate.

$$L(\mathbf{s_i}, \mathbf{v_j}) = \frac{1}{N_s} \sum_i L_s(\mathbf{s_i}, \mathbf{s_i^*}) + \frac{\lambda}{N_v} L_v(\mathbf{v_j}, \mathbf{v_j^*}) \tag{5}$$

where $\mathbf{s_i}$ is the predicted probability of anchor i being a true text, $\mathbf{s_i^*} = \{0, 1\}$ is the ground truth. j is the index of an anchor in the set of positive anchors($\mathbf{s_i^*} = 1$). $\mathbf{v_j}$ and $\mathbf{v_j^*}$ are the prediction and ground truth coordinates associated with the j-th anchor.

As for the training of second stage, we adopt the pre-train technique[16] following Chen et al. [6]. Typically, we first pre-train the T-Net and M-Net separately and then fine-tune the second stage in an end-to-end way. To form the training set **Q**, we crop the text regions from our synthetic dataset based on their ground truth bounding boxes. Moreover, to make nets robust, we randomly pad zeros around these cropped text regions.

T-Net pre-train. To train T-Net, based on the training set **Q**, we first generate the trimap ground truth by dilating the ground truth alpha mattes. We augment training samples by randomly rotation and horizontal flipping to avoid overfitting. We employ the cross-entropy loss $\mathcal{L}_t$ for training T-Net.

M-Net pre-train. To train M-Net, we still use the training set **Q** and corresponding generated trimap ground truth. During the training, we combine the 3-channel input image with the corresponding generated trimap ground truth as the input. We also augment training samples by randomly rotation and horizontal flipping to avoid overfitting. Following Xu et al. [34], we adopt the alpha prediction loss $\mathcal{L}_p$ and compositional loss $\mathcal{L}_c$ to train M-Net.

$$\begin{gathered} \mathcal{L}_m = \gamma \mathcal{L}_p + (1-\gamma)\mathcal{L}_c \\ \mathcal{L}_p = \|\alpha_p - \alpha_g\|_1 \quad \mathcal{L}_c = \|c_p - c_g\|_1 \end{gathered} \tag{6}$$

where α_p and α_g are the prediction alpha matte and ground truth alpha matte, respectively. c_p and c_g are the prediction compositional image and ground truth compositional image, respectively. We set $\gamma = 0.5$ in the paper.

Second stage end-to-end training. We initialize the T-Net and M-Net with their pre-trained parameters. We fine-tune the second stage nets with the training set **Q** and corresponding generated trimap ground truth. Following Chen et al. [6], we utilize the total loss $\mathcal{L}_{total}$.

$$\mathcal{L}_{total} = \mathcal{L}_m + \theta \mathcal{L}_t \tag{7}$$

We set $\theta = 0.01$ in this paper.

5. Experiments

5.1. Experiments Setup

In this section, we evaluate our method on the synthetic text matting dataset and apply our method to real movie

posters. Our synthetic dataset contains 45,384 training images and 905 testing images, which are generated from the engine introduced in section 3. We collect real movie posters from the Internet. The movie posters contain a wide range of artistic text matting foreground contexts such as various languages, textures, positions, *etc.*

We evaluate the predicted alpha matte by four metrics: the sum of absolute differences (SAD), mean square error (MSE), gradient error, and connectivity error. SAD and MSE are correlated to our training loss, while gradient error and connectivity are proposed by [27] to measure matting quality observed by a human. We normalize the ground truth and predicted alpha matte to the range of [0, 1]. We compute four metrics on the whole image and average metric values by the image size instead of only by the unknown area of an image.

We compare our method with 9 state-of-the-art image matting methods to show the effectiveness of our method. They are Closed Form (CF) matting[18], Learning Based (LB) matting [37], Global matting [13], Alpha matting [11], Comprehensive Sampling Sets (CSS) [28], Knn matting [7], Deep Image matting (DIM) [34], alphagan [23], Semantic Human matting [6]. Specifically, except for Semantic Human matting, other methods are interactive matting methods that need extra interactive trimaps as inputs. To compare fairly, we provide these interactive methods with the generated trimaps from our pipeline. We denote these methods as DT + X, where X represents previous methods excluding Semantic Human matting. For Semantic Human matting, we train the network on our dataset following their implementation.

5.2. Performance Comparison

We compare our method with the state-of-the-art automatic matting methods and interactive matting methods with generated trimaps on the synthetic text matting testing dataset. The trimaps of interactive matting methods are generated from our method.

The quantitative results are shown in Table 1, where αM for Alpha matting and Gan for alphagan. From the table, alphagan and DIM are better than other interactive matting methods because both of them utilize the deep matting networks and have a strong capacity to capture complicated patterns in the image. Our method outperforming other methods is mainly due to the attentional text region detector, deep networks, and joint training. The attentional text region detector provides the attention guidance for latter T-Net and M-Net. The joint training calibrates the T-Net and M-Net well. The qualitative result is shown in Figure 4. Our result looks better than those of other methods. Our result has fine details for each word and less false positive errors for the word boundary.

We further compare our method with the interactive methods with the trimap ground truth. The trimap ground truth is generated following the method we use to train T-Net. We denote these methods as GT + X, where X represents previous state-of-the-art interactive matting methods, including CF matting, LB matting, Global matting, Alpha matting, CSS, Knn matting, Deep Image matting, alphagan. The quantitative result is shown in Table 2. Our method outperforms most of baselines and our qualitative results are also superior as shown in Figure 5. Our method is quantitatively slightly weaker than GT + alphagan and the qualitative results of both methods are similar. However, alphagan needs GT trimaps as the inputs.

We also compare our method with the automatic matting method, Semantic Human matting. Table 1 shows that our method outperforms the Semantic Human matting method. It is mainly because we have an attentional text region detector that provides attentional guidance to find the area which tends to contain text. Latter networks predict the trimap and matte values in the text region and do not pay

Table 1. The quantitative analysis on synthetic text matting testing dataset. "DT+Method" means the method utilizes the generated trimap from our method. The scale of numbers in this table is 10^{-3}. The best results are emphasized in bold. M for MSE; S for SAD; G for gradient; C for connectivity

Methods	M	S	G	C
DT+CF	9.49	12.51	37.72	12.12
DT+LB	8.53	11.55	34.05	11.12
DT+Global	6.28	8.57	25.37	8.24
DT+αM	6.55	8.74	27.17	8.47
DT+CSS	11.66	15.62	36.98	15.39
DT+Knn	5.23	8.67	24.02	8.24
DT+DIM	5.12	6.04	29.35	6.01
DT+Gan	4.64	5.75	27.12	5.72
SHM	3.95	5.62	18.77	5.54
Ours	**1.47**	**2.77**	**7.10**	**2.64**

Table 2. The quantitative results on synthetic text matting testing dataset. "GT+Method" means the method utilizes the ground truth trimap from the dataset. The scale of numbers in this table is 10^{-3}. The best results are emphasized in bold. M for MSE; S for SAD; G for gradient; C for connectivity

Methods	M	S	G	C
GT+CF	7.12	10.46	30.96	10.08
GT+LB	6.55	9.81	29.02	9.40
GT+Global	3.83	6.59	17.76	6.28
GT+αM	3.90	6.12	19.22	6.02
GT+CSS	8.62	12.75	34.33	12.54
GT+Knn	2.45	4.89	13.21	4.65
GT+DIM	1.71	2.60	8.20	2.59
GT+Gan	**1.02**	**1.95**	**5.42**	**1.98**
SHM	3.95	5.62	18.77	5.54
Ours	1.47	2.77	7.10	2.64

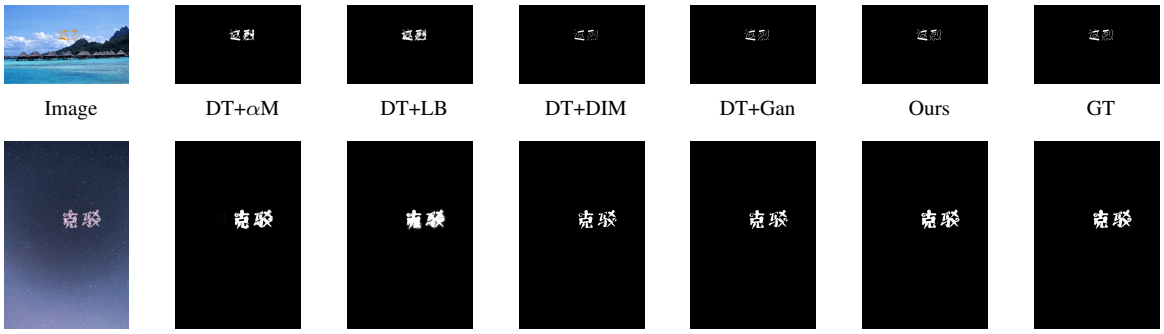

Figure 4. The qualitative results of our method and interactive matting methods with generated trimap from our pipeline on the synthetic text matting testing dataset

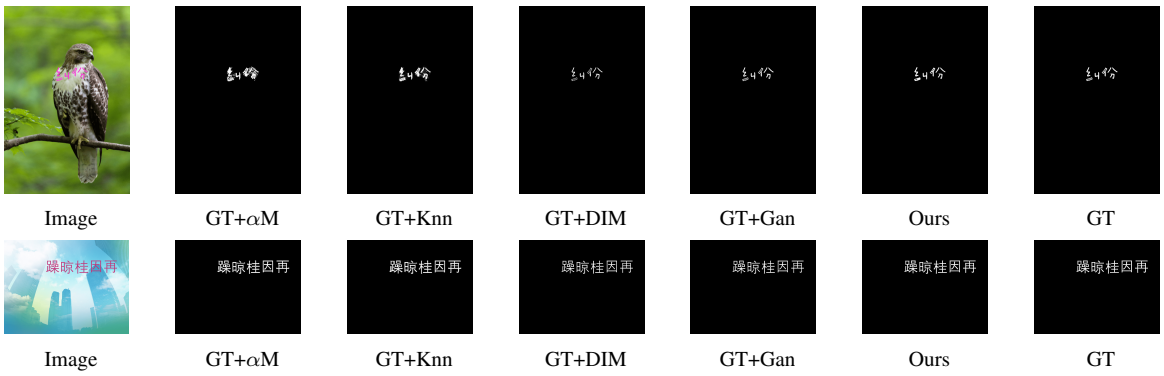

Figure 5. The qualitative results of our method and interactive matting methods with trimap ground truth on the synthetic text matting testing dataset

Figure 6. The qualitative results of our method and automatic matting method SHM on the synthetic text matting testing dataset

attention to the large background area. The qualitative result in Figure 6 shows our method is qualitatively better and has less false positive errors on the word boundary.

5.3. Ablation Study

Attentional Text Region Detector. To validate the effectiveness of attentional text region detector, we compare with a baseline that removes attentional text region detector and post-processing (SHM). We train T-Net and M-Net for the whole image with the same objective function. We can see from Table 3, our method performs better than the baseline on all four metrics. The reason is that the attentional text region detector detects text regions and filters out most background areas by the attentional mechanism. Therefore, the latter networks only need to focus on detected text regions.

Table 3. The quantitative results of ablation study on synthetic text matting testing dataset. The scale of numbers in this table is 10^{-3}. The best results are emphasized in bold. M for MSE; S for SAD; G for gradient; C for connectivity

Methods	M	S	G	C
No Joint Training	8.03	10.60	35.68	9.33
No $\mathcal{L}_t$	4.69	6.40	20.44	6.23
No Detection (SHM)	3.95	5.62	18.77	5.54
No Post Filtering	4.07	5.81	24.74	5.63
Ours_Matte Averaging	1.74	3.19	8.16	3.11
Ours_Matte Ranking	**1.47**	**2.77**	**7.10**	**2.64**

Post-processing Module. To demonstrate the effectiveness of the post-processing module, we design a baseline

Figure 7. The qualitative results of our method and baselines with generated trimaps from our pipeline on real Chinese and English posters, where GM is Global matting

without the post-processing module. T-Net and M-Net predict the matte value on the detected region with the highest confidence score. We can see from Table 3, ours with matte averaging or matte ranking post-processing methods are both better than the baseline without the post-processing module. These results show the superiority of our post-processing methods.

Joint Training of T-Net and M-Net. To show the effectiveness of joint training, we design a baseline with the pretrained T-Net and M-Net. We denote the baseline without joint training of T-Net and M-Net as no joint training. The comparison is shown in Table 3. The performance of the joint training network is better than no joint training network. The result shows the effectiveness of joint training of T-Net and M-Net. The reason is that the joint training of T-Net and M-Net integrates two sub-networks well. M-Net adapts the input from the trimap ground truth to the trimap prediction from T-Net, which boosts the performance.

Constraint of $\mathcal{L}_t$. To investigate the effect of $\mathcal{L}_t$ constraints in text matting, we use the pretrained T-Net and M-Net and remove the constraint of $\mathcal{L}_t$ in the joint training process. We denote joint training without $\mathcal{L}_t$ as No $\mathcal{L}_t$. Table 3 shows the performance of No $\mathcal{L}_t$ is better than no joint training and worse than that of our method. This constraint is useful in the joint training process that adapts M-Net to the trimap prediction from T-Net well.

5.4. Applying on Real Images

The images in our dataset are synthetically generated, so we validate our method on movie posters to show the generalizability of our method on real images for qualitative analysis. Matte results are shown in Figure 7. Although our model is trained on the synthetic dataset, our method still performs well on real movie posters, since the text shape and texture are well recovered by our synthetic dataset.

Furthermore, we use our pipeline to extract the movie titles from posters and re-composite the title matte with a new background to generate new movie posters. Figure 1 presents an example of a poster generation system with our automatic text matting system. Our poster generation system composes the movie poster title and a new creative background together to generate a new movie poster. The final composition result has high visual quality.

6. Conclusion and Discussion

In this paper, we focus on the text matting problem which shows great importance for a wide variety of applications. To overcome the three challenges of text matting, we propose a two-stage attentional text matting pipeline to solve the text matting problem. We utilize the attentional text region detector, automatic matting system, and post-processing module to obtain the matte prediction of the input image automatically. Furthermore, we construct a very large text matting dataset with high-quality annotations. Benefiting from the model structure and dataset, our automatic attentional text matting demonstrates its superiority over previous state-of-the-art image matting methods on the task of text matting.

This paper provides a good baseline system for future work. And our framework is so flexible that it can be easily adapted to a new scenario, such as curved/multilingual text matting by changing the attentional text region detection modules of the framework. For example, if we want to conduct curved text matting, we can utilize CRAFT [2] as our attentional text region detector. Moreover, if we want to build a multilingual text matting system, we can use pyramid mask text detector [20] as our attentional text region detector.

References

[1] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 29–37, 2017.

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 618–626. ACM, 2018.

[7] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013.

[8] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision*, pages 626–643. Springer, 2016.

[9] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *CVPR (2)*, pages 264–271, 2001.

[10] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Eduardo SL Gastal and Manuel M Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library, 2010.

[12] Dafang He, Xiao Yang, Chen Liang, Zihan Zhou, Alexander G Ororbi, Daniel Kifer, and C Lee Giles. Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3519–3528, 2017.

[13] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR 2011*, pages 2049–2056. IEEE, 2011.

[14] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2165–2172. IEEE, 2010.

[15] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055, 2017.

[16] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[17] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[18] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30:228–242, 2008.

[19] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] Jingchao Liu, Xuebo Liu, Jie Sheng, Ding Liang, Xin Li, and Qingjie Liu. Pyramid mask text detector. *arXiv preprint arXiv:1903.11800*, 2019.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088*, 2018.

[24] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[25] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[27] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1826–1833. IEEE, 2009.

[28] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643, 2013.

[29] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *European Conference on Computer Vision*, pages 92–107. Springer, 2016.

[30] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 315–321. ACM, 2004.

[31] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016.

[32] Jue Wang and Michael F Cohen. Optimized color sampling for robust matting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[33] Yue Wu and Prem Natarajan. Self-organized text detection with minimal post-processing via border learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5000–5009, 2017.

[34] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2970–2979, 2017.

[35] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4159–4167, 2016.

[36] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[37] Yuanjie Zheng and Chandra Kambhamettu. Learning based digital matting. In *2009 IEEE 12th international conference on computer vision*, pages 889–896. IEEE, 2009.

[38] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.