TFMNet: Trimap-free Real-time Image Matting Algorithm Based On Deep Learning

Ge Peng and Jingzong Yang*

School of Big Data, Baoshan University, Baoshan Yunnan 678000, China

** Corresponding author. E-mail: yjingzong@foxmail.com*

Received: Jan. 16, 2023; Accepted: Aug. 31, 2023

The conventional image matting algorithms needed priori manual Trimap information to produce excellent matting results which made real time matting impossible. To tackle the problem, a Trimap-free image matting network, TFMNet, is proposed in this paper. The proposed network consists of four modules, ConvNeXt backbone module for image features extraction, Trimap prediction module for normalized Trimap generation, glance matting module for rough matting results prediction, and post-processing module for exact matting results production. To further optimize the training process of the proposed model, an improved Loss function based on frequency domain information is proposed. In experiment, Sets of Experiments designed by variable controlling approach prove that the proposed TFMNet do well in real time image matting. The TFMNet model achieves 8.99, 0.011, 12.31, 11.15 in the accuracy metrics of SAD, MSE, GRAD, CONN, respectively, costs 51ms for one image averagely which meet the real-time requirements, and model size is 671M. Besides, further experiments conducted by comparing with five state-of-the-art models based on three typical matting databases demonstrate the superiority of the proposed algorithm.

Keywords: real-time image matting; image semantic segmentation; convolution neural network without pooling; image processing

http://dx.doi.org/10.6180/jase.202404_27(4).0009

1. Introduction

Image matting [1], an essential research direction in image processing, aimed to extract foreground targets from input images or videos, and then synthesized the foreground targets into new scenes to produce synthesized images or videos. It was widely used in image processing and movie production. Such as, Passport photo synthetization which had no requirement for real-time performance, video background substitution and virtual background for video conferencing which needed image matting algorithm with better performances on real-time. In numerous occasions, the real-time performance of the image matting algorithm was particularly important.

On the image matting task, input image I was understood as generated by the foreground image F, the background image B, and the transparency $\alpha \in [0,1]$ according to Eq. (1), where the symbol "·" denoted matrix dot product.

$$I = \alpha \cdot F + (1-\alpha) \cdot B \tag{1}$$

The image matting algorithm tried to estimate the foreground image $\widehat{F}$ from the input image I accurately. According to Eq. (2), $\widehat{F}$ could be generated based on the transparency prediction $\hat{\alpha}$ and input image I. That was, the core of the image matting algorithm was to approximate $\hat{\alpha}$ from I.

$$\widehat{F} = \hat{\alpha} \cdot I \tag{2}$$

According to Eq. (1), it would be an unsolvable problem

to solve $\hat{\alpha}$ with only the image I known. Only more priori information added could be attached to find the approximate $\hat{\alpha}$. Trimap was commonly used as priori information to get accurate matting results in image matting tasks, where the Trimap referred to a grayscale map consisting of three pixel values (0, 128, 255), which was used to exactly identify the background pixels, or uncertain pixels, or foreground pixels in an image. On the basis of priori Trimap information, Xu et al. [2] proposed DIM (Deep Image Matting) algorithm based on supervised Fully Convolutional Networks (FCN), which obtained good results in the image matting task for the first time based on deep learning, but the matting accuracy needed to be improved further. Park et al. [3] proposed MatteFormer matting model, in which a pre-trained transformer model was used as the backbone to further improve the matting accuracy. However, the two algorithms mentioned above required priori Trimap information as the model input to guide better matting results, which meant that manual annotation of Trimap information was required before each matting, and real-time image matting could not be accomplished. Another typical prior information was the background information. Lin et al. [4] proposed a background based image matting algorithm, achieved satisfied real-time image matting results. However, it was limited to the case of video matting task where the background of input image tended to be constant, and it was unable to perform the case with various background.

Therefore, scholars devoted to Trimap generation algorithms. Generally, the input images were classified into three categories: (1) images with significant and non-transparent foreground targets ($\alpha = 0$ or 1), such as human portraits and animals; (2) images with significant and transparent foreground targets $(0 \leq \alpha < 1)$, such as glass cups and plastic bags; (3) images with insignificant foreground targets $(0 \leq \alpha < 1)$, such as mesh images such as spider webs. Henry and Lee [5] and Li et al. [6] worked on Trimap automatic generation algorithm and proposed Trimap generation algorithms without human intervention, but the algorithms could only generate Trimap for class (1) images more accurately, but not for class (2) and (3) images. Another part of scholars [7, 8] tried to implement an "end-to-end" image matting algorithm, but still could not handle class (2) and (3) images well. The Unet [9–11] was wildly used in image generation, image classification and image segmentation. Li et al. [12] proposed a normalized Trimap generation method to deal with the Trimap generation problem for three kinds of images, and implemented Trimap prediction based on the supervised ResNet-Unet semantic segmentation model, so as to achieve accurate real-time matting. However, it led to wrong results for the reason that the ResNet-Unet semantic segmentation model was difficult to achieve the ideal Trimap generation results in practice. To improve the performance of image semantic segmentation, scholars [13, 14] had tried to use networks based on the swin-transformer model to improve performances in tasks of medical images and traffic images semantic segmentation. However, the main barrier of swin-transformer was that the input images must be with same size. Although this problem could be circumvented by cropping, it was still impossible to really break this barrier. Liu et al. [15] proposed ConvNeXt model, and it was demonstrated that the convolutional neural network could achieve equivalent results as swin-transformer by tuning parameters in their work, and it also broken through the input image size limitation barrier.

In summary, there were two main challenges in current image matting algorithm: The Trimap-based image matting algorithm performed satisfied results but could not meet the real-time requirements; it was difficult to generate Trimap accurately for all types of images. Thus, A Trimap-free image matting algorithm, TFMNet, is proposed in this paper to solve the two challenges through the following four contributions.

(1) Using ConvNeXt as the backbone of image matting model, and using two different up-sampling networks to complete image semantic segmentation task and glance matting task, respectively, further improve the matting accuracy.

(2) Using normalized Trimap generation algorithm to train the model in an end to end way, which eliminates too much manually Trimap labeling and makes real-time image matting possible.

(3) Proposing non-pooling CNN network as the output layer for image matting model for the first time. Non-pooling CNN layer retains the location information of pixels more accurately and improves the matting accuracy.

(4) Proposing an improved regression loss function based on frequency domain information to improve the matting accuracy and coffin performance.

Based on this, a group of experiments about network training complex, training time-consuming, testing accuracy, efficiency [16, 17] is performed to demonstrate the properties of real-time and effectiveness by comparing the training and testing results based on one synthetic databases and two real-world databases.

2. Related works

2.1. Real-time Image Matting

Real-time performance is a key metrics for image matting tasks because of it makes real-time video processing pos-

sible. A Real-time image matting algorithm [4] should be meeting the following two requirements.

(1) Matting should not need any external input but original images. In detail, a Trimap is not necessary for the image matting process.

(2) Matting frame rate should attain 20 frame per second at least. It means that image matting algorithm can meet the lowest video frame rates, that is real-time matting.

The real-time performances of the proposed algorithm would be judged with the two points above.

2.2. The Normalized Trimap Generation Algorithm

Trimap was a manually annotated generated grayscale map, consisting of three pixel values (0, 128, 255), which identified the background pixels, uncertain pixels and foreground pixels of an image accurately. Li et al. [12] proposed a normalized Trimap generation method to reduce the workload of manually Trimap labeling. They classified images into three categories according to types of foreground targets, and Trimaps were generated with method corresponding to categories of the input images.

(1) Images with significant non-transparent foreground, called Salient Opaque (SO) image, always contained foreground, background and uncertain regions, and the Trimap of this category could be generated from α by image erosion and expansion;

(2) Images with significant transparent foreground, called Salient Transparent/Meticulous (STM) image: contained background and uncertain regions, and the Trimap could generated by labelling all pixels with $\alpha > 0$ as *128*, as *0* otherwise;

(3) Images with insignificant foreground, called Non-Salient (NS) image, contained only uncertain regions, the Trimap should be equal to *128* constantly.

Fig. 1. Three types of images, and their trimaps respectively

As shown in Fig. 1, the three types of normalized Trimap could be generated based on the method mentioned above. The black, gray and white colors denoted as the values of 0, 128 and 255, respectively.

2.3. ConvNeXt Network

Image semantic segmentation networks consisted of an Encoder layer for image features extracting and a Decoder layer for image dimensions recovering. Encoder included a backbone network loading pre-trained models, where ResNet network [18] was used most typically. With the continuous advancement of deep learning in the field of image processing, much more convolutional networks such as MobbileNet [19], EfficientNet [20], RegNet [21] were used as backbone in semantic segmentation tasks. Swin-transformer model [22, 23] performed outstanding in image semantic segmentation tasks. However, one of the shortcomings of the Swin-transformer model was that it could not accept input image with different size, which severely reduced its usefulness in downstream tasks. Liu et al. [15] demonstrated that CNN networks possessing a similar structure to the Swin-transformer could achieve comparable performance to Swin-transformer. Moreover, the ConvNeXt network possessed all the advantages of CNN networks, including being robust to image size. In summary, the ConvNeXt network would be used as the backbone of the proposed TFMNet model in this paper.

2.4. Conventional Regression Loss

The image matting algorithm aimed at predicting $\alpha \in [0, 1]$ of the input image, thus it was a regression task. L_1 Loss and L_2 Loss were the typical loss functions for the regression tasks. This part would briefly elaborate on the two loss functions.

L_1 Loss, Mean Absolute Error, was calculated as shown in Eq. (3). It had stable gradient, but was easy to produce oscillation when the error value was small, which affected the convergence of the model.

$$L_1(y, \hat{y}) = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \tag{3}$$

L_2 Loss, Mean Square Error, was calculated as shown in Eq. (4). L_2 Loss decreased as the error value decreasing. And if the sample error value was less than 1, the training speed would tend to be slow.

$$L_2(y, \hat{y}) = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{4}$$

Where $\hat{y}$ denoted the predicted value, y denoted the true value, and N denoted the number of samples. To calculate the matting loss of α using L_2 Loss would lead to slow training speed. An improved image matting Loss calculation method will be proposed in section 3.5

3. The proposed image matting model

In this paper, we propose an end-to-end Trimap-free matting algorithm: TFMNet, which consists of four core modules: ConvNeXt backbone module, Trimap prediction module, glance matting module and post-processing module as shown in Fig. 2. This section will discuss the four modules and the Loss function of TFMNet.

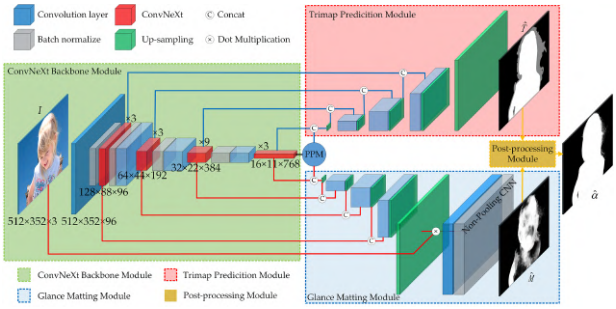

Fig. 2. The proposed TFMNet model

3.1. ConvNeXt Backbone Module

As discussion above, ConvNeXt network can be used to process input images with arbitrary size and has comparable feature extraction performance with Swin-transformer. A pre-trained with ImageNet database ConvNeXt network will be used as backbone to complete image feature extraction in this paper.

3.2. Trimap Prediction Module

The Trimap prediction module contains a set of up-sampling networks based on transposed convolution. To combining Trimap prediction module with ConvNeXt backbone module, a U-shaped image semantic segmentation network is formed for Trimap prediction. Based on the database composed of the original images and Trimap labels mentioned in section section 2.2, the Loss of U-shape network can be calculated for network training. Thus, the Trimap prediction module can produce Trimap predictions, as shown in Fig. 3, for real-time matting.

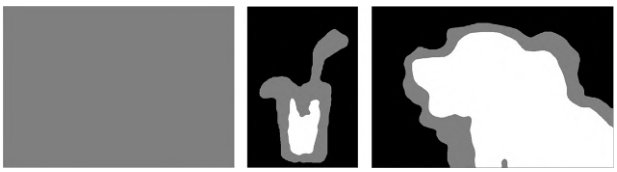

Fig. 3. Trimap prediction results

3.3. Glance Matting Module

The glance matting module consists of a set of up-sampling networks and CNN without pooling layers. Similarly, Joining the ConvNeXt backbone module with the glance matting module, a U-shaped network will be formed for glance matting. As we all know, the pooling layer of CNN can increase the network perceptual field and reduce the model parameters, which is important for the image classification but bad for the regression contrarily. Because of that the pooling layer will weaken the location information of pixels which will lead to a wrong regression prediction results. Therefore, we propose a novel non-pooling CNN network, as shown in Fig. 4, to improve the performance of glance matting module. Based on a dataset consisting of the original images with alpha labeling, we are able to get the Loss of the glance matting network to train the matting network.

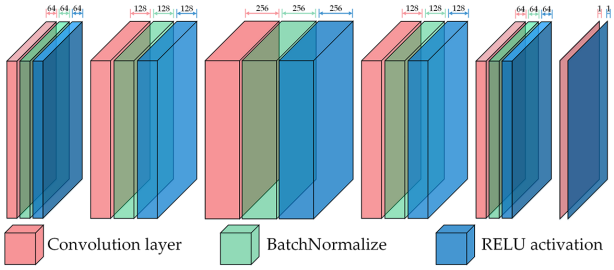

Fig. 4. Non-pooling CNN network

From Fig. 5, it can be seen that the glance matting module can produce image details, like hair, well but significant region of foreground. In order to further improve the matting accuracy, a post-processing module based on the Trimap predictions and the glance matting results is proposed in section 3.4

Fig. 5. Glance matting results

3.4. Post-processing Module

In accordance with the discussion in section 2.2, Trimap prediction results can accurately distinguish the foreground, background, and uncertain regions, glance matting results can produce results containing details correctly. To produce the satisfied matting results, a post-processing module is proposed in this paper as shown in Fig. 6 and Eq. (5).

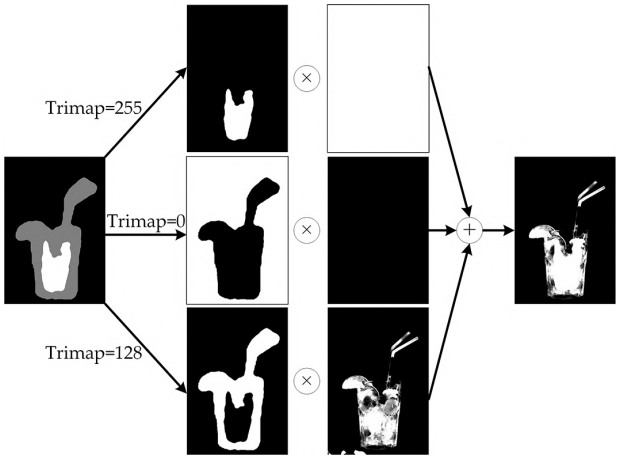

Fig. 6. Post-processing module

$$\hat{\alpha} = \widehat{\alpha_F} + \widehat{\alpha_B} + \widehat{\alpha_N} = \begin{cases} \widehat{\alpha_F} = 1, \widehat{T} = 255 \\ \widehat{\alpha_B} = 0, \widehat{T} = 0 \\ \widehat{\alpha_N} = \widehat{M}, \widehat{T} = 128 \end{cases} \quad (5)$$

Where $\hat{\alpha}$ denotes the matting result, $\widehat{\alpha_F}$, $\widehat{\alpha_B}$, and $\widehat{\alpha_N}$ denote the matting results of foreground, background, and uncertain pixel points, respectively. $\hat{T}$ denotes the Trimap prediction result, and $\widehat{M}$ denotes the glance matting result.

3.5. Loss Function

As shown in Eq. (6), we can obtain $\hat{T}$, $\widehat{M}$ and $\hat{\alpha}$ based on TFMNet with original input image I. The Loss of the TFMNet model L_α should be consisted of the Trimap prediction Loss L_T and the glance matting Loss L_M.

$$(\widehat{T}, \widehat{M}) = \text{TFMNet}(I) \quad (6)$$

3.5.1. Loss of Trimap Prediction

Trimap consists of foreground, background and uncertain regions, and the distribution of the them is relatively uniform. Thus, Loss of Trimap prediction L_T can be calculated using Cross Entropy loss as shown in Eq. (7). Where T denotes the true value of normalized Trimap.

$$L_T = \text{CrossEntropy}(T, \hat{T}) \quad (7)$$

3.5.2. Loss of Glance Matting

Mechrez et al. [24] confirmed that L_1 and L_2 Loss will lead to elimination of details information which are important for image matting. A novel Loss function, denoted as FDL_1, retaining more detail information by using frequency domain information, is proposed in this section. a mean filter is used to obtain high-frequency information (containing more image details) and low-frequency information of α. And then, the FDL_1 tries to protect more detail information by controlling the rate of high frequency information loss in total Loss L_M using two hyper parameters β_1 and β_2 as given in Eqs. (8) and (9).

$$L_M = FDL_1\left(\widehat{M_H}, \widehat{M_L}\right) \begin{cases} \widehat{M_L} = \text{AveFilter}\left(\widehat{M}\right) \\ \widehat{M_H} = \widehat{M} - \widehat{M_L} \end{cases} \quad (8)$$

$$FDL_1 = \beta_1 L_1\left(M_H, \widehat{M_H}\right) + \beta_2 L_1\left(M_L, \widehat{M_L}\right) \quad (9)$$

Where M_L and M_H denote the low-frequency and high-frequency information of α, respectively. $\widehat{M_L}$, $\widehat{M_H}$ denote the low-frequency and high-frequency information of $\widehat{M}$. *AveFilter* denotes the mean filter. β_1 and β_2 are used to set the proportion of low-frequency and high-frequency information in L_M. In addition, $\beta_1 = 0.6$ and $\beta_2 = 0.4$ in this paper.

3.5.3. Total Loss for Multiple Tasks

Generally, the total loss of multiple tasks is calculated by additive approach simply. However, since the two tasks in this paper have different value scales, so we calculate the total Loss L_α based on Eq. (10). Two hyper parameters a_T and a_M can be set to a suitable values to perform better convergence for both tasks. In addition, $a_T = 1$ and $a_M = 5$ in this paper.

$$L_\alpha = a_T L_T + a_M L_M \quad (10)$$

Above all, an end-to-end training and testing for the proposed TFMNet can be completed base on the four modules and the improved Loss function.

4. Experimental results and discussions

4.1. Experimental Databases

Three classical databases for the image matting tasks, the DIM dataset [2], the AM-2k dataset [7], and the AIM-500 dataset [12], are used as the experimental databases in this paper as shown in Table 1

Table 1. Information of databases

Database	Image types	Size of training set	Size of testing set
DIM	SO, STM, NS	49300	1000
AM-2K	SO	1800	200
AIM-500	SO, STM, NS	——	500
Total	SO, STM, NS	51100	1700

The DIM database is a synthetic image database. In this database, 493 accurately labeled foreground images, including SO image, STM image, and NS image are synthesized with 100 random background images from COCO-2014

Table 2. Parameters setting of Loss function comparison experiments

Model	Backbone	Up-sampling network	Output layer	Loss function
Ct-NP-TFMNet-L1	ConvNeXt-tiny	Trimap prediction + Non-pooling glance matting	Post-processing	L_1
Ct-NP-TFMNet-L2				L_2
Ct-NP-TFMNet-FD				FDL_1

dataset [25] to generate 49300 images based on Eq. (1) as the training set, and 1000 images generated in a same way based on 50 foreground images and 20 background images as the test set.

The AM-2K database is a real image database. It consists of 2000 images including monkeys, elephants, rabbits and other 20 types of foreground (SO image) with different backgrounds. Of these, 1800 images are used as the training set and 200 images are used as the test set.

The AIM-500 database is a real image database. It consists of 500 accurately labeled images with three types of foregrounds including SO, STM, and NS. All images are used as the training set.

In summary, the experimental database used in this paper consists of 52800 images, of which 50000 will be used as the training set, 1100 as the validation set, and 1700 as the test set. It should be noted that the longer side of the image is set to 512 in order to standardize the image size as much as possible. Later experiments are all based on this database.

4.2. Experimental Environment

The experiments were finished on Windows using the Pytorch deep learning framework with hardware configurations as follow: Intel(R) Core(TM) i7-12700KF 3.60 GHz CPU, NVIDIA Geforce RTX 3090 Ti GPU, and 32G RAM.

4.3. Experimental Settings

In order to verify the effectiveness of the proposed algorithm, three comparison experiments are designed as follow: Loss function comparison experiments, non-pooling CNN comparison experiments and backbone comparison experiments, which are used to prove the effectiveness of FDL_1 Loss function, non-pooling CNN layer and ConvNeXt Backbone, respectively.

4.3.1. Settings of Loss function comparison experiments

Three experiments based on the TFMNet network consisting of ConvNeXt-tiny back-bone module, Trimap prediction module, pooling-free glance matting module and post-processing module but with different Loss function are designed to compare the three types of matting Loss functions L_1, L_2 and FDL_1, designated by Ct-NP-TFMNet-L1, Ct-NP-TFMNet-L2, and Ct-NP-TFMNet-FD as described in Table 2. In addition, FDL_1 has superior performance according to the discussion in section 4.4, so unless specifically stated later, the FDL_1 is used in TFMNet by default.

4.3.2. Settings of non-pooling CNN comparison experiments

Two experiments based on the TFMNet network with different glance matting module are designed to demonstrate the effectiveness of Non-pooling CNN layer, denoted by Ct-TFMNet, Ct-NP-TFMNet as shown in Table 3.

4.3.3. Settings of backbone comparison experiments

In this section, six experiments based on the TFMNet network with different back-bone are designed to compare the effectiveness of ConvNeXt-tiny, ConvNeXt-base, ConvNeXt-large, ResNet-18, ResNet-34 and ResNet-50 backbone, denoted by Ct-NP-TFMNet, Cb-NP-TFMNet, Cl-NP-TFMNet, Res18-NP-TFMNet, Res34-NP-TFMNet, and Res50-NP-TFMNet, as shown in Table 4. In addition, the glance matting module with non-pooling CNN layer is used in this section.

4.4. Discussions of experimental results

Based on the above three sets of comparative experiments, results of specific experiments are discussed in this section.

4.4.1. Evaluation indicators for image matting results

Firstly, four indicators are used for evaluating the experimental alpha matte results of image matting algorithm in this paper. The four evaluation metrics [26–29] include: SAD (Sum of Absolute Difference), MSE (Mean Squared Error), GRAD (Gradient) and CONN (Connectivity) proposed by Rhemann et al. [30].

(1) SAD represents the total absolute differences of the alpha matte prediction from ground truth as shown in Eq. (11).

$$SAD(\alpha, \hat{\alpha}) = \sum |a - \hat{\alpha}| \tag{11}$$

(2) MSE represents the mean squared error of the alpha matte prediction from ground truth as shown in Eq. (12).

$$\text{MSE}(\alpha, \hat{\alpha}) = \text{Mean}\left((a - \hat{\alpha})^2\right) \tag{12}$$

(3) GRAD represents the gradient error of the alpha matte prediction from ground truth as shown in Eq. (13), and the smaller GRAD, the more similarity.

Table 3. Parameters setting of non-pooling CNN comparison experiments

Model	Backbone	Up-sampling network		Output layer
		Trimap prediction	Glance matting	
Ct-TFMNet Ct-NP-TFMNet	ConvNeXt-tiny	Trimap prediction	– Non-pooling CNN	Post-processing

Table 4. Parameters setting of backbone comparison experiments

Model	Backbone	Up-sampling network	Output layer
Ct-NP-TFMNet	ConvNeXt-tiny	Trimap prediction + Non-pooling CNN glance matting	Post-processing
Cb-NP-TFMNet	ConvNeXt-base		
Cl-NP-TFMNet	ConvNeXt-large		
Res18-NP-TFMNet	ResNet-18		
Res34-NP-TFMNet	ResNet-34		
Res50-NP-TFMNet	ResNet-50		

$$GRAD(\alpha, \hat{\alpha}) = \sum(\nabla a - \nabla \hat{\alpha})^2 \tag{13}$$

(4) CONN represents the degree of connectedness of alpha matte prediction from ground truth as shown in Eq. (14) discussed in Radosavovic et al. [21], and the smaller CONN, the more exact prediction is.

$$\text{CONN}(\alpha, \hat{\alpha}) = \sum(\varphi(a, \Omega), \varphi(\hat{a}, \Omega))^2 \tag{14}$$

4.4.2. *Results and discussions of Loss function comparison experiments*

Based on the three models in section 4.3.1, the experimental results derived after 50 Epochs are shown in Table 5 in detail.

Table 5. Loss function comparison experimental results

Model	SAD	MSE	GRAD	CONN
Ct-NP-TFMNet-L1	14.16	0.026	18.25	14.51
Ct-NP-TFMNet-L2	16.27	0.031	**17.80**	16.00
Ct-NP-TFMNet-FD	**13.88**	**0.024**	18.13	**14.18**

From image matting accuracy, Ct-NP-TFMNet-FD performs better than the others. The reason is that the FDL_1 Loss function retains more detail information. Wherein, Ct-NP-TFMNet-L2 performs weakest because of that L_2 Loss leads to slower training speed and does not reach convergence within 50 Epochs.

From training process, the processes of testing Loss convergence of the three models within 50 Epochs are shown in Fig. 7. It is evident that FDL_1 converge faster than L_1 and L_2, and achieve better training results by focusing more on low-frequency information.

For more intuitive representations of matting performances, some examples of experimental results are given in Fig. 8. From the results, it can be seen that the Ct-NP-TFMNet-FD model produces better results within less training epochs. Thus, the FDL_1 Loss will be used by default if no specifically stated or marked latter.

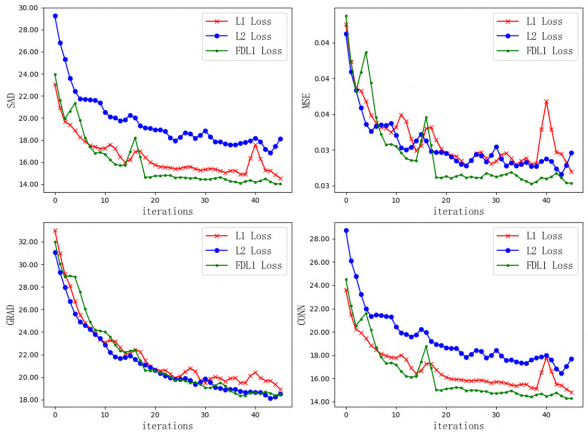

Fig. 7. Training results within 50 epochs

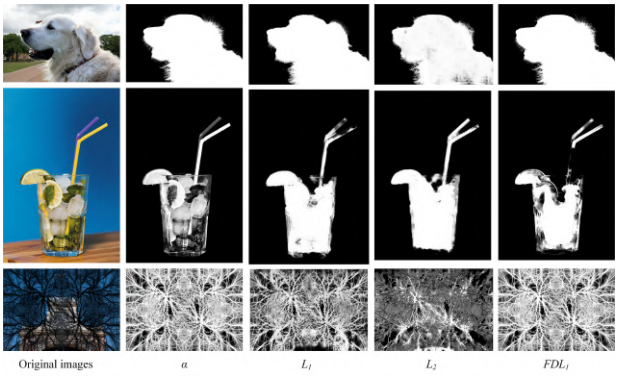

Fig. 8. Samples of Loss function comparison experimental results

4.4.3. *Results and discussions of non-pooling CNN comparison experiments*

Based on the two models in section 4.3.2, the experimental results derived after 50 Epochs are shown in Table 6 and Fig. 9 in detail.

Table 6. Non-pooling CNN comparison experimental results

Model	SAD	MSE	GRAD	CONN
Ct-TFMNet	50.33	0.125	56.45	60.23
Ct-NP-TFMNet	**13.88**	**0.024**	**18.13**	**14.18**

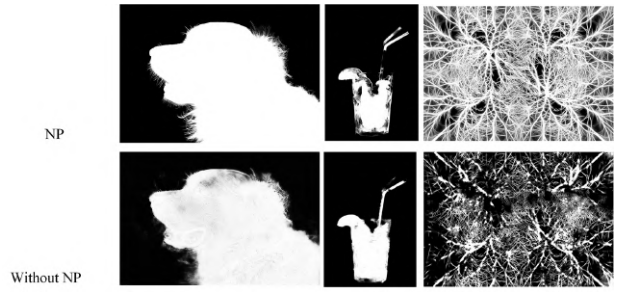

Fig. 9. Samples of non-pooling CNN comparison experimental results

From the results, The Ct-TFMNet model can be seen to perform substantially worse than Ct-NP-TFMNet for the reason that the non-pooling CNN layer further improves the matting performance.

4.4.4. *Results and discussions of backbone comparison experiments*

Based on the six experiments with different backbone, Ct-NP-TFMNet, Cb-NP-TFMNet, Cl-NP-TFMNet, Res18-NP-TFMNet, Res34-NP-TFMNet, and Res50-NP-TFMNet, the experimental results derived after 50 Epochs are shown in Table 7 and Fig. 10 in detail.

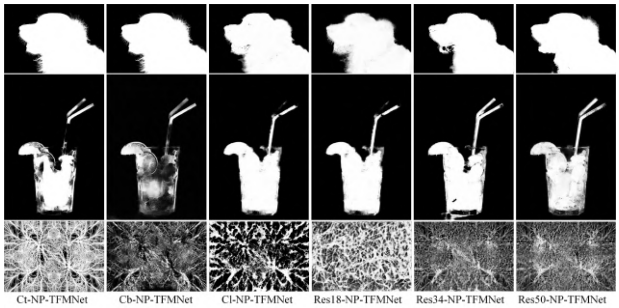

Fig. 10. Samples of backbone comparison experimental results

From image matting accuracy, Cb-NP-TFMNet model produces the highest matting accuracy, Ct-NP-TFMNet the second and Cl-NP-TFMNet the worst within 50 training Epochs. It obvious that the matting accuracy does not increase consistently with the size of model in limited training epochs. The models based on the ResNet network can be seen to give worse performances than the ConvNeXt-based model. In particular, the Res34-NP-TFMNet model achieves comparable matting results with Ct-NP-TFMNet. Overall, the Cb-NP-TFMNet achieves the best matting accuracy from Table 7 and Fig. 10.

From image matting efficiency, the six models mentioned above do not require manual Trimap information, do have the conditions to complete real-time matting. In contrast, the smaller Ct-NP-TFMNet and Res18-NP-TFMNet are more efficient with an average matting time of 32ms and 20ms for one image, respectively. Cb-NP-TFMNet, with the best matting accuracy, takes 51ms per image averagely what achieves a quasi-real-time matting of 20 frames per sec.

4.5. Comparison with the state-of-the-art

Above all, the proposed TFMNet model consisting of ConvNeXt-base Backbone, Trimap prediction module, pooling-free CNN glance matting module and FDL_1 Loss function performs best in the matting comparison experiments. In order to further discuss the effectiveness of the proposed algorithm, a set of comparison experiments between the proposed algorithm and the current state-of-the-art image matting algorithms, DIM [2], Matteformer [3], MODNet [8], GFM [7], and AIM [12], is conducted based on the experimental databases in this paper. The detailed comparison experimental results are listed in Table 8 and Fig. 11 provides some samples matting results of different algorithms for the three types of images.

In terms of image matting accuracy, the DIM model and the Matteformer model, both of which require manual Trimap as auxiliary input, achieve performance for matting accuracy. The Matteformer model inherits the advantages of the Swin-Transformer model and achieves better matting accuracy than the rest of the "Trimap-free" models including the TFMNet model with the help of manual Trimap. In addition, TFMNet will provide comparable results to Matteformer if accurate manual Trimap is provided. However, the Trimap-based algorithms cannot tackle the problem of real time as discussed previously.

Focus on the Trimap-free image matting algorithms, a group of comparison experiments between MODNet, GFM model, AIM model, and the proposed TFMNet is conducted in this paper. From Table 8 and Fig. 11, it is obvious that the performances of MODNet and GFM model is not

Table 7. Backbone comparison experimental results

Model	SAD	MSE	GRAD	CONN	Model size	Time cost per image
Ct-NP-TFMNet	13.88	0.024	18.13	14.18	214M	32ms
Cb-NP-TFMNet	**8.99**	**0.011**	**12.31**	**11.15**	671M	51ms
Cl-NP-TFMNet	15.54	0.031	22.09	15.63	1.46G	75ms
Res18-NP-TFMNet	40.25	0.096	38.26	38.65	**78M**	**20ms**
Res34-NP-TFMNet	14.21	0.028	18.14	17.33	119M	45ms
Res50-NP-TFMNet	33.71	0.108	32.24	33.97	905M	77ms

Table 8. Experimental results compared with state-of-the-art

Model	SAD	MSE	GRAD	CONN	Model size	Time cost per image
DIM+Trimap [2]	6.33	0.008	6.38	6.41	292M	-
Matteformer+Trimap [3]	**4.15**	**0.005**	**3.75**	**4.13**	513M	-
MODNet [8]	37.79	0.086	28.97	37.76	**25M**	11ms
GFM [7]	41.23	0.096	32.69	38.95	480M	46ms
AIM [12]	29.71	0.066	28.16	26.59	211M	32ms
TFMNet	**8.99**	**0.011**	**12.31**	**11.15**	671M	51ms
TFMNet+Trimap	4.23	0.005	4.01	4.19	671M	-

really good. The reason is that the two models are focus on SO images (human portrait or animal) while the matting results of human foreground and animal foreground are remarkable doubtlessly. In other words, MODNet and GFM model have made great contributions for matting task of SO image but with drawback of not working for STM and NS images. The AIM model proposes a normalized Trimap generation method for SO, STM and NS images to make real-time matting practicable. However, it can be seen that the model has much more misjudgments in image semantic segmentation from the matting results of AIM model for the reason that the performance of ResNet Backbone is not as good as that of ConvNeXt used in this paper which would lead to the misjudgments to a certain extent. The Trimap-free TFMNet model proposed in this paper achieves a better matting accuracy, which is not as good as the Trimap-based model but meet the demand basically. The good performances should be attributed to ConvNeXt-based Backbone, the non-pooling CNN layer, post-processing module, and the improved matting loss function proposed in this paper.

In terms of image matting efficiency, the DIM model and Matteformer model cannot execute real-time matting because of the requirements of manual interaction information. Comparing the remaining four models in Table 8, we can find that MODNet takes the shortest time, with an average matting time cost of 11ms per image, which can accomplish matting for 100 images per second. TFMNet takes the longest time, 51ms, for a single image, and achieves a real-time matting of 20 frames/s on GPU. Focusing on the model size, the proposed TFMNet occupies 671M memory space, which is larger than the rest of the models. The first reason is that ConvNeXt-base is chosen as the model Backbone, which has relatively large model size itself. The more important reason is that the non-pooling CNN layer makes the model parameters and computational weight increase significantly. These reasons will be taken as a direction for our later work. Yet, in terms of the average time consumed for a single image, TFMNet is able to achieve quasi-real-time matting at 20 frames/second.

In summary, by comparing with five representative image matting algorithms, the proposed TFMNet can be proved to achieve better performance of matting accuracy and effectiveness, and it can meet the two requirements mentioned in section 2.1, namely a real-time image matting algorithm.

5. Conclusions

To address the problem of that image matting tasks suffered from the reliance on manual Trimap information and that current algorithms could not generate Trimap for various types of images accurately and automatically, a Trimap-free deep learning mat-ting algorithm, TFMNet, is proposed in this paper. TFMNet uses a multiple task deep learning model based on ConvNeXt Backbone to perform accurate and effective real-time matting. Accordingly, we focus on the study of the model output layer and the Loss function, proposing non-pooling CNN output layer, post-processing module and improved regression Loss function based on frequency domain information. In the experiments, based on three typical matting databases DIM, AM-2k and AIM-500, the accuracy and real-time performance of the proposed TFMNet model is verified to achieve better performance of matting accuracy and effectiveness by comparing

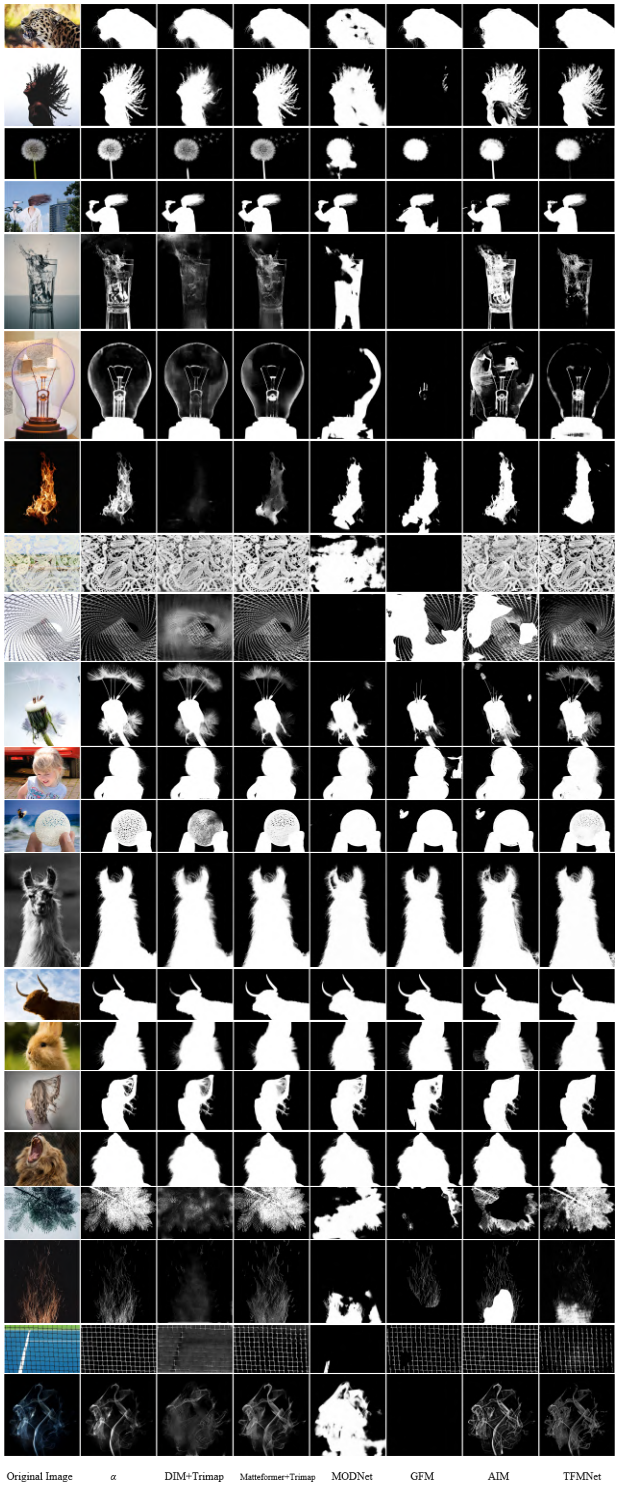

Fig. 11. Samples of experimental results

with five representative matting models DIM, Matteformer, MODNet, GFM and AIM.

However, there are still some problems in the efficiency of the proposed model that the current matting accuracy far from being able to be applied to practice, and the model size of the proposed algorithm needs further optimization. And the limitations of the proposed improved loss function based on frequency domain information are not discussed extensively. The future work should be focus on the problems above.

6. Acknowledgements

This work was funded by Natural Science Research Projects of Baoshan University (Grant No. ZKMS202104), 10th batches of Baoshan young and middle-aged leaders training project in academic and technical (Grant No. 202109), Baoshan Xingbao Young Talent Training Project (Grant No. 202303).

References

[1] R. Brinkmann. *The art and science of digital compositing: Techniques for visual effects, animation and motion graphics*. Morgan Kaufmann, 2008.

[2] N. Xu, B. Price, S. Cohen, and T. Huang. "Deep image matting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2970–2979. DOI: 10.1109/CVPR.2017.41.

[3] G. Park, S. Son, J. Yoo, S. Kim, and N. Kwak. "Matteformer: Transformer-based image matting via prior-tokens". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11696–11706. DOI: 10.1109/CVPR52688.2022.01140.

[4] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman. "Real-time high-resolution background matting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8762–8771. DOI: 10.1109/CVPR46437.2021.00865.

[5] C. Henry and S.-W. Lee, (2019) *"Automatic trimap generation and artifact reduction in alpha matte using unknown region detection"* **Expert Systems with Applications *133***: 242–259. DOI: 10.1016/j.eswa.2019.05.019.

[6] J. Li, G. Yuan, and H. Fan, (2020) *"Robust trimap generation based on manifold ranking"* **Information Sciences *519***: 200–214. DOI: 10.1016/j.ins.2020.01.017.

[7] J. Li, J. Zhang, S. J. Maybank, and D. Tao, (2022) *"Bridging composite and real: towards end-to-end deep image matting"* **International Journal of Computer Vision *130***(2): 246–266. DOI: 10.1007/s11263-021-01541-0.

[8] Z. Ke, J. Sun, K. Li, Q. Yan, and R. W. Lau. "Modnet: Real-time trimap-free portrait matting via objective decomposition". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. ***36***, 1140–1147. DOI: 10.48550/arXiv.2011.11961.

[9] A. Bilal, L. Zhu, A. Deng, H. Lu, and N. Wu, (2022) *"AI-based automatic detection and classification of diabetic retinopathy using U-Net and deep learning"* **Symmetry *14***(7): 1427. DOI: 10.3390/sym14071427.

[10] A. Bilal, G. Sun, S. Mazhar, A. Imran, and J. Latif, (2022) *"A Transfer Learning and U-Net-based automatic detection of diabetic retinopathy from fundus images"* **Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization *10***(6): 663–674. DOI: 10.1080/21681163.2021.2021111.

[11] A. Bilal, M. Shafiq, F. Fang, M. Waqar, I. Ullah, Y. Y. Ghadi, H. Long, and R. Zeng, (2022) *"IGWO-IVNet3: DL-Based Automatic Diagnosis of Lung Nodules Using an Improved Gray Wolf Optimization and InceptionNet-V3"* **Sensors *22***(24): 9603. DOI: 10.3390/s22249603.

[12] J. Li, J. Zhang, and D. Tao. "Deep automatic natural image matting". In: *Proceedings of International Joint Conferences on Artificial Intelligence Organization, Montreal-themed virtual reality*, 800–806. DOI: 10.48550/arXiv.2107.07235.

[13] E. K. Aghdam, R. Azad, M. Zarvani, and D. Merhof, (2022) *"Attention swin u-net: Cross-contextual attention mechanism for skin lesion segmentation"* **arXiv preprint arXiv:2210.16898**: DOI: 10.1007/978-3-031-25066-8_9.

[14] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang. "Swin-unet: Unet-like pure transformer for medical image segmentation". In: *European conference on computer vision*. Springer, 205–218.

[15] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986. DOI: 10.48550/arXiv.2201.03545.

[16] J. Xiao, S. A. Suab, X. Chen, C. K. Singh, D. Singh, A. K. Aggarwal, A. Korom, W. Widyatmanti, T. H. Mollah, and H. V. T. Minh, (2023) *"Enhancing assessment of corn growth performance using unmanned aerial vehicles (UAVs) and deep learning"* **Measurement *214***: 112764. DOI: 10.1016/j.measurement.2023.112764.

[17] R. Thukral, A. Arora, A. Kumar, and Gulshan. "Denoising of thermal images using deep neural network". In: *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2021*. Springer, 827–833. DOI: 10.1007/978-981-16-7118-0_70.

[18] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. DOI: 10.48550/arXiv.1512.03385.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, (2017) *"Mobilenets: Efficient convolutional neural networks for mobile vision applications"* **arXiv preprint arXiv:1704.04861**: DOI: 10.48550/arXiv.1704.04861.

[20] M. Tan and Q. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *Proceedings of International conference on machine learning*. PMLR, 6105–6114. DOI: 10.4236/ojmsi.2021.93017.

[21] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. "Designing network design spaces". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10428–10436. DOI: 10.48550/arXiv.2003.13678.

[22] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, and L. Dong. "Swin transformer v2: Scaling up capacity and resolution". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12009–12019. DOI: 10.48550/arXiv.1405.0312.

[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022. DOI: 10.48550/arXiv.2103.14030.

[24] R. Mechrez, I. Talmi, and L. Zelnik-Manor. "The contextual loss for image transformation with non-aligned data". In: *Proceedings of the European conference on computer vision (ECCV)*, 768–783. DOI: 10.48550/arXiv.1803.02077.

[25] 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: *Proceedings of European conference on computer vision*. Springer, 740–755. DOI: 10.48550/arXiv.1405.0312.

[26] S. Chauhan, M. Singh, and A. K. Aggarwal, (2021) *"Data science and data analytics: artificial intelligence and machine learning integrated based approach"* **Data Science and Data Analytics: Opportunities and Challenges *1***: DOI: 10.1201/9781003111290.

[27] A. Aggarwal, (2020) *"Enhancement of GPS position accuracy using machine vision and deep learning techniques"* **Journal of Computer Science *16***(5): 651–659. DOI: 10.3844/jcssp.2020.651.659.

[28] A. Kaur, A. P. S. Chauhan, and A. K. Aggarwal. "Machine learning based comparative analysis of methods for enhancer prediction in genomic data". In: *Proceedings of 2019 2nd International Conference on Intelligent Communication and Computational Techniques*. IEEE, 142–145. DOI: 10.1109/ICCT46177.2019.8969054.

[29] A. Kaur, A. P. S. Chauhan, and A. K. Aggarwal, (2022) *"Dynamic deep genomics sequence encoder for managed file transfer"* **IETE Journal of Research**: 1–13. DOI: 10.1080/03772063.2022.2060869.

[30] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. "A perceptually motivated online benchmark for image matting". In: *Proceedings of 2009 IEEE conference on computer vision and pattern recognition*. IEEE, 1826–1833. DOI: 10.1109/CVPR.2009.5206503.