VideoMatt: A Simple Baseline for Accessible Real-Time Video Matting

Jiachen Li[1], Marianna Ohanyan[3], Vidit Goel[3], Shant Navasardyan[3], Yunchao Wei[2], Humphrey Shi[1,3]
[1]SHI Lab @ University of Oregon & UIUC, [2]BJTU, [3]Picsart AI Research (PAIR)

Abstract

*Recently, real-time video matting has received growing attention from academia and industry as a new research area on the rise. However, most current state-of-the-art solutions are trained and evaluated on private or inaccessible matting datasets, which makes it hard for future researchers to conduct fair comparisons among different models. Moreover, most methods are built upon image matting models with various tricks across frames to boost matting quality. For real-time video matting models, simple and effective temporal modeling methods must be explored better. As a result, we first composite a new video matting benchmark that is purely based on publicly accessible datasets for training and testing. We further empirically investigate various temporal modeling methods and compare their performance in matting accuracy and inference speed. We name our method as **VideoMatt**: a simple and strong real-time video matting baseline model based on a newly-composited accessible benchmark. Extensive experiments show that our VideoMatt variants reach better trade-offs between inference speed and matting quality compared with other state-of-the-art methods for real-time trimap-free video matting. We release the VideoMatt benchmark at* https://drive.google.com/file/d/1QT4KHeGW3YrtBs1_7zovdCwCAofQ_GIj/view?usp=sharing.

1. Introduction

Video matting is the task of estimating the alpha matte for each frame of a given video sequence input. It has received considerable attention from both industry and academia in recent years. Given a video sequence $I = \{I_1, I_2, ..., I_T\}$, each frame I_i can be viewed as a composition of unknown foreground image F_i and background image B_i [38] with coefficient map alpha matte $\alpha_i \in [0, 1]$

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \tag{1}$$

Since the goal of video matting is to predict alpha mattes $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$, it becomes an under-constrained problem with only 3 equations from Eq. 1 and 7 unknowns

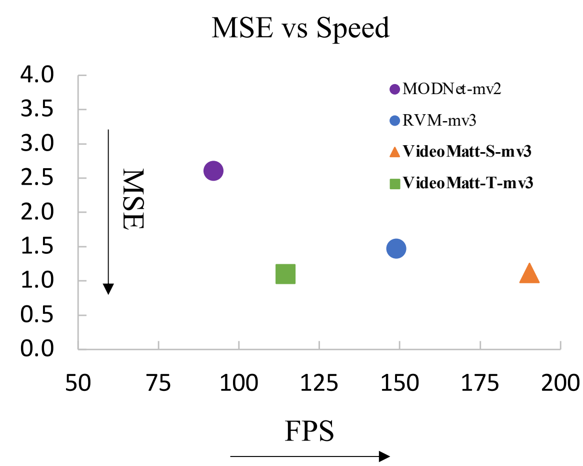

Figure 1. Comparisons between state-of-the-art *trimap-free* real-time video matting methods and our VideoMatt variants in terms of *MSE* (lower is better) and *FPS* (higher is better). MSE (Mean Squared Error) is used for evaluating the accuracy of alpha matte predictions. All models are tested under LR (low resolution) inputs in a single RTX 2080 GPU and more details are presented in Table 2. VideoMatt variants show better trade-offs between inference speed and matting quality.

from α_i, F_i, B_i for each pixel. Previous solutions expect users to provide a *trimap*, which is a segmentation map of foreground, background and unknown regions of images, to add constraints and estimate alpha matte through iterative nonlinear optimization [26]. Deep learning based approaches [29, 40–42, 44] for image matting take inputs of images with corresponding trimaps and estimate alpha mattes in an end-to-end manner through deep convolutional neural networks, which outperform tranditional solutions by a large margin. When it comes to video matting solutions, recent deep learning based methods like DVM [37] and TAM [43] add different attention-based modules for temporal aggregation. Since adding trimap annotations to video sequences is expensive and inconvenient, recent works mainly explore *trimap-free* solutions. BGM [36] and BGMv2 [30] adds background images at first frame and provides efficient solutions under high-resolution inputs. MODNet [24] uses self-supervised strategy for postprocessing to get more smoothing outputs. RVM [31] further involves segmentation data for training and make the network robust to real-world scenes.

However, for current trimap-free video matting methods, we notice that a fair and accessible video matting benchmark is missing. MODNet [24] proposes PPM-100 to evaluate different matting methods while the test set of this benchmark is image-based and the training set is not released. BGMv2 [30] proposes video matting dataset VideoMatte240K which only includes foregrounds and alpha mattes. RVM [31] further adds video backgrounds from DVM [37] to composite training set, but image backgrounds are crawled from the internet and not publicly available. Meanwhile, we also observe that simple and effective temporal modeling techniques are not well-explored for trimap-free video matting models. DVM [37] and TAM [43] prove that trimap-based video matting models can benefit from temporal modeling based on attention mechanism. However, for trimap-free models, selecting a proper temporal modeling method to boost matting quality while maintaining real-time inference speed is still an open question to the community.

As a result, we first composite a new video matting benchmark with VideoMatte240K [30] as video foregrounds, DVM [37] as video backgrounds and BG20K [28] as image backgrounds, respectively. These datasets are all publicly accessible and can be used to evaluate different video matting methods under a fair comparison. Then, we propose **VideoMatt**: a simple and strong real-time trimap-free video matting baseline which is built upon this newly-composited benchmark. It is based on U-Net [35] design that has an encoder for feature extraction and a decoder to finish alpha matte prediction. Furthermore, we empirically investigate different temporal modeling methods based on VideoMatt in terms of alpha matte quality and build a series of VideoMatt-T models. Experiments show that our VideoMatt baseline outperforms other trimap-free solutions by the trade-off between the accuracy of alpha matte prediction and inference speed as shown in Figure 1.

Our contributions can be summarized as follows:

- We composite a new video matting benchmark that is purely based on publicly accessible datasets for comparing different models.
- We propose a simple and strong baseline VideoMatt on this newly-composited video matting benchmark and empirically evaluate different temporal modeling methods based on VideoMatt.
- Our VideoMatt variants reach better trade-offs between inference speed and matting quality compared with other state-of-the-art solutions.

2. Related Works

2.1. Image Matting

Previous image matting solutions started from color sampling-based methods, which sample pixels nearby foregrounds and backgrounds to group alpha maps in the transition region [9, 11, 13, 15, 23]. Then, affinity-based methods, which estimate the alpha matte from unknown to known ones [1–3, 6, 14] that are more robust when dealing with complex images. Traditional methods mainly focus on low-level features to estimate alpha maps of images. In the deep-learning era, DIM [41] proposed an encoder-decoder network to estimate alpha matting in an end-to-end manner with trimaps. DeepMattePropNet [40] further involves encoder-decoder design within propagation-based matting. LDN [44] proposes a mobile design for fast deep matting and GCA [29] extends the idea to natural image matting with guided attention. Some other works also explore trimap-free image matting without the need for extra inputs of trimaps. SHM [5] uses segmentation networks to solve alpha matting with single image input. MRN and QUN [32] were proposed to augment human matting quality with coarse annotations. HDMatt [42] employs cross-patch context module for high-resolution image matting. HAttMatting [33] uses spatial and channel-wise attention to integrate appearance cues.

2.2. Video Matting

Video matting is a relatively new track compared with image matting since temporal information can be introduced to augment matting quality. Similar to image matting, trimap-based video matting methods DVM [37] involve spatio-temporal feature aggregation module for temporal feature fusion and alignment. TAM [43] also uses attention on adjunct frames for feature aggregation. Trimap-free method background matting [36], which takes an input of the background image as the first frame and it provides an important cue for predicting the alpha matte. Following work BGMv2 [30] provides solutions to high-resolution real-time video matting. MODNet [24] only takes images as inputs and uses a self-supervised strategy for modeling temporal consistency. RVM [31] further trains the video matting model on segmentation data and make the matting quality robust on real-world data. Vision transformer models [7, 8, 21, 22] adopt full transformers into image segmentation tasks, and VMFormer [27] also leverage a vision transformer as the solution to trimap-free video matting and achieves competitive performance.

3. Accessible Benchmark Composition

In this section, we mainly introduce how we construct the new accessible video matting benchmark by publicly available foreground video matting dataset Video-

Figure 2. Selected video clips from the composited test set. Please zoom in for details.

Matte240K [30], image background dataset BG20K [28] and video background dataset DVM [37]. We first give a review of these three datasets about the data statistics and then introduce how we composite the training and testing data based on them separately.

3.1. Dataset Overview

VideoMatte240K [30]: it collects 484 high-resolution green screen videos including annotations of alpha matte and foregrounds. 384 of them are in 4k resolution and the rest 100 are in HD resolution. The authors split the videos by 479/5 as training and test sets for evaluation. BG20K [28]: it contains 20000 high-resolution clean background images with no salient objects included. The average solution of BG20K is 1180×1539. The background scenes include city, mountain, urban, and other outdoor environments. The authors split it by 15000/5000 as training and test sets. DVM [37]: it collects over 6500 free video clips of natural scenarios, city views, and indoor environments. Most of them are HD videos and a few are 4k videos. The authors treat 6400 video clips as a training set and 248 video clips as a test set. Compared to other datasets, it mainly provides rapidly-moving objects for challenging video matting evaluation. All three datasets are publicly accessible for compositing a new video matting benchmark with the training and test set.

3.2. Composited Benchmark

Training Set To composite the training set, we first divide VideoMatte240K following the 479/5 split and further move 4 video clips from the training set to the validation set. We further separate the BG20K into a 15000/500/4500 for training, validation, and test set. Then, DVM is added and split into 3080/37/162 video sequences following RVM [31]. VideoMatte240K, BG20K and DVM provide video foreground sequences with corresponding alpha matte sequences, image background sequences and video background sequences, respectively. During training, the model randomly picks up a video foreground sequence $F = \{F_1, F_2, ..., F_T\}$ with length T and alpha matte $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$ from the training set of VideoMatte240K, then an image or a video background sequence $B = \{B_1, B_2, ..., B_T\}$ is randomly chosen for composition. Then, a composited $I = \{I_1, I_2, ..., I_T\}$ is used for training with ground truth $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$. For each composited video clip I, we also provide corresponding foreground $F = \{F_1, F_2, ..., F_T\}$ and background $B = \{B_1, B_2, ..., B_T\}$ to meet the needs of different models.

Test Set During testing, we composited a test set that has 200 video clips in which each clip contains 100 frames. 50 of them are composited based on test sets of VideoMatte240K and DVM, which mainly contain video foregrounds and video backgrounds that are sampled from real-world videos. The rest 150 of them are composited based on test sets of VideoMatte240K and BG20K, which provide video foregrounds and image backgrounds from the real-world. We hope the diverse backgrounds bring more challenges for robust video matting to the community and we select some clips from the test set for visualization in Figure 2. The benchmark is purely synthetic since the annotations for alpha matte of per-frame video are expensive, and previous video matting benchmarks are also purely syn-

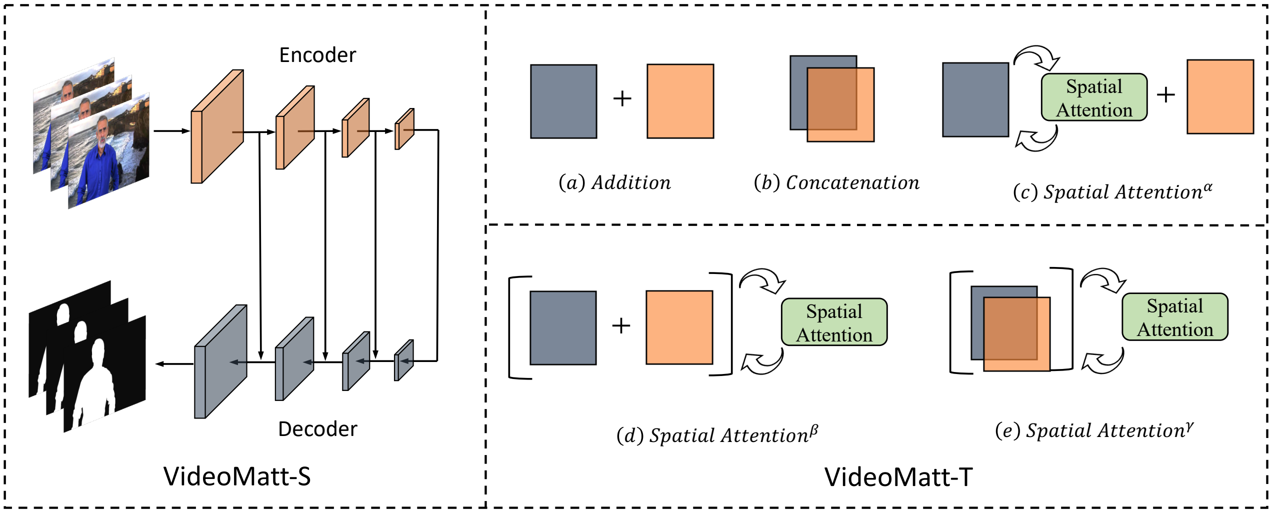

Figure 3. **The overall architecture of VideoMatt-S and VideoMatt-T**. VideoMatt-S is a simple encoder-decoder based single-frame video matting model without temporal modeling technique. For VideoMatt-T, we evaluate different temporal modeling methods. Given feature maps F_n and F_{n+1} from two consecutive frames I_n and I_{n+1}, the updated F'_{n+1} is formulated by (a) Addition: $F_n + F_{n+1}$ (b) Concatenation: $Conv(Concate(F_n, F_{n+1}))$ (c) Spatial Attention$^\alpha$: $F_{n+1} + SpatialAttn(F_n)$ (d) Spatial Attention$^\beta$: $SpatialAttn(F_n + F_{n+1})$ (e) Spatial Attention$^\gamma$: $SpatialAttn(Conv(Concate(F_n, F_{n+1})))$. Table 1 further shows that Spatial Attention$^\gamma$ brings greatest improvement among all metrics especially on temporal coherence and thus is selected for temporal modeling module of VideoMatt-T.

thetic [31, 36, 43]. We release the link to the VideoMatt benchmark both in the abstract and the supplemental file.

4. Our Method

In this section, we mainly introduce our VideoMatt-S model and compare different temporal modeling techniques in VideoMatt-T. Then, we describe how we train and evaluate all VideoMatt variants.

4.1. VideoMatt-S

The framework of VideoMatt-S is illustrated in Figure 3, which is a single-frame baseline without temporal modeling. Given a video sequence $I = \{I_1, I_2, ..., I_T\}$ as input and T is the number of frames, the encoder generates feature pyramids $f = \{f_1, f_2, f_3, f_4\}$. Here the feature pyramids start from a high-resolution feature map since video matting is a per-pixel prediction task and high-resolution feature maps are favorable for accurate prediction of alpha matte. f_4 is then sent to an atrous spatial pyramid pooling layer [4] and becomes f'_4. For the decoder part, in each up-scaling block we have

$$f'_i = Deconv(f'_{i+1}) + f_i \tag{2}$$

which outputs $f' = \{f'_3, f'_2, f'_1\}$ accordingly. Finally, the predictions of the corresponding alpha matte sequences $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$ and foreground images $F = \{F_1, F_2, ..., F_T\}$ for the video sequence are based on a combination of up-scaling and convolution layers on top of F'_1.

$$[\alpha, F] = Conv(Deconv(f'_1)) \tag{3}$$

Then prediction of $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$ are used for evaluation of matting accuracy. We use VideoMatt-S as a baseline model to investigate the efficacy of various temporal modeling techniques in VideoMatt-T. Unlike BGMv2 [36], which requires background image input, and RVM [31], which incorporates ConvGRU temporal modules, VideoMatt-S utilizes the U-Net architecture and does not rely on such mechanisms. By leveraging this baseline model, we aim to identify effective ways of incorporating temporal information into video matting models and improve the accuracy of alpha matte predictions.

4.2. VideoMatt-T

Temporal modeling is about utilizing temporal information to augment matting quality across frames by reducing flickers and revising wrong predictions. To evaluate the effectiveness of different temporal methods for video matting, we try five different implementations based on our strong baseline VideoMatt-S and they are illustrated in Figure 3. For two consecutive frames I_n and I_{n+1}, given two corresponding feature maps F_n and F_{n+1} at the same level in the decoder, the simplest way to model their relation is to add them for temporal aggregation,

$$F'_{n+1} = F_n + F_{n+1} \tag{4}$$

F'_{n+1} denotes for updated feature map of frame I_{n+1}. A similar implementation is to concatenate feature maps with

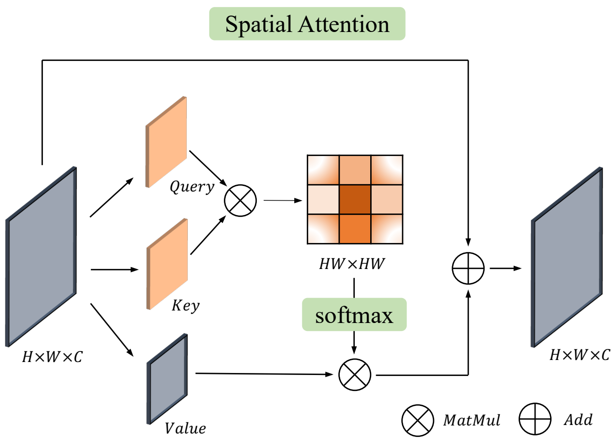

Figure 4. Illustration of Spatial Attention used in VideoMatt-T.Given an input feature map F, it generates query, key, and value matrix by convolution layers, then spatial attention map is calculated by batch matrix-matrix product of query and key with a following softmax function for normalization. Finally, it is projected on the value matrix and the projected value matrix is added to the original F.

a following convolution layer for channel reduction,

$$F'_{n+1} = Conv(Concate(F_n, F_{n+1})) \tag{5}$$

Compared with VideoMatt-S baseline, the addition operation makes little improvements on accuracy and temporal connectivity in terms of MAD, Grad, Conn and dtSSD metrics. We further consider more complicated temporal modeling techniques based on the attention mechanism introduced as a non-local block [39] for video classification. To save memory and computation, we apply a simplified version of spatial attention [19] for semantic segmentation and apply it between two consecutive frames iteratively. We design three versions α, β, γ of applying spatial attention to F_n. The implementation of Spatial Attention$^\alpha$ is

$$F'_{n+1} = F_{n+1} + SpatialAttn(F_n) \tag{6}$$

which implies that the temporal information from the attention map of the previous frame is added to the current frame. The Spatial Attention operation is illustrated in Figure 4 in details, which generates a self-attention based feature map. To make the temporal information learnable on both consecutive frames, we further apply Spatial Attention$^\beta$ and Spatial Attention$^\gamma$,

$$F'_{n+1} = SpatialAttn(F_n + F_{n+1}) \tag{7}$$

$$F'_{n+1} = SpatialAttn(Conv(Concate(F_n, F_{n+1}))) \tag{8}$$

They apply spatial attention to outputs of addition and concatenation of two consecutive frames. It shows that spatial attention$^\gamma$ achieve the greatest improvements compared with the other two implementations. As a result, we use spatial attention$^\gamma$ to build VideoMatt-T model. We only apply it to half channels of feature pyramids F'_4, F'_3 and F'_2 to save computation, which reaches a better trade-off between accuracy and inference speed.

4.3. Training and Testing

Training Stage During training, we follow the short-to-long principles from RVM [31] and break the whole training pipeline into two stages. In the first stage, we train the network based on low-resolution and short video sequences for 20 epochs. When the training is well converged, we extend the video sequence length and train the network for another 5 epochs in the second stage for the final comparisons.

Loss Function The loss function we used is adopted from RVM, which is a combination of individual loss on alpha matte and foreground prediction,

$$L = L_\alpha \; + \; L_F \tag{9}$$

L_α is loss for alpha matte, which is

$$L_\alpha = \|\alpha_n - \hat{\alpha}_n\|_1 \; + \; \lambda_\alpha \| \frac{\partial \alpha_n}{\partial t} - \frac{\partial \hat{\alpha}_n}{\partial t} \|_2 + \sum_{i=1}^{5} 2^{i-1} \|L^i(\alpha_n) - L^i(\hat{\alpha}_n)\|_1 \quad n \in [0, T-1] \tag{10}$$

here α_n is the prediction of alpha matte and $\hat{\alpha}_n$ is ground truth. The L_α involves L1 loss, temporal consistency loss and Laplacian loss used in [17,31,37]. λ_α is the loss weight for Laplacian loss and is set to be 5 for balanced training. For foreground loss L_F,

$$L_F = \|F_n - \hat{F}_n\|_1 \; + \; \lambda_F \| \frac{\partial F_n}{\partial t} - \frac{\partial \hat{F}_n}{\partial t} \|_2 \tag{11}$$

similarly, L_F contains L1 loss and temporal consistency loss for foreground prediction. λ_F is also set to be 5 for balanced training procedure.

Testing Stage During inference stage, we resize the inputs into three different resolutions: 512×288 (LR), 1920×1080 (HD) and 3840×2160 (4K) for evaluation. The network takes the whole video clip as input and run inference frame by frame. During inference on each frame, temporal feature maps are saved as intermediate result for input of next frame. More experimental details are introduced in the next section.

5. Experiments

In this section, we first review the datasets and evaluation metrics we used for experiments. Then, we specify the experimental setting in detail and make ablation studies on various temporal modeling methods. We also test the inference speed and model size of different models for comparison. We further compare our VideoMatt variants with other

Temporal Modeling	Metrics				
	MAD↓	MSE↓	Grad↓	Conn↓	dtSSD↓
VideoMatt-S	6.75	1.46	1.20	0.52	1.85
+Addition	6.57	1.49	1.17	0.50	1.75
+Concatenation	7.48	2.80	1.21	0.62	1.78
+Spatial Attention$^\alpha$	6.51	1.45	1.15	0.49	1.73
+Spatial Attention$^\beta$	6.77	1.48	1.13	0.45	1.75
+Spatial Attention$^\gamma$	**6.50**	**1.45**	**1.19**	**0.49**	**1.70**

Table 1. Ablation study on different temporal modeling methods based on VideoMatt-S. The performance of temporal coherence (dtSSD) benefits from the added attention operator. All numbers are evaluated only after the first stage of training as described in Section 4.3.

Model	Backbone	FPS (4K/HD/LR)	Params
BGMv2	MobileNetV2	30.6/121.6/159.0	4.991M
MODNet	MobileNetV2	2.9/11.2/92.1	6.487M
RVM	MobilenetV3	72.9/114.7/149.0	3.749M
RVM	ResNet50	54.2/74.5/81.8	26.890M
VideoMatt-S	MobilenetV3	**99.0/136.8/190.4**	**3.296M**
VideoMatt-S	ResNet50	61.6/83.4/96.0	25.990M
VideoMatt-T	MobilenetV3	65.2/89.4/114.6	3.304M
VideoMatt-T	ResNet50	51.4/66.4/72.5	26.008M

Table 2. Inference speed & Params Comparisons between different models on a single RTX 2080 GPU. **Bold** indicates the highest FPS and the least number of parameters.

trimap-free methods and visualize some video matting results.

5.1. Datasets and Evaluation Metrics

Datasets The datasets we use for compositing training, validation and test sets are from VideoMatte240K, BG20K and DVM. We give detailed descriptions in Section 3 about how we composite the training set and conduct an evaluation of the test set.

Evaluation Metrics The evaluation metrics are mainly focused on the quality of predicted alpha mattes. It involves Mean Absolute Difference (MAD), Mean Squared Error (MSE), Gradient (Grad), Connectivity (Conn) [34] and Sum of Squared Differences (dtSSD) [10] for evaluating quality and temporal consistency of alpha mattes. We scale MAD, MSE, Grad, Conn, dtSSD by 10^3, 10^3, 10^{-3}, 10^{-3} and 10^2 respectively.

5.2. Experimental Setting

Training Setting For training the network, we use 4 RTX A6000 GPU with batch size at 1 video clip per GPU. The optimizer is Adam with different learning rates at different modules of the network. The initial learning rate for the encoder is 0.0001 and 0.0002 for the decoder, which are further scaled down to 0.00005 and 0.0001 at stage 2. We use random resize, center crop, horizontal flip, color jittering, image blurring and sharpening for data augmentation. For backbone selection, we use ImageNet [25] pre-trained Mobilenetv3-Large [18] as encoders to train VideoMatt variants. Most other training hyper-parameters and settings are adopted from RVM [31] for fair comparisons.

Runtime Setting During inference, the test sets are precomposited for stable and fast testing. We compare current trimap-free video matting models and VideoMatt variants under both our synthetic test sets and the test sets used in RVM [31] to test both the matting quality and robustness of our model to different backgrounds. In detail, we test these models on a single RTX 2080 GPU to compare inference speed under inputs of different resolutions and report the number of parameters of these models. The framework is based on Pytorch 1.9.1 and CUDA 11.1. The system is Ubuntu 20.04 with AMD EPYC 7662 as the CPU.

5.3. Ablation Study

Temporal Modeling To evaluate the effectiveness of different temporal modeling methods, we first build the VideoMatt-S baseline which is a single-frame version without any temporal modeling technique. It is trained on inputs with short and low-resolution video sequences for the first stage of training as described in Section 4.3. Then we add different temporal modeling methods as shown in Table 1 and trained all these models under the same settings. It shows that for simple temporal modeling, the addition operator is more effective than the concatenation operator. For more complicated ones, spatial attention based on concatenation is the most effective solution among them, especially on the temporal consistency metric dtSSD which drops from 1.85 to 1.70. As a result, we select Spatial Attention$^\gamma$ for the temporal modeling module and build VideoMatt-T on top of VideoMatt-S accordingly.

Backbone Selection We evaluate VideoMatt variants mainly under MobilnetV3-Large [18] that represent lightweight CNN-based backbone. ResNet50 [16] is a relatively larger and deeper CNN-based backbone.

Inference Speed To evaluate the inference speed of different real-time video matting models fairly, we used pretrained weights of three most recent models BGMv2 [30], MODNet [24] and RVM [31] for comparisons under inputs of three resolutions in Table 2. All models are evaluated on a single RTX 2080 GPU with inputs under 3840×2160 (4K), 1920×1080 (HD) and

Model	Backbone	Temporal Modeling	Metrics MAD ↓	MSE ↓	Grad ↓	Conn ↓	dtSSD ↓
BGMv2	MobilenetV2	✗	33.90	28.39	2.38	4.52	2.72
MODNet	MobilenetV2	✗	7.36	2.60	1.58	0.60	3.75
RVM	MobilenetV3	✓	6.36	1.47	1.03	0.45	1.68
VideoMatt-S	MobilenetV3	✗	6.02	1.12	0.94	0.40	1.68
VideoMatt-T	MobilenetV3	✓	**5.90**	**1.10**	**0.94**	**0.39**	**1.57**

Table 3. Comparison on the composited testing set. VideoMatt-S: Single-frame baseline version without temporal modeling; VideoMatt-T: VideoMatt-S with temporal modeling based on spatial attention$^{\gamma}$. All numbers are evaluated after the two stages of training as described in Section 4.3.

Model	Backbone	Training Data Segmentation	Matting	Metrics MAD ↓	MSE ↓	Grad ↓	Conn ↓	dtSSD ↓
BGMv2 [31]	MobilenetV2	✓	✓	25.19	19.63	2.28	3.26	2.74
DeepLabV3 [4]	ResNet101	✓	✓	14.47	9.67	8.55	1.69	5.18
MODNet [31]	MobileNetV2	✓	✓	9.28	4.17	1.76	0.79	2.10
FBA [12]	ResNet50	✓	✓	8.36	3.37	2.09	0.75	2.09
RVM [31]	MobilenetV3	✓	✓	6.08	1.47	**0.88**	**0.41**	**1.36**
VideoMatt-S	MobilenetV3	✗	✓	6.52	1.54	1.15	0.48	1.76
VideoMatt-T	MobilenetV3	✗	✓	**6.06**	**1.27**	1.09	0.42	1.60

Table 4. Robustness evaluation on the test set in RVM. We directly evaluate VideoMatt-S/T on RVM's test set **without** re-training the models on more segmentation data.

512×288 (LR). We report Frame-Per-Second (FPS) and number of parameters of these models. Considering this table and the following tables on matting quality, it shows that our VideoMatt variants reaches better trade-offs between accuracy and inference speed compared to other methods, since VideoMatt-S is fastest with comparable matting accuracy.

Mobile Device Inference To estimate the performance and inference speed of VideoMatt-S/T on the mobile device, we refer to the AI-Benchmark [20], where the MobilNetV3 runs 66ms on the Apple A15 Bionic chip. As a result, the estimation of VideoMatt-S/T on the Apple A15 Bionic would be around 94ms/132ms, considering the inference of MobileNetV3 takes 70%/50% of the total inference time in Table 2.

5.4. Comparison to State-of-the-art Methods

Composited test set The composited test set we used for comparison is introduced in Section 3, which contains 200 video clips for evaluation. We compare our VideoMatt variants with most recent state-of-the-art real-time video matting models BGMv2 [30], MODNet [24] and RVM [31]. To make fair comparisons, we reproduced them with their original design based on their open-sourced codes and trained them on our composited training data. All experimental results are listed in Table 3. For the VideoMatt-S and VideoMatt-T, we further trained them with the second stage as described in Section 4.3, and the performance improves compared to the numbers in Table 1. Our evaluation mainly focuses on matting qualify of alpha matte predictions and we evaluate all models under inputs of 512×288. VideoMatt-T outperforms VideoMatt-S in all metrics especially on the temporal consistency (dtSSD) from 1.68 to 1.57.

Other benchmarks To test the robustness of our VideoMatt variants, we further test them on the test set used in RVM [31] as shown in Table 4 without re-training the VideoMatt models. We directly run inference and evaluation of VideoMatt models on this test set with the pretrained weights on our training set, while all other models BGMv2, MODNet, FBA, DeeplabV3 and RVM are re-trained as reported in RVM [31]. It shows that VideoMatt reach comparable performances to other pre-trained state-of-the-art solutions.

Figure 5. Visualization of alpha matte predictions from MODNet, RVM, VideoMatt-S and VideoMatt-T under challenging frames from the composited test set. Please zoom in for details.

5.5. Visualization

In this section, we select some challenging consecutive video frames from the composited test set for comparisons and visualize the video matting results in Figure 5. It shows that our VideoMatt variants can distinguish ambiguous backgrounds from foregrounds and temporal modeling in VideoMatt-T further removes some inaccurate predictions on foregrounds compared with MODNet and RVM.

6. Conclusion

In this paper, we discuss current bottlenecks for real-time video matting solutions. Firstly, it lacks a fair and accessible video matting benchmark, for making comparisons between different algorithms. Secondly, temporal modeling is not well-explored for trimap-free video matting models. Motivated by these observations, we first composite a new video matting benchmark that is based on all public accessible datasets for comparing different models. Then, we investigate various temporal modeling methods and compare their performance on matting accuracy and temporal consistency. Our benchmark and method are named as VideoMatt: a simple and strong real-time trimap-free video matting model that is trained and evaluated on our new video matting benchmark. Extensive experiments show that our VideoMatt variants reach better trade-offs between accuracy of alpha matte predictions and inference speed compared with other state-of-the-art solutions. For the future work, we mainly focus on improving the robustness of VideoMatt to real-world data and scenarios.

References

[1] Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 2

[2] 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

[3] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 4, 7

[5] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 618–626, 2018. 2

[6] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 2

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 2

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[9] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001. 2

[10] Mikhail Erofeev, Yury Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, pages 99–1, 2015. 6

[11] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. A cluster sampling method for image matting via sparse coding. In *European Conference on Computer Vision*, pages 204–219. Springer, 2016. 2

[12] Marco Forte and François Pitié. f, b, alpha matting. *arXiv preprint arXiv:2003.07711*, 2020. 7

[13] 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. 2

[14] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005. 2

[15] 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. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4130–4139, 2019. 5

[18] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 6

[19] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 5

[20] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 7

[21] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. *arXiv preprint arXiv:2211.06220*, 2022. 2

[22] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021. 2

[23] Levent Karacan, Aykut Erdem, and Erkut Erdem. Image matting with kl-divergence based sparse sampling. In *Proceedings of the IEEE international conference on computer vision*, pages 424–432, 2015. 2

[24] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson WH Lau. Is a green screen really necessary for real-time portrait matting? *arXiv preprint arXiv:2011.11961*, 2020. 1, 2, 6, 7

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 6

[26] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007. 1

[27] Jiachen Li, Vidit Goel, Marianna Ohanyan, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. Vmformer: End-to-end video matting with transformer. *arXiv preprint arXiv:2208.12801*, 2022. 2

[28] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, pages 1–21, 2022. 2, 3

[29] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *Proceedings of the AAAI Conference*

on Artificial Intelligence, volume 34, pages 11450–11457, 2020. 1, 2

[30] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8762–8771, 2021. 1, 2, 3, 6, 7

[31] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. *arXiv preprint arXiv:2108.11515*, 2021. 2, 3, 4, 5, 6, 7

[32] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian-sheng Hua. Boosting semantic human matting with coarse annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8572, 2020. 2

[33] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13676–13685, 2020. 2

[34] 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. 6

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[36] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2291–2300, 2020. 1, 2, 4

[37] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. Deep video matting via spatio-temporal alignment and aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6975–6984, 2021. 1, 2, 3, 5

[38] Jue Wang and Michael F Cohen. Image and video matting: a survey. 2008. 1

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 5

[40] Yu Wang, Yi Niu, Peiyong Duan, Jianwei Lin, and Yuanjie Zheng. Deep propagation based image matting. In *IJCAI*, volume 3, pages 999–1006, 2018. 1, 2

[41] 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. 1, 2

[42] Haichao Yu, Ning Xu, Zilong Huang, Yuqian Zhou, and Humphrey Shi. High-resolution deep image matting. *arXiv preprint arXiv:2009.06613*, 2020. 1, 2

[43] Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-sheng Hua, Hujun Bao, Qixing Huang, and Weiwei Xu. Attention-guided temporal coherent video object matting. *arXiv preprint arXiv:2105.11427*, 2021. 1, 2, 4

[44] Bingke Zhu, Yingying Chen, Jinqiao Wang, Si Liu, Bo Zhang, and Ming Tang. Fast deep matting for portrait animation on mobile phone. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 297–305, 2017. 1, 2