Video Instance Matting

Jiachen Li[1], Roberto Henschel[2], Vidit Goel[2], Marianna Ohanyan[2], Shant Navasardyan[2], Humphrey Shi[1,2]
[1]SHI Labs @ Georgia Tech & Oregon & UIUC, [2]Picsart AI Research (PAIR)

Abstract

*Conventional video matting outputs one alpha matte for all instances appearing in a video frame so that individual instances are not distinguished. While video instance segmentation provides time-consistent instance masks, results are unsatisfactory for matting applications, especially due to applied binarization. To remedy this deficiency, we propose **Video Instance Matting (VIM)**, that is, estimating alpha mattes of each instance at each frame of a video sequence. To tackle this challenging problem, we present **MSG-VIM**, a Mask Sequence Guided Video Instance Matting neural network, as a novel baseline model for VIM. MSG-VIM leverages a mixture of mask augmentations to make predictions robust to inaccurate and inconsistent mask guidance. It incorporates temporal mask and temporal feature guidance to improve the temporal consistency of alpha matte predictions. Furthermore, we build a new benchmark for VIM, called **VIM50**, which comprises 50 video clips with multiple human instances as foreground objects. To evaluate performances on the VIM task, we introduce a suitable metric called Video Instance-aware Matting Quality (VIMQ). Our proposed model MSG-VIM sets a strong baseline on the VIM50 benchmark and outperforms existing methods by a large margin. The project is open-sourced at https://github.com/SHI-Labs/VIM.*

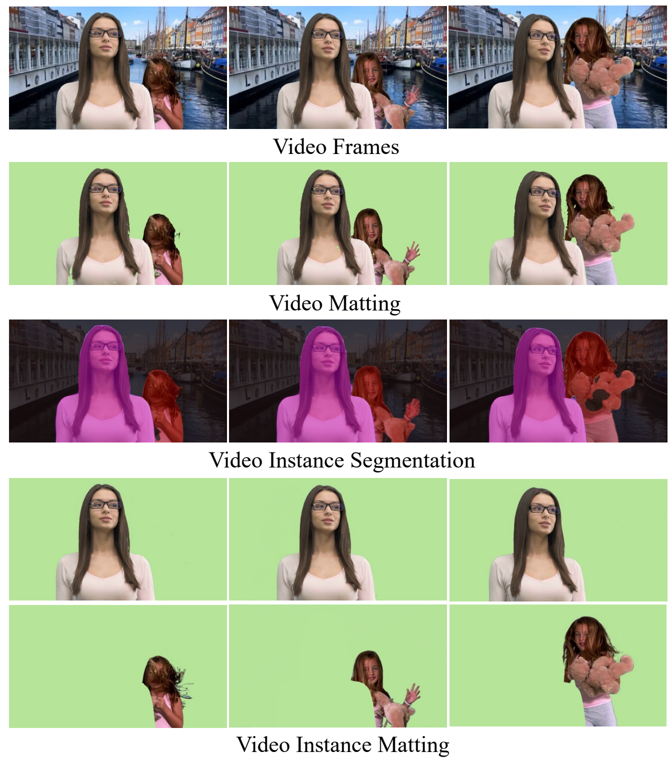

Figure 1. **Video Instance Matting compared with related tasks.** Depicted are three frames of the VIM50 benchmark. The results of Video Matting, Video Instance Segmentation, and Video Instance Matting are obtained from RVM [32], SeqFormer [44], and MSG-VIM, respectively. Video Instance Matting is the task of recognizing and tracking each foreground instance, then estimating the alpha matte of the corresponding instance at each frame of the video sequence.

1. Introduction

Recently, video matting has drawn much attention from industry and academia as it is widely used in video editing and video conferencing [32]. Current deep learning based video matting methods [22, 26, 31, 32, 35] output a single alpha matte for all instances appearing in the foreground for each frame. However, different applications require sequences of alpha mattes *and* the separation into the instances. One approach towards instance-aware video matting is to employ video instance segmentation [2,4,5,24,40], which segments and tracks each object instance appearing in a video sequence. Unfortunately, the generated masks are binary and coarse at the outline of an instance, making them inadequate for high-quality instance-aware video editing. Several image matting works [38, 49] have thus focused on converting segmentation masks into alpha mattes by adopting mask guidance from instance segmentation. However, such approaches do not track instances and are thus not applicable to video editing. In summary, there is no off-the-shelf solution for high-quality instance-aware video matting.

Motivated by these observations, we extend video matting to a multi-instance scenario in Section 3, called **Video Instance Matting (VIM)**, a new task aiming at estimating alpha mattes of each instance at each frame given a video sequence, as shown in Figure 1. To tackle this task, we propose in Section 4 a new baseline method, called *Mask Sequence Guided Video Instance Matting* (**MSG-VIM**), which takes mask sequences from a video instance segmentation (VIS) method as guidance and transforms them into time-consistent, high-quality alpha matte sequences. In more detail, we employ a VIS method to obtain a sequence of coarse binary masks for each instance. Then, masks are concatenated with corresponding video frames and passed to an encoder-decoder-based model, which returns a sequence of high-quality alpha mattes for each instance. This mask-guided architecture allows our method to benefit from future advances in VIS *without* re-training the MSG-VIM model. Furthermore, we propose a mixture of mask augmentations during training, which make MSG-VIM less susceptible to error propagation from mask sequence guidance caused by the employed VIS method. We then apply temporal mask guidance (TMG) and temporal feature guidance (TFG) to the model, incorporating temporal information for the alpha matte creation. Accordingly, MSG-VIM can compensate for individual incorrect input masks, leading to improved matting quality.

In order to evaluate the performance of MSG-VIM and other methods on the VIM task, we establish in Section 3 **VIM50**, a benchmark for VIM, which comprises 50 video clips with multiple human instances as foregrounds for evaluation. We further propose the *Video Instance-aware Matting Quality* (VIMQ) metric to evaluate video instance matting performance. It simultaneously considers recognition, tracking, and matting quality. We compare MSG-VIM to video matting, video instance segmentation, and image instance matting methods on the proposed VIM50 benchmark. The experiments show that the VIM task is not sufficiently well-handled by existing approaches, justifying the focus on this challenging task. Moreover, the experiments demonstrate that carefully incorporating mask sequence guidance and temporal modeling is crucial to obtain accurate results. As shown in Figure 1, MSG-VIM shows not only better instance-level matting quality, but also conventional video matting quality if we merge all instance mattes.

To summarize, our contributions are as follows:

- We propose Video Instance Matting, a new task aiming at predicting alpha mattes of each foreground instance at each frame given a video sequence as input.
- We establish a benchmark for the proposed VIM, called VIM50, which comprises 50 videos with multiple human instances as foregrounds. Furthermore, we propose VIMQ as an evaluation metric for VIM.
- We propose MSG-VIM, a Mask Sequence Guided Video Instance Matting network, as a simple and strong baseline model for the VIM50 benchmark.

2. Related Works

2.1. Image Matting

Image matting [41], *i.e.*, the overlay of a foreground image with a background image, is an important technique in photography and filmmaking and a classical problem in computer vision. To tackle the task, several methods have focused on detecting the transition areas of the alpha mattes using low-level features [1, 3, 9, 11, 15]. Recently, deep learning based methods have been proposed that tackle image matting end-to-end using guidance from a manually-created trimap [13, 30, 34, 42, 45, 48, 50, 52]. Applicability has been simplified by exploring trimap-free approaches [8], *e.g.*, using coarse segmentation masks [6, 27, 33, 46, 49] as guidance. To eliminate the drawback of image matting methods, which output exactly one alpha matte for an image, HIM [38] proposes multi-instance matting. To this end, instance-level masks are generated from MaskRCNN [16] and further refined by incorporating corresponding image data, resulting in instance-level alpha mattes. In contrast to VIM, HIM works on a frame-level so that the sequence of alpha mattes per instance is not provided. We show in our experiments that simply connecting the frame-wise results of HIM is not sufficient to obtain time-consistent high-quality results.

2.2. Video Matting

Compared to image matting, the task of video matting is to estimate sequences of alpha mattes in a video sequence. By leveraging temporal context the quality of predictions improves. Trimap-based methods add spatial-temporal feature aggregation [39, 51] to improve the accuracy and consistency of alpha matte predictions at each frame. Most trimap-free solutions use a trimap [36] only for the first frame, or background images [31, 35]. MODNet [22], RVM [32] and VideoMatt [28] directly predict mattes from a video. VMFormer [26] adopts the transformer to solve the video matting task. Yet, these methods output only one alpha matte per frame, which covers all foreground instances. When used in a multi-instance scenario, these video matting methods are incapable of distinguishing alpha mattes for different instances. In contrast, VIM methods output instance-aware mattes, which is crucial for various video editing applications such as instance-selective human removal in videos.

2.3. Video Instance Segmentation

The goal of video instance segmentation (VIS) is to simultaneously perform detection, tracking and segmentation of all instances appearing in a video sequence. The baseline approach MaskTrackRCNN [47] adds a tracking head to Mask RCNN [16] to achieve tracking ability. Subsequent works [2, 4, 5, 14, 24, 40] have progressively improved the performance with better representation learning and unified

Task	Alpha Matte	Instance
Video Instance Segmentation	✗	✓
Video Matting	✓	✗
Video Instance Matting	✓	✓

Table 1. High-level comparison between video instance matting and related tasks.

architectures such as transformers [10, 20, 21, 43, 44]. Yet, resulting masks are not suitable for matting tasks as (i) they are too coarse at the outlines of instances and (ii) they are binary.

3. Video Instance Matting

3.1. Problem Definition

We consider a video sequence $\mathbf{I} \in \mathbb{R}^{T\times H\times W\times 3}$ with T frames, where each frame $I_t, t \in \{1, \ldots, T\}$ has spatial dimension $H \times W$. The conventional alpha matting [41] is defined for a single image I_t. The task is to find a composition into a foreground image F_t and background image B_t together with an alpha matte $\alpha_t \in [0, 1]^{H\times W\times 1}$, *i.e.*,

$$I_t = \alpha_t \circ F_t + (\mathbf{1} - \alpha_t) \circ B_t. \quad (1)$$

Here, $\circ$ denotes the Hadamard product and $\mathbf{1}$ is the all-one matrix of appropriate dimension. Our proposed video instance matting task extends image-based matting to sequences of mattings for multiple instances. Hence, assuming that N instances appear in $\mathbf{I}$, the task is to find alpha mattes $\alpha^i_t \in [0, 1]^{H\times W\times 1}$ for all $i \in \{1, \ldots, N\}$, for all $t \in \{1, \ldots, T\}$ such that

$$I_t = \sum_{i=1}^{N} \alpha^i_t \circ F^i_t + (\mathbf{1} - \sum_{i=1}^{N} \alpha^i_t) \circ B_t, \quad (2)$$

which is a natural extension of (1) by assuming that

$$F_t = \sum_{i=1}^{N} F^i_t,\ \alpha_t = \sum_{i=1}^{N} \alpha^i_t,\ \alpha^i_t \circ F^j_t = 0\,\forall i \neq j. \quad (3)$$

Thus, the alpha mattes and foreground images of an image I_t are dissected, according to the appearing instances. A video instance matting method thus has to recognize and track each instance, and to estimate the alpha matte of the corresponding instance at each frame. Compared to (i) video matting and (ii) video instance segmentation, it requires (i) instance-level alpha mattes and (ii) accurate mask predictions without binarization. We show a high-level comparisons between video instance matting and other tasks in Table 1.

3.2. VIM50 Benchmark

There is no benchmark that provides real-world video clips with instance-level alpha matte annotations for each frame. To be able to validate VIM approaches, we thus create a new benchmark for evaluation. To this end, we leverage instance-level foregrounds and backgrounds contained in existing single-instance matting datasets to form our multi-instance video matting benchmark VIM50. Specifically, we use VideoMatte240k [31], which provides high-resolution foreground human instances with alpha matte annotations, and the background video dataset DVM [39] to composite the VIM50 benchmark. We randomly select 50 consecutive background frames and select N times a clip of 50 consecutive frames from the foreground sequences, each showing a single human. We save corresponding backgrounds, foregrounds, and alpha mattes. We composite 50 testing clips of resolution 1920×1080, each consisting of 50 frames. VIM50 comprises 35,10 and 5 video clips with two, three and four human instances, respectively, to cover different levels of crowdedness. 35 video clips depict partially occluded persons, thus posing challenges inputs regarding tracking and matting quality. Exemplary frames of VIM50 are presented in Figure 1 and in the Appendix A.

3.3. Evaluation Metric

We propose a new metric, which we term *Video Instance-aware Matting Quality* (VIMQ), to evaluate the proposed video instance matting task. It combines the instance-level matting quality (MQ) [38], tracking quality [40] (TQ), and recognition quality [25] (RQ) via

$$\mathrm{VIMQ} = \mathrm{RQ} \cdot \mathrm{TQ} \cdot \mathrm{MQ}. \quad (4)$$

The metrics require a minimum-cost maximal matching between predicted and ground truth instance-sequences of alpha mattes. To this end, we apply the Hungarian algorithm [10] on sequence-averaged L1 distances. The true positive set TP comprises the matched sequences of alpha mattes $(\alpha^i_1, \ldots, \alpha^i_T)$, whose average intersection over union between the binarized masks $[\alpha^i] := ([\alpha^i_1], \ldots, [\alpha^i_T])$ and corresponding binarized ground truth masks $[\bar{\alpha}^i] := ([\bar{\alpha}^i_1], \ldots, [\bar{\alpha}^i_T])$ is above ρ. Binarization is done via $\alpha > 0$. We denote by N_{TP}, N_{FP} and N_{FN} the number of corresponding true positives, false positives and false negatives, respectively. Finally, we compute for each true positive instance its frame-averaged intersection over union to the ground truth masks and obtain RQ, an IoU-weighted F1 score of the alpha mattes:

$$\mathrm{RQ} = \frac{\sum_{i\in TP} \mathrm{IoU}([\alpha^{\mathrm{i}}], [\bar{\alpha}^{\mathrm{i}}])}{N_{TP}} \cdot \frac{N_{TP}}{N_{TP} + \frac{1}{2}N_{FP} + \frac{1}{2}N_{FN}}. \quad (5)$$

To measure the Tracking Quality (TQ), we compute a frame-wise minimum-cost maximal matching between TP and ground truth. A deviating assignment compared to the sequence-wise matching is counted as an ID switch error, since it fails to track the corresponding instance at the respective frame. The tracking quality is thus defined as

$$\mathrm{TQ} = 1 - \frac{\sum_{i\in TP} \sum_{t=1}^{T} \mathrm{IDS}(i, t)}{N_{TP} \cdot T}, \quad (6)$$

where $\mathrm{IDS}(i, t) = 1$ in the case of an ID switch error for i at frame t, and 0 otherwise. In order to evaluate the matting quality, we compare the ground truth alpha matte $\bar{\alpha}_t^i$ with prediction α_t^i using a similarity metric S defined by

$$S(\bar{\alpha}_t^i, \alpha_t^i)) = 1 - \min(1, \omega\xi(\bar{\alpha}_t^i, \alpha_t^i)) \in [0, 1]. \quad (7)$$

ω is a manually set weight and ξ computes the mean distance between true positive predicted and ground-truth mattes, inspired by IMQ [38]. We utilize Mean Absolute Difference (MAD), Mean Squared Error (MSE), and direct temporal gradients on Sum of Squared Differences (dtSSD) [12] as metrics. MSE and MAD are used to evaluate the frame-level accuracy, while dtSSD shows temporal consistency of the estimations. Finally, we introduce the Matting Quality (MQ) as

$$\mathrm{MQ} = \frac{\sum_{i \in TP} \sum_{t=1}^{T} S(\bar{\alpha}_t^i, \alpha_t^i)}{N_{TP} \cdot T}. \quad (8)$$

We chose $\rho = 0.5$, $\omega = 50$ for $\mathrm{VIMQ}_{\mathrm{mse}}$ and $\mathrm{VIMQ}_{\mathrm{mad}}$, and $\omega = 10$ for $\mathrm{VIMQ}_{\mathrm{dtssd}}$ in all experiments.

4. Method

4.1. MSG-VIM

To demonstrate feasibility of the VIM task, we propose a baseline model. To this end, we take advantage of the progress made in video instance segmentation and use their mask sequence predictions as auxiliary input. The subsequent network converts them into alpha mattes, thus focusing on improving matting quality. Specifically, given a video clip $\mathbf{I} \in \mathbb{R}^{T \times H \times W \times 3}$ showing N human instances, we employ a VIS method to obtain mask predictions $\mathbf{m}^i = (m_1^i, \dots, m_T^i)$ for instance $i \in N$, where m_t^i is the prediction for instance i at frame t. To enable the matting model to focus on improving the matting result, we split the whole mask sequences into two groups: (i) $\mathbf{m}_{\mathbf{tar}}^i := \mathbf{m}^i$ and (ii) $\mathbf{m}_{\mathbf{ref}}^i := \sum_{j \neq i}^{N} \mathbf{m}^j$. Reference masks have been proven to reduce false positive predictions of non-selected instances [38], which help the subsequent matting model to focus on refining the target mask. Then, we apply a mixture of mask-oriented augmentation in Section 4.2 to make the model robust to inaccurate and misleading mask guidance. We introduce temporal guidance in Section 4.3, which enables the model to exploit matte signals across frames, leading to improved results. Finally, target and reference masks are concatenated with $\mathbf{I}$, resulting in

$$\mathbf{I}^i = \mathrm{Concat}(\mathbf{I}, \mathbf{m_{tar}}^i, \mathbf{m_{ref}}^i) \in \mathbb{R}^{T \times H \times W \times 5}. \quad (9)$$

We utilize a neural network U, which takes $\mathbf{I}^i$ as input. It uses an encoder-decoder design to estimate alpha mattes of the target and reference instances as shown in Figure 2. The encoder is based on ResNet34 [30], which extracts sequences of feature maps at multiple resolutions. Then, the feature maps with the lowest resolution are sent to an Atrous Spatial Pyramid Pooling (ASPP) [7] module and upsampled to higher resolutions in the decoder. We also build skip connections between the feature maps of the same resolution at the encoder and the decoder. For the prediction of the alpha mattes of the target and reference instances, we adopt Progressive Refinement Module [49], and add a light convolution layer upon the feature maps at the decoder to obtain the target and reference mattes. To exploit temporal context and improve time-consistency of the predictions, we add temporal feature guidance between consecutive frames with a lightweight recurrent neural network, see Section 4.3. Finally, the alpha matte sequences $\boldsymbol{\alpha}_{\mathbf{tar}}{}^i$ and $\boldsymbol{\alpha}_{\mathbf{ref}}{}^i$ of instance i are predicted jointly from the upsampled feature maps at the decoder of U:

$$(\boldsymbol{\alpha}_{\mathbf{tar}}{}^i, \boldsymbol{\alpha}_{\mathbf{ref}}{}^i) := U(\mathbf{I}^i). \quad (10)$$

4.2. Mixture of Mask Augmentations

During training, we randomly choose foregrounds and backgrounds and composite training clips on-the-fly. To keep training efficient, we do not use VIS inference to obtain mask guidance. Instead, we convert a labeled alpha matte α into a mask m_α via binarization. To mimic mask sequence guidance from a VIS model, we randomly apply erosion and dilation to the binarized mattes before creating the inputs for (9). Still, resulting masks are often more accurate than typical results from a VIS model. Considering that mask sequence guidance can be inaccurate during inference, we apply a mixture of mask-oriented clip-level augmentations during training to make the MSG-VIM model robust to such guidance, as shown in Figure 2. The first strategy is to use Mask Erase, *i.e.* randomly erasing parts of the mask guidance at each frame during training. Then, we adopt Mask Paste that randomly selects two mask regions, and paste one region to the other one at each frame to add perturbations to the input. We further apply Mask Merge, which randomly picks frames and merges the whole target and reference mask at the selected frame to make the model robust to joint predictions of different instances. With the proposed mask augmentation strategy, the model becomes robust to errors induced by inaccurate mask guidance. We observe significant performance improvements especially on RQ and MQ, as shown in Table 3.

4.3. Temporal Guidance

Using temporal information during matte creation is expected to improve quality. To this end, MSG-VIM utilizes temporal guidance w.r.t. mask sequences and feature maps. **Temporal Mask Guidance** We introduce a temporal mask guidance to make the model robust to individual localisation errors caused by the video instance segmentation method. To this end, during training, we consider a mask sequence $\mathbf{m} = (m_1, \dots, m_T)$. For each m_t at frame t, we randomly

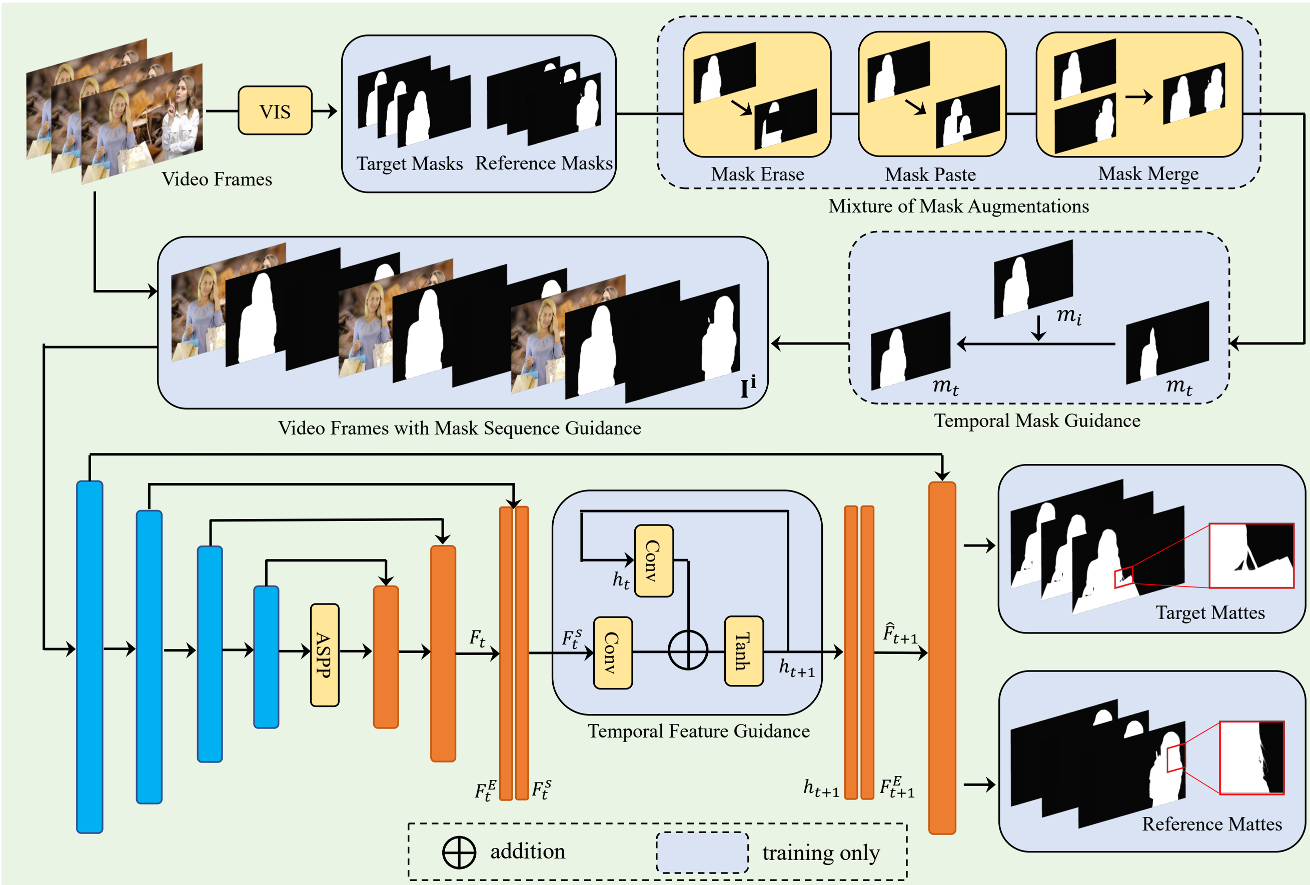

Figure 2. **The Architecture of MSG-VIM.** A VIS method is used to obtain mask sequences of target and reference instances. The sequences of mask guidance are concatenated with video frames as inputs to the matting network, which outputs the refined alpha matte predictions of both target and reference instances. Details of the MSG-VIM are described in Section 4 and illustrated in the Appendix C.

choose $i \in \{1, \ldots, T\}$ and merge the corresponding masks, *i.e.* we set $m_t := m_t + m_i$. Our experiments show that MQ, TQ and RQ are improved with temporal mask guidance, as shown in Table 2.

Temporal Feature Guidance We further add a lightweight convolutional recurrent network at the second highest feature map resolution at the decoder to exploit temporal information, see Figure 2. For timestep t, the feature map, which we denote by F_t, is split into F_t^S and F_t^E by the first and second half of the channels. We compute the recurrent state h_{t+1}, which is used in the next iteration to add temporal context to F_{t+1}, The hidden state is given via $h_{t+1} = \tanh(\mathrm{Conv}(h_t) + \mathrm{Conv}(F_t^S))$. The update is performed via $\hat{F}_{t+1} := \mathrm{Concat}(h_{t+1}, F_{t+1}^E)$. With the temporal feature guidance, the model can exploit temporal context so that the predictions can benefit from it. As shown in Table 4, it further improves the temporal consistency of the predictions across frames.

4.4. Loss Function

We apply a loss function to the predictions of target alpha matte $\boldsymbol{\alpha}_{\mathbf{tar}}$ and reference alpha mattes $\boldsymbol{\alpha}_{\mathbf{ref}}$ simultaneously. For frame t, we use L1 loss L_α^t, pyramid Laplacian loss L_{lap}^t, and composition loss L_{com}^t [18, 32, 38, 39]. The composition loss at frame t is defined as

$$L_{com}^t = \| \sum_{i=1}^{N} \alpha_t^i \circ F_t^i + (\mathbf{1} - \sum_{i=1}^{N} \alpha_t^i) \circ B_t - I_t \|_1. \quad (11)$$

The total loss averages the losses over all frames:

$$L = T^{-1} \sum_{t=1}^{T} L_\alpha^t + L_{lap}^t + L_{com}^t. \quad (12)$$

4.5. Inference

For inference, we apply a VIS method to the video sequence and obtain mask sequences of each instance. We then run inference iteratively for each instance $i \in N$, forming $\mathbf{m_{tar}}^i$ and $\mathbf{m_{ref}}^i$, sending them to the MSG-VIM network, which output the alpha mattes $\boldsymbol{\alpha}_{\mathbf{tar}}{}^i$ and $\boldsymbol{\alpha}_{\mathbf{ref}}{}^i$ for instance i. Finally, all $\boldsymbol{\alpha}_{\mathbf{tar}}{}^i$ are merged as final outputs.

MMA	TMG	TFG	RQ↑	TQ↑	MQ_{mse}↑	VIMQ_{mse}↑
-	-	-	63.28	91.32	42.86	24.77
✓	-	-	70.92	92.16	51.91	33.92 (+9.15)
✓	✓	-	71.67	92.55	55.81	37.02 (+12.25)
✓	✓	✓	**72.72**	**93.17**	**56.52**	**38.29** (+13.52)

Table 2. Ablation on the MSG-VIM Setup. MMA: Mixture of Mask Augmentations, TMG: Temporal Mask Guidance, TFG: Temporal Feature Guidance.

Method	RQ↑	TQ↑	MQ_{mse} ↑	VIMQ_{mse} ↑
MMA	**70.92**	**92.16**	**51.91**	**33.92**
– Mask Erase	61.64	91.70	47.87	27.06 (-6.86)
– Mask Merge	62.15	92.71	48.32	27.84 (-6.08)
– Mask Paste	64.06	92.05	50.97	31.62 (-2.30)

Table 3. Ablation on the Mixture of Mask Augmentations.

Method	RQ↑	TQ↑	MQ_{dtssd} ↑	$\text{VIMQ}_{\text{dtssd}}$ ↑
MMA	70.92	92.16	24.92	16.28
+ TMG	71.67	92.55	27.76	18.41 (+2.13)
+ TFG	**72.72**	**93.17**	**28.51**	**19.32** (+3.04)

Table 4. Ablation on the Temporal Guidance. We select *dtSSD* as the similarity metric to highlight the improvement in the temporal consistency of the predictions.

5. Experiments

5.1. Implementation Details

Datasets Since there is no video instance matting dataset, we select foregrounds and background data from different benchmarks separately to train our model. For the foregrounds, we choose high-resolution clip-level instances from the remaining part of VideoMatte240k [31], which excludes the human instances used in VIM50. For the backgrounds, we choose clip-level video backgrounds without human instances from the remaining part of DVM [39], excluding the backgrounds used in VIM50. Also, we select 20,000 frame-level image backgrounds from BG20k [29] to make the model robust to diverse environments. During training, we select two to four instances as foregrounds and iteratively add them to the video backgrounds [38] to composite the training data, which follows the same practice we used when compositing the VIM50 benchmark.

Training Setting During training, we use eight RTX A6000 GPUs with two video clips per GPU as bath size, each containing 10 consecutive frames. We empoy the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.99$. We use cosine learning rate decay with an initial learning rate of $1e^{-3}$. Training lasts for $2 \cdot 10^4$ iterations, with warm-up at the first $2 \cdot 10^3$ iterations. For the data augmentation, we first generate mask sequence guidance for each instance from the corresponding alpha matte sequence, by binarization followed by random erosion and dilation. We separately apply RandomAffine and RandomCrop to the foregrounds, masks, alpha mattes, and backgrounds. Then, they are composited iteratively and concatenated to train the matting network, see Section 4.2.

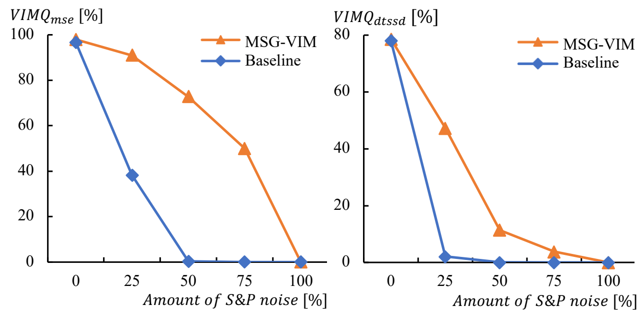

Figure 3. Performance of MSG-VIM and the baseline model w.r.t. VIMQ metrics on VIM50.

5.2. Ablation Studies

MSG-VIM Setup We analyze the impact of the proposed model improvements of Section 4. The first line of Table 2 is the baseline model, as described in Section 4.1, on VIM50 under mask guidance obtained from MaskTrack-RCNN, without any augmentation or temporal guidance. When we gradually add our mixture of mask augmentations (MMA), Temporal Mask Guidance (TMG), and Temporal Feature Guidance (TFG) to the baseline model, the performance gains are 9.15, 12.25 and 13.52 w.r.t. VIMQ over the baseline model, respectively. We refer to MSG-VIM as the model including all these improvements.

Mixture of Mask Augmentation We further analyze the influence of the different methods involved in our proposed mixture of mask augmentations, as shown in Table 3. Without Mask Erase, Mask Merge and Mask Paste, the performance drops 6.86, 6.08 and 2.30 w.r.t. VIMQ, respectively. It shows that MSG-VIM benefits from each mask-oriented data augmentation to improve its robustness to inaccurate mask guidance from VIS methods.

Temporal Guidance Finally, we analyze in detail the impact of temporal mask guidance (TMG) and temporal feature guidance (TFG) as discussed in Section 4.3. We use dtSSD as metric ξ, which evaluates the effectiveness of the temporal guidance. The results presented in Table 4 show that using TMG and TFG further improves both the tracking score and temporal consistency of alpha matte predictions to 18.41 $\text{VIMQ}_{\text{dtssd}}$ and 19.32 $\text{VIMQ}_{\text{dtssd}}$, justifying the relevance of these temporal guidance approaches.

Robustness Analysis We analyze the dependency of MSG-VIM on the accuracy of the mask sequence guidance input. We turn ground truth alpha mattes of VIM50 into masks via binarization and track the performance of MSG-VIM on the perturbed input as we add noise. As shown in Figure 3, when no noise is applied, both models achieve about the same accuracy. When the input data is perturbed, the performance drops for both models. However, the impact is less severe when temporal mask guidance, temporal feature

Method	Backbone	FPS	RQ ↑	TQ ↑	$\mathrm{MQ}_{\mathrm{mse}}$ ↑	$\mathrm{VIMQ}_{\mathrm{mse}}$ ↑	$\mathrm{MQ}_{\mathrm{mad}}$ ↑	$\mathrm{VIMQ}_{\mathrm{mad}}$ ↑	$\mathrm{MQ}_{\mathrm{dtssd}}$ ↑	$\mathrm{VIMQ}_{\mathrm{dtssd}}$ ↑
Video Instance Segmentation										
MTRCNN [47]	ResNet50	24.5	37.43	78.00	15.49	4.52	8.45	2.47	2.21	0.65
SeqFormer [44]	ResNet50	75.7	79.83	98.01	36.01	28.18	23.69	18.53	4.13	3.23
Video Matting										
MODNet [23]	MobileNetV3	124.0	32.67	65.52	18.72	4.01	13.02	2.79	7.85	1.68
RVM [32]	MobileNetV3	131.5	37.29	74.44	20.87	5.79	14.43	4.00	5.00	1.39
Mask-Guided Image Matting										
MGMatting* [49]	ResNet34-UNet	31.4	56.08	87.39	26.88	13.17	17.74	8.69	14.16	6.94
MGMatting† [49]	ResNet34-UNet	31.4	70.83	96.49	43.13	29.48	27.84	19.03	25.33	17.31
InstMatt* [38]	ResNet34-UNet	27.2	65.63	92.63	42.26	25.69	30.45	18.51	21.19	12.88
InstMatt† [38]	ResNet34-UNet	27.2	82.57	97.88	64.39	52.04	47.94	38.74	33.32	26.93
Video Instance Matting										
MSG-VIM*	ResNet34-UNet	30.7	72.72	93.17	56.52	38.29	40.49	27.43	28.51	19.32
MSG-VIM†	ResNet34-UNet	30.7	**91.21**	**98.34**	**78.87**	**70.74**	**59.60**	**53.46**	**46.40**	**41.62**

Table 5. Performance of SOTA methods on the VIM50 benchmark. Models with * and † use mask guidance from MTRCNN and SeqFormer, separately.

Model	MAD↓	MSE↓	Grad↓
BGMv2 [31]	20.35	14.26	22.79
MODNet [37]	11.13	5.54	15.30
RVM [32]	6.57	1.93	10.55
MSG-VIM	**6.47**	**1.73**	**10.40**

Table 6. Performance on the video matting benchmark of RVM [32]. MSG-VIM is evaluated without retraining.

Model	MAD↓	MSE↓	Grad↓	dtSSD ↓
MODNet [37]	24.04	15.53	38.88	12.35
RVM [32]	27.50	21.31	34.18	17.16
MSG-VIM	**20.30**	**13.91**	**22.03**	**9.77**

Table 7. Evaluation of video matting methods and MSG-VIM on VIM50 under video matting metrics.

guidance, and the mixture of mask augmentations are all used. For instance, when we apply Salt-and-Pepper noise to 25% of the pixels of the mask guidance, the baseline model achieves a $\mathrm{VIMQ}_{\mathrm{mse}}$ value of around 40%, while for the full MSG-VIM model, the performance drops only slightly from 99% to 91%. It shows the effectiveness of our model improvements, which mitigate input errors due to the exploitation of temporal information.

5.3. Comparisons to SOTA methods

To make comprehensive comparisons to other methods, we select state-of-the-art methods from three relevant categories: video instance segmentation, video matting, and mask-guided image matting. We then evaluate these methods on the VIM50 benchmark as shown in Table 5. For video instance segmentation, we use the ResNet-50 [17] based checkpoints of the two well-established VIS models: CNN-based MaskTrackRCNN [47] and the most recent state-of-the-art transformer-based SeqFormer [44]. For video matting methods, considering that VIM50 does not provide trimap annotations and trimap-free inference is closer to real-world applications, we select two state-of-the-art trimap-free video matting models: MobileNetV3 [19] based MODNet [22] and RVM [32] for evaluation. Since they do not decompose alpha mattes into corresponding instances, we perform the assignment into instances in the post-processing step. To this end, we binarize alpha mattes into masks, identify in each frame the connected components, and then link each component across time via overlap maximization using the Hungarian algorithm to form for each instance the alpha matte sequence. We also compare our method with mask-guided image matting methods MGMatting [49] and InstMatt [38]. Since both methods are designed for image matting, we apply mask guidance frame-by-frame during inference. We re-train ResNet34-UNet [30] based MGMatting and InstMatt on our training set and evaluate the performance of both methods using MaskTrackRCNN and SeqFormer for the mask guidance. MGMatting and InstMatt successfully refine the mask predictions into instance-level alpha matte in each frame. Yet, MSG-VIM performs much better, especially on the metrics evaluating temporal consistency, *e.g.* $\mathrm{VIMQ}_{\mathrm{dtssd}}$. MSG-VIM also benefits from the more accurate mask guidance from SeqFormer *without* re-training the MSG-VIM model. We also evaluate the inference speed of each model under a single A6000 with input size at 512×288 without extra optimization during inference.

5.4. Video Matting Extension

Video matting benchmark While MSG-VIM is designed for video instance matting, it can also be applied to conventional video matting by merging alpha mattes of all instances into one alpha matte at each frame. We evaluate MSG-VIM on the high-resolution test set used in RVM [32]. The performance in Table 6 shows that MSG-VIM reaches better results compared to previous video matting works. It can thus be considered as high-quality VIM method, while still being useful for video matting, where it delivers state-of-the-art results.

Difficulty Analysis of VIM50 Any video instance matting dataset can be transformed into a video matting dataset by ignoring all instance-related information. Thus, to estimate

Figure 4. Qualitative comparisons of different methods with colored matting predictions of each instance in VIM50.

Figure 5. Qualitative comparisons of different modules with colored matting predictions of each instance in VIM50.

the difficulty of VIM50, we evaluate state-of-the-art video matting methods RVM [32], MODNet [37] as well as MSG-VIM on VIM50 from the video matting perspective. Comparing the same error metrics of Table 7 with Table 6 indicates that the VIM50 benchmark is indeed challenging, already on the video matting task. For the state-of-the-art video matting benchmark presented in RVM [32], the MAD metric ranges between 6.47 (MSG-VIM) and 20.35 (BGMv2 [31]) and the MSE error between 1.73 (MSG-VIM) and 14.26 (BGMv2).

Figure 6. Qualitative comparisons on alpha matte quality of real-world data.

5.5. Qualitative Results

VIM50 benchmark We select frames of the VIM50 benchmark and visualize them in Figure 4 and Figure 5 with colored matting results of different methods and modules. While the mask guidance is inaccurate from MTRCNN, our method is able to correct most errors and produces more accurate alpha mattes than the results from competing methods. The errors are also gradually reduced with the proposed modules cumulatively added.

Real-world data We further select the real-world images with multiple human instances from HIM2K [38], and visualize the prediction of alpha mattes under SeqFormer, MGMatting, and MSG-VIM in Figure 6.

6. Conclusion

In this paper, we propose Video Instance Matting (VIM), a new task aiming at estimating the alpha mattes of each instance at each frame of a video sequence. Furthermore, we establish the VIM50 benchmark and the VIMQ metric to evaluate the performance of different models on the new task. We propose MSG-VIM, a Mask Sequence Guided Video Instance Matting network, as a baseline model for the VIM50 benchmark. It benefits from our mixture of mask augmentations, temporal mask guidance, and temporal feature guidance. It outperforms previous methods by a large margin on our VIM50 benchmark and current methods on the video matting task. We anticipate numerous new applications arising from the proposed task and that the presented dataset will drive research in this direction. In addition, research conducted on VIM may lead to advances in classical video matting, as our studies suggest, due to the great performance of MSG-VIM in this field.

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

[2] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *European Conference on Computer Vision*, pages 158–177. Springer, 2020. 1, 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] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9739–9748, 2020. 1, 2

[5] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2020. 1, 2

[6] Guowei Chen, Yi Liu, Jian Wang, Juncai Peng, Yuying Hao, Lutao Chu, Shiyu Tang, Zewu Wu, Zeyu Chen, Zhiliang Yu, et al. Pp-matting: High-accuracy natural image matting. *arXiv preprint arXiv:2204.09433*, 2022. 2

[7] 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

[8] 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

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

[10] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 3

[11] 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

[12] Mikhail Erofeev, Yury Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, pages 99–1, 2015. 4

[13] Marco Forte and François Pitié. f, b, alpha matting. *arXiv preprint arXiv:2003.07711*, 2020. 2

[14] Vidit Goel, Jiachen Li, Shubhika Garg, Harsh Maheshwari, and Humphrey Shi. Msn: efficient online mask selection network for video instance segmentation. *arXiv preprint arXiv:2106.10452*, 2021. 2

[15] 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

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[17] 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. 7

[18] 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

[19] 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. 7

[20] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 3

[21] 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. 3

[22] 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, 7

[23] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1140–1147, 2022. 7

[24] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020. 1, 2

[25] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 3

[26] 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. 1, 2

[27] Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. *arXiv preprint arXiv:2306.05399*, 2023. 2

[28] Jiachen Li, Marianna Ohanyan, Vidit Goel, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. VideoMatt: A simple baseline for accessible real-time video matting. In *CVPR Workshops*, 2023. 2

[29] 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*, 130(2):246–266, 2022. 6

[30] 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. 2, 4, 7

[31] 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 8762–877f1, 2021. 1, 2, 3, 6, 7, 8

[32] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. *arXiv preprint arXiv:2108.11515*, 2021. 1, 2, 5, 7, 8

[33] 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

[34] 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

[35] 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

[36] Hongje Seong, Seoung Wug Oh, Brian Price, Euntai Kim, and Joon-Young Lee. One-trimap video matting. *arXiv preprint arXiv:2207.13353*, 2022. 2

[37] Jiayu Sun, Zhanghan Ke, Lihe Zhang, Huchuan Lu, and Rynson WH Lau. Modnet-v: Improving portrait video matting via background restoration. *arXiv preprint arXiv:2109.11818*, 2021. 7, 8

[38] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Human instance matting via mutual guidance and multi-instance refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2647–2656, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[39] 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. 2, 3, 5, 6

[40] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7942–7951, 2019. 1, 2, 3

[41] Jue Wang and Michael F Cohen. Image and video matting: a survey. 2008. 2, 3

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

[43] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021. 3

[44] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021. 1, 3, 7

[45] 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. 2

[46] Dogucan Yaman, Hazım Kemal Ekenel, and Alexander Waibel. Alpha matte generation from single input for portrait matting. *arXiv preprint arXiv:2106.03210*, 2021. 2

[47] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 2, 7, 11, 12

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

[49] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. Mask guided matting via progressive refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1154–1163, 2021. 1, 2, 4, 7, 11

[50] Zijian Yu, Xuhui Li, Huijuan Huang, Wen Zheng, and Li Chen. Cascade image matting with deformable graph refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7167–7176, 2021. 2

[51] 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. 2

[52] 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. 2