Transformer Based Video Matting

Jackson Stokes

June 6, 2022

Abstract

Video Background Matting has shown increasing promise over the previous years for reliable, lightweight video background augmentation. However, existing methods rely on convolutional and recurrent models with limited time horizons, limiting their performance in difficult scenes. Meanwhile, transformer-based vision models have achieved SOTA results on other tasks through their extended receptive windows. In this work, we explore how Transformer architectures can be applied to high performance video matting. We contribute a novel spatio-temporal transformer based model, which achieves SOTA results on the alpha matting tasks. Additionally, we contribute empirical results of intermediate architecture ablations studies, to better inform research in other video applications.

1 Introduction

Background Alpha Matting is a task by which the foreground is separated from the background with high fidelity and granularity, in the hopes of recomposing images on different backgrounds. In order to do so, models learn to take in an image and produce an alpha matte of the image, in which each pixel in the image is assigned a float from 0 to 1 representing if the pixel is background (0), foreground (1), or some intermediary value (e.g. 0.4). This alpha matte allows an image to be viewed as a linear combination of the foreground and background based on this alpha, mimicing how light interacts with semi-translucent surfaces in the real world. Figure 1 shows an image with its corresponding alpha matte, along with a treconstructed image with a blurred background for a portrait effect. Images are constructed from an alpha matte, foreground image, and background image, using the formula:

$$I = \alpha(F) + (1 - \alpha)B$$

In order effectively extract an alpha matte from difficult scenes, a model needs to understand how each element fits into the scene as a whole. For example, a model which just looks for human-like objects may falsely identify a painting in the background as being part of the foreground, or it may erroneously leave out objects that are held in the hand. While this context is best derived from total scene understanding, existing approaches to alpha matting leverage convolutional neural networks which have a narrow receptive field. As such, they struggle with difficult cases such as crowds in the background. These problems are exacerbated when this is extended to the video case, where several frames in a sequence need to have their alpha matte extracted with a high degree of coherence.

Previous work has attempted to improve video alpha matting by using temporal information under inference. For example, Lin et al [1] uses a recurrent model in the decoder in order to pass forward the alpha matte inferred from previous frames, resulting in SOTA results for video matting. However, this model still struggles when people or moving objects are present in the background, due to the limited spatial and temporal horizons of its convolutional recurrent architecture.

This problem of long-range sequence understanding was present in similar recurrent networks used in language modeling, where the vanishing gradients of recurrent models caused the model to forget context. This was solved with the introduction of transformer models, which allow for long-range sequence attention.

The promising results of transformers in modeling sequences of words suggests that they can also drive improvements in the alpha matting case. In this persuit, this quarter I researched how vision transformers can be leveraged to improve video alpha matting, finding that with the use of attention mechanims both spatially and temporally, we are able to achieve state of the art results in alpha matting.

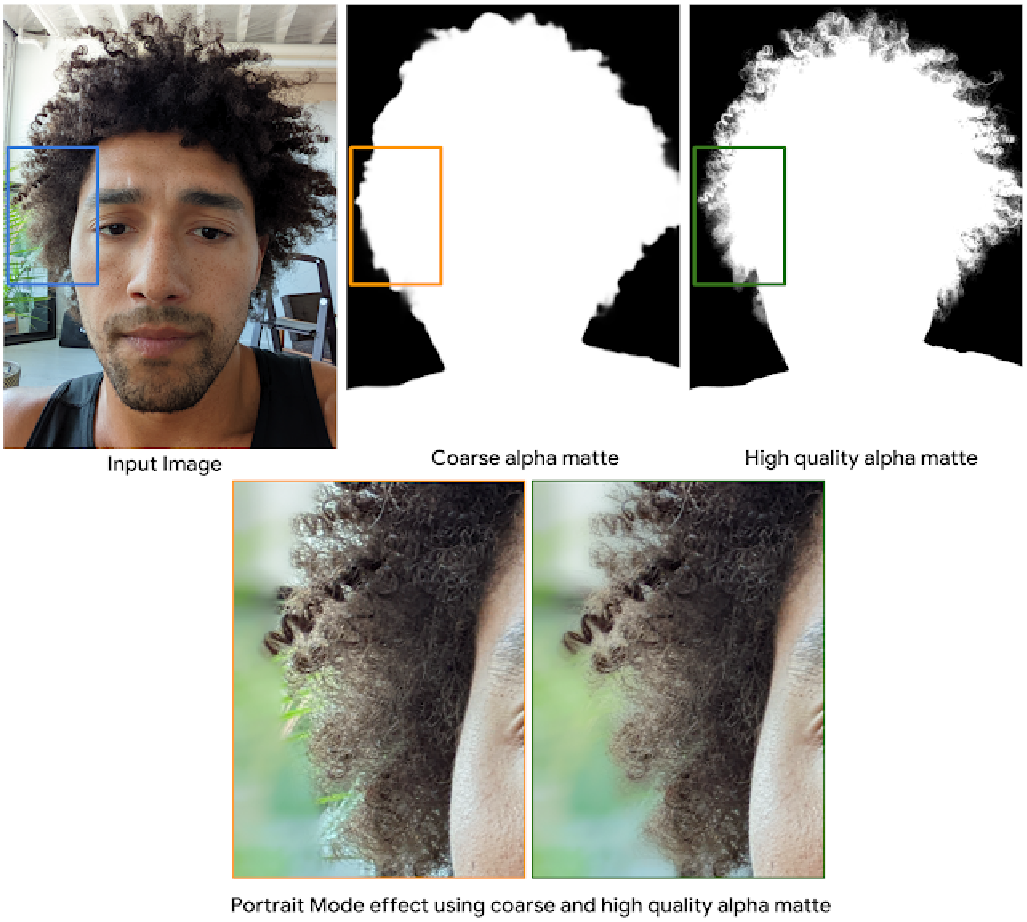

Input Image
Coarse alpha matte
High quality alpha matte
Portrait Mode effect using coarse and high quality alpha matte

Figure 1: The high fidelity alpha matte allows for the blurred regions around the hair to look natural. Cite: Google Camera Research

2 Related Work

This work inherits on two primary research areas:

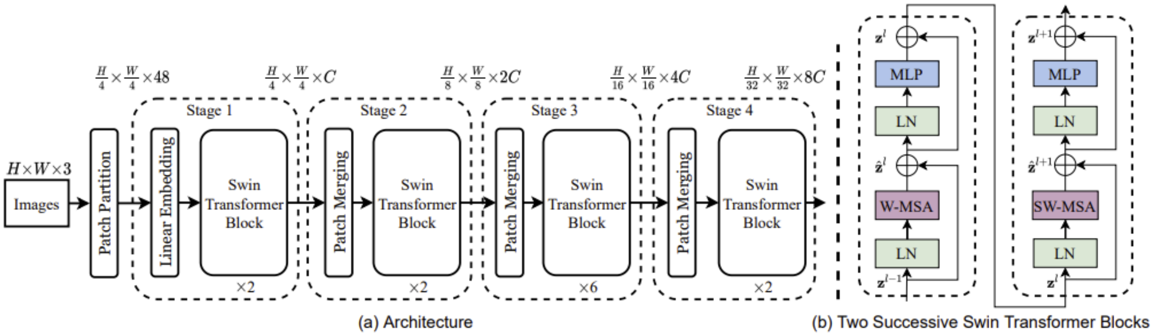
- Video Background Matting [8, 7, 6]
- Vision Transformers [1, 4, 9, 11, 12]

2.1 Video Background Matting

In order to perform video image matting, we now have to perform alpha matting on a large number of frames with a high degree of coherence between these frames. For example, if frames 6, 7, and 9 all predict a certain area's alpha as 0.9, then frame 8 should not predict that area's alpha as 0.1. Otherwise, this can result in a flickering effect as found in [6], which occurs because frames are treated as individual images. One advantage, however, is that the additional temporal information can be used to discern the background from the foreground in tricky cases.

Generally there are two ways to incorporate temporal information under inference on a particular frame. One natural approach is to perform convolutions over the temporal domain, in addition to the spatial convolutions that are performed to extrapolate the high level features of the image. One issue with this approach, however, is that the structure of how temporal information changes can vary a lot. for example, a very different structure of temporal convolution is required for relatively static videos, i.e. a person sitting at a desk, vs dynamic videos, i.e. someone dancing.

Another approach to be considered is using a recurrent neural network to tie together frames. This is the subject of the paper Robust Video Matting [8] which uses a Recurrent Decoder paired with a



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Figure 2: (a): The full architecture of the Swin Transformer without decoder; (b): Information Flow within Swin Blocks. Cite: Microsoft Research Asia

lightweight encoder in order to generate temporally aware alpha matting for videos. This paper relates significantly to the topics presented in the course due to the sheer amount of data it takes to learn to generate effective alpha mattes.

One weakness with their training setup is that the training examples are consistent in having large amounts of movement in the videos. This may seem like an additional difficulty, however in actuality the increased movement provides more signal than if the subject of the video is static, because the lack of temporal changes makes mimic how the background may not move. This makes the model much less effective for applications such as effective background replacement on video calls, or more cinematic video. The problem is not isolated to the training examples, however. The recurrent architecture that they use for the decoder is inherently poorly structured for static videos, because the temporal signal in these videos follows a much sparser distribution.

This problem occurred similarly in language modeling, where sequences of text follow a similar sparse distribution in subjects. Recurrent architectures had problems with vanishing gradients, and struggled to remember key information including long-range context. For example, a noun such as ”the driver” may refer to sever different things; A computer driver, a driver of a car, or a golf driver, and the context may have occurred in the distant past, beyond the reach of recurrent or convolutional models. This issue was remedied with the introduction of transformers for language modeling [11].

For this work, we take heavy inspiration from transformer models for text, as their ability to model long range temporal interactions is useful in our case as well.

2.2 Vision Transformer

There have been several developments in recent years to bring the magic of transformers into both the image and video case. One of the original, effective forays into this space was the paper *An image is worth 16x16 words* [4] which breaks images up into (you guessed it) 16x16 blocks then computes self attention across these. The authors found that this architecture delivered results comparable to SOTA convolutional models, while using far fewer resources to train.

This model is not without its flaws, however. Two main issues arise:

- The scale of objects varies greatly (objects may spread across multiple chunks), and
- The resolution of pixel level information is much higher compared to words in text.

In explaining the first issue, as you can image, faces of individuals close to the camera get treated differently from faces far from the camera, which fit in a single image block. To highlight the second point of increased resolution, **A single 4k image has 10 times the number of pixels as there are words in Shakespeare’s entire works**. As a result, the scale of data that is being operated on under the vision transformer sense can become massive. In order to address some of these pitfalls, we introduce the second seminal work we inherit from in this research.

2.2.1 Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

The Swin transformer builds on vision transformer research to solve some of the issues that arise in bringing transformers to computer vision. In order to solve the issues with different scales of features in images, it builds upon the pyramid vision transformer [12] which computes features and self attention at different resolution of blocks. This means that features with take up a third of the image can be considered in addition to features at lower resolution. However, that work still had a problem with placement; it still looks for features within certain regions because it simply merges adjacent blocks at each. Swin transformer attempted to solve this by taking a lesson from convolutional image networks, which effectively slide kernels at different areas throughout the image.

The Swin transformer works by computing joint self-attention across patches of various resolution, while also offsetting these patches at different levels to consider features which may occur in between patch blocks. The architecture shown in Figure 2 details how Shifted transformer blocks are considered at reductions of 4x4, 8x8, 16x16, and 32x32. This encoding at different resolutions allows for features to be considered across many different scales, while keeping the computational complexity from exploding. By computing self attention across these different patches in the image, at different resolution, This model becomes much better at correlating features in different regions of the image, where convolutions may fail. It also effectively deals with the massive scale of image data; both with the size of individual 4K and HD images, as well as the large training scale of ImageNet used to train these models.

2.2.2 Video Transformers

While the aforementioned models provide self-attention for individual images, there are some necessary modifications to set these up specifically for embedding videos. The first expands on the original vision transformer by splitting temporal video sequences into blocks and correlating across them [1]. The Swin transformer authors also perform their shifted window trick across temporal blocks, showing a slight improvement in [10]. Both of these fail however to process real-time video, which is necessary for use in downstream tasks like video call background replacement. While this is out of the scope of my current research, it is an interesting future direction that warrants more exploration.

2.3 Motivation

In this research, we explore how transformers can improve alpha matting models to solve the two main issues with current baselines:

- Convolutional encoders struggle to extract features with appropriate scene context.
- Recurrent neural networks decoders are unable to incorporate distant temporal information, and they suffer from vanishing gradients.

The result of these two issues is that current baseline models struggle to generate alpha mattes for difficult scenes, or scenes with weak temporal signal. In this work, we uncover that the use of transformers can reach new levels of performance, through helping remedy these two issues.

3 Contributions

In this work, we present two main contributions:

- A novel video matting model, using both spatial and temporal self attention mechanisms. Our model achieves state of the art results on the video matting task.
- Empirical analysis of various attention structures for alpha matting, to contribute to the burgeoning space of video transformer research.

Additionally, we contribute the learnings we acquired along the way with various stepping-stone models.

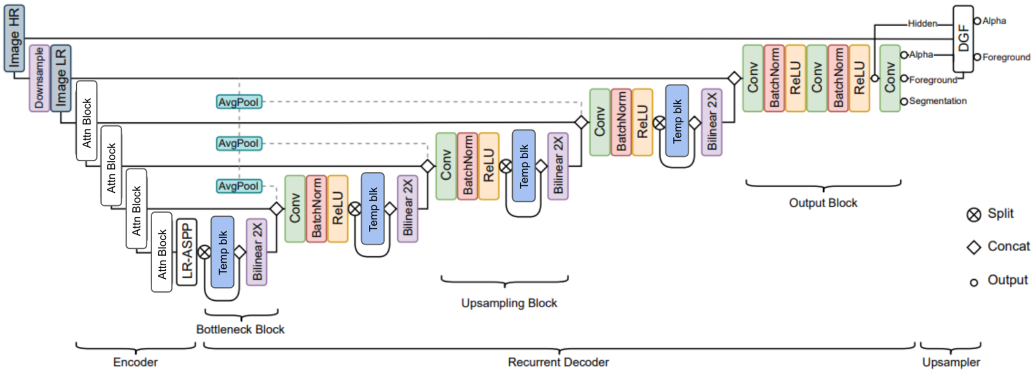


Figure 3: Our model architecture. Swin transformer ”attn blocks” extract features with full image context, and our custom ”temporal” transformer blocks aggregate feature information across the entire sequence.

4 Model architecture

Our architecture consists of three main sections: First, a spatial transformer extracts features independently for each frame at various resolutions. The feature channel dimension of the lowest resolution is reduced using a LRASPP bottleneck. Our decoder then reconstructs an alpha matte form these features, using a temporal attention mechanism to aggregate features across frames. Our full model architecture can be seen in figure 3.

In addition to this model, we have constructed several other model architectures with different attention structures.

4.1 Spatial Transformer Encoder

Our encoder is a vision transformer model which takes in the entirety of the frame to infer local features. We find the best results using the Swin transformer. As mentioned in related work, this chunks the image into patches, computes self attention across the other patches, then merges these patches to repeat the process with higher feature depth and lower resolution. We find that using a joint spatial-temporal transformer encoder at this step led to reduced performance. As such, we retain only spatial encoding here. This has the added benefit of model performance from lower overhead.

4.2 Temporal transformer decoder

Our decoder consists of custom temporal self attention blocks with relative patch embedding, In addition to 2d convolutions for feature aggregation and smoothing. The main work here is in the temporal attention block. Taking in feature input from the previous block, these transformer blocks first apply linear projections to generate embeddings for the Query, Key, and Value:

$$Q, K, V =_{3*E} (\text{inputs})$$

Where E is the embed dim. After we have these embeddings, we scale the query embeddings by a fixed parameter to ensure gradient consistency, take the dot product of the keys and queries, and add our relative positional embeddings:

$$(QK^{\top} + E_{pos})$$

We source the relative positional embeddings as a learnable parameter. By relative, we mean that each frame in a sequence is assigned a vector of embeddings representing the temporal difference to each other frame in the sequence, i.e.

$$\text{Embed}_T = ..., E_{t-2}, E_{t-1}, E_t, E_{t+1}, E_{t+2}, ...$$

Note that this means that for a sequence of length N there must be $2N-1$ possible relative embeddings, which look all the way back across the sequence and all the way forward. In order to store these

embeddings while still allowing sequences of varying length, they stored in a learnable parameter of up to relative distance of 64. We chose this size due to the effective training limitations of GPU sizes (for 512 resolution, we cap out at around 50 frames effective training on A40 GPUs.
After computing the attention matrix and adding the relative position matrix, we can finally compute the self attention, normalized based on the embed dimension d:

$$\text{Attention}(QKV) = softmax(\frac{QK^\top}{\sqrt{d_k}})V$$

This represents our new sequence values for this block. Under reconstruction, this provides a temporal smoothing which adds a high degree of cohorence to the frames, as well as being able to track features across frames. For our highest performing model, we compute the temporal attention independently per patch location, rather than also attending on all other patches.

4.3 LR-ASPP bottneck and deep guided filter

In addition to the key structures mentioned above, we use an Atrous Pyramid Pooling Module [3] to reduce channel dimension at the bottleneck, and a deep-guided filter as a high resolution refiner.

5 Datasets and Evaluation

Our training sequnce uses various publicly available datasets, of varying size and quality.

- VideoMatte (VM240k): 484 pairs of high-resolution alpha matte and foreground video clips, constituting 240,709 unique frames.
- Adobe Image Matting (AIM): 49k Image matte pairs
- ImageNet-S: 1.2 Million images, 50k semantic segmentation masks.
- Coco panoptic: 380,000 images with segmentation masks.
- Youtube-vis: 232k annotations.

Generating VideoMattes is a complicated process. VM240k was created using green-screen video samples owned and licensed by the University of Washington. These Video frames had corresponding alpha mattes generated using chromatic software (the green screen becomes the background in the alpha matte, and edge regions consider both the amount of green in that pixel as well as nearby pixels to get true color. These alpha mattes are still relatively coarse, however. In order to expose the model to higher quality alpha mattes, I will be training Adobe's high quality image matting dataset, as well as additional sources using segmentation passes as well as single frame image matting. This is similar to the training setup used in [8].

We will be evaluating our models both quantitatively, using the L1 loss of the generated alpha matte from ground truth, as well as qualitatively, to see which video clips are particularly challenging for each model to best expose their weaknesses. We train each model primarily on 512 by 512 resolution clips, with additional high resolution passes.

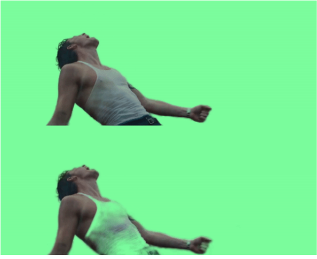


Figure 4: We see how our model (top) outperforms the baseline RVM model (Bottom) on a video test case where scenes change rapidly.

Model	MAD
Ours	**1.32e-3**
RVM [8]	1.70e-3
RVM-small	2.03e-3
MODnet [6]	2.74e-3

Table 1: Comparisons of model performance on held out VM240k data. Our model beats RVM, the current alpha matting benchmark.

6 Results

We find that our model beats the state of the art alpha matting model on held-out VM240k data. Mean average distances of test sets are reported in table 1. Under qualitative review of outputs, we see that our model performs especially better on videos where scenes are changing rapidly. This can be attributed to the fact that our temporal transformer is bidirectional, meaning when evaluating a frame it can use information from following frames. This contrasts with the recurrent decoder in the RVM baseline, which can only consider frames in the past. The result is that when video clips rapidly change scenes, the recurrent decoder struggles to "catch up". This is shown in figure 4.

We also find that our model has comparable size and inference rates compared to RVM using the resnet50 encoder, with both achieving around 45 FPS under inference on a single GPU. This shows that our model's accuracy does not come at a cost of speed or resource usage. RVM has the option to use a lightweight MobileNetV3 [5] encoder which is smaller and has higher FPS rates under inference, though this comes at a cost of accuracy as shown in table 1.

7 Additional Contributions: Ablation studies of several architectures

Additionally, we would like to present some of the learnings found from our earlier experiments throughout this research, to provide better understanding of how each part plays a role.

7.1 Experiment 1: Single Image Matting with Swin Encoder

We began with a sanity check to make sure that the tranformer would perform at least comparably to the lightweight convolutional encoders. In order to keep this ablation as simple as possible, This section of the experimentation is based off of the current SOTA model for trimap-free, single image portrait matting, Modnet [6].

To test the transformer's effectiveness relative to SOTA, We built a simple model based on MODnet[6] which replaces their convolutional encoder (MobileNetV3) with the Swin-T transformer. While structured similarly, this ends up being a significantly larger model due to both the increased encoder size as well as the increased channel dimensions of skip connections, requiring a larger decoder. The transformer encoder is likely to match or outperform the convolutional encoder because of it's effectiveness at scene understanding compared to convolutional encoders which operate at similar feature scales. However, the important spatial locality is still preserved with the convolutional decoder.

We trained this model for around 20 epochs with single-frame, synthetic RVM data at a 512×512 resolution. We find that the transformer encoder effectively works as a replacement for MobileNetV3 [5] (the details of these results are presented in the results section.) This finding serves as justification that the architecture works, allowing us to move forward with our next two experiments.

7.1.1 Experiment 1 Results

We find that the Swin transformer outperforms the baseline convolutional encoder used in MODnet. Across held-out synthetic $VM240k$ data, **Model 1** achieves a mean squared error of **7.442e-3**, compared with MODnet's MSE of **7.925e-3**. This is a relatively modest improvement, however as single frame matting does not take into account any temporal information from previous video frames (where we expect the transformer to really out-perform) the purpose of this experiment was really just to make sure we didn't get worse results than baseline. This result gives us the justification to build upon these models and expand into the video domain.

7.2 Experiment 2: Single Image Matting with Full Transformer Network

Following the success of the transformer encoder in replacing the convolutional encoder for single image matting, We sought to learn if replacing the convolutional *decoder* with a transformer based model delivers an improvement. For this, we built a full E2E transformer alpha matte model, based on a similar model experimented with for medical segmentation [2]. We based the decoder of this model off of the Swin transformer encoder, building in separate patch expanding blocks to replace the patch merging operation performed in original Swin encoder. These blocks work by combing 2×2 blocks of patches at each step, merging the feature values into one patch then performing spatial self attention across the other merged patches. The Swin transformer blocks are left unchanged. The resulting model is similar to a U-NET, but with transformer blocks and patch merging instead of convolutions and upsampling.

7.2.1 Experiment 2 Results

We find that our full transformer U-net model performs worse than the convolutional baseline. Across held-out synthetic $VM240k$ data, **Model 2** achieves a mean squared error of **9.831e-3**, compared with MODnet's MSE of **7.925e-3**. Moreover, we can see this poor performance visually by inspecting the model predictions against ground truth, shown in Figure 5a. We see that when using swin as a decoder, patchy, block-like artifacts can appear. we believe that this is due to patch errors from lower-resolution layers are being propogated up to the full resolution alpha matte. This also makes sense, because the transformer can lack the spatial emphasis that convolutional models have under decoding. From a high level, while the encoder is meant to extract feature information, under alpha matting the decoder's main goal is to incorporate the full pixel-level information from the image with the feature information from the encoder, and smooth the combination to create a visually appealing image. Because the feature information is handled by the encoder, the act of locally combining these aspects requires an emphasis on spatial locality. It makes sense then that the convolutional architecture is a better option for a decoder.

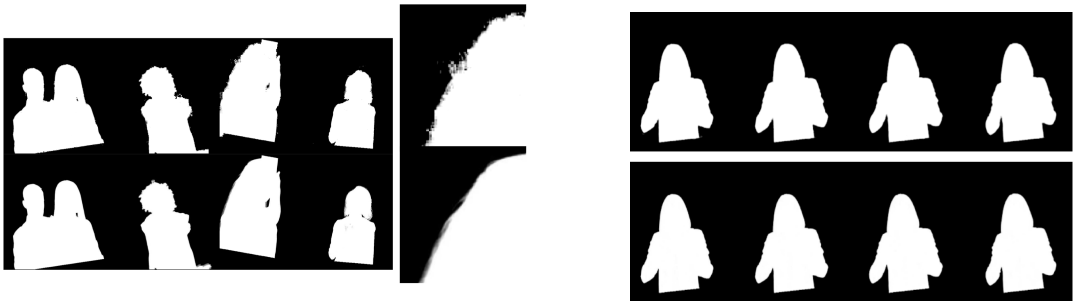
7.3 Experiment 3: Video Image Matting

After validating that the transformer architecture is suitable for use as an encoder in experiment 1, we moved on to the video domain. For ablation, we started off with the SOTA video matting network, and experimented with how introducing spatial / temporal self attention can impact results. In order to understand the various ablation steps, it's useful to first get a deep-dive into the Robust Video Matting (RVM) architecture. The model contains 3 main components:

- Convolutional single frame backbone
- ASPP bottleneck
- Recurrent Conv-GRU decoder

While the recurrent decoder leverages the recurrent information from previous frames, both the convolutional backbone and ASPP module operate on a single frame at a time. This makes it relatively easy replace the backbone with another single frame backbone (though it will require some modification to make channel dimensions match up). Based on this, as well as the results from the Swin transformer outperforming MobileNet in the first experiment, for my initial experiment I simply replaced the single frame convolutional backbone with the swin-transformer backbone I trained in experiment 1. The forward pass of the resulting model is detailed as such: (Note B refers to the number of video clips per batch, T refers to the number of frames in each clip, H, W are the dimensions of each frame and C is the number of embed channels.)

1. The dataloader loads a single clip, with dimensions $[B, T, 512, 512, 3]$



(a) Experiment 2: Full Transformer Results. Note the poor-quality, patchy details.

(b) Experiment 3: Video Results.

Figure 5: Visual comparisons of **predicted** (top) vs **ground truth** (bottom) alpha mattes for experiments 2 and 3.

2. The Swin Transformer Backbone independently encoder each frame, at various feature scales for the bottleneck and skip connections. These outputs are of shape:

$$[B, T, 128, 128, 96], [B, T, 64, 64, 192], [B, T, 32, 32, 384], [B, T, 16, 16, 768]$$

3. The ASPP bottlneck operates on the lowest resolution, 768 channel outputs.
4. The decoder iterates over each frame in the encoded clip (dimension T,) passing the recurrent state from previous frames. Starting from the lowest resolution, bottleneck outputs, it repeatedly upsamples each frame then convolves it with the skip connections of the same (now 2x larger) resolution. As such, the reconstruction of the alpha matte continually incorporates the low-resolution feature information with the progressively higher resolution information from the skip connection provided by the transformer.

We chose to keep the SOTA convolutional decoder initially for this step, due to the result in experiment 2, which suggests that the transformer architecture does not perform as well for use as a decoder.

7.3.1 Experiment 3 Results: We beat the SOTA model on the synthetic VM240k video model

Finally, we find that **Model 3** outperforms the SOTA video matting network RVM [8]. Our model achieved an MSE of **6.58e-3** compared with RVM's 6.92e-3 MSE. Note that both of these models outperform their single frame counterparts because temporal cues can provide valuable information to improve the alpha matte, based on what was inferred in previous frames. Visual samples of this can be seen in figure 5b. Similar to experiment 1, it makes sense that this model outperforms the models which use convolutional encoders as the baseline, as the transformer encoder has shown to be more effective at feature extraction.

8 Discussion

Our transformer model outperforms baseline convolutions and recurrent networks for a few main reasons:

- Transformer Encoders allow features to be aware of the entire image, providing a better ability to segment regions as background and foreground.
- The temporal aggregation of features allows our model to keep better track of objects over time and identify them as foreground objects.
- The bidirectional nature of the temporal transformer allows the model to learn from the future.

To explain the first point, consider the case where a face appears in a crowd in the background. When a convolutional encoder passes over that region of the image, it is unaware of what's happening in the rest of the image, so it partially learns to just recognize subject-like objects, including faces. This was expressed as a limitation of the convolutional baseline, RVM [8]. However, our model allows for long range interactions made possible by the use of a spatial transformer [?]. This addition of better feature context is likely the biggest reason for the improvement, partially as seen in ablation experiment 2, where we see the improvement of just the transformer encoder alone still delivering significant improvements.

To explain the second and third points, imagine the case where an individual in a video call and briefly moves their hand down out of frame. At that point, it would likely be dropped from the recurrent state of the decoder, however our model is able to attend on the distant past and future frames where the hand is present and recognized as foreground. This expanded temporal horizon allows for much better feature tracking for the duration of the model. We were able to validate this through a qualitative analysis of videos that these models performed the worst on.

Finally, we must consider the finding that convolutions are still useful in providing a local smoothing effect in the decoder. We believe that this brings light to the best use case of convolutional models: While they may be outperformed in feature understanding by vision transformers, their ability to aggregate local information is unmatched. The alpha matting task requires a precise balance of feature understanding (to 'see' the foreground out of the background) as well as the precision to still create a smooth alpha matte.

8.1 Result Limitations

We'd like to reiterate that these results come from models that are trained and tested on $VM240k$ synthetic data. We use this dataset in order to rapidly iterate with architecture experiments, however in order to build a full production model, it's better to do a more detailed experimentation including custom datasets of both images and video. Previous papers in the space do not do this, as it is infeasible for research timelines. [8]

9 Future work

In the scope of this quarter, we focused on building the best possible model we could. However, there are still other attention mechanisms and strategies that are worth exploring. For example, local attention structures, which include convolutions over the embedded patches, could potentially allow for better object tracking, giving better accuracy around moving hands. Additionally, in order to make this project paper-ready for submission, We will need to perform further analysis of our model against more baselines.

10 Conclusion

In this work, we show that transformers are able to serve as effective replacements for convolutional encoders and recurrent neural networks. We present a model which achieves state of the art results on video alpha matting, along with ablation studies to understand how each component plays a role. Finally, we present our analysis for the performance of this model, considering the various architecture choices. Our findings suggest that transformers are significantly more effective at modeling the contextual interactions relevant for alpha matting, while still maintaining the localized precision useful for alpha matte reconstruction. We are excited to see how others build on this work and our code to further push the boundaries of video transformer research.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[2] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation, 2021.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2016.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[5] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[6] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W. H. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition, 2020.

[7] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. *CoRR*, abs/2012.07810, 2020.

[8] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[10] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[12] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021.