ReBotNet: Fast Real-time Video Enhancement

Jeya Maria Jose Valanarasu[1,3*] Rahul Garg[1] Andeep Toor[1] Xin Tong[1] Weijuan Xi[1]
Andreas Lugmayr[2] Vishal M. Patel[3] Anne Menini[1]
[1]Google [2]ETH Zurich [3]Johns Hopkins University

Abstract

*Most video restoration networks are slow, have high computational load, and can't be used for real-time video enhancement. In this work, we design an efficient and fast framework to perform real-time video enhancement for practical use-cases like live video calls and video streams. Our proposed method, called **Re**current **Bot**tleneck Mixer **Net**work (**ReBotNet**), employs a dual-branch framework. The first branch learns spatio-temporal features by tokenizing the input frames along the spatial and temporal dimensions using a ConvNext-based encoder and processing these abstract tokens using a bottleneck mixer. To further improve temporal consistency, the second branch employs a mixer directly on tokens extracted from individual frames. A common decoder then merges the features form the two branches to predict the enhanced frame. In addition, we propose a recurrent training approach where the last frame's prediction is leveraged to efficiently enhance the current frame while improving temporal consistency. To evaluate our method, we curate two new datasets that emulate real-world video call and streaming scenarios, and show extensive results on multiple datasets where ReBotNet outperforms existing approaches with lower computations, reduced memory requirements, and faster inference time. Project site: https://jeya-maria-jose.github.io/rebotnet-web/*

1. Introduction

Video enhancement has several use-cases in surveillance [63, 14, 59], cinematography [83, 27], medical imaging [34, 68], virtual reality [87, 54, 23, 81], sports streaming [11, 107], and video streaming [105, 105]. It also facilitates downstream tasks such as analysis and interpretation [60], e.g., it improves accuracy of facial recognition algorithms, allows doctors to diagnose medical conditions more accurately, and helps in better sports analysis by understanding player movements and tactics. Also, the recent rise of hybrid work has led to an immense increase in video conferencing, where poor video quality due to a low quality camera, poor lighting conditions, or a bad network connection can obscure non-verbal cues and hinder communication and increase fatigue [16]. Thus, there lies a significant interest in developing methods that can perform real-time video enhancement.

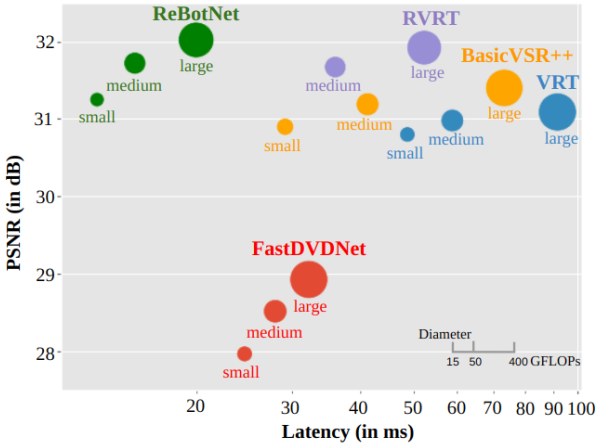


Figure 1: A comparison between the performance of ReBotNet with state-of-the art video restoration networks across different FLOPs regimes on a NVIDIA A100 GPU for PortraitVideo dataset. ReBotNet is observed to give the best performance while having the least latency.

Unlike individual restoration tasks like denoising [18, 74], deblurring [102, 61], super-resolution [90, 44] which focus on restoring videos affected by a single degradation; generic video enhancement techniques focus on improving the overall quality of videos and make them look better [93]. In this setup, there are multiple degradations that can interact in a complex way, e.g., compression of a noisy video, camera noise, motion blur etc. mirroring the real world scenarios. Video restoration methods can be adopted for video enhancement by training on a dataset that includes multiple degradations. However, from our experiments we found that they are computationally complex and have a high inference time and are not suitable for real-time applications. Also, many methods take past and future frames as input which will introduce latency in streaming video.

*Parts of the work was done during an internship at Google.

In this paper, we develop an efficient video enhancement network that achieves state of the art results and enables real-time processing. At the core of our method is a novel architecture using convolutional blocks at early layers and MLP-based blocks at the bottleneck. Following [67] which uses a convolutional encoder for initial feature extraction followed by a transformer network in the bottleneck, we propose a network where the initial layers extract features using ConvNext [47] blocks and a bottleneck consisting of MLP mixing blocks [76]. This design avoids quadratic computational complexity of vanilla attention [82], while maintaining a good performance. We also tokenize the input frames in two different ways to enable the network to learn both spatial and temporal features. Both these token sets are passed through separate mixer layers to learn dependencies between these tokens. We then use a simple decoder to predict the enhanced frame. To further improve efficiency and improve temporal consistency, we exploit the fact that real world videos typically have temporal redundancy implying that the prediction from previous frame can help current frame's prediction. To leverage this redundancy, we use a frame-recurrent training setup where the previous prediction is used as an additional input to the network. This helps us carry forward information to the future frames while being more efficient than methods that take a stack of multiple frames as input. We train our proposed network in this recurrent way and term our overall method **Re**current **Bot**tleneck Mixer **Net**work (ReBotNet).

To evaluate our method, we curate and introduce two new datasets for video enhancement. The existing video restoration datasets focus on a single task at a time, e.g., denoising (DAVIS [35], Set8 [72], etc.), deblurring (DVD [69], GoPro [50], etc.), and super-resolution (REDS [49], Vid4 [41], Vimeo-90k-T [93], etc.). These datasets do not emulate the real-world case where the video is degraded by a mixture of many artifacts. Also, rise in popularity of video conferencing calls for datasets that have semantic content similar to a typical video call. Single image enhancement methods are often studied on face images [46, 32] because human perception is very sensitive to even slight changes in faces. However, a comparable dataset for video enhancement research has yet to be established. To this end, we curate a new dataset called *PortraitVideo* that contains cropped talking heads of people and their corresponding degraded version obtained by applying multiple synthetic degradations. The second dataset, called *FullVideo*, contains a set of degraded videos without face alignment and cropping. We conduct extensive experiments on these datasets and show that we obtain better performance with less compute and faster inference than recent video restoration frameworks. In particular, our method is 2.5x faster while either matching or in some cases obtaining a PSNR improvement of 0.2 dB over previous SOTA method. This shows the effectiveness of our proposed approach and opens up exciting possibilities of deploying them in real-time applications like video conferencing.

In summary, we make the following major contributions:

- We work towards **real-time** video enhancement, with a specific focus on practical applications like video calls and live streaming.
- We propose a new method: Recurrent Bottleneck Mixer Network (ReBotNet) , an efficient deep neural network architecture for real-time video enhancement.
- We curate two new video enhancement datasets: *PortraitVideo*, *FullVideo* which emulate practical video enhancement scenarios.
- We perform extensive experiments where we find that ReBotNet matches or exceeds the performance of baseline methods while being significantly faster.

2. Related Work

Image and video restoration [15, 19, 20, 22, 103, 104, 110, 96, 55, 94, 97, 98] is a widely studied topic where CNN-based methods have been dominating over the past few years. For video restoration, most CNN-based methods take a sliding window approach where a sequence of frames are taken as input and the center frame is predicted [69, 72, 73, 75, 89, 110]. To address motion between frames, many methods explicitly focus on temporal alignment [7, 111, 8, 75, 89], with optical flow being a popular alignment method [4, 31, 42, 42, 71, 93]. Dynamic upsampling filters [29], spatio-temporal transformer networks [36], and deformable convolution [75] have been proposed for multi-frame optical flow estimation and warping. Aside from sliding window approaches, another widely used technique is a recurrent framework where bidirectional convolutional neural networks warp the previous frame prediction onto the current frame [7, 9, 21, 24, 26, 28, 51, 53, 66, 109, 62]. These recurrent methods usually use optical flows to warp the nearby frames to create the recurrent mechanism. Unlike these works that require compute intensive optical flow, we develop a simple and efficient frame-recurrent setup with low computational overhead. As most of these methods use synthetic datasets, recent works have looked into adopting these methods for real-world application [95, 10]. One recent work attempted to solve multiple degradation problem that includes blur, aliasing and low resolution with one model [5] but it is still computationally intensive.

Since the introduction of transformers [82] for visual recognition [17, 45], transformers have been widely adopted for many restoration tasks [80, 12, 91, 39, 70, 57, 106, 6, 108, 65]. Deformable attention [85] has been proposed for video super-resolution. Video restoration transformer (VRT) introduced a parallel frame prediction model leveraging long-range temporal dependency modelling abil-

ities of transformers [38]. Recurrent video restoration transformer (RVRT) [40] introduced a globally recurrent framework with processing neighboring frames. At the time of writing, it is worth mentioning that RVRT stands as the SOTA method for most video restoration datasets. Unlike above methods, we focus on developing real-time solutions for generic video enhancement with a focus on practical applications like live video calls.

3. Method

3.1. Recurrent Bottleneck Mixer

Transformers [17] form the backbone of current state of the art video restoration methods [38, 40] due to their ability to model long-range dependencies but suffer from high computational cost due to the quadratic complexity of attention mechanism. Attention with linear complexity [86, 101, 33] reduces performance while still not achieving real-time inference. On the other hand, [100, 43, 84] show that attention can be replaced by other mechanisms with marginal regression in quality, e.g., [76] replaces self-attention with much more efficient token mixing multi-layer perceptrons (MLP-Mixers). Mixers have been subsequently shown to be useful for multiple tasks [77, 79, 99, 58, 78, 48]. However, Mixers do not work out-of-the box for video enhancement, as (i) they lead to a significant regression in quality (in our experiments in supplementary material) compared to transformer-based approaches, and (ii) while more efficient, they still do not yield real-time inference on high resolution imagery. Also, videos are processed using transformers by either representing them as tubelets or patch tokens [1]. However, tubelet tokens [1] and image tokens [17] can be complementary with different advantages and disadvantages. Tubelet tokens can compactly represent spatio-temporal patterns. On the other hand, image tokens or patch tokens extracted from an individual frame represents only spatial features without spending capacity on modeling motion cues. These issues motivates us in developing a new backbone for video enhancement with mixers at its core while combining tubelets and image tokens in a single efficient architecture.

After motivating the design, we now explain our proposed network architecture: Recurrent Bottleneck Mixer Network (ReBotNet) in detail. First, we give an idea about the overall network architecture and then delve into the details of tokenization, bottleneck, and the recurrent setup. An overview of ReBotNet can be found in Fig. 3. ReBotNet takes two inputs: the previous predicted frame (y_{t-1}) and the current frame (x_t). We use an encoder-decoder architecture where the encoder has two branches. The first branch focuses on spatio-temporal mixing where we tokenize the input frames as tubelets and then process these spatio-temporal features using mixers in the bottleneck. The output features of this mixer block has information processed along both the spatial and temporal dimensions. The second branch extracts just the spatial features using linear layers from individual frames. These tokens contain only spatial information as the frames are processed independently. These spatial features are forwarded to another mixer bottleneck block which learns the inter-dependencies between these tokens. This mixer block captures temporal information by extracting the relationship between tokens from individual frames, thereby encoding the temporal dynamics. The resultant features from both branches are added and are forwarded to a decoder which consists of transposed convolutional layers to upsample the feature maps to the same size as of the input. We output a single prediction image (y_t) which is the enhanced image of the current frame (x_t).

3.2. Encoder and Tokenization

Tokenization is an important step in pre-processing data for transformer-based methods as it allows the model to work with the input data in a format that it can understand and process [56]. For our network, we use two different ways of doing tokenization: i) tubelet tokens and ii) image tokens.

Branch 1 - Tubelet tokens: Tubelet tokens are extracted across multiple frames, in our case, the current frame and the previous predicted frame, and encode spatio-temporal data. Convolutional layers can be advantageous in extracting tokens as they can capture more informative features compared to linear layers due to their inductive bias [92]. Hence, we stack the input images: y_{t-1}, x_t across the channel dimension and directly forward them to ConvNext blocks [47], which are more efficient and powerful than vanilla convolutional layers. Each ConvNext block consists of a depth-wise convolution layer [13] with kernel size of 7×7, stride 1 and padding 3 followed by a layer normalization [2] and a point-wise convolution function. The output of this is activated using GeLU [25] activation and then forwarded to another point-wise convolution to get the output. More details of this why this exact setup is followed can be found in supplementary. We also have downsampling blocks after each level in the ConvNext encoder. These tubelet tokens compromise the first branch of ReBotNet where we do spatio-temporal mixing. These tokens are further processed using a bottleneck mixer to enhance the features and encode more spatio-temporal information.

Branch 2 - Image tokens: The individual frames y_{t-1}, x_t are from different time steps. Although tubelet tokens encode temporal information, learning additional temporal features can only improve the stability of the enhanced video and help get clearer details for enhancement. We do this by extracting individual image tokens and learn the correspondence between them. To this end, we tokenize the

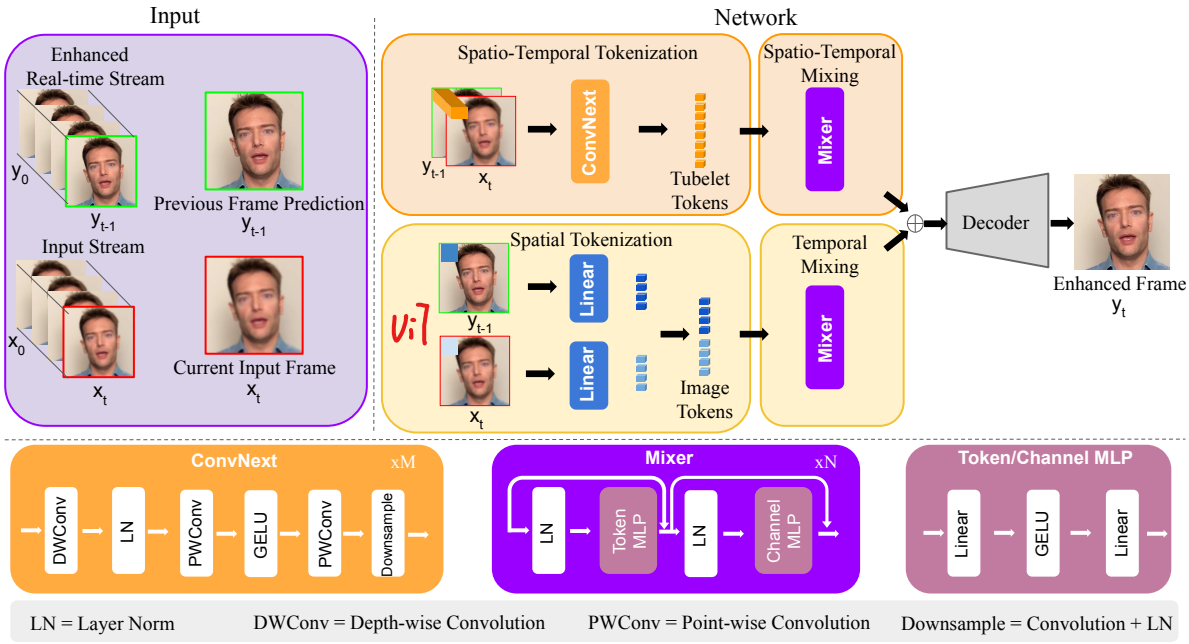


Figure 2: Overview of the proposed Recurrent Bottleneck Mixer Network. The inputs to the network are the previous frame prediction and the current input frame. These are tokenized in two different ways: Tubelet tokens and image tokens . The tubelet tokens are processed using a Mixer to learn spatio-temporal features while image tokens are processed using a Mixer to learn temporal features. These features are passed through an upsampling decoder to get the output enhanced frame.

images individually by converting them into patches and using linear layers like in ViT [17]. In this branch, we use linear layers instead of ConvNext blocks for the sake of efficiency although ConvNext blocks extract more representative and useful features. The main goal of this block is to ensure that the temporal features of the input data remain consistent. Note that high quality spatial features necessary for enhancing spatial quality, is handled in the first branch. To this end, the mixer bottleneck learns to encode the temporal information between these image tokens extracted from individual frames.

We ensure that the tubelet tokens and image tokens have the same dimensions of $N \times C$, where N is the number of tokens and C is the number of channel embeddings. To achieve this, we max-pool image tokens to match the dimensions of tubelet tokens.

3.3. Bottleneck

The bottleneck of both the branches consist of mixer networks with the same basic design. The mixer network takes in tokens T as input and processes them using two different multi-layer perceptrons (MLPs). First, the input tokens are normalized and then mixed across the token dimension. The process can be summarized as:

$$T_{TM} = MLP_{TM}(LN(T_{in})) + T_{in}, \quad (1)$$

where T_{TM} represents the tokens extracted after Token Mixing (TM), T_{in} represents the input tokens, and LN represents layer normalization [2]. Note that there is also a skip connection between the input to the mixer and the output from token mixing MLP. Token mixing encodes the relationship between individual tokens. Afterwards, the tokens are flipped along the C axis and fed into another MLP to learn dependencies in the C dimension [76]. This is called channel mixing and is formulated as follows:

$$T_{out} = MLP_{CM}(LN(T_{TM})) + T_{TM}, \quad (2)$$

where T_{out} represents the output tokens and CM denotes channel mixing. The MLP block comprises of two linear layers that are activated by GeLU [25]. The initial linear layer converts the number of tokens/channels into an embedding dimension, while the second linear layer brings them back to their original dimension. The selection of the embedding dimension and the number of mixer blocks for the bottleneck is done through hyperparameter tuning.

3.4. Recurrent training

Recurrent setups generally refer to a type of configuration or arrangement that is repeated or ongoing [64]. In real-time video enhancement, the original video stream has to be enhanced on-the-fly which means we have the information of all the enhanced frame till the current time instance. The enhanced frames from the previous time step has valuable information that could be leveraged for the current prediction for increased efficiency. Leveraging previous frame prediction can also help in increasing the temporal stability

of the predictions as the current predictions gets conditioned on the previous predictions. Although it is possible to use multiple previous frames in a recurrent setup, we have chosen to only use the most recent prediction for the sake of efficiency. An overview of this setup is illustrated in Fig 3.

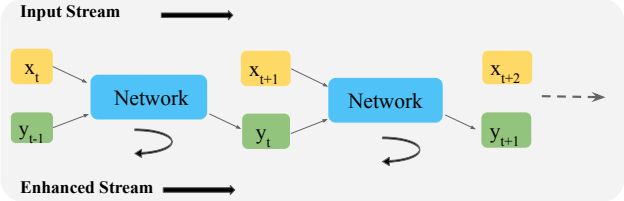


Figure 3: Overview of the proposed recurrent setup where x_t is the current input frame, y_{t-1} is the previous prediction, and y_t represents the current prediction.

In the following, we elucidate how we leverage the recurrent setup to output the enhanced frame. Let us define the original input stream as $X = \{x_0, x_1, ..., x_t\}$ where X denotes the video and x denotes the individual frames. The frames start from the initial frame x_0 to the current time frame x_t. Similarly, we also define the enhanced video stream represented as $Y = \{y_0, y_1, ..., y_{t-1}\}$ where Y denotes the enhanced video stream and y denotes the individual enhanced frames. These enhanced frames go from the initial time step y_0 to the previous time frame y_{t-1}. So, to find the enhanced prediction of the current frame y_t, we make use of current degraded frame x_t and the previous enhanced frame y_{t-1}. These images are sent to the network to output y_t. In the context of training, a single feed forward step involves using the input values x_t and y_{t-1} to make a prediction for the output value y_t. When processing a video, multiple feed forward steps are used in a sequential manner to predict the output values for all frames in the video. Similarly, during backpropagation, the gradients are propagated backwards through the network, starting from the last frame and moving towards the first frame of the video. Note that there is a corner case for the first frame while predicting y_0. To circumvent it, we use the ground truth frame as the initial prediction to kick-start the training.

4. Experiments and Results

4.1. Datasets

We note that there exists several video super-resolution, deblurring, and denoising datasets like REDS [49], DVD [69], GoPro [50], DAVIS [35], Set8 [72], etc. However, these datasets focus on just one degradation at a time like deblurring or denoising. As we focus on the problem of generic video enhancement of live videos, presence of multiple degradations is very common. Also, a major use case for real-time video enhancement is video conferencing where the video actually contains the torso/face of the person. To reflect these real-world scenarios, we curate two datasets for the task of video enhancement: i) PortraitVideo and ii) FullVideo.

PortraitVideo: We build PortraitVideo on top of TalkingHeads [88] which is a public dataset that uses Youtube videos and processes them using face detectors to obtain just the face. Here, the frame is fixed allowing only the movement of the head to be captured, which simulates a scenario where the camera is fixed during video calls. The face region is then is cropped similar to face image datasets like FFHQ [32]. Also, we note that TalkingHeads consists of a lot of non-human faces like cartoons and avatars as well. Further, a lot of videos are of very low quality and hence unsuitable for training or evaluation of video restoration. So, we curate PortraitVideo by skipping low quality videos and pick 113 face videos for training and 20 face videos for testing. We fix the resolution of the faces to 384×384. The videos are processed at 30 frames per second (FPS) with a total of 150 frames per video. We use a mixture of degradations like blur with varying kernels, compression artifacts, noise, small distortions in brightness, contrast, hue, and saturation. The exact details of these degradations can be found in the supplement.

FullVideo: We develop this dataset using high quality videos collected from Youtube videos. The video IDs are taken from TalkingHeads dataset however we do not use any of the pre-processing techniques from the TalkingHeads dataset so that the original information of the scene is maintained. We also manually filter to keep only high quality videos. There are 132 training videos and 20 testing videos, and all videos are 720×1280, 30 FPS and 128 frames long. We apply similar degradations as PortraitVideo for this dataset. The major difference is that this dataset is of a higher resolution and captures more context around the face, including the speaker's body and the rest of the scene.

4.2. Implementation Details

We prototype our method using PyTorch on NVIDIA A100 GPU cluster. ReBotNet is trained with a learning rate of $4e^{-4}$ using Adam optimizer, and a cosine annealing learning rate scheduler with a minimum learning rate of $1e^{-7}$. The training is parallelized across 8 NVIDIA A100 GPUs, with each GPU processing a single video. The model is trained for 500,000 iterations. For fair comparison with existing methods, we only use the commonly used Charbonnier loss [3] to train all models. More configuration details of the architecture can be found in the supplementary.

4.3. Comparison with previous works

We compare ReBotNet against multiple recent methods. Recurrent Video Restoration Transformer (RVRT) [40] is the current SOTA method across many tasks like deblurring, denoising, super-resolution, and video-frame interpo-

Table 1: Comparison of quantitative results of ReBotNet with previous methods. † represents the default configuration from paper and public code. S, M, L represent the small (∼10G), medium (∼50G), and large (∼400G) FLOPs regimes.

Method	GFLOPs (↓)	Latency (in ms) (↓)	Param (in M) (↓)	PortraitVideo		FullVideo	
				PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)
FastDVDNet † [73]	367.81	36.23	1.12	28.88	0.8516	29.56	0.8577
VRT † [38]	2054.32	781.15	19.62	31.70	0.8835	33.49	0.9140
BasicVSR++ † [9]	157.53	49.55	9.3	31.26	0.8739	33.10	0.9078
RVRT † [40]	396.29	52.30	13.57	31.92	0.8870	33.79	0.9191
FastDVDNet (S) [73]	15.85	30.51	**0.46**	27.97	0.8384	28.16	0.8459
VRT (S) [38]	15.22	48.73	2.45	30.80	0.8681	31.85	0.8901
BasicVSR++ (S) [9]	19.05	29.08	1.76	30.90	0.8705	32.78	0.8950
RVRT (S) [40]	-	-	-	-	-	-	-
RebotNet (S)	**13.02**	**13.15**	3.8	**31.25**	**0.8778**	**33.45**	**0.9113**
FastDVDNet (M) [73]	64.51	33.89	**0.68**	28.52	0.8405	29.35	0.8528
VRT (M) [38]	60.18	58.89	3.99	30.98	0.8701	32.35	0.8987
BasicVSR++ (M) [9]	60.93	41.18	4.29	31.19	0.8729	33.04	0.9051
RVRT (M) [40]	62.42	35.93	2.66	31.60	0.8821	**33.59**	0.9145
RebotNet (M)	**56.06**	**15.02**	6.86	**31.85**	**0.8865**	33.45	**0.9168**
FastDVDNet (L) [73]	416.90	37.14	**1.19**	28.93	0.8537	29.68	0.8593
VRT (L) [38]	419.32	91.74	20.96	31.09	0.8729	32.68	0.9014
BasicVSR++ (L) [9]	403.22	73.32	24.55	31.40	0.8775	33.31	0.9126
RVRT (L) [40]	396.29	52.30	13.57	31.92	0.8870	**33.79**	0.9191
RebotNet (L)	**363.76**	**19.98**	41.3	**32.13**	**0.8902**	33.65	**0.9199**

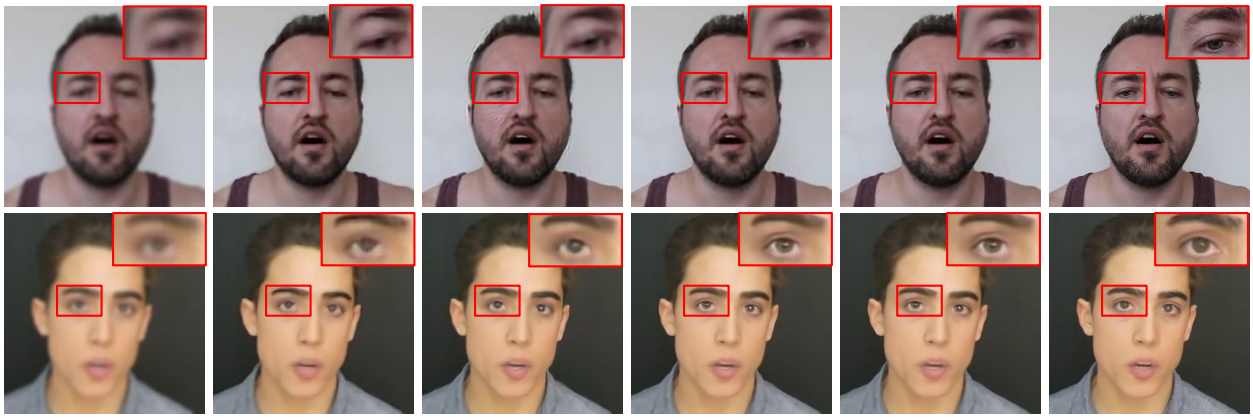
lation. We also compare against Video Restoration Transformer (VRT) [38], the SOTA convolution-based video super-resolution method BasicVSR++ [10], and the fastest deblurring method FastDVD for fair comparison. We retrain all these methods on the new datasets PortraitVideo and FullVideo using their publicly available code.

Table 2: Comparison of ReBotNet with previous methods on public datasets. Numbers correspond to PSNR / SSIM.

Method	DVD [69]	GoPro [50]
DeepDeblur [50]	29.85 / 0.8800	38.23 / 0.9162
EDVR [89]	31.82 / 0.9160	31.54 / 0.9260
TSP [52]	32.13 / 0.9268	31.67 / 0.9279
PVDNet [66]	32.31 / 0.9260	31.98 / 0.9280
VRT [38]	34.24 / 0.9651	34.81 / 0.9724
RVRT [40]	34.30 / 0.9655	34.92 / 0.9738
ReBotNet	34.28/ 0.9656	34.90 / 0.9734

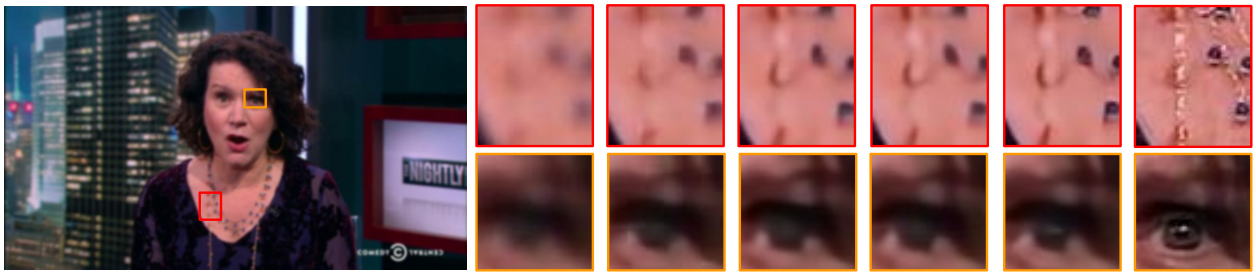
Initially, we conducted experiments on the new datasets PortraitVideo and FullVideo using the default configurations of VRT, RVRT, BasicVSR++, and FastDVD, as provided in their publicly available code as seen in the first few rows of Table 1. It is important to mention that these models have different levels of floating-point operations (FLOPs). Therefore, to ensure a fair comparison, we assessed the performance of ReBotNet in different FLOP regimes in comparison to the previous methods. This approach helped us gain a comprehensive understanding of the performance of these models across different levels of FLOPs. We pick the embedding dimension across different levels of the network as the hyper-parameter to change the FLOPs [37]. We acquire different configurations of FLOPs regimes of Small (10Gs), Medium (50Gs), and Large (400Gs). The exact configuration details can be found in the supplementary material. Note that RVRT does not have a S configuration as it is infeasible to scale down the model near 10 GFLOPs due to its inherent design. It should also be noted that for each configuration, we ensured that the computational complexity of ReBotNet remained lower than that of the other models being compared. To provide an example, when evaluating models in the medium regime, we compared ReBotNet, which had a complexity of 56.06, with VRT, which had a complexity of 60.18, and RVRT, which had a complexity of 62.42. In all of our experiments, we used a consistent number of frames, which was set to 2 for all models except for FastDVD, which was designed to process 5 frames. To evaluate the models, we compute the PSNR and SSIM for each individual frame of a video, and then average these values across all frames within the video. We then calculated the mean PSNR and SSIM across all videos and present these results in Table 1. We use the high-quality frame as the first frame for all these methods while performing the inference. Additionally, we measure the inference time for each method by forwarding 2 frames of dimensions $(384, 384)$ through the network. To obtain the latency, we perform GPU warm-up for 10 iterations and then feed-forward the clip 1000 times, reporting the average. The latency was recorded on a NVIDIA A100 GPU. We also report the number of parameters for each model.

Table 1 demonstrates that our method outperforms most previous approaches in terms of PSNR and SSIM, while using less computational resources across most regimes for both datasets. A significant advantage of our model is its fast inference speed, which is 2.5x faster than the previous best performing method, RVRT. We also note that we get better results than the original implementations which have



Input VRT BasicVSR++ RVRT ReBotNet GT

Figure 4: Qualitative Results on *PortraitVideo* dataset. Please zoom in for better visualization.



Degraded Frame Input VRT BasicVSR++ RVRT ReBotNet GT

Figure 5: Qualitative Results on *FullVideo* dataset. Please zoom in for better visualization.

way more computations (as seen in first few rows of Table 1) . The efficiency of ReBotNet comes because of its effective design while also employing token mixing mechanisms by using mixers. The main contribution towards computation in transformer-based methods like RVRT and VRT come from the self-attention mechanism acting at original scale of the image. Note that we do not use self-attention but replace it with a careful design choice that matches (or even exceeds) its performance. We also conduct experiments on single degradation public datasets like DVD, GoPro and report the results in Table 2 . For this, we use ReBotNet (L) and compare against the default configurations of previous methods. It can be observed that we obtain a competitive performance in spite of low latency of our model, which can be already seen in Table 1.

In Figures 4 and 5, we present qualitative results from PortraitVideo and FullVideo dataset. It can be observed that our method is better than previous methods in terms of quality. The enhanced details are much visible in ReBotnet when compared to other methods. The results are taken from the medium configurations of each model. More results can be found in the supplementary material.

4.4. User Study

To validate the perceptual superiority of ReBotNet for video enhancement, we conducted a user study on the M

Table 3: User study results on PortraitVideo dataset.

Method	Preference for ReBotNet	95% Confidence Interval
FastDVDNet	+ 1.83	0.059
VRT	+ 1.61	0.088
BasicVSR++	+ 1.63	0.105
RVRT	+ 0.08	0.073

configuration models on PortraitVideo dataset. We compare our approach to each competing method in a one-to-one comparison. We recruited 3 experts with technical experience in conference video streaming and image enhancement. Each expert evaluated on average 80 video comparisons across four baseline methods. For each comparison, we showed output videos of our method and one competing method, played both videos simultaneously and asked the user to rate which video had a higher quality with the corresponding scores (”much worse”, -2), (”worse”, 1), (”same”, 0), (”better”, 1) and (”much better”, 2). We calculated the mean score and 95% confidence intervals for paired samples and report them in Table 3. The user study demonstrates the superiorty of our method. Despite RVRT being the closest second, our method is still preferred over it while also being more efficient and faster.

5. Discussion

Analysis on ReBotNet: In order to elucidate our design decisions for ReBotNet, we carry out a set of experiments using various parameter configurations, which af-

Table 4: Analysis on the (a) number of embedding dimension in Mixer (b) depth of the botteneck (c) number of frames taken.

Embedding	PSNR (↑)	SSIM (↑)	GFLOPs (↓)	Latency (↓)
128	31.79	0.8851	55.50	14.85
256	31.85	0.8865	56.06	15.02
512	31.90	0.8869	56.60	15.27
728	31.89	0.8869	57.14	15.36

(a)

Depth	PSNR (↑)	SSIM (↑)	GFLOPs (↓)	Latency (↓)
2	31.83	0.8864	55.50	14.67
4	31.85	0.8865	56.06	15.02
6	31.87	0.8866	57.14	15.31
8	31.81	0.8861	58.08	16.34

(b)

Frames	PSNR (↑)	SSIM (↑)	GFLOPs (↓)	Latency (↓)
1	29.56	0.8586	55.50	14.85
2	31.85	0.8865	56.06	15.02
3	31.88	0.8871	57.14	15.16
4	31.92	0.8874	58.08	15.40

(c)

fect both the performance and computational aspects of the model. These experiments are conducted on the PortraitVideo dataset, using ReBotNet (M) as the base configuration. Table 4 illustrates these results where gray rows correspond to the configuration of the actual implementation of ReBotNet (M). We analyze the performance along with computation and latency on different configurations of embedding dimension in Mixer (Table 4.a), depth of the bottleneck (Table 4.b), and the number of frames (Table 4.c).

Ablation Study: In order to investigate the contribution of each component proposed in the work, we conduct an ablation study using the PortraitVideo dataset. The results of these experiments are shown in Table 5. First, we use the Tubelet tokens extracted from spatio-temporal branch where we use ConvNext encoder directly with a decoder to get the prediction. Then, we consider a configuration where we use image tokens extracted using linear layers from the spatial branch directly forwarded to decoder to get the prediction. This configuration obtains the best latency however suffers from a significant drop in performance. Next, we fuse features extracted from both these branches and use the common decoder. This shows a relative improvement in terms of performance without much addition in computation. Note that here the FLOPs of fused configuration is not direct addition between FLOPs of tubulet tokens and image tokens as the decoder's computation was common in both the previous setups. Next, we add the bottleneck mixers which obtains an improvement in performance with little increase in compute. Finally, we add the recurrent training setup which adds no increase in compute but improves the performance. Our findings indicate that each individual component in ReBotNet plays a vital role.

Table 5: Ablation study on PortraitVideo dataset.

Tub. Tok.	Img Tok.	Bot. Mix.	Rec. Setup	PSNR (↑)	SSIM (↑)	GFLOPs (↓)	Latency (↓)
✓	×	×	×	31.24	0.8768	54.94	14.27
×	✓	×	×	28.01	0.8295	41.94	5.63
✓	✓	×	×	31.41	0.8792	55.50	14.67
✓	✓	✓	×	31.59	0.8822	56.06	15.02
✓	✓	✓	✓	31.85	0.8865	56.06	15.02

FPS and Peak Memory Usage: In Figure 6, we provide a comparison of ReBotNet's frames per second (FPS) rate and peak memory usage with previous methods. For this analysis, we consider feed forward of 2 frames of resolution 384×384 and consider RebotNet (L) configuration with original implementations for the previous methods. It can be observed that our method has a FPS that is real-time while also not occupying much memory. We note that 30 FPS is considered real-time for applications like video conferencing. Also, ReBotNet has one of the least memory requirements compared to other methods due to its efficient design and implementation.

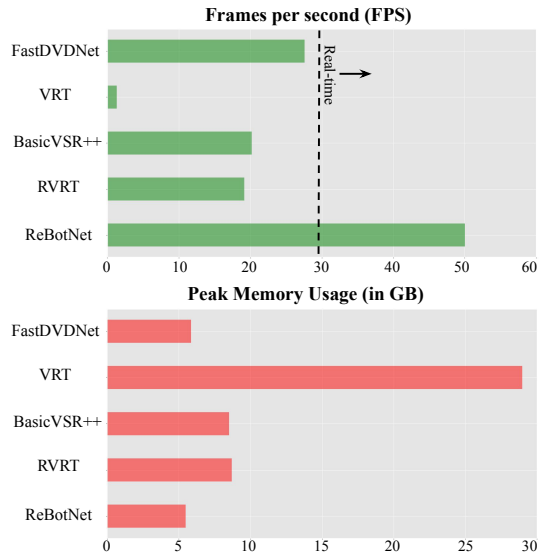


Figure 6: Comparison chart of ReBotNet (L) against default configurations of previous methods for Frames Per Second (FPS) and Peak Memory Usage (in GB), as measured on NVIDIA A100 GPU for $2 \times 3 \times 384 \times 384$ resolution.

Limitations: Our method is not ideal in terms of the number of parameters. This is not a concern for applications such as video calls or live streams, where processing is usually performed in the cloud. However, if the method is to be used for edge applications, it is necessary to optimize the number of parameters. To focus on improvement due to the proposed architecture alone, we only used Charbonnier loss to train all models. Additional losses like the perceptual loss [30] can be applied to further improve the results.

6. Conclusion

In this paper, we proposed a novel approach for real-time video enhancement by proposing a new framework: Recurrent bottleneck mixer network (ReBotNet). ReBotNet combines the advantages of both recurrent setup and bottleneck models, allowing it to effectively capture temporal dependencies in the video while reducing the computational complexity and memory requirements. We evaluated the performance of ReBotNet on multiple video enhancement datasets. The results showed that our proposed method outperformed state-of-the-art methods in terms computational efficiency while matching or outperforming them in terms

of visual quality.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[4] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4778–4787, 2017.

[5] Jiezhang Cao, Jingyun Liang, Kai Zhang, Wenguan Wang, Qin Wang, Yulun Zhang, Hao Tang, and Luc Van Gool. Towards interpretable video super-resolution via alternating optimization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 393–411. Springer, 2022.

[6] Jiezhang Cao, Qin Wang, Yongqin Xian, Yawei Li, Bingbing Ni, Zhiming Pi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. *arXiv preprint arXiv:2212.04362*, 2022.

[7] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4956, 2021.

[8] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 973–981, 2021.

[9] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981, 2022.

[10] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5962–5971, 2022.

[11] Shih-Fu Chang, Di Zhong, and Raj Kumar. Real-time content-based adaptive streaming of sports videos. In *Proceedings IEEE workshop on content-based access of image and video libraries (CBAIVL 2001)*, pages 139–146. IEEE, 2001.

[12] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021.

[13] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[14] Dandan Ding, Junchao Tong, and Lingyi Kong. A deep learning approach for quality enhancement of surveillance video. *Journal of Intelligent Transportation Systems*, 24(3):304–314, 2020.

[15] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.

[16] Nicola Döring, Katrien De Moor, Markus Fiedler, Katrin Schoenenberg, and Alexander Raake. Videoconference fatigue: A conceptual analysis. *International journal of environmental research and public health*, 19(4):2061, 2022.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Michael Elad, Bahjat Kawar, and Gregory Vaksman. Image denoising: The deep learning revolution and beyond–a survey paper–. *arXiv preprint arXiv:2301.03362*, 2023.

[19] Yuchen Fan, Honghui Shi, Jiahui Yu, Ding Liu, Wei Han, Haichao Yu, Zhangyang Wang, Xinchao Wang, and Thomas S Huang. Balanced two-stage residual networks for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 161–168, 2017.

[20] Yuchen Fan, Jiahui Yu, Ding Liu, and Thomas S Huang. Scale-wise convolution for image restoration. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10770–10777, 2020.

[21] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3476–3485. IEEE, 2019.

[22] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5407–5416, 2020.

[23] Jungong Han, Dirk Farin, and Peter HN de With. A real-time augmented-reality system for sports broadcast video

enhancement. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 337–340, 2007.

[24] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3897–3906, 2019.

[25] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[26] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. *Advances in neural information processing systems*, 28, 2015.

[27] Satoshi Iizuka and Edgar Simo-Serra. Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[28] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8008–8017, 2020.

[29] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3224–3232, 2018.

[30] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[31] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE transactions on computational imaging*, 2(2):109–122, 2016.

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[33] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[34] Efklidis Katsaros, Piotr K Ostrowski, Krzysztof Włódarczak, Emilia Lewandowska, Jacek Ruminski, Damian Siupka-Mróz, Lukasz Lassmann, Anna Jezierska, and Daniel Wkesierski. Multi-task video enhancement for dental interventions. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VII*, pages 177–187. Springer, 2022.

[35] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14*, pages 123–141. Springer, 2019.

[36] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 106–122, 2018.

[37] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16020–16030, 2021.

[38] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022.

[39] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.

[40] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. *arXiv preprint arXiv:2206.02146*, 2022.

[41] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2013.

[42] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2507–2515, 2017.

[43] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

[44] Hongying Liu, Zhubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu, Linlin Yang, and Radu Timofte. Video super-resolution based on deep learning: a comprehensive survey. *Artificial Intelligence Review*, 55(8):5981–6035, 2022.

[45] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[46] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[47] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[48] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point

cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

[49] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[50] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[51] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8102–8111, 2019.

[52] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3043–3051, 2020.

[53] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 109–125. Springer, 2020.

[54] Naama Pearl, Tali Treibitz, and Simon Korman. Nan: Noise-aware nerfs for burst-denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12672–12681, 2022.

[55] Malsha V Perera, Wele Gedara Chaminda Bandara, Jeya Maria Jose Valanarasu, and Vishal M Patel. Transformer-based sar image despeckling. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 751–754. IEEE, 2022.

[56] Shengju Qian, Yi Zhu, Wenbo Li, Mu Li, and Jiaya Jia. What makes for good tokenizers in vision transformer? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[57] Qin Qin, Jingke Yan, Qin Wang, Xin Wang, Minyao Li, and Yuqing Wang. Etdnet: An efficient transformer deraining model. *IEEE Access*, 9:119881–119893, 2021.

[58] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, and Tao Mei. Mlp-3d: A mlp-like 3d architecture with grouped time mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3062–3072, 2022.

[59] Arathy Rajan and VP Binu. Enhancement and security in surveillance video system. In *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, pages 1–5. IEEE, 2016.

[60] Yunbo Rao and Leiting Chen. A survey of video enhancement techniques. *J. Inf. Hiding Multim. Signal Process.*, 3(1):71–99, 2012.

[61] Siddhant Sahu, Manoj Kumar Lenka, and Pankaj Kumar Sa. Blind deblurring using deep learning: A survey. *arXiv preprint arXiv:1907.10128*, 2019.

[62] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6626–6634, 2018.

[63] Wei-wei Shen, Lin Chen, Shuai Liu, and Yu-Dong Zhang. An image enhancement algorithm of video surveillance scene based on deep learning. *IET Image Processing*, 16(3):681–690, 2022.

[64] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[65] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *arXiv preprint arXiv:2207.08494*, 2022.

[66] Hyeongseok Son, Junyong Lee, Jonghyeop Lee, Sunghyun Cho, and Seungyong Lee. Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. *ACM Transactions on Graphics (TOG)*, 40(5):1–18, 2021.

[67] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021.

[68] Paul F Stetson, F Graham Sommer, and Albert Macovski. Lesion contrast enhancement in medical ultrasound imaging. *IEEE transactions on medical imaging*, 16(4):416–425, 1997.

[69] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1279–1288, 2017.

[70] Fuxiang Tan, YuTing Kong, Yingying Fan, Feng Liu, Daxin Zhou, Long Chen, Liang Gao, Yurong Qian, et al. Sdnet: mutil-branch for single image deraining using swin. *arXiv preprint arXiv:2105.15077*, 2021.

[71] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 4472–4480, 2017.

[72] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809. IEEE, 2019.

[73] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1354–1363, 2020.

[74] Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan. Deep learning for image denoising: A survey. In *Genetic and Evolutionary Computing: Proceedings of the Twelfth International Conference on Genetic and Evolutionary Com-*

puting, December 14-17, Changzhou, Jiangsu, China 12, pages 563–572. Springer, 2019.

[75] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3360–3369, 2020.

[76] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

[77] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[78] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022.

[79] Jeya Maria Jose Valanarasu and Vishal M Patel. Unext: Mlp-based rapid medical image segmentation network. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V*, pages 23–33. Springer, 2022.

[80] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022.

[81] Reid Vassallo, Hidetoshi Kasuya, Benjamin WY Lo, Terry Peters, and Yiming Xiao. Augmented reality guidance in cerebrovascular surgery using microscopic video enhancement. *Healthcare technology letters*, 5(5):158–161, 2018.

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[83] Ziyu Wan, Bo Zhang, Dongdong Chen, and Jing Liao. Bringing old films back to life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17694–17703, 2022.

[84] Guangting Wang, Yucheng Zhao, Chuanxin Tang, Chong Luo, and Wenjun Zeng. When shift operation meets vision transformer: An extremely simple alternative to attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2423–2430, 2022.

[85] Hai Wang, Xiaoyu Xiang, Yapeng Tian, Wenming Yang, and Qingmin Liao. Stdan: deformable attention network for space-time video super-resolution. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[86] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[87] Tianhang Wang and Lu Zhao. Virtual reality-based digital restoration methods and applications for ancient buildings. *Journal of Mathematics*, 2022, 2022.

[88] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021.

[89] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[90] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

[91] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021.

[92] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[93] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.

[94] Ren Yang. Ntire 2021 challenge on quality enhancement of compressed video: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–666, 2021.

[95] Xi Yang, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4781–4790, 2021.

[96] Rajeev Yasarla, Jeya Maria Jose Valanarasu, and Vishal M Patel. Exploring overcomplete representations for single image deraining using cnns. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):229–239, 2020.

[97] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, Tao Lu, Xin Tian, and Jiayi Ma. Omniscient video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4429–4438, 2021.

[98] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3106–3115, 2019.

[99] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S2-mlp: Spatial-shift mlp architecture for vision. In *Pro-*

ceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 297–306, 2022.

[100] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.

[101] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11799–11808, 2022.

[102] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9):2103–2130, 2022.

[103] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3262–3271, 2018.

[104] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.

[105] Yinjie Zhang, Yuanxing Zhang, Yi Wu, Yu Tao, Kaigui Bian, Pan Zhou, Lingyang Song, and Hu Tuo. Improving quality of experience by adaptive video streaming with super-resolution. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1957–1966. IEEE, 2020.

[106] Dong Zhao, Jia Li, Hongyu Li, and Long Xu. Hybrid local-global transformer for image dehazing. *arXiv preprint arXiv:2109.07100*, 2021.

[107] Ling Zhao. Motion track enhancement method of sports video image based on otsu algorithm. *Wireless Communications and Mobile Computing*, 2022, 2022.

[108] Zhihang Zhong, Mingdeng Cao, Xiang Ji, Yinqiang Zheng, and Imari Sato. Blur interpolation transformer for real-world motion from blur. *arXiv preprint arXiv:2211.11423*, 2022.

[109] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 191–207. Springer, 2020.

[110] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2482–2491, 2019.

[111] Qiang Zhu, Haoyu Zhang, Shuyuan Zhu, Guanghui Liu, Bing Zeng, and Xiaozhen Zheng. Deep video super-resolution with flow-guided deformable alignment and sparsity-based temporal-spatial enhancement. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2022.

A. Configurations of ReBotNet

In the main paper, we mentioned we conducted experiments with different FLOPs regimes for all the methods. We did that by controlling the depth of the bottleneck and the embedding dimension of different methods to get the required FLOPs. In Tables 6, 7, and 8 we provide the exact configurations of ReBotNet - S,M, and L respectively. More analysis on the dependence of these parameters were provided in the main paper.

Table 6: Configuration of ReBotNet-S.

Block	Type	Value
Branch I	Number of Layers	4
	Depths per layer	[4,4,4,4]
	Embedding dimensions	[28,36,48,64]
Branch II	Patch size	1
	Embedding Dimension	256
	Depth	4
Bottleneck	Input Dimension	64
	Hidden Dimension	728

Table 7: Configuration of ReBotNet-M.

Block	Type	Value
Branch I	Number of Layers	4
	Depths per layer	[4,4,4,4]
	Embedding dimensions	[64,80,108,116]
Branch II	Patch size	1
	Embedding Dimension	256
	Depth	4
Bottleneck	Input Dimension	116
	Hidden Dimension	728

Table 8: Configuration of ReBotNet-L.

Block	Type	Value
Branch I	Number of Layers	4
	Depths per layer	[5,5,5,4]
	Embedding dimensions	[172,180,188,196]
Branch II	Patch size	1
	Embedding Dimension	256
	Depth	4
Bottleneck	Input Dimension	64
	Hidden Dimension	728

B. Configuration of Baselines

We used the publicly available codes for the original implementations of FastDVDNet, BasicVSR++, VRT, and RVRT; the results of which can be seen in Table 1 of the

Table 9: Configurations of VRT.

Method	Embedding Dimension
VRT - S	[24,24,24,24,24,24,24,24,24,24]
VRT - M	[48,48,48,48,48,48,48,48,48,48]
VRT - L	[180,180,180,180,180,180,120,120,120,120]
VRT - OG	[180,180,180,180,180,180,120,120,120,120,120,120,120]

Table 10: Configurations of RVRT.

Method	Embedding Dimension
RVRT - S	-
RVRT - M	[36,36,36]
RVRT - L	[192,192,192]
RVRT - OG	[192,192,192]

Table 11: Configurations of FastDVDNet.

Method	Embedding Dimension
FastDVDNet - S	[32, 48, 72, 96]
FastDVDNet - M	[64, 80, 108, 116]
FastDVDNet - L	[96, 112, 132, 144]
FastDVDNet - OG	[80, 96, 132, 144]

Table 12: Experiment on pure mixers.

Method	PSNR	SSIM	GFLOPs	FPS
VRT	34.24	0.9651	2054.32	1
VRT (Mixers)	32.14	0.9429	1495.06	2

main paper. For the S,M and L configurations we use the same configurations of the original implementations but change the embedding dimensions. These changes have been illustrated in Tables 9, 10, and 11. OG means the original implementation. Note that RVRT does not have a S configuration as even with embedding dimensions of $[1, 1, 1]$, the FLOPs does not hit the range of 10 GFLOPs.

C. Experiments on Pure Mixers

We observed that MLP-Mixers tend to exhibit a noticeable decline in quality when applied directly for video enhancement compared to transformer-based approaches. Using Mixers directly on large size images still takes a lot of compute and makes it difficult to achieve real-time speed. In Table 12, we conduct an experiment where we take VRT as the base network and convert all the transformer blocks in it to MLP-Mixers. The experiment is conducted on the DVD dataset. It can be observed that the although the computation reduces, the performance also drops significantly. And still the computation is far away from obtaining a real-time FPS. This motivates us to work towards our design of ReBotNet as seen in the main paper.

D. More Qualitative results

In Figures 7 and 8, we provide more qualitative results on PortraitVideo and FullVideo datasets respectively.

E. Degradations

In Table 13, we provide the detailed configurations of degradations that we use in PortraitVideo and FullVideo dataset. In all the rows where there is a range, we choose a random value in the range. To get the final degradation of a sample image at hand, we choose a random combination of the degradations from Table 13. These values were decided to emulate degradations possible in real-world and after consulting experts working in the field of video conferencing.

Table 13: Degradations used in PortraitVideo and FullVideo datasets.

Type of Degradation	Value
Eye Enlarge ratio	1.4
Blur kernel size	15
Kernel Isotropic Probability	0.5
Blur Sigma	[0.1,3]
Downsampling range	[0.8,2.5]
Noise amplitude	[0,0.1]
Compression Quality	[70,100]
Brightness	[0.8,1.1]
Contrast	[0.8,1.1]
Saturation	[0.8,1.1]
Hue	[-0.05,0.05]

F. Reasons behind design choices in Branch I

We pick ConvNext blocks over basic ConvNet blocks as it has been shown that they are both efficient and effective than ConvNets. Each ConvNext block first consists of a depth-wise convolution block with kernel size of 7×7, stride 1 and padding 3. Using a large kernel size is to have a larger receptive field similar to non-local attention. It was observed in [47] that the benefit of larger kernel sizes reaches a saturation point at at 7×7. It is followed by a layer normalization and a point-wise convolution function. The point-wise convolution is basically a convolution layer with kernel size 1×1. The output of this is activated using GeLU activation and then forwarded to another point-wise convolution to get the output. More details of this why this exact setup is followed can be found in [47]. We also have downsampling blocks after each level in the ConvNext encoder. The number of ConvNext blocks is a hyperparameter. However, for simplicity we fixed the number of total levels as 4 which means the downsampling is done only 4 times throughout the encoder.

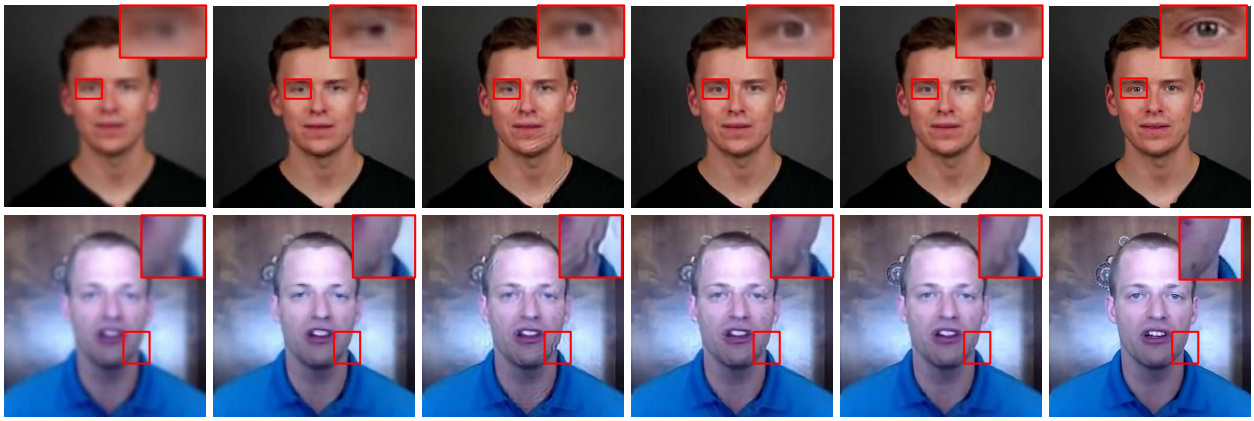


Figure 7: Qualitative Results on *PortraitVideo* dataset. Please zoom in for better visualization.

Figure 8: Qualitative Results on *FullVideo* dataset. Please zoom in for better visualization.