Evaluating Video Frame Interpolation and Audio Guidance Models With Distanced Frame Inputs

Kyung Myung Ko
Purdue Univdersity
ko112@purdue.edu

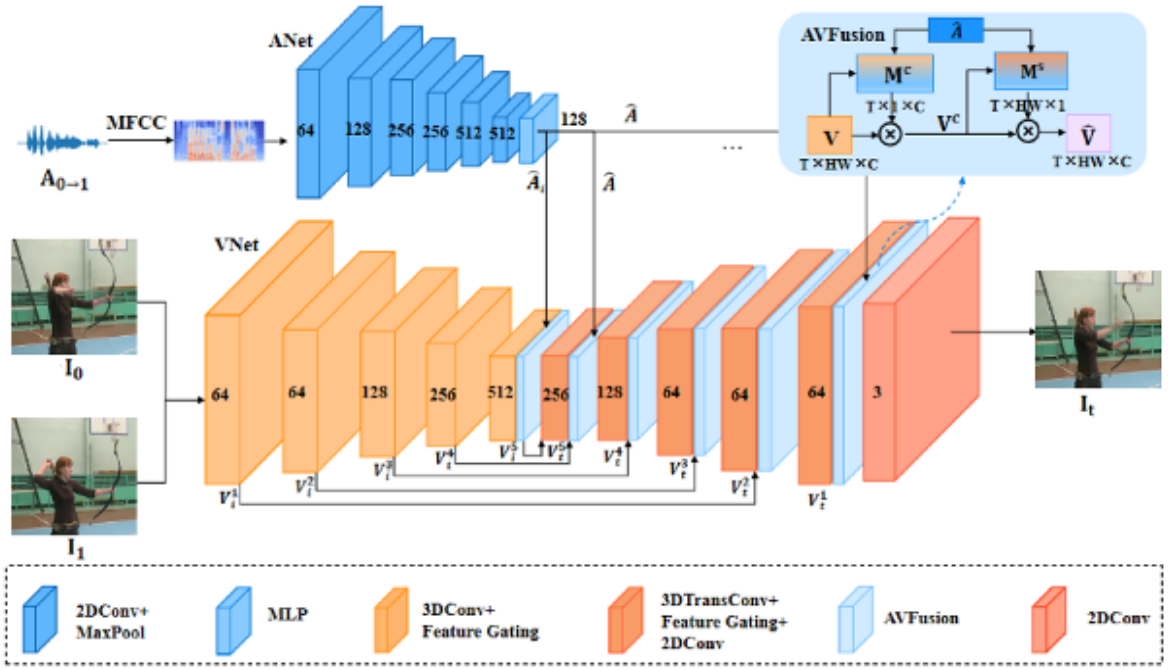
Abstract

Previous works have shown the effectiveness of video enhancement from Video Frame Interpolation (VFI). While the progress have been promising, it suffers from the precondition of requiring many number of frames. This problem setting concerns whether this could be applied in a setting with less number of frames to start with. Individual frames are the major components of the video stream that leads to memory overhead. With continuous accessibility to higher-resolution frames, there must be the need to reduce this memory overhead. Still, modern videos also contain audio as part of its stream due to its multimodality. This becomes a strong cue to leverage information in the visual stream, while maintaining relative reasonable file size. In this work, we investigate the potential to leverage audio stream for the variation of video frame interpolation task. As our goal is to utilize audio to fill in the information gap that arises from less number of frames, we frame it Audio-Guided Video Frame Interpolation (AGVFI). We first test our hypothesis by obtaining the results from the state-of-the-art video frame interpolation framework tailored to our problem setting. We then evaluate our baseline performance with 3D-UNet multimodal architecture that can handle both video and audio stream as input. We follow the evaluation methods proposed in the video frame interpolation task, but we also propose possible strategies that can better evaluate the approach for the problem we are interested in.

1. Introduction

Videos are expensive, requiring substantial storage and bandwidth due to the costly video stream data. A 6-second video clip of (320×240) resolution of an action sequence takes up 400 KB in size when encoded with H.264. This is the output of compressing 160 frames of 22 MB and audio stream of 1 MB. As higher-resolution frames become more accessible, the need for better codec methods are needed. Modern advanced codecs such as H.264, HEVC, and AV1 have made great progress in reducing video file sizes while preserving the visual quality of the contents for human vision [2, 35, 40]. Yet, they present limitations including quality degradation at high compression and increased file size with higher resolutions. Major bottleneck that prevents efficiently compressing the video files arise from the need to save numerous number of frames. More number of frames in the video will increase the FPS, which makes the transition between the motions become more clear. Video Frame Interpolation (VFI) has progressed over the past years as to apply its contribution to video enhancements. Recent works have shown promising results in this application [18, 33]. However, it is questionable whether this approach can solve the problem of interpolation between sparse frames, as the assumption to this task is the access to frames that are close to each other in the time step. Specifically, the problem setting can be formulated as, given a high FPS video, interpolate the in-between frames to increase FPS even higher. This assumption and formulation contradicts the setting we are proposing. This raises us the question: can the video files be saved with only a subset of frames from the video stream data and generate the missing information at the time it is played, while preserving the reasonable amount of information?

Modern video frame interpolation task takes the opposite approach to the question we are proposing, generating the in-between missing frames for natural flow of visual information. Inherently, interpolating frames accurately across dynamic transitions poses a challenge from the necessity of sufficient contextual frames. Taking a closer look at the structure of the video file, there is more than just the frames. The audio stream, though relatively small in file size, carries meaningful temporal and contextual cues that assist humans to better understand the continuous video stream information. This leads us to explore an alternative method to answer our original question. Essentially, we want to leverage audio to replace the necessity of sufficient number of frames for video interpolation. We frame this task as audio-guided video frame interpolation.



(a) An example of a subfigure.

Figure 1. The overall framework of the ASVFI. The audio encoder(ANet) extracts the audio feature to perform AVFusion from the last layer of the video decoder(VNet), modified from the ASVFI architecture.

In this work, we investigate the potential of the audio-guided video frame interpolation method and establish a baseline performance with the model architecture that matches our problem setting. To this end, we attempt to prove our hypothesis: the modern VFI methodologies are not tailored to AGVFI task. To do this, we leverage the state-of-the-art video interpolation method by passing in two sparse frames. We specifically curate video samples from UCF101 [34]. The assumption to this set up is that a large motion is occuring during in the video stream, as non-interactive actions or slight movements would not make any difference to interpolating frames that are close in timeline. We hypothesize that the modern interpolation algorithms that are trained solely on the nearby frames close in temporal dimension may struggle to interpolate the scenes where the audio becomes a specific cue to a visual appearance, such as musical performance or action scenes from the animated movies. We then replicate the most closely related work to our objective from scratch. Finally, we evaluate the results with two criteria: video perceptual quality and image quality.

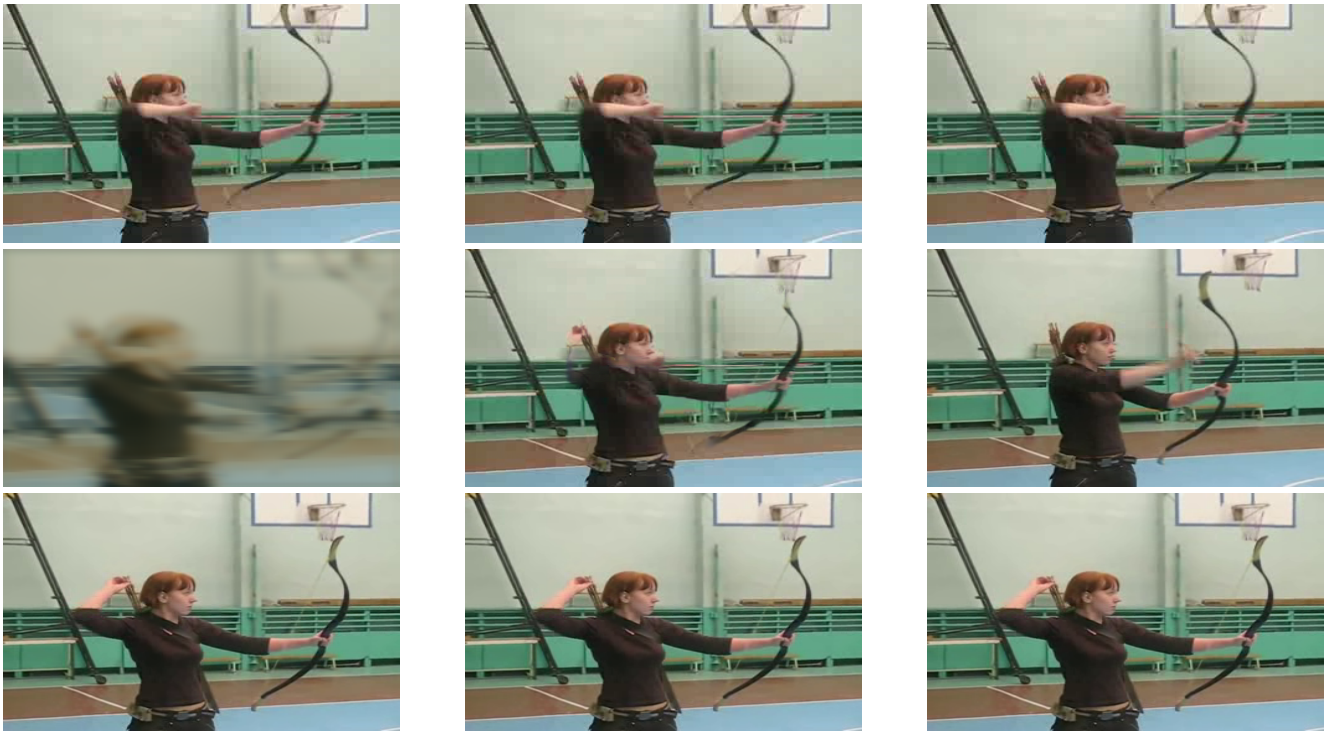
Our contributions are as follows:

- We analyze the weaknesses of the current state-of-the-art methodology in Video Frame Interpolation task to demonstrate the difference in the problem we intent to solve.
- We perform baseline evaluation for our target task with ASVFI model using UCF101 dataset and analyze our findings.

2. Related Works

2.1. Video Frame Interpolation

Video frame interpolation(VFI) is a task in computer vision that synthesizes the intermediate frames given the two frames as input. This task can be applied real-world scenarios, including video enhancement [22, 32], slow-motion rendering [14, 41, 43], and frame rate conversion [24]. It has wide variety of applications in diverse fields with the availability of high-resolution frames rises. Over the recent years, video frame interpolation methods have progressed towards deep learning-based methods. The two main categories are flow-based and kernel-based. Flow-based methods generate interpolated frames based on optical flow estimation, efficiently utilizing bi-directional flows, [16, 27, 30]. While flow-based methods present promising performances, they are often unreliable in dynamic texture scenes. Kernel-based methods mitigate the issue and predict locally adaptive convolution kernels and synthesizes the output pixels [17, 25, 28]. There have also been attempts to combine flows and kernels to perform end-to-end frame synthesis [3, 15]. Furthermore, with the rise



(a) ASVFI (b) PerVFI (c) Ground Truth

Figure 2. Comparison of transition of actions in ASVFI and PerVFI. ASVFI's prediction is lower quality overall, but still demonstrates relevant motion transition as compared to PerVFI.

of diffusion models [9], some works have employed them and achieved promising results [7, 11]. However, the challenges still remained from unavoidable motion errors and misalignment in supervision. Recent work mitigates these challenges by blending intermediate features, such that the reference frame emphasizes the primary content, whole the other contribute to complementary information [42]. While these methods have continuously improved upon the previous state-of-the-art on the image perceptual metrics such as Peak Signal-to-Noise Ration (PSNR) or Structural Similarity Index Measure (SSIM) [39], previous work claimed that PSNR metric is not reflective to represent perceptual quality in interpolated videos [6]. This has driven extended works on providing meaningful metrics to measure video perceptual quality [5, 10, 38].

2.2. Audio-driven Video Generation

With the rise of text-to-image diffusion models [31], audio-to-image models have also been explored and gained attention. Previous work has shown that selecting highly correlated frame-audio segment as train data to jointly train the audio and image encoder can generate an image that corresponds to the audio [1, 36]. These works focus on using the temporal dynamics in sound as a source for vivid video generation. Instead of generating images solely from the audio, attempts have been made to edit the input image with the audio [20, 21]. With the success on image generation, attempts have been made to generate video conditioned on image and audio [13]. StyleGAN has also been employed to generate coherent video based on audio [12, 19]. These promising works suggest that audio is capable of guiding information that does not exist in the visual stream by leveraging neural networks. Although human speech is commonly involved in audio, these work tend to focus more on the audio that does not inclue human speech. Our work is also oriented towards audio that do not include human speech.

3. Methodology

In this section, we present the technical details of the model architecture we used to generate the baseline results. We first discuss our problem setting. Given two reference frames I_0, $I_T \in \mathrm{R}^{C_I \times H \times W}$, and corresponding waveform $A_{0 \to T} \in \mathrm{R}^{C_A \times N_A}$, where $C_I = 3$, $C_A = 2$, $H, W = 224$, and N_A the number of samples in the audio, our goal is to reconstruct I_t where $t = \frac{T}{2}$. Strictly speaking, our input frames are extracted at each extreme of the video stream, where I_0 represents the frame extracted at 0 seconds and I_T at T seconds of the video stream, whereas previous Video Frame Interpolation works set $T < 1$. We set $T = 6$, upon analyzing the distribution of the collected waveform

Figure 3. ASVFI result on predicting the marching motion of the band.

dataset. The overall architecture of the model, which is illustrated in the original paper, is depicted in Figure 1.

The first step is to extract each inputs separately to corresponding audio encoder and video encoder. For the audio, it utilizes VGGish, as it demonstrated strong performance in the audio classification task [8]. For video, it concatenates I_0 and I_T to pass onto the video network, which is a 3D U-Net of 5-layer encoder and 6-layer decoder. The concatenated input propagates through the encoder to obtain the output at the last layer $v_i^5 \in \mathbb{R}^{T\times(h_{in}*w_{in})\times C_{in}}$, which is then passed onto the fusion network with the extracted audio feature. The output of the fusion network becomes the input for the video decoder. The output of each decoder is concatenated with the encoded video feature in the backward sequence, of which is passed as input to the next decoder layer. This series of step continues until the 4^{th} decoder layer. At the 5^{th} decoder layer, the concatenated output from the previous layer propagates to generate v_1^t, which then propagates to the last decoder layer of 2D convolution upon another fusion. The last layer aggregates the propagated information with 2D convolution and sets the number of frames to output, which is essentially performing split across the final output embedding.

3.1. Video Module

Each layer of the encoder consists of 2 blocks of 3D convolution and Feature Gating modules to accurately capture the temporal dynamics between the input frames [15]. Each 3D convolutions is a 5-dimensional filter with kernel size of $3 \times 3 \times 3$. Stride and padding in the temporal and spatial dimensions are set all as 1 except for the third and fourth layers that have stride of 2 in the spatial dimension. ReLU activation function is applied after the first convolution and at the end of each blocks throughout the layers, along with skip connections to preserve feature information across the layers. To focus on the relevant dimensions of the feature maps that learns meaning cues for frame interpolation such as motion boundaries, Feature Gating is applied after every layer. It works as a form of self-attention mechanism by applying spatio-temporal 3D average pooling on the input video feature $v_i \in \mathbb{R}^{C_{in}\times T\times h_{in}\times w_{in}}$ that is passed onto 3D convolution $W_c \in \mathbb{R}^{C_{in}\times C_{out}}$ $b_c \in \mathbb{R}^{C}_{out}$ with stride and padding of 1. Sigmoid activation function is applied after the convolution. The final output is the element-wise product of the attended map with the input video feature.

$$v_o = \sigma(W_c \cdot pool(v_i) + b_c) \odot v_i \tag{1}$$

The objective of decoder is to construct the output frames from the latent representation captured by the encoder via multi-scale feature upsampling. Each layer of decoder consists of 3D transpose convolution, Feature Gating module, and 2D convolution. Each has kernel size of $3\times3\times3$, stride of $3 \times 4 \times 4$, and padding of 1, except for the first and the 4^{th} layer with kernel size of 3 and stride of 1. After obtaining the 3D feature map from 3D transpose convolution and Feature Gating module at each layer, 2D convolution is applied by combining the feature map's temporal and channel dimensions to aggregate the information present in multiple frames. The final decoder layer is a 2D convolution with kernel size of 7×7 with 2D reflection padding to obtain the original input shape of the video. At this point, the the conventional way to diverge the frames is to perform split on the channel dimension to generate $k-1$ output frames, where k represents the scale. However, we only generate a single frame, as it does not align with the objective of our task. We will discuss this further in Section 5.

3.2. Fusion Module

The objective of the fusion module is to model the visual features with the guidance of the audio features. Previous works have demonstrated that audio signals are capable of attaining this task [29, 37]. To be specific, this module exploits audio signals to guide visual attention across spatial and channel dimensions to obtain the enhanced video feature. It generates channel-wise attention maps to emphasize informative features and spatial attention maps for the channel-attentive features to highlight sounding regions, performing matrix multiplication with the obtained features from Eq.4 and Eq.5 to obtain the channel-attentive visual features as a final product.

$$v^{cs} = M^s \otimes (v^c)^T \tag{2}$$

3.2.1 Channel-wise Attention

Given audio feature $a \in \mathbb{R}^{T \times C_a}$ and video feature $v \in \mathbb{R}^{T \times (h_{in} * w_{in}) \times C_v}$, each features first are projected onto the same channel dimension with a feed-forward network $U_a^c \in \mathbb{R}^{C_a \times C}$ and $U_v^c \in \mathbb{R}^{C_v \times C}$ with ReLU activation function. Then, the output of the element-wise multiplication of the projected features is applied global average pooling on the channel dimension. Lastly, the output is passed onto two fully-connected network $U_1^c \in \mathbb{R}^{C \times 256}$ and $W_1 \in \mathbb{R}^{256 \times C}$ layers with a sigmoid activation function. Final channel attention map v^c is obtained by element-wise multiplication between M^c and v with a residual connection.

$$M^c = \sigma(W_1 U_1^c(\delta_a(U_a^c a \odot U_v^c v)) \tag{3}$$

$$v^c = v \odot (M^c + 1) \tag{4}$$

3.2.2 Spatial Attention

Following a similar fashion, spatial attention map is obtained. σ here is hyperbolic tangent activation function, with $W_2 \in \mathbb{R}^{1 \times 256}$ is a learnable parameter, $U_a^s \in \mathbb{R}^{256 \times C_v}$ and $U_v^s \in \mathbb{R}^{256 \times C_a}$ the fully-connected layers with ReLU activation function.

$$M^s = softmax(\sigma(W_2(U_a^s a \odot U_v^s v^c))) \tag{5}$$

Loss Function The model is trained via L1 pixel level loss between the predicted and the ground truth frame, where N represents the size of mini-batch used in training.

$$L(I_{pred}, I_{gt}) = \frac{1}{N} \sum_{i=1}^{N} ||I_{pred}^i - I_{gt}^i||_1 \tag{6}$$

4. Experimental Setup

4.1. Dataset

The problem we aim to solve must involve audio that has high relevance with the visual information, or a clear cue to triggering motions. This may include such actions as playing an instrument or sports. UCF101 is a dataset with 101 categories of actions that belong to Human-Object Interaction, Body-Motion, Human-Human interaction, Playing Music Instruments or Sports, which matches with our target domain [34]. We sample a subset of the dataset and split with 8:2 ratio to generate 5469 train and 1368 test samples. This includes categories such as blowing hair dryer, bowling, and boxing a punching bag.

4.2. Implementation Details

Our procedure is as follows: (1) Extract two sparse frames, I_0 and I_1, that are at the each extreme of the video by discretizing frames with their default configured fps rate. (2) Extract $A_{0 \to 1}$ with the default sampling rate.

Audio Feature Extraction. First, we determine the duration of the audio based on the distribution of our obtained samples. The average duration was 6.8, median of 5.9, with a standard deviation of 3.7. The ideal setting to our problem is to use the whole duration of each audio sample, as the audio that corresponds to I_1 exists towards the end of the stream. However, to balance the memory overhead, we set the duration rounded down to the nearest integer of the average duration. We pad and truncate the audio samples that are shorter or longer than our determined duration. We pass our pre-processed audio sample to the VGGish extractor to obtain the MFCC feature to pass as input to VGGish model. This creates $17 \times 96 \times 64$ feature representation.

Video Feature Extraction. The first dimension of the audio feature essentially becomes the number of frames we must match our video data with. To do this, we perform trilinear interpolation to increase the number of frames from 2 to 17. For training, we center crop the images with size of (224×224) and perform random horizontal and vertical flips. In the end, the dataset consists of the video features, the ground truth frame features, and the corresponding audio features.

Training Details. Our goal is to evaluate the performance of ASVFI network in our problem setting. However, the model is too complex compared to the dataset we train with. We attempted to preserve the original architecture as much as possible while preventing the model from overfitting. Denote that we do not have any access to pre-trained weights, all models are trained from scratch. We set the training epoch to 1, as we have observed the model overfitting quickly. We use AdamW optimizer [23] with initial learning rate of 2e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a batch size of 1. We evaluate the model on the test set every 10% of the iteration and drop the learning rate by a factor of 0.1 with patience of 1.

Metrics. Our evaluation metrics follow two main criteria: video perceptual quality and image perceptual quality. To measure the video perceptual quality, we initially measured FloLPIPS, VFIPS [10], and Frechet Video Distance (FVD). Each of these methods utilize pre-trained networks to obtain the feature embedding at the hidden layers to compute the scores. FloLPIPS leverages optical flow from the previous frame to compute the weighted spatial average of LPIPS. VFIPS leverages Swin Transformer to measure the perceptual distance at multiple scales. FVD utilizes pre-trained I3D network to compute the FID score over the distribution of frames of the video. However, we excluded VFIPS in our evaluation as it produced scores outside the the typical range. It is reported in the original paper that VFIPS works well in setting with a minimum number of 12 frames. We interpolated the output video to match this setting, but the numbers were still not promising. To measure the image

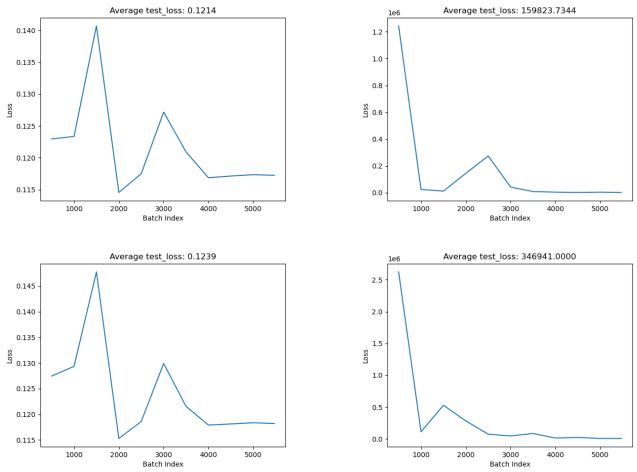


Figure 4. Test loss every 10% of iteration of train set, compared between AdamW Optimizer(Top) and Adam(Bottom) with initial learning rate is set to 0.0002(Left) and 0.0006(Right).

perceptual quality, we utilize Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). These metrics compute the scores based on pixel-level differences. PSNR measures the log ratio of the max pixel intensity over the mean squared error between the reference and the distorted frames. SSIM computes the difference in luminance and contrast over the sliding window to emphasize the similarity in the local structures, which PSNR fails to capture. However, our main focus is oriented towards perceptual video quality than image quality, as we want to assess whether the predicted frames are coherent with regards to the neighbor frames.

Training Experiments. With ASVFI architecture tailored to be trained on large datasets, our goal was prevent the model from overfitting to our relatively smaller dataset. Without modifying the original architecture, we ran experiments in two different settings: the optimizer and the initial learning rate. The comparison between the learning rate is shown in Figure 2 and between the optimizer is shown in Figure 4. We also present the performance in video quality and image quality metrics for different training settings in Table 2. We do not include the metrics for learning rate higher than our optimal setting as their results were similar to arbitrary noise.

5. Results

5.1. Quantitative Analysis

We present the evaluation results and analysis on the baseline ASVFI architecture and PerVFI's performances on UCF101 [34]. The result is shown in Table 1. PerVFI, while obtaining overall higher performance than our current ASVFI, obtains half of the performance than its performance on Vimeo-90k [44]. Vimeo-90k dataset's configuration is similar to our dataset in terms of video resolution and the category of samples.

	FloLPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
AdamW	0.353	2158.594	12.541	0.420
Adam	0.357	2254.949	12.169	0.415

Table 1. Metric comparison between AdamW and Adam optimizer on the subset of UCF101 [34].

	FloLPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑
PerVFI	0.177	801.894	17.148	0.573
ASVFI	0.353	2158.594	12.541	0.420

Table 2. Performance comparison between PerVFI and ASVFI across the subset of UCF101 [34].

5.2. Qualitative Analysis

We also present the output video as series of frames. Our problem setting is to predict a single intermediate frame given the input frames at each extreme of the video sample. The result is shown in Figure 2. We also wanted to point out that predicting the reasonable transition in action is a challenging task. We show one of the best results on ASVFI in Figure 3. In human's eyes, we can easily predict what will come in the intermediate of the frames. However, it is difficult to model this process, as higher score will be given to the frame that captures more information from the inputs rather than making reasonable transition in motion. This also occurs in PerVFI in 5. The silhouette of the person from both frames in the intermediate frame results in high score of FloLPIPs and low FVD score.

6. Limitation and Future Work

We originally anticipated the baseline performance on ASVFI to be slightly higher on video coherency measures than the state-of-the-art video interpolation models, as audio itself in these videos can directly infer the visual scenes for humans. However, we were not able to fully optimize training at this point. Such problems let constrained us from being able to utilize the full capacity of the model. This is due to the architecture itself is tailored to train on large datasets. The original paper trains and evaluates the model on Vox2Celeb [4] and HDTF [45], where each dataset contains more than 100K samples, with HDTF even higher resolutions. Furthermore, we were not able to balance the memory overhead. Loading a single batch involves 5-D tensor video, 4-D tensor image, and 4-D tensor audio. This makes the training insufficient, as the maximum batch size

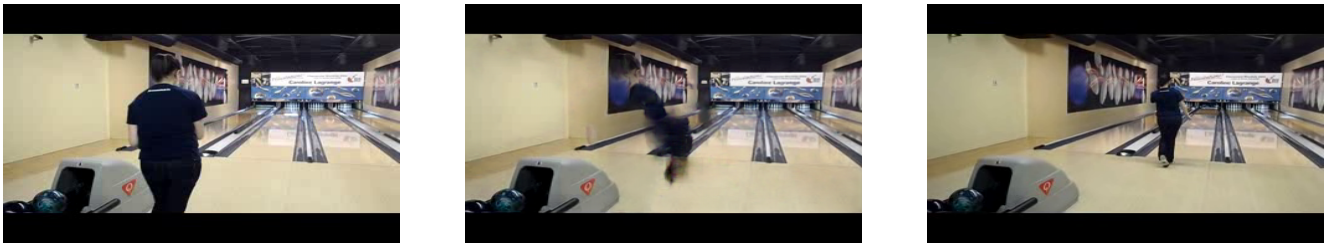


Figure 5. PerVFI result on predicting the motion in between walking.

we could set to was 1. Still, we were able to generate reasonable results with the current configuration on image quality metrics, compared to the performance of PerVFI.

Furthermore, next step to this work can be extended to two major work. First, we need to develop a meaningful metric that is able to tell the ground truth motion based on audio. As we stated in the previous section, the current video quality metrics tend to give high scores to the frames that simply embeds information in the input frames. While this metric would make sense in the problem setting of close frames in the time line, it is not able to give much information about what we aim to measure. We essentially want to measure how natural the motion transition from I_0 to I_1. Simply moving from I_0 to I_1 in the frame-level does not reflect our intention. Audio source localization is work focused on predicting the location in the visual information given the corresponding audio. These have demonstrated high correlation in audio and visual information leads to better performance in locating the source of sound in images [26]. We aim to leverage these pre-trained networks in order to compute score by leveraging the audio to determine the ground truth location of objects in the frame.

As we have established the baseline performance without the modification to the original ASVFI network, we plan to modify the architecture in a way that we are able to further utilize our dataset without overfitting. At the high-level, we can simply add dropout layers in between the numerous convolution layers to check its performance, then develop further by adding new modules, such as modifications to the fusion modules, etc. We also plan to modify PerVFI architecture tailored to our task. As it still demonstrates strong performance in generating intermediate frames, we believe the addition of audio input with appropriate fusion mechanism will make this architecture be suitable to answering our problem. We first would like to make the minimum change as possible while being able to accept the input to see how the performance changes from the baseline. We believe that with the directions we posed upon this work, we will be able to demonstrate that audio can become a strong tool to solve the major problems in computer vision task.

References

[1] Moitreya Chatterjee and Anoop Cherian. Sound2sight: Generating visual dynamics from sound and context. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 701–719. Springer, 2020. 3

[2] Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, et al. An overview of core coding tools in the av1 video codec. In *2018 picture coding symposium (PCS)*, pages 41–45. IEEE, 2018. 1

[3] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020. 2

[4] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 6

[5] Duolikun Danier, Fan Zhang, and David Bull. Flolpips: A bespoke video quality metric for frame interpolation. In *2022 Picture Coding Symposium (PCS)*, pages 283–287. IEEE, 2022. 3

[6] Duolikun Danier, Fan Zhang, and David Bull. A subjective quality study for video frame interpolation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1361–1365. IEEE, 2022. 3

[7] Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1472–1480, 2024. 3

[8] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 4

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[10] Qiqi Hou, Abhijay Ghildyal, and Feng Liu. A perceptual quality metric for video frame interpolation. In *European Conference on Computer Vision*, pages 234–253. Springer, 2022. 3, 5

[11] Siddhant Jain, Daniel Watson, Eric Tabellion, Ben Poole, Janne Kontkanen, et al. Video interpolation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7341–7351, 2024. 3

[12] Dasaem Jeong, Seungheon Doh, and Taegyun Kwon. Träumerai: Dreaming music with stylegan. *arXiv preprint arXiv:2102.04680*, 2(4):10, 2021. 3

[13] Yujin Jeong, Wonjeong Ryoo, Seunghyun Lee, Dabin Seo, Wonmin Byeon, Sangpil Kim, and Jinkyu Kim. The power of sound (tpos): Audio reactive video generation with stable diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7822–7832, 2023. 3

[14] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018. 2

[15] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2071–2082, 2023. 2, 4

[16] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022. 2

[17] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5316–5325, 2020. 2

[18] Sungho Lee, Narae Choi, and Woong Il Choi. Enhanced correlation matching based video frame interpolation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2839–2847, 2022. 1

[19] Seung Hyun Lee, Gyeongrok Oh, Wonmin Byeon, Chanyoung Kim, Won Jeong Ryoo, Sang Ho Yoon, Hyunjun Cho, Jihyun Bae, Jinkyu Kim, and Sangpil Kim. Sound-guided semantic video generation. In *European Conference on Computer Vision*, pages 34–50. Springer, 2022. 3

[20] Seung Hyun Lee, Wonseok Roh, Wonmin Byeon, Sang Ho Yoon, Chanyoung Kim, Jinkyu Kim, and Sangpil Kim. Sound-guided semantic image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3377–3386, 2022. 3

[21] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19108–19118, 2022. 3

[22] Wenhao Li, Guangyang Wu, Wenyi Wang, Peiran Ren, and Xiaohong Liu. Fastllve: Real-time low-light video enhancement with intensity-aware look-up table. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8134–8144, 2023. 2

[23] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[24] Guo Lu, Xiaoyun Zhang, Li Chen, and Zhiyong Gao. Novel integration of frame rate up conversion and hevc coding based on rate-distortion optimization. *IEEE Transactions on Image Processing*, 27(2):678–691, 2017. 2

[25] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022. 2

[26] Shentong Mo and Pedro Morgado. Localizing visual sounds the easy way. In *European Conference on Computer Vision*, pages 218–234. Springer, 2022. 7

[27] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5437–5446, 2020. 2

[28] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*, pages 261–270, 2017. 2

[29] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 801–816. Springer, 2016. 4

[30] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14539–14548, 2021. 2

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[32] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5114–5123, 2020. 2

[33] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Transactions on Image Processing*, 30:277–292, 2020. 1

[34] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5, 6

[35] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. 1

[36] Kim Sung-Bin, Arda Senocak, Hyunwoo Ha, Andrew Owens, and Tae-Hyun Oh. Sound to visual scene generation by audio-to-visual latent alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6430–6440, 2023. 3

[37] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 247–263, 2018. 4

[38] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 3

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[40] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003. 1

[41] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: Backward accumulation for long-range optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12119–12128, 2023. 2

[42] Guangyang Wu, Xin Tao, Changlin Li, Wenyi Wang, Xiaohong Liu, and Qingqing Zheng. Perception-oriented video frame interpolation via asymmetric blending. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2753–2762, June 2024. 3

[43] Haoming Xu, Runhao Zeng, Qingyao Wu, Mingkui Tan, and Chuang Gan. Cross-modal relation-aware networks for audio-visual event localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3893–3901, 2020. 2

[44] 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. 6

[45] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3661–3670, 2021. 6