CSE353 Milestone 4: Final Report

Hyeon Joon Lee & DongHun Kim

Scope and Purpose

The primary objective of our group's final project is to enhance the performance of existing approaches of frame interpolation. Many notable approaches rely on 3D images and depth data that can be obtained from them. Some others, who employ 2D images, also estimate the depth and use the data for obtaining the loss functions. Having initially planned to incorporate such depth data from 3D videos into our algorithm, our group has shifted to using human join data through utilizing Openpose, since Milestone 3.

The remaining sections of this paper will explain the progress up until Milestone 4, our final implementation, and how we have arrived to such progress. Furthermore, we will discuss some successes, in addition to failures, and state our conclusion.

Progress

One notable change since Milestone 3 is that we have decided to use all frames in a video, regardless of human detection, for training and validating our model. Then, we have altered our dataloader, in *dataloader.py*, to change data fetching methods. In Milestone 3, we only collected frames that contained human skeleton, and from such frames, we picked three frames, not necessarily consecutive, for computing the loss functions and ultimately calculating the intermediate frame. In this previous approach, both the interpolated frame and the loss functions were highly biased, as apparent in the psnr (peak signal-to-noise ratio). We removed factors that led to such a bias in Milestone 4.

Additionally, we have adjusted the sequential container for the perceptual loss, adding seven more layers to the previous one and recalibrated the coefficients for the five terms in our global loss function. By empirical methods, we have arrived to a more optimal equation.

Finally, we have created a new python file, *create_video.py*, for loading the trained model and performing frame interpolation into a given video file. While the python file for training our model, *train_keypoints.py*, uses 60fps videos for the ground truth data, *create_video.py* utilizes 30fps videos and double the number of frames, ultimately making a 2x slomo video. Interpolating videos with configurations other than 30fps, for instance, 60fps, some adjustments to *create_video.py* are necessary.

Success & Failure

In all aspects, our final implementation performs superior to that of Milestone 3. Both the training loss and the validation loss decrease smoothly, while the psnr gradually increases after iterations of training.

Algorithms for preprocessing data, in addition to training and applying the saved model to perform an actual interpolation all works as expected. The losses and psnr results, for both baseline algorithm and our algorithm are given in the figures next page.

For comparing the performances between the two algorithms, only one epoch each were performed. However, when actually saving a model, three epochs of training were performed.

Baseline Algorithm Training and Validating

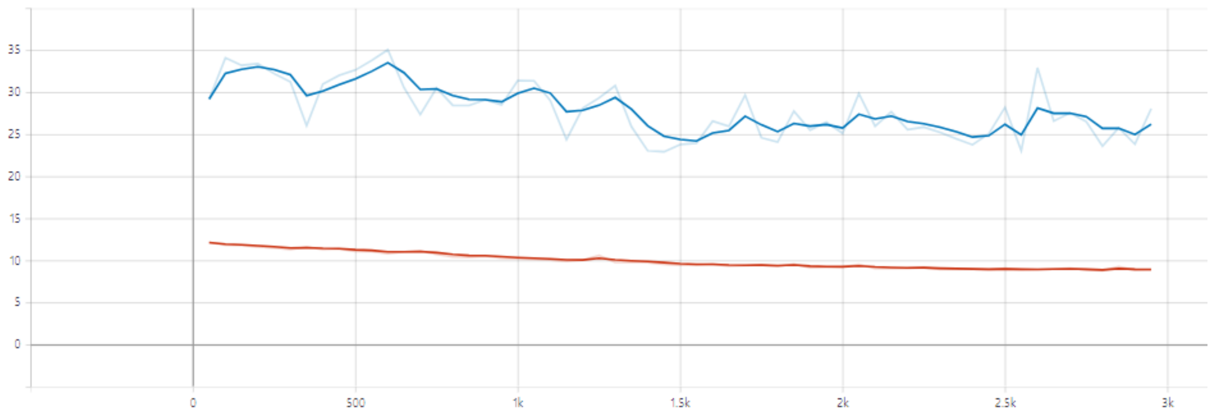

Fig 1: Train & Validation Loss – Blue refers to the train loss and the red the validation loss

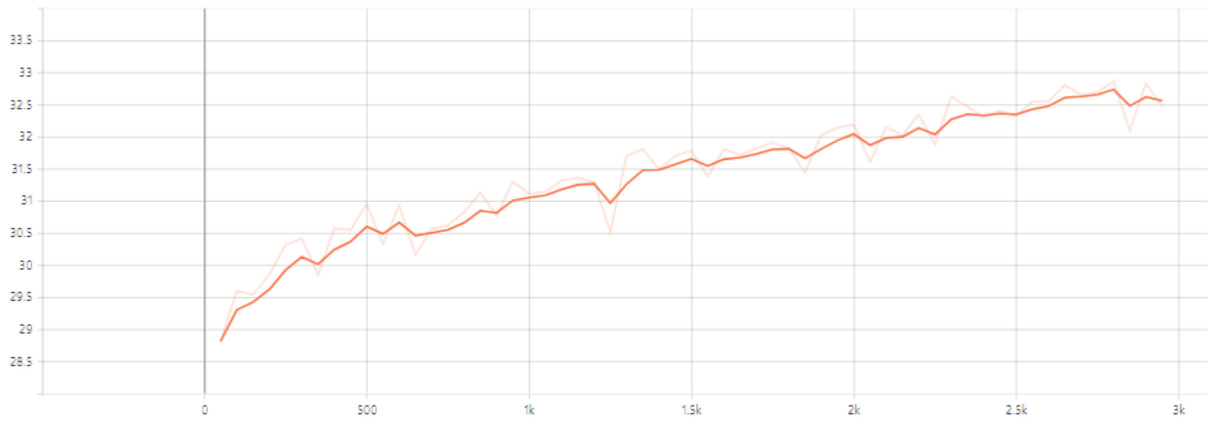

Fig 2: PSNR – Maximum 32.8 (the transparent line is the actual psnr and the solid line is due to smoothing)

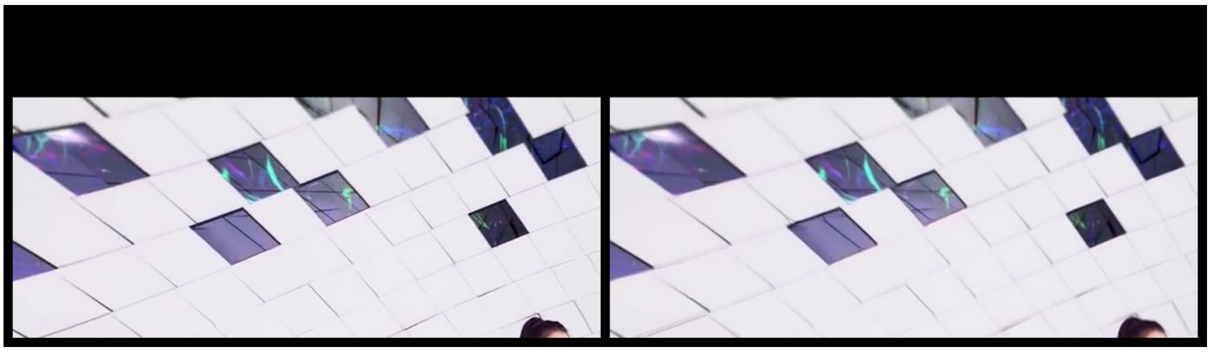

Fig 3: Comparison of Intermediate & Interpolated – 1 (left: ground-truth intermediate frame / right: interpolated frame)

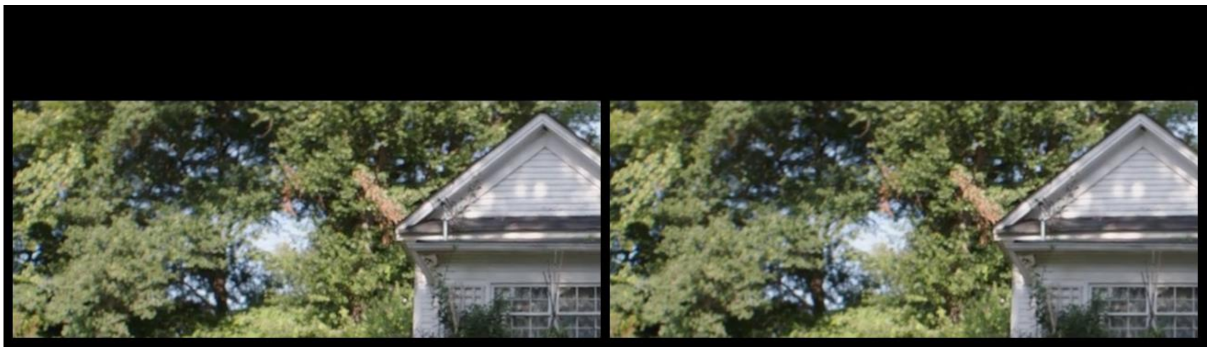

Fig 4: Comparison of Intermediate & Interpolated – 2 (left: ground-truth intermediate frame / right: interpolated frame)

Our Algorithm Training and Validating

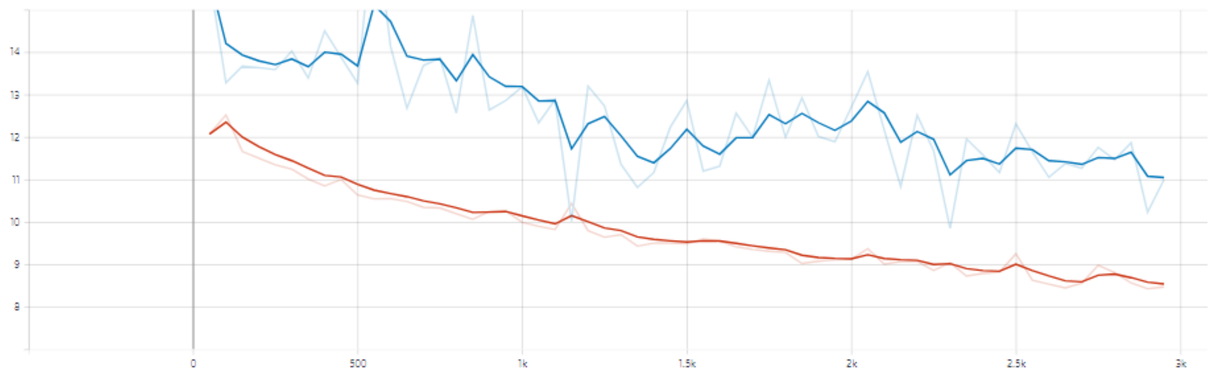

Fig 5: Train & Validation Loss - Blue refers to the train loss and the red the validation loss

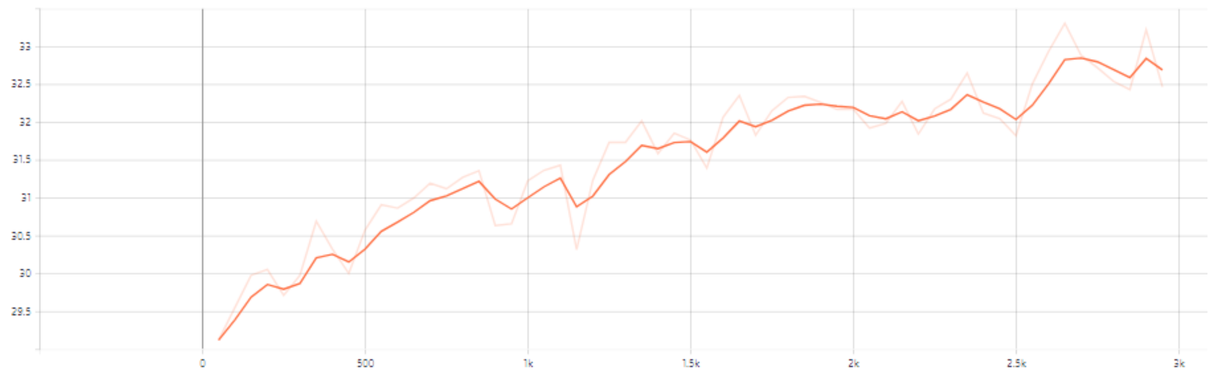

Fig 6: PSNR – Maximum exceeds 33 (transparent line is the true psnr)

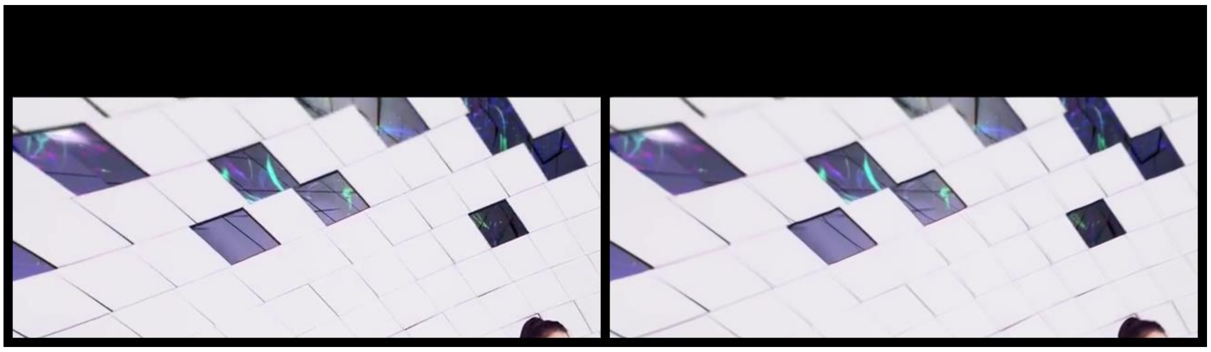

Fig 7: Comparison of Intermediate & Interpolated – 1 (left: ground-truth intermediate frame / right: interpolated frame)

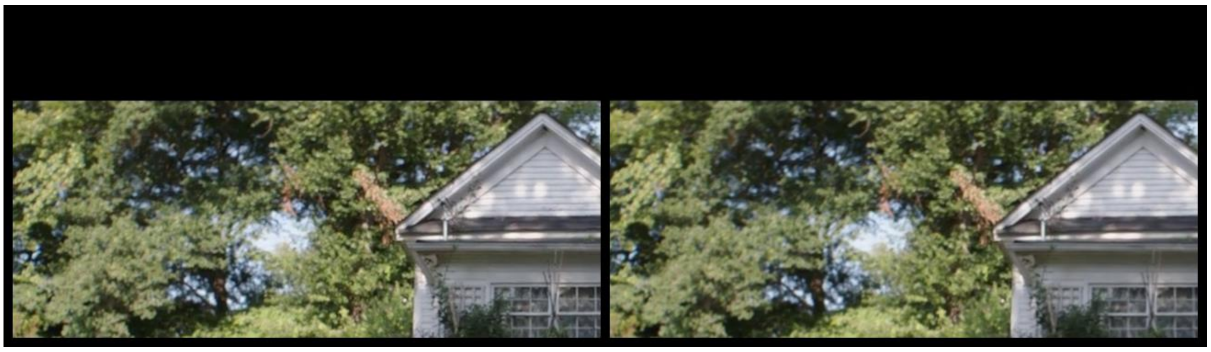

Fig 8: Comparison of Intermediate & Interpolated – 2 (left: ground-truth intermediate frame / right: interpolated frame)

Conclusion

We have made notable changes since Milestone 3 and worked hard to improve our results. The resulting outcome can be easily observed through comparing the losses and psnr values with those from Milestone 3. While we admire the results of the existing papers on frame interpolation, we believe their impressive outcomes may have resulted from using high fps videos in the first place. We have tried our algorithms on various videos with differing frame rates, from 30fps to 240fps, and the results vary immensely. Using videos with high frame rates in the beginning, we believe we could have reached more admirable outcomes.