Saliency-Guided Video Codec Pre-Processing

Alaskar Alizada

[1]Computer Science, Stanford

Challenge and Background

The rapid increase in video quality is becoming unsustainable to transmit over a network. For instance, a second of raw HD video at 30 frames-per-second would require approximately 150MB to store and transmit!
Intelligent video encoding-decoding (codec) algorithms are crucial to keeping video consumption sustainable and widely accessible. But a good codec is a balancing act between reducing the file size and preserving the video quality.
The focus of this project was centered around leveraging Computer Vision techniques to pre-process videos in order to improve the efficiency of industry standards such as HEVC and H.264. Specifically, the attempt was to efficiently extract a saliency map of each frame in the video, blur the video based on said saliency map, and then compress it using H.264.

Background and Problem statement

There have been numerous methods applied in this domain, such as:

- Using reinforcement learning to guide Quantization Parameter of H.264
- Extracting saliency maps using hand-crafted features
- Recording eye tracking data of people watching videos

Many of these algorithms required a lot of compute or time, such as the work requiring recording a lot of eye tracking data before compression.
Thus, we tried to create a very simple and quick algorithm that could derive a rough saliency map and pre-process videos for H.264 compression.
The algorithm takes as input any uncompressed video in the form of a numpy array. As output, the algorithm returns a numpy array of the same shape, but having been pre-processing with blurring. Therefore, to be able to adequately evaluate the efficacy of this pre-processing step, we used the following metrics:

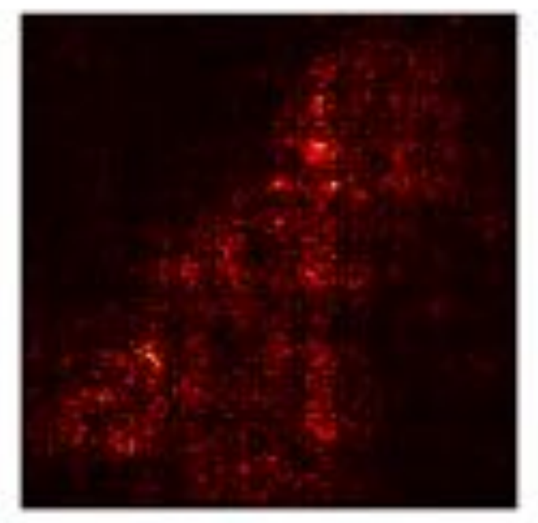
- A comparison of file sizes after H.264 compression
- Peak Signal-to-Noise Ratio (PSNR)
- Structural Similarity (SSIM)
- Qualitative surveying

Dataset

The dataset is taken from HMDB5, a human motion databases consisting of 7000 short clips distributed in 51 actions. Of those, we selected 30 clips across 15 actions (2 clips per action) to test the efficacy of the algorithm. The 15 actions were: throwing, sword exercise, swinging a baseball bat, smoking, shooting a gun, kicking a ball, riding a bike, riding a horse, brushing hair, fencing, playing golf, picking something up, hitting something, drinking, and catching something. Each of the selected clips was compressed using H.264 as a baseline for the pre-processing algorithm.

Methods

The rough saliency map was derived by passing each frame of a video through GoogLeNet and backpropagating. The produced shape was reduced to a 2D mask by only keeping the highest absolute value of the 3 color channels for each pixel.



We see that while this produced a good rough outline of the object of interest, it missed some spots within the bounds of the object that are also salient. Thus, we apply applied an averaging 3x3 convolution to every value in the generated rough saliency map, where each value in the convolution is preset to $\frac{1}{9}$. This preprocessing convolution ensures that neighboring pixels are of similar values, thus creating smoother regions of saliency and non-saliency.
The following formula was then applied to the saliency map to derive standard deviations for each pixel i, j;

$$\mathit{std}_{i,j} = \sigma((\frac{\max \mathit{Saliency\ Map}}{2} - \mathit{Saliency\ Map}_{i,j}))$$

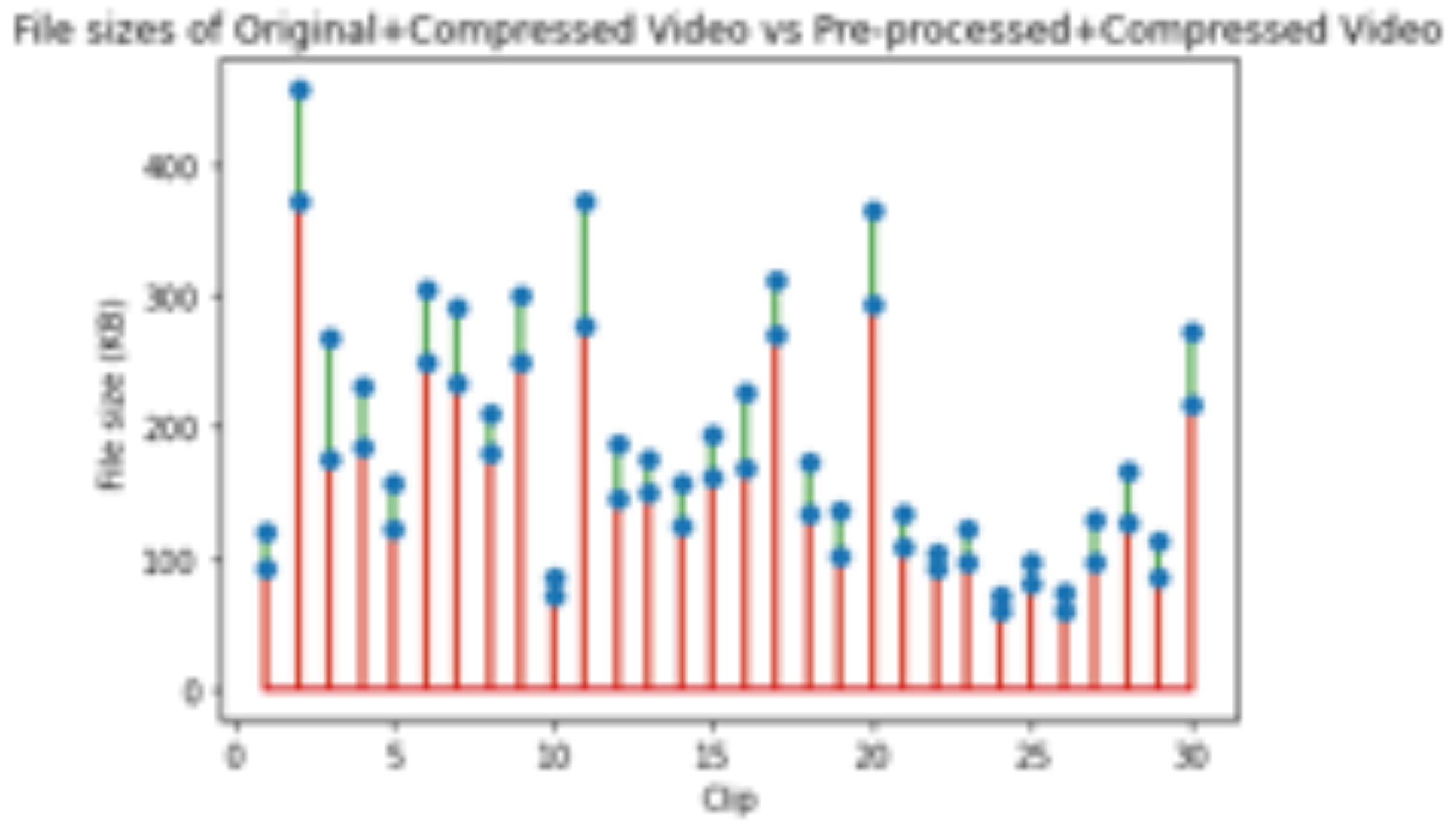
where $\sigma()$ is the sigmoid function. We then produced and applild a Gaussian convolution with standard deviation $\mathit{std}_{i,j}$ on each pixel (i, j). Once that was done, the video was ready to be compressed by H.264

Experiments and Analysis

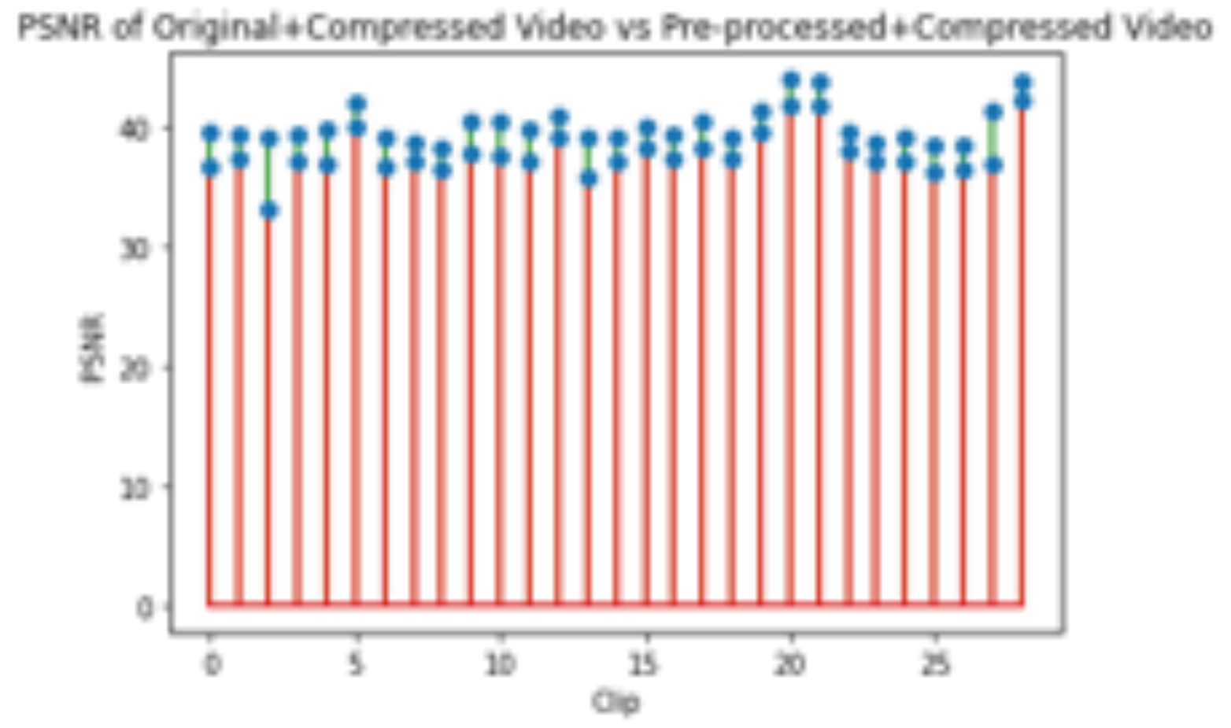
We compared the pre-processed+compressed videos to original+compressed videos, as well as testing different strengths of blurring in pre-processing. Specifically, the following two formulas were used to shrink and expand the difference in blurring:

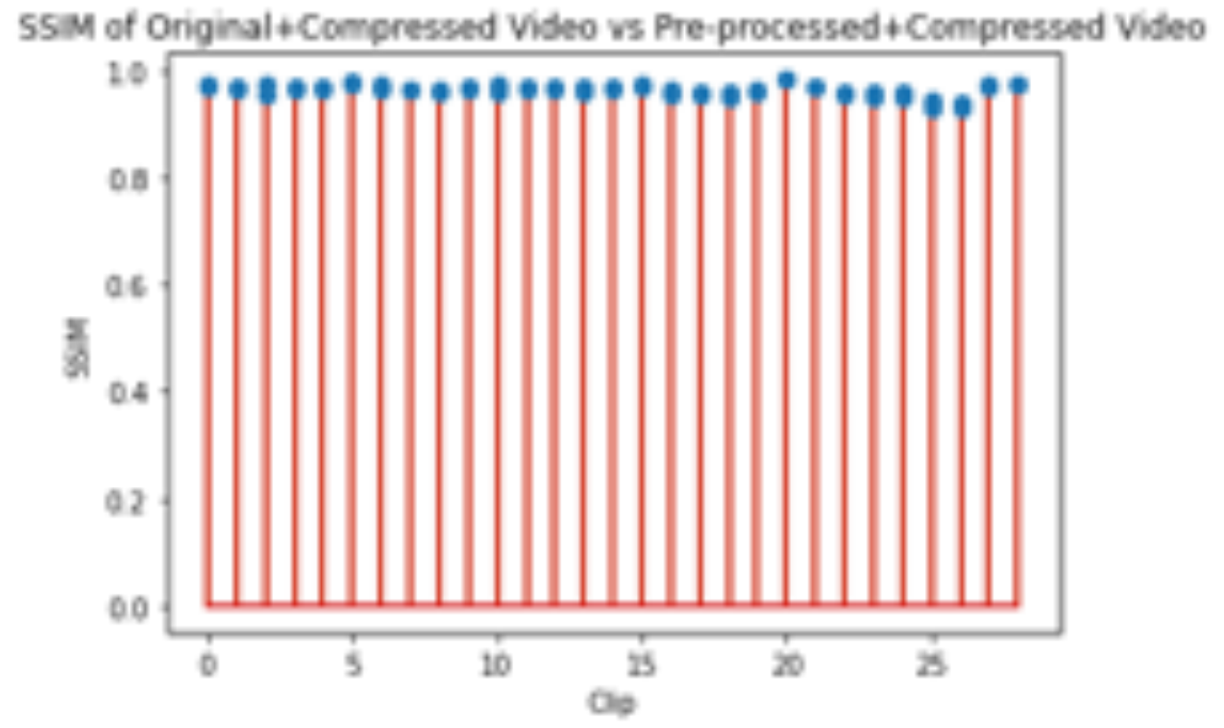
$$\mathit{std}_{i,j} = \sigma((\frac{\max \mathit{Saliency\ Map}}{2} - \mathit{Saliency\ Map}_{i,j})10)$$

$$\mathit{std}_{i,j} = \sigma((\frac{\max \mathit{Saliency\ Map}}{2} - \mathit{Saliency\ Map}_{i,j})0.1)$$

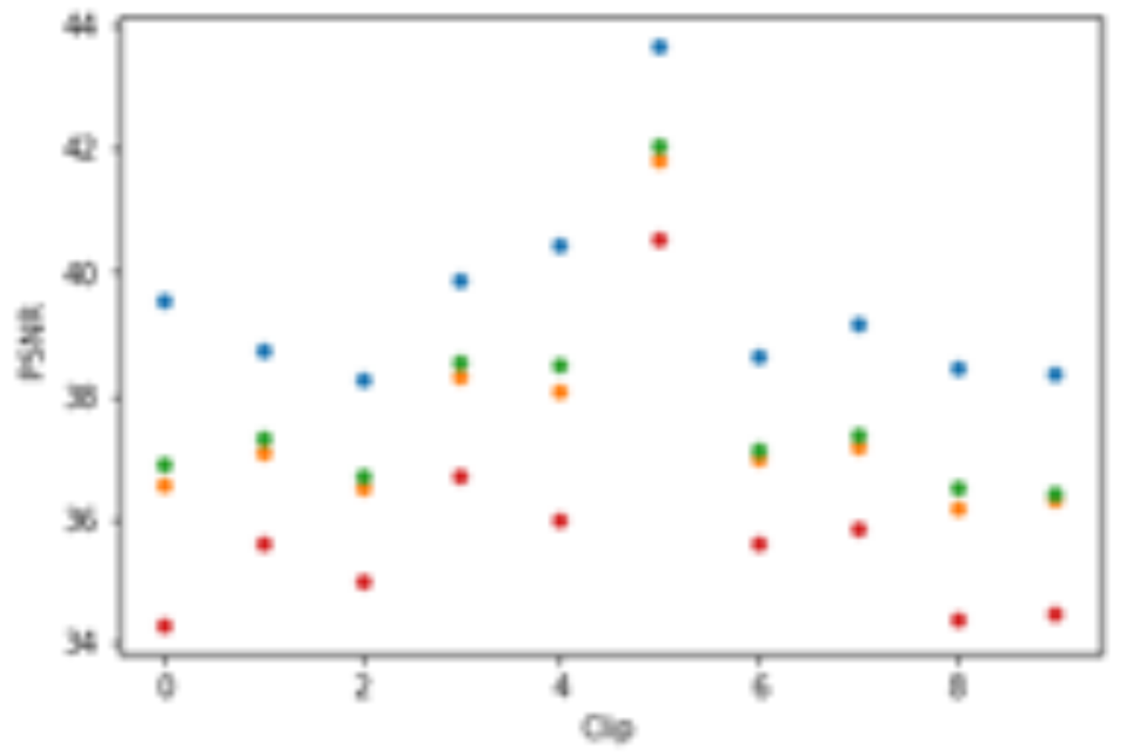


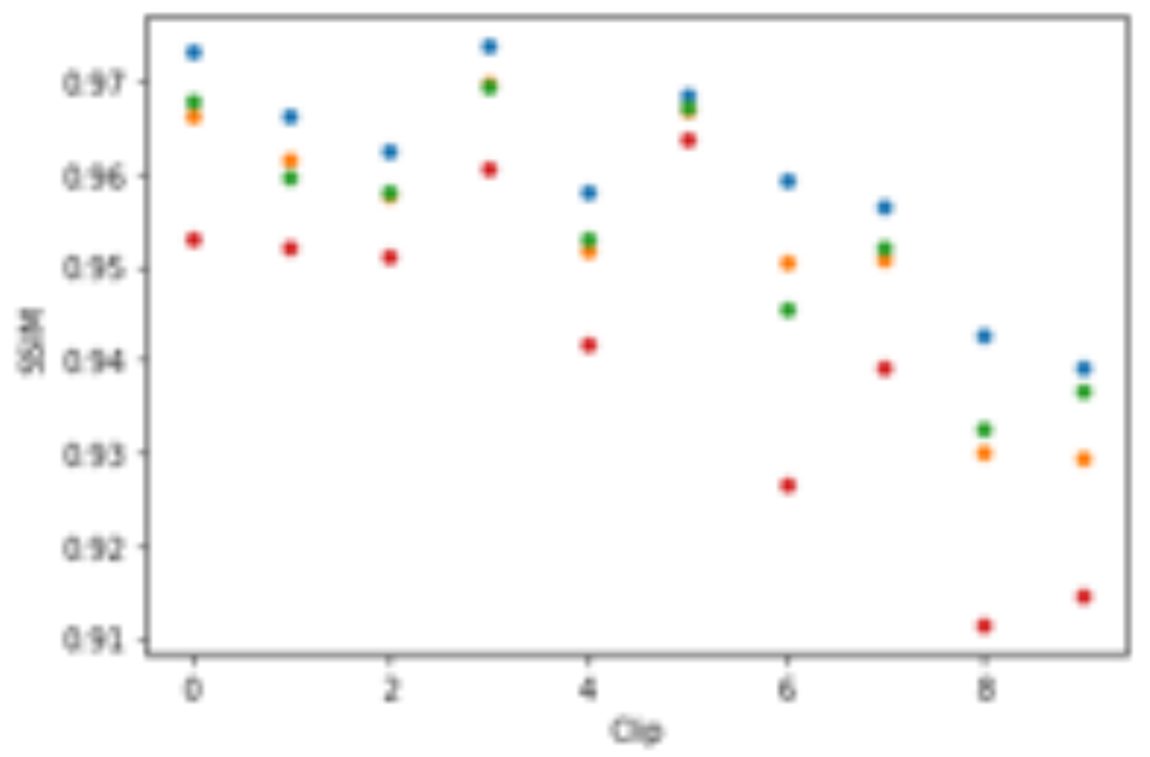
As seen in the figure comparing the file size, the pre-processing approach reduced the post-compression file size on every video (20.28% on average).





As we see the in the figures above, the difference in PSNR beetween pre-processed+compressed vs original+compressed is quite large. However. SSIM reports more favorable relative results.





We similarly compared PSNR and SSIM scores of different intensities of pre-processing and found that there is a tradeoff between video quality and file size depending on how much intensely Gaussian blurring is applied.
Lastly, for the qualitative results, after showing people video and asking to pick which one is better, or neither, participants picked:

- Original+compressed was better: 1
- Pre-processed+compressed was better: 0
- Neither: 10

Conclusions

Overall, the considerable reduction in size compared with a small decrease in SSIM makes a promising case for the pre-processing step. However, it's important to realize that the dataset was full of bad quality videos which might have affected the metrics. Moreover, there are certain limitation to using GoogLeNet , such as the fact that it was trained on ImageNet, which contains only 1000 categories.
As further work, it would be interesting to use transformer-based image captioning models and see if we could derive a saliency map by looking at the attention weights.

[1] H Kuhne et al. "HMDB: A Large Video Database for Human Motion Recognition". In: IEEE International Conference on Computer Vision (ICCV). 2011
[2] Christian Szegedy et al. "Going deeper with convolutions". In: Proceedings of the IEEE conference on computer vision and pattern recog- nition. 2015, pp. 1–9