Saliency-Guided Video Codec Pre-Processing

Alaskar Alizada
Stanford University
Stanford, CA
alaskar@cs.stanford.edu

June 3, 2022

Abstract

With the rapid increase in high quality video content, video compression algorithms must adapt to ensure that content on the internet remains accessible and sustainable. In this paper we use a video pre-processing technique to improve the industry standard H.264 compression algorithm. The pre-processing of videos will involve quickly generating a saliency map describing which parts of the video are important to preserve and which parts are not, where important is defined as the region consumers are most likely to look at. This saliency map is then used to blur parts of the video deemed ”irrelevant”, and the product is then inputted into H.264 algorithm to compress. With this technique we observed a significant reduction in file size (20.28%) and only a marginal decrease in quality, as measured by PSNR (5.84%) and SSIM (1.03%).

1 Introduction

The challenge of efficient content streaming is becoming an increasingly pressing issue with an increase in video quality and an abundance of video streaming services. For instance, a second of raw HD video at 30 frames-per-second would require approximately 150MB to be sent over the network. Now, with the development of higher quality videos (4K, 8K) and an increase in their demand, this problem is expected to be exacerbated even further. Thus, intelligent video encoding-decoding (codec) algorithms are crucial to keeping video consumption sustainable and widely accessible. However, the problem of video codec is two-fold. First, a codec algorithm needs to reduce bandwidth and storage costs of videos by reducing their bitrate. Second, once decoded, a codec algorithm needs to preserve user Quality of Experience (QoE). Since QoE is often times reduced as a result of video distortion produced by the encoding-decoding phases of a codec algorithm, the two aspects of video codec are negatively correlated, posing a significant challenge. This means that an intelligent video encoding algorithm is one that is able to find a good trade-off of the two.

The current industry standard for video codec is H.264. In this paper, we focus on pre-processing videos to improve the efficacy of H.264, such that once these pre-processed videos are subsequently processed by H.264, we can observe a lower bitrate and a higher QoE. We do so by first deriving salient regions - regions in a vidoe that the user is most likely to pay attention to - of each frame in a video. Next, we add a targeted Gaussian blur to the video by blurring the regions that are low saliency and preserving the regions that have high saliency. Therefore, the algorithm take a video as input, and outputs a video with targeted blurring, which is then passed through H.264 for the final result.

2 Related Work

Since the inception of video streaming, a lot of research has been put into trying to improve on video compression algorithms. Countless techniques have been applied in an attempt improve pre-processing, encoding, and decoding algorithms, or any combination of the three, and strike a good balance between bitrate and QoE.

First such work is one done by researchers at DeepMind/ Google [10]. They proposed Reinforcement Learning (RL) method using which each frame could be assigned a quantization parameter (QP), which is a parameter used in establishing the aforementioned tradeoff between video size and video quality, after which they would input the video with per-frame QPs into H.264.

Work done by Li et al. [7] suggested using a Graph-Based Visual Saliency (GBVS) to derive an initial saliency map, after which a Gaussian function is used to refine the map. GBVS relies on hand-crafted features that are meant to mimic human attention to extract important features, and normalize over the whole frame. This creates a mask that could be incorporated in the HEVC codec (a successor to H.264) that would allow HEVC to attend to different regions of each frame depending on the region's probability of attracting human attention. Similarly, Ku et al. [5] focused on first extracting low-level features of illumination, color and texture of each frame using color spaces proposed by the international commission on illumination. They then extracted features based on inter-frame correlation of the low-level features, and subsequently fused all of the features together to obtain a saliency map. This was then once again used as input into HEVC.

Other work, done by Zhu et al. [16] used (Deep Neural Network) DNN based saliency algorithm to extract regions of interest an guide bit allocation in the HEVC algorithm. Specifically, they used a 10-layer Convolutional Neural Nets (CNN) to compute spatial saliency and combined it with an extracted temporal saliency from motion information to form video's saliency map. Similarly, Sun et al. [14] relied on DNNs to create a saliency map for HEVC. They first trained a deep convolution network (DCN) model to extract the static saliency map by highlighting semantically salient regions. Then they developed moving object segmentation technique to automatically extract the dynamic salient regions for each frame. Finally, they combined the two to create an adaptive bit rate allocation for HEVC.

Moreover, work has been done to predict salient regions by getting volunteers to watch videos and tracking their gaze while they are watching. Polakovic et al. [12] as well as Lyudvichenko et al. [9] focused on recording eye tracking information and using it to generate true saliency maps, that can then be used to blur parts "irrelevant" parts of the video. Once again, the video was then processed by video codec, H.264.

A different method of human attention prediction is using texture analysis. For example, Cross et al. [1] used Markov Random Fields to describe a textured image, while Jain et al. [4] used Gabor filters to derive texture representation of an image. Both could be used as an alternative to create an accurate saliency map. Lastly, there has also been plenty of work done in improving the codec algorithm, specifically using Artificial Neural Nets. For example, Li et al. [8] CNNs to efficiently encode similarities between frames, while Hu et al. [3] used Recurrent Neural Nets (RNN), and both implemented their approaches into HEVC.

3 Methods

Our method focused on generating a fast and simple saliency map that could then be used to blur appropriate parts of the video, and compress it using H.264. The first step was to generate a rough saliency map for each frame. This step was inspired by the work done by Simonyan et al. [13], in which they explored visualization techniques for image classification models. In our algorithm, we first used a GoogLeNet [15] model that had been pretrained on the ImageNet dataset [2]. Each frame of the video was passed into the model generating a distribution of scores over the 1000 categories in ImageNet. The category with the highest score was then used as a pseudo-label for that specific frame. We then use the GoogLeNet scores of the frame and the associated pseudo-label for the frame to backpropagate through GoogLeNet and generate gradients in the shape of the frame. We then took the absolute value of all values in the generated set of gradients and, for each pixel, we took the maximum of the three RGB channels. Intuitively, the gradients represented the relationship between pixel values and the generated scores, and the result frame (after taking absolute value and max over the three channels) provides a rough indication of where the GoogLeNet focused most in the original frame when it generated its scores.

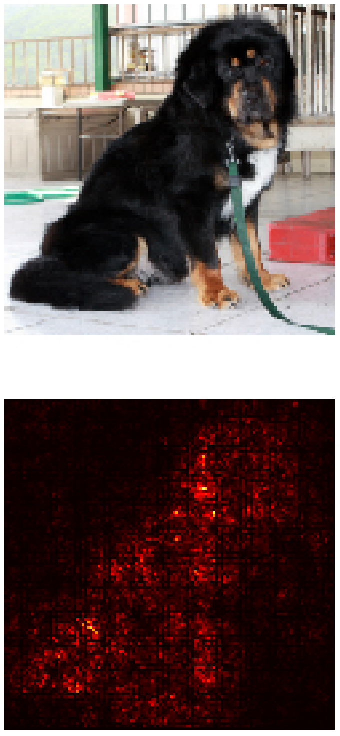

Figure 1: Visualization original image and the result rough saliency map

As seen in 1, the GoogLeNet creates a compelling rough saliency map, outlining the object of interest in the image.

Once a rough saliency map is generated for a frame, the next step is to smooth it out. As seen in 1, the generated saliency map gives a good rough outline of the object of interest, but often misses spots within the bounds of the object that are also salient. Thus, we apply apply 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.

Next, we used the generated saliency map for a given frame to guide how we blur that frame. Specifically, for each pixel in the frame, we used the corresponding value of the saliency map to create a 3x3 Gaussian filter, and use the filter as a convolution applied to the pixel. To do so, we transformed the values of the saliency map into appropriate standard deviations that were then used for the Gaussian filter. We used the following formula to transform the saliency map into standard deviations for each pixel:

$$std_{i,j} = \sigma\left(\left(\frac{\max Saliency\ Map}{2} - Saliency\ Map_{i,j}\right)\right)$$

where $\sigma()$ is the sigmoid function and $std_{i,j}$ is the standard deviation that was used to create the Gaussian convolution filter for pixel (i, j). We subtracted half the maximum value of the saliency map from each element to create a range centered at 0. Next we multiplied each element by -1 to flip the values, since we want the higher saliency regions to have smaller blur. Lastly, we applied the sigmoid function to squish all of the values in the range 0 and 1, with more salient regions being less than 0.5 and less salient regions being more than 0.5.

To account for frame size discrepancy in each video, and a fixed frame size requirement for GoogLeNet, we clipped each frame to the required size of 224x224, and after generating the rough saliency map we padded the saliency map with values such that it starts with the values at the edge of the original, clipped rough saliency map and linearly gets smaller as it gets to the edge of the padded rough saliency map.

4 Dataset

The dataset we will be using is HMDB5 1 [6], 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. The clips vary in duration and quality, as shown in 2 and 3. Each of the selected clips was compressed using H.264 as a baseline for the pre-processing algorithm.

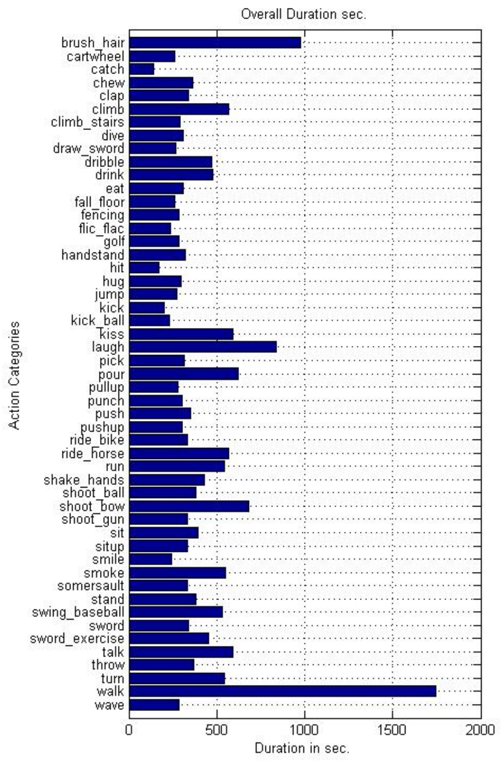

Figure 2: Clip duration distribution [6]

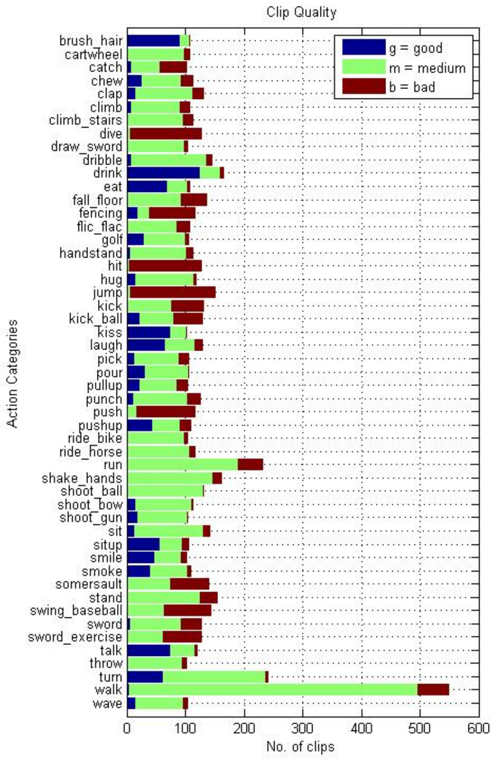

Figure 3: Clip quality distribution [6]

5 Experiments and Results

To evaluate the efficacy of the algorithm, we first compressed each of the selected 30 clips using H.264 to use as baseline. Then we pre-processed each of the videos using the algorithm described in the Methods section. To further evaluate the algorithm, we used two more calculations for the standard deviations of the per pixel Gaussian blur. The first one was

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

, for each pixel (i, j) where $\sigma()$ is the sigmoid function. This was done to shrink the difference between high and low saliency regions in the amount of Gaussian blur that is applied. Similarly, the next calculation was

$$std_{i,j} = \sigma((\frac{\max Saliency\ Map}{2} - Saliency\ Map_{i,j})*10)$$

which was meant to widened the difference between high and low saliency regions.

The metrics we will be using to assess the efficacy of the algorithm are a combination of quantitative and qualitative. The first quantitative metric is directly comparing file sizes of pre-processed videos compressed with H.264 and original videos compressed with H.264. Then, we will be using common compression evaluation metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). PSNR is a metric commonly used to assess the quality of image reconstruction in lossy compression codecs (such as H.264), and is a ratio between the maximum possible value of a signal, in our case 255, and the value of distorting noise. Formula for PSNR is:

$$PSNR = 10 * \log_{10}(\frac{255^2}{MSE})$$

$$MSE = \frac{1}{H*W}\sum_{i=0}^{H}\sum_{j=0}^{W}(R_{i,j} - D_{i,j})$$

, where $R_{i,j}$ is the (i,j) pixel of the reference image, and $D_{i,j}$ is the (i,j) pixel of the distorted/compressed image. For video analysis, this metric is commonly averaged over all frames. However, as shown in the work by Nasrabadi et al. [11], the following formula is more correlated with subjective test results (human viewing) and will thus be used in this paper:

$$PSNR = 10 * \log_{10}(\frac{255^2}{\frac{1}{N}\sum_{i=0}^{N} MSE_i})$$

Similarly, SSIM is another commonly used metric to assess quality of reconstruction in images. However, SSIM is widely regarded to be more accurate in evaluating subjective QoE than is PSNR. The formula for SSIM is as follows:

$$SSIM(R,D) = \frac{(2\mu_R\mu_D + c_1)(2\sigma_{R,D} + c_2)}{(\mu_R^2 + \mu_D^2 + c_1)(\sigma_R^2 + \sigma_D^2 + c_2)}$$

where $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ for $k_1 = 0.01, k_2 = 0.03$ and L is the dynamic range of the pixel-values. R and D are once again the reference and distorted images, respectively.

Finally, for the qualitative metric, we will be showing the pre-processed+compressed videos as well as the original+compressed videos to 11 individuals and later give them an option to pick one of: video A (pre-processed+compressed) is higher quality, video B (original+compressed) is higher quality, undecided. We will not tell the respondents up-front that we will be asking about video quality because we don't want them to inspect the videos with that premise in mind, but rather look at the videos as they would naturally.

First, we will look at the difference in file sizes between pre-processed+compressed videos and original+compressed videos.

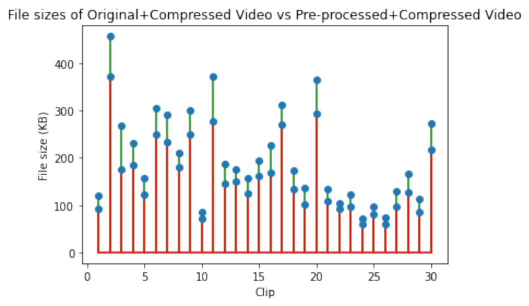

Figure 4: File sizes (KB) of pre-processed+compressed videos (red) and original+compressed videos (green)

As we see in 4, original videos that were compressed with H.264 are larger than the videos that have been pre-processed before being compressed in every video. The average size of the original video that was compressed with H.264 is 200.03 KB, while the average size of the videos that had been pre-processed before being compressed with H.264 is 159.47. This is a significant decrease of 20.28% of the average size of a video that had been pre-processed over the one that hasn't, before being compressed with H.264.

Next, we look at the PSNR scores of original+compressed videos and compare them with pre-process+compressed videos.

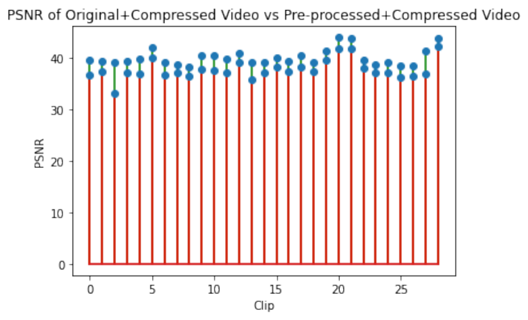

Figure 5: PSNR of pre-processed+compressed videos (green) and original+compressed videos (red)

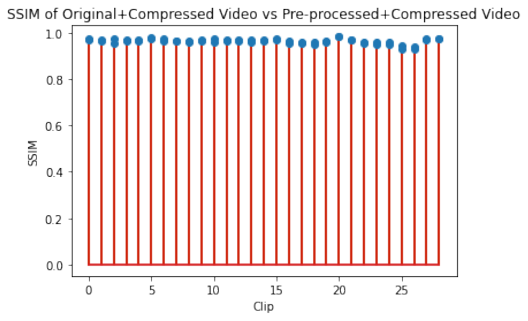

Figure 6: SSIM of pre-processed+compressed videos (green) and original+compressed videos (red)

As we see in 5, the original+compressed video has a slight advantage in the PSNR score over pre-processed+compressed videos. The average original+compressed PSNR is 40.04 and the average pre-processed+compressed PSNR is 37.7, so the lack of pre-processing results in better reconstruction, on average, according to PSNR. If we look at SSIM, we see a similar pattern, where original+compressed video has a slight advantage over pre-processed+compressed videos, as shown in 6. However, this difference is very small. Specifically, the average SSIMs are 0.97 and 0.96, original+compressed video and pre-processed+compressed videos, respectively.

We also looked at PSNR and SSIM when we shrunk and expanded the difference between the high and low saliency regions.

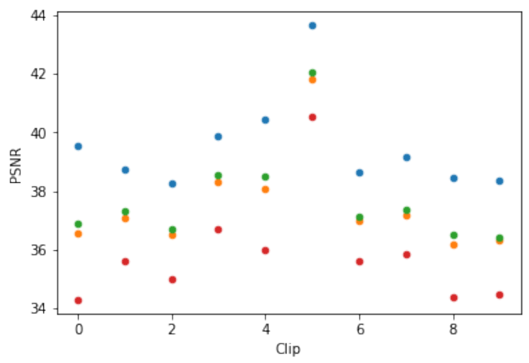

Figure 7: PSNRs: Original(Blue), Pre-Pro(Orange), Pre-Pro*0.1(Green), Pre-Pro*10(Red)

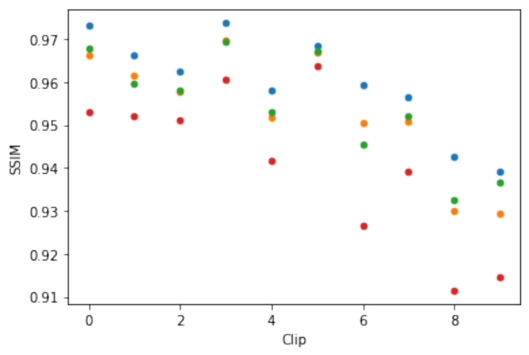

Figure 8: SSIMs: Original(Blue), Pre-Pro(Orange), Pre-Pro*0.1(Green), Pre-Pro*10(Red)

As we see from 7 and 8, the original+compressed video (blue) PSNR and SSIM is better than all other variations, as expected. Moreover, when we expanded the difference between high and low saliency regions (red), we got an overall much lower scores in both PSNR and SSIM. This is likely due to over-blurring parts of the video. However, we also see that when we shrink the difference between high and low saliency regions (green), both PSNR and SSIM are very similar to the regular pre-processing (orange). This points to the fact that there exists a trade-off of file-size and video quality, where over-blurring regions can result in a steep drop in video quality and a marginal decrease in file size as compared to the regular blurring step, while under-blurring can result in a similar drop in video but a smaller decrease in file-size as compared to the regular blurring step.

Finally, when 11 responds were shown the videos and after that asked to pick which one is higher quality, or neither, we saw that only 1 person selected original+compressed video as the higher quality and 10 selected undecided. This tells us that while they may be some small distortions when pre-processing the video before compressing it with H.264, it is unnoticeable to a human eye.

Overall, While the cost incurred by video quality from a decrease in PSNR by 5.84% is significant, the reduction in file size of 20.28% makes a compelling case for using the pre-processing technique. Moreover, the reduction in SSIM was proportionally much smaller (1.03%), and since, as mentioned earlier, SSIM is widely regarded as the more indicative metric of the two, the technique does seem promising. By contrast, while the qualitative results seemed to be in support of the pre-processing, since almost nobody noticed the difference in video quality, it could also very likely be due to the fact that the quality of the original videos were not good to begin with, which is why a relative decrease in quality went unnoticed.
Lastly, we saw that certain clips performed better than others, in terms of a larger reduction in size and a smaller decrease in PSNR/SSIM. This was likely due to the fact that some clips contained objects that are not part of the 1000 classes in ImageNet that GoogLeNet was trained on. As a result, the saliency map that was generated was likely more sparse and less accurate than the one generated for objects familiar to GoogLeNet, creating a more random blur.

6 Conclusion and Further Work

In conclusion, we saw that using saliency to guide Gaussian blurring can significantly reduce file size generated by H.264. However, it comes a at a cost (albeit smaller than the reduction in size) of video quality, as seen from decrease PSNR and SSIM. We also saw that there exists a middle ground for blurring regions based on saliency, and over-blurring frames comes with a greater cost to video quality than is the reduction in size.
Overall, we think that PSNR and SSIM results could be improved with a better calculation of a saliency. For future work, it would be interesting to look at other methods for generating a saliency map. In particular, we would be curious to explore if we can leverage a pre-trained transformer-based image captioning model, extract the attention weights of the encoder and use them to guide the creation of a saliency map.

References

[1] George R. Cross and Anil K. Jain. "Markov Random Field Texture Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5.1 (1983), pp. 25–39. DOI: 10.1109/TPAMI.1983.4767341.

[2] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[3] Yueyu Hu et al. "Progressive spatial recurrent neural network for intra prediction". In: *IEEE Transactions on Multimedia* 21.12 (2019), pp. 3024–3037.

[4] A.K. Jain and F. Farrokhnia. "Unsupervised texture segmentation using Gabor filters". In: *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*. 1990, pp. 14–19. DOI: 10.1109/ICSMC.1990.142050.

[5] ChungWen Ku et al. “Bit Allocation based on Visual Saliency in HEVC”. In: *2019 IEEE Visual Communications and Image Processing (VCIP)*. 2019, pp. 1–4. DOI: 10.1109/VCIP47243.2019.8965753.

[6] H. Kuhne et al. “HMDB: A Large Video Database for Human Motion Recognition”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2011.

[7] Yiming Li et al. “Saliency based perceptual HEVC”. In: *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. 2014, pp. 1–5. DOI: 10.1109/ICMEW.2014.6890644.

[8] Yue Li et al. “Convolutional Neural Network-Based Block Up-Sampling for Intra Frame Coding”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.9 (2018), pp. 2316–2330. DOI: 10.1109/TCSVT.2017.2727682.

[9] Vitaliy Lyudvichenko et al. “A semiautomatic saliency model and its application to video compression”. In: *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2017, pp. 403–410. DOI: 10.1109/ICCP.2017.8117038.

[10] Amol Mandhane et al. “MuZero with Self-competition for Rate Control in VP9 Video Compression”. In: *arXiv preprint arXiv:2202.06626* (2022).

[11] A. T. Nasrabadi et al. “Investigating the PSNR calculation methods for video sequences with source and channel distortions”. In: *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. 2014, pp. 1–4. DOI: 10.1109/BMSB.2014.6873482.

[12] Adam Polakovič et al. “An Approach to Video Compression Using Saliency Based Foveation”. In: *2018 International Symposium ELMAR*. 2018, pp. 169–172. DOI: 10.23919/ELMAR.2018.8534631.

[13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *arXiv preprint arXiv:1312.6034* (2013).

[14] Xuebin Sun et al. “Content-aware rate control scheme for HEVC based on static and dynamic saliency detection”. In: *Neurocomputing* 411 (2020), pp. 393–405. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.06.003. URL: https://www.sciencedirect.com/science/article/pii/S0925231220309668.

[15] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[16] Shiping Zhu and Ziyao Xu. “Spatiotemporal Visual Saliency Guided Perceptual High Efficiency Video Coding with Neural Network”. In: *Neurocomput.* 275.C (Jan. 2018), pp. 511–522. ISSN: 0925-2312.