Assignment 2

Coding for streaming.

Student Number: 122115091

Name: Melanie Leonard

Introduction

For the purpose of this assignment I will be analysing the differences between the two separate Capped CRF encodings of the PeterPan.webm video provided. I have separated the encodings into different files so that the psnr.log, ssim.log and probe.csv files could be contained separately, with their respective jupyter notebooks for each assessment as this was easier to manage.

Determining Average Bitrates

In order to determine the average bitrates for the CRf values I first encoded the audio stream separately, specifying the AAC format and a bitrate of 128k. I chose this rate because it adds all of the important data, and anything higher usually shows diminishing returns as it becomes harder to notice the difference, making the increased bitrate unreasonable.

After this the ffmpeg tool was used to specify the target output, outlining the the keyint_min value to be the same as the GOP length, enforcing the Instantaneous Decoder Refresh (IDR). The CRF was then set to 21/24, and the audio was removed from the encoding process using -an. The target output was specified with no buffer (see fig.1.). When this was run the average bitrate shows in the output data, (see fig 2.) . Because the algorithm works on a GOP level I then checked the average bitrate for the GOP's (see fig.3.). I decided to use the overall average rather than the GOP average as the overall was higher and I thought that was a safer bet.

I then opened new notebooks to test the capped versions. The only difference was that -b:v was set to the average bitrate for each CRF, the max rate was set to the same and the buffersize to double this(see fig.4).

Perceptual Differences, PSNR Values & Quality Consistency

The SSIM.log was chosen to test the perceptual differences as this measurement was designed to test based on the Human Visual System[2]. The results of the SSIM test showed no real perceptual difference between the encodings, meaning that to the viewer, the content would be practically indistinguishable.

I then tested the PSNR values between the reference and the encoded versions. The maximum range for all of them was thrown by the last few images which were just a black screen, potentially due to the fact that I worked with a frame rate of 24 rather than 23.98, the result returned Inf (see fig.5). I searched through the PSNR log files to find the highest PSNR avg in each. The maximum average PSNR for the Capped CRF 21 was 58.55, with an average bitrate of 2368.93kb/s , the Capped CRF 24 had an average PSNR of 56.15dB and a bitrate of 1762.63kb/s,respectively, showing just over 2dB difference between the two. The higher the PSNR the better the quality, but an increase of only 2dB is not likely to be worth the 600+kb/s rise in bitrate.

The content is extremely consistent from a SSIM perspective throughout both encodings,

	SSIM	Average PSNR	Maximum Average PSNR	Minimum Average PSNR
CRF 21 Uncapped	0.95	26.41	inf/55.99	7.89
CRF 24 Uncapped	0.95	26.41	inf/56.18	7.88
CRF 21 Capped	0.95	26.41	inf/58.55	7.89
CRF 24 Capped	0.95	26.42	inf/56.13	7.89

Different CRF Suggestions

Lower CRF Values usually lead to higher quality outputs, anywhere within the range of 17–28 is generally considered to be almost lossless, or at least there is very little perceptual difference.[3] An increase of + 6 CRF generally results in a halving of the bitrate, so I would suggest changing the CRF values to 21 and 27, as both of these values are in the range where the human visual system will notice very little if any change, yet the bitrate can be halved. This would also reduce oscillations in quality as there is a significant difference between the two.

VBV Buffer Size

As mentioned, the maxrate was specified to be the average bitrate for each encoding. The data which the buffer receives must be at least the size as the maxrate +1 to ensure that there is enough data to present for the next segment. This helps to avoid interruptions in playback. I decided to make the buffer size double the maxrate, which would slightly delay content presentation, however it is more likely to ensure continuous playback of the content.

GOP Choices

For this encoding process I worked with a frame rate of 24fps rather than 23.98 as recommended in class. I decided on a GOP length of 72, which equates to a duration of 3 seconds. I made this decision firstly because there is a lot of fast changing information in the content, and a shorter GOP length preserves this content more accurately and also adapts more quickly to bandwidth changes[4]. It is however less efficient. The second reason for my decision was that this would force an IDR every 3 seconds, making the segment duration 3 seconds as the durations under 2 seconds perform badly and noticeable dain stops after 4 seconds, making between 2-4 seconds the optimal choice.

GOP Bitrates

The GOP bitrate variability was tested using ‘visualise_GOP_bitrates’ from lab 9. The results for both encodings show potential problem areas (see figures 6&7). The variability in CRF 21 would likely not be acceptable for streaming platforms as they tend to allow only up to 10% variability. The presence of the buffer can help to reduce the impact of this bitrate fluctuation, however there does come a point when this is not enough to ensure smooth playback. A more stable bitrate than that seen in CRF 21 would likely be necessary, but given that the buffer size is doubled, the CRF 24 variability is likely acceptable. To ensure a more consistent bitrate you would have to lower the CRF value, which is a viable option in this case as both 21 and 24 are reasonably high.

Figures

---

crf 21 stream

```
%%bash -s "$REFERENCE" "$CRF21TARGET"
ffmpeg -hide_banner -i $1 -c:v libx264 -pix_fmt yuv420p -an -b:v 0 -keyint_min 72 -g 72 -crf 21 -y $2
```

Figure 1.

```
[libx264 @ 0x14d707290] kb/s:2843.67
```

Figure 2.

```
average = gop_dataframe["Bitrate"].mean()
print(average)
```

```
2.3647379272448066
```

Figure 3.

```
%%bash -s "$REFERENCE" "$CAPPEDCRF24TARGET"
ffmpeg -hide_banner -i $1 \
-c:v libx264 -pix_fmt yuv420p -an \
-keyint_min 72 -g 72 -crf 24 \
-b:v 2.1M \
-maxrate 2.1M \
-bufsize 4.2M \
-y $2
```

Figure 4.

```
00b0] PSNR y:24.714296 u:39.930252 v:41.314885 average:26.419153 min:7.878832 max:inf
```

Figure 5.

```
: visualise_GOP_bitrates(gop_dataframe, 2.1, 72, 23.98)
```

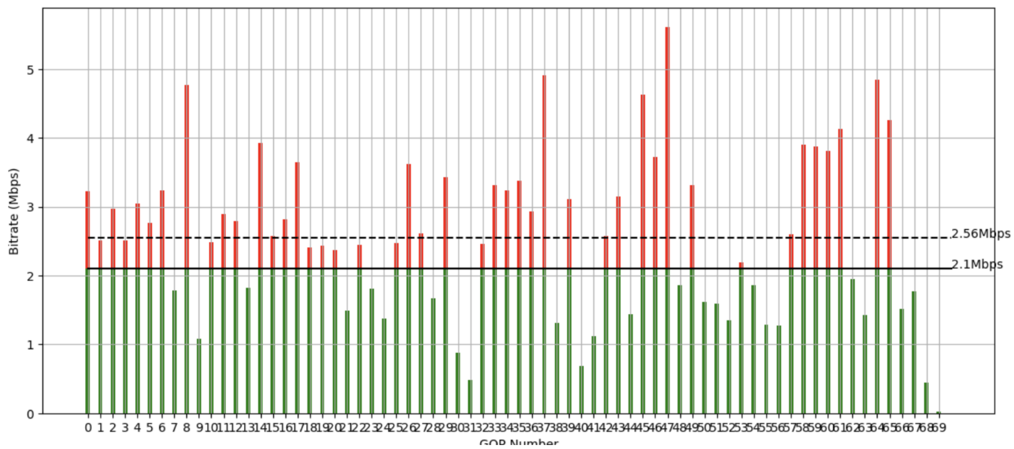

Figure 6. Bitrate Variability CRF 21.

```
: visualise_GOP_bitrates(gop_dataframe, 2.1, 72, 23.98)
```

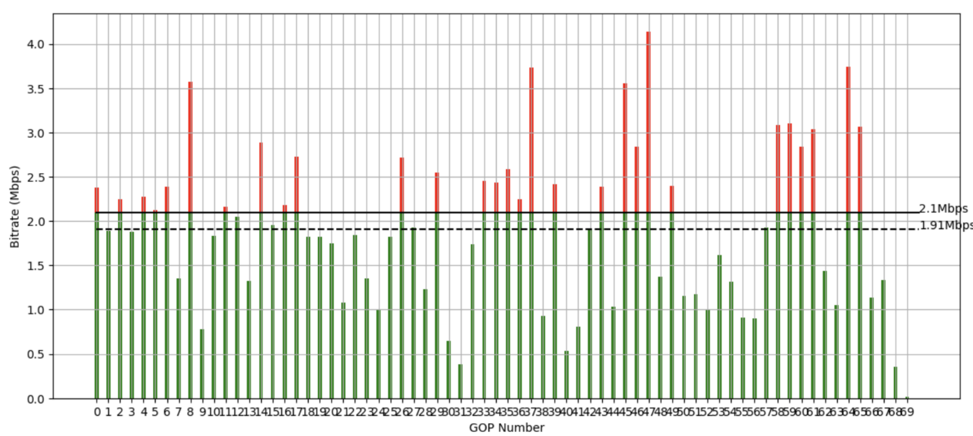

Figure 7. Bitrate Variability CRF 24.

Bibliography

1. https://slate.com/news-and-politics/2007/04/can-anyone-tell-the-difference-between-music-files-at-128-kbps-and-256-kbps.html
2. Setiadi, D.I.M. PSNR vs SSIM: imperceptibility quality assessment for image steganography. *Multimed Tools Appl* 80, 8423–8444 (2021). https://doi.org/10.1007/s11042-020-10035-z
3. https://trac.ffmpeg.org/wiki/Encode/H.264
4. https://www.veneratech.com/understanding-gop-what-is-group-of-pictures-and-why-is-it-important/