E9 208 Digital Video: Perception and Algorithms
Assignment 3

Vineeth S
M. Tech Artificial Intelligence
SR. No. 16543

Abstract

In this assignment, we explore the principal components of natural images and the compression of video frames. In the first task, we extract a large number of patches from natural images from the Berkeley segmentation dataset and visualize the principal components as images. In the second task we try to understand the bit rate below which it is better to downsample the video spatially by 2, compress the video and then upsample rather directly compress at the desired bit rate. Documented code will also be available at `https://github.com/vineeths96/Video-Compression-and-PCA`.

Problem 1: Principal components of natural images

In this section, we discuss about the principal components of natural images. We use the train images in the Berkeley segmentation dataset as our natural images. We extract 8×8 patches from each of these images and perform Principal Component Analysis (PCA) on the patches. We then visualize the principal components as images. We try with 100 (Fig 1), 1000 (Fig 2), 10000 (Fig 3), and 25000 (Fig 4) patches from every image (~ 200) in the dataset. Note that we consider random overlapping patches within an image. As we would expect, the higher the number of patches, better the results are. We observe that the results closely agrees with the current literature.

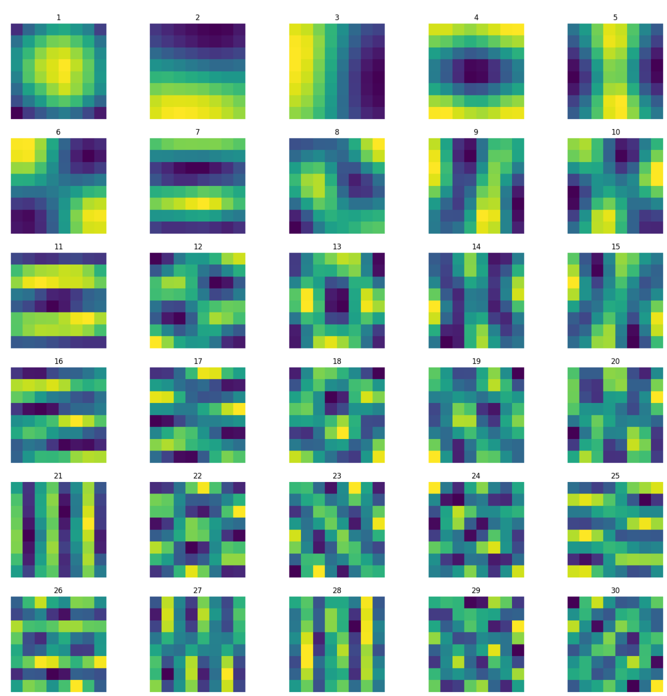


Figure 1: 100 Patches per Image

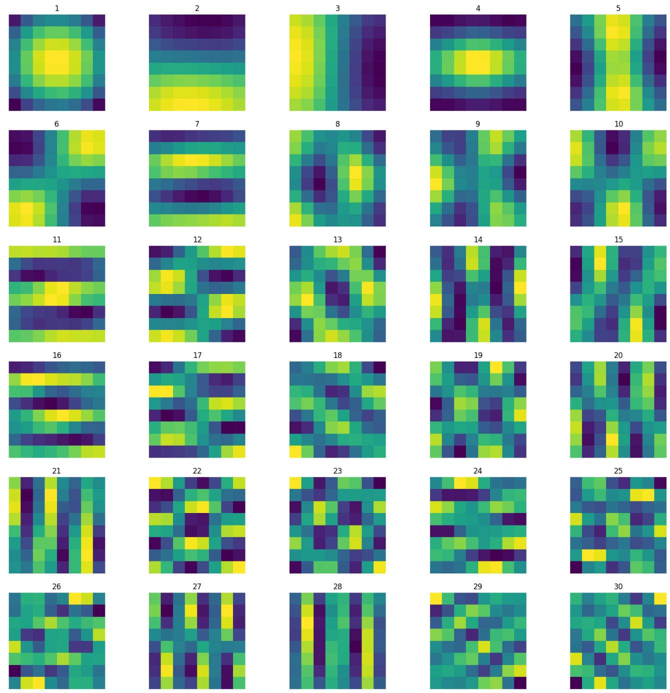


Figure 2: 1000 Patches per Image

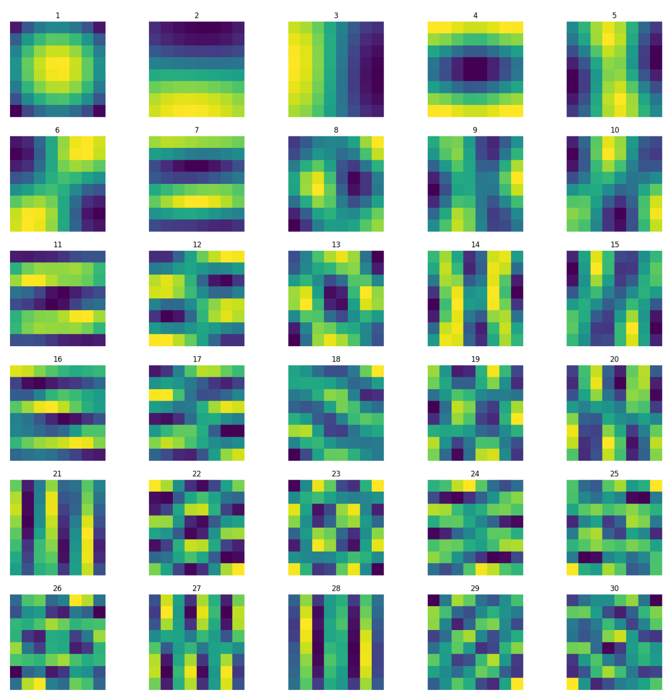


Figure 3: 10000 Patches per Image

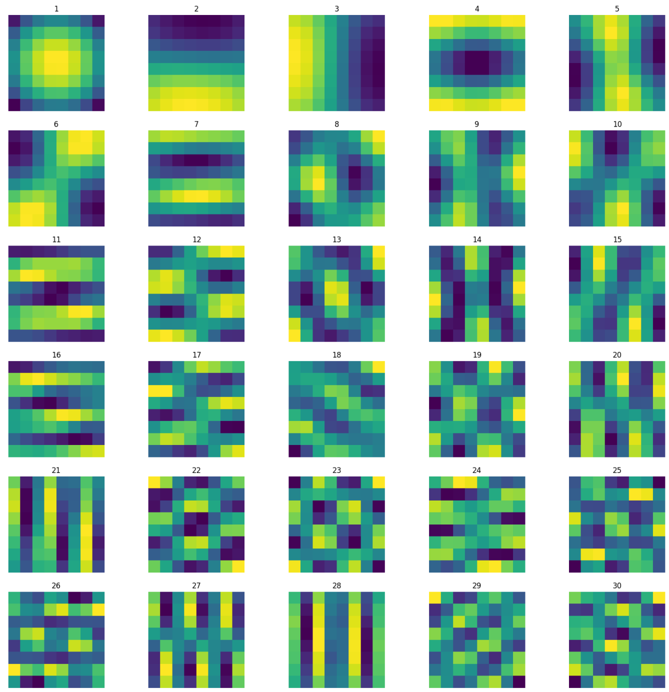


Figure 4: 25000 Patches per Image

Problem 2: Image compression and Resolution

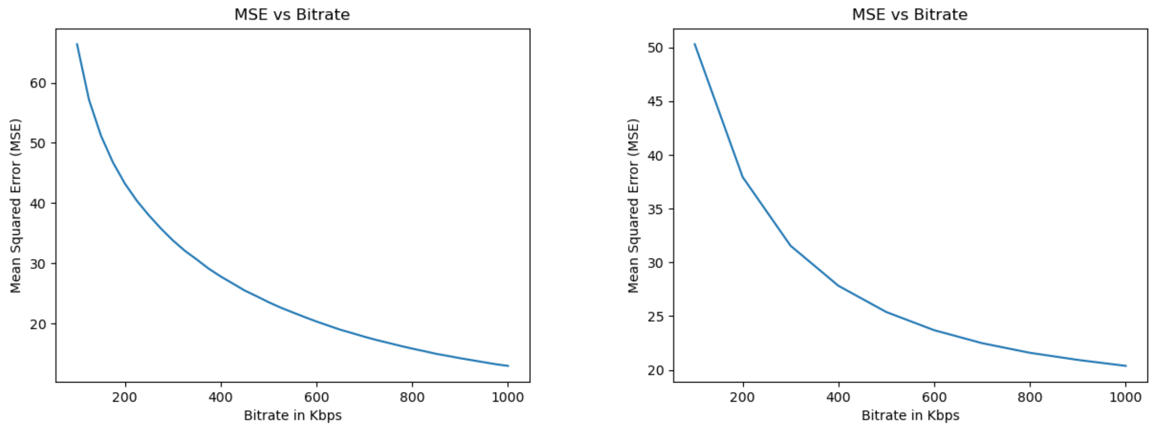
In this section, we discuss about the effects of bit rate used in video compression on the quality of the videos. More specifically, we try to to understand the bit rate below which it is better to downsample the video spatially by 2, compress the video and then upsample rather directly compress at the desired bit rate. We use the *pa1_25f ps.yuv* provided with this assignment and use H.264 as our compression scheme. We utilize the popular compression software *ffmpeg* to carry out our experiments. We compress all the videos with one I frame and rest being P frames. We measure the quality in terms of mean squared error in the luminance between the original frame and reconstructed frame.

The table 1 shows the settings we use for *ffmepg* for H.264 compression. We execute all the calls from Python using os.system() calls (in code). We choose the values for *keyint* and *min-keyint* such that we compress using one I frame and the rest being P frame. This is also verified by using the *ffprobe* (in code).

Table 1: Compression Settings

Bit rate	100 - 1000 Kbps (step 25)
Video Codec	H.264
Audio Codec	Copy (default)
Key Int	300
Min Key Int	300
No Scenecut	1
B Frames	0
Log level	Panic
All other settings	ffmpeg default

Fig 5 shows the plot between mean squared error and the bit rate for both the methods we explore. Fig 5a shows the plot where we compress and decompress the frames in the original dimensions. Fig 5b shows the plot where we downsample the frames spatially by 2, compress, decompress, and upsample the frames spatially by 2.



(a) Compress - Decompress (b) Downsample - Compress - Decompress - Upsample

Figure 5: MSE - Bit rate plot

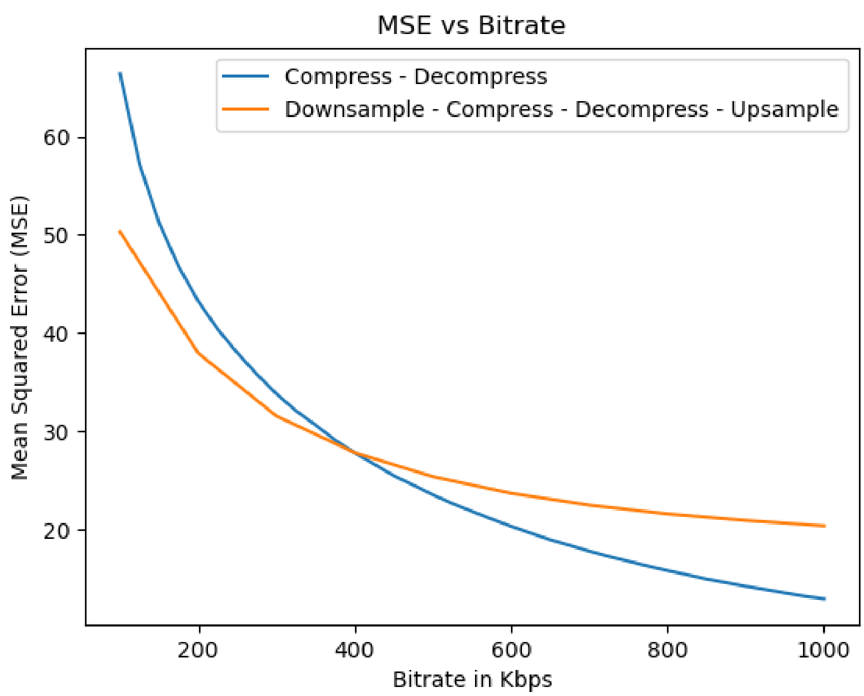


Figure 6: MSE - Bit rate

Fig 6 shows the plots between mean squared error and the bit rate of both the methods for comparison. Clearly, we can see that there is a switching between the two methods. The switching bit rate is around 400 Kbps — above which compression-decompression in original dimensions is better and below which downsampling - compression - decompression - upsampling is better.

Problem 3: Quality Assessment

I choose the option to take part in the subjective study on Image Quality Assessment of Low Light Restored Images.