Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Radioelectronics

Performance Analysis of Lossy Image Compression Algorithms

Jan Ošťádal

Supervisor: Ing. Karel Fliegel, Ph.D.
August 2020

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Ošťádal Jan** Personal ID number: **474444**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Radioelectronics**

Study program: **Electronics and Communications**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Performance Analysis of Lossy Image Compression Algorithms

Bachelor’s thesis title in Czech:

Analýza účinnosti algoritmů ztrátové komprese obrazu

Guidelines:

Provide an overview of current image codecs, including a comparison of the compression methods used and available implementations. For the selected application scenario, analyze and compare the performance of the selected codecs on suitable image data.

Bibliography / sources:

[1] Wu, H. R., Rao, K. R., Digital Video Image Quality and Perceptual Coding, CRC Press, 2006.
[2] Gonzalez, R. C., Woods, R. E., Digital image processing, Pearson, 2018.

Name and workplace of bachelor’s thesis supervisor:

Ing. Karel Fliegel, Ph.D., Department of Radioelectronics, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **12.02.2020** Deadline for bachelor thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

Ing. Karel Fliegel, Ph.D. Supervisor’s signature	doc. Ing. Josef Dobeš, CSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

I would like to thank my supervisor Ing. Karel Fliegel, Ph.D. for his patience with me in the challenging times this work was created in and also to my friends from VV Art Production for providing me some photographs used in this work.

Declaration

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Place:
Date:
Signature:

Abstract

The goal of this work is to examine compression quality and efficiency of 5 different image formats and their lossy compression. For this purpose from a database of hundreds high resolution photographs, 20 images were chosen based on their complexity and based on to what extent those images represent the typical image content on the internet, mainly on the social media. Those images were then compressed either into specific file size with the use of a Matlab script that was written for this work or by using a quality option that was a part of a codec. To describe the quality differences between the formats, several objective quality assessment metrics were used. As the last step those differences between individual metrics and file formats were described and compared between each other for specific images with the use of Bjøntegaard metric.

Keywords: Bjøntegaard's metric, BPG, HEIC, JPEG, JPEG 2000, Quality assessmen, WebP

Supervisor: Ing. Karel Fliegel, Ph.D.
Katedra radioelektroniky FEL,
Technická 2,
166 27 Praha 6

Abstrakt

Cílem této práce je prozkoumat kvalitu komprese a efektivitu 5 různých obrazových formátů a jejich ztrátových kompresí. Pro tento účel bylo z databáze stovky fotografií s vysokým rozlišením vybráno 20 obrázků na základě jejich komplexity a toho, v jaké míře tyto obrázky reprezentují typický obrazový obsah na internetu, zejména na sociálních sítích. Poté byly tyto obrázky komprimovány buď do specifické velikosti souboru pomocí skriptu v Matlabu, který byl pro tuto práci vytvořen, nebo pomocí nastavení kvality, které bylo součástí konkrétního kodeku. K popisu rozdílů v kvalitě mezi kodeky bylo použito několik metrik hodnocení kvality. Jako poslední krok byly tyto rozdíly mezi jednotlivými metrikami a mezi formáty porovnány pomocí Bjøntegaardovy metriky.

Klíčová slova: Bjøntegaardova metrika, BPG, HEIC, Hodnocení kvality, JPEG, JPEG 2000, WebP

Překlad názvu: Analýza účinnosti algoritmů ztrátové komprese obrazu

Contents

Figures

Tables

Introduction

Capturing moments was always a part of the human nature. Since the beginning of the human age humans have the need to record their achievements, important events or the beauty of their surroundings. In the modern era this purpose is bestowed upon photographs. With the development of digital cameras there is also a need for storing and reading the images on the computers and later on the internet and smartphones. The most common and known image format is JPEG which was created in 1992. Since then it has become the most common lossy compression format. However, today it is not the only format available for use whenever one is making a photo album, creating website or uploading images onto social media as Facebook or Instagram. People can choose from a large variety of formats and codecs and it is not always an easy task to choose the optimal codec/format for one's need. The goal of this work is to extend current awareness about modern compression formats and compare the efficiency and compression quality of those modern formats. This work should act as a recommendation on which file format is optimal for one's use.

The first chapter describes basic principles used in lossy image compression and represents file formats that are subject for testing. Second chapter explains how the reference images are chosen based on their complexity and how are those images processed with used codecs. It also describes which quality metrics are used and how the Matlab implementation of the assessment works. The results are discussed in the last chapter with examples of 2 particular reference images.

Chapter 1

Basic principles

The resulting product of lossy image compression contains less data then the original image and therefore saving capacity demand is reduced. However, this data loss leaves traces on the compressed images in form of more or less visible artefacts. This chapter covers basic principles used in compression techniques as well as an overview of image formats used for this work. Only basic information is given with a link to external literature and websites with more information about each principle and file format. The last section provides overview of already conducted researches on the matter of compression quality.

1.1 Transform coding

Transform coding in image processing is used to change pixels of the original image into the transform domain. The advantages of transform coding as well as more information about types of transform coding can be found in [1]. Those advantages are: easier application of a function on the transformed function, it may provide data reduction, the transform of an equation may be easier to solve, and the transform of a function may give additional information about the original function. [1]

Widely used are 2 transforms: Discrete cosine transform and Discrete wavelet transform.

1.1.1 Discrete cosine transform

Discrete cosine transform (DCT) convert an image into its equivalent frequency domain by partitioning image pixel matrix into blocks of size NxN, N depends upon the type of image [2]. It is similar to the discrete Fourier transform (DFT), but using only real numbers. The definition can be found at [2]. DCT has its derived form, discrete sine transform, which is sometimes used in compression.

1.1.2 Discrete wavelet transform

Wavelet transform uses wavelets instead of matrix of numbers to transform an image. To call a particular function a wavelet system, it has to fulfil three properties defined in [3]. Wavelet transform produces as many coefficients as there are pixels in the image. These coefficients can then be compressed more easily because the information is statistically concentrated in just a few coefficients. Unlike DCT, it captures both frequency and location information.[4]

1.2 Chroma subsampling

Compressing colour images has one more step to be done. It is usually using more bandwidth for luma information that for chroma. This can be done due to less sensitivity of human visual system to chroma. Usually the signal is divided into 1 luma component and 2 chroma components. The subsampling scheme is expressed as a three part ratio A:b:c A- horizontal sampling reference b- number of chrominance samples c- number of changes of chrominance samples between first and second row of A pixels.

Usually c must be equal to zero or b. There are a few special cases that have c equal to different number.

1. 4:4:4 Each of the three components have the same sample rate, thus there is no chroma subsampling.
2. 4:2:2 The horizontal chroma resolution is halved. This reduces the bandwidth of a signal by one third with very little visual difference.
3. 4:1:1 The horizontal colour resolution is quartered, and the bandwidth is halved.
4. 4:2:0 The horizontal sampling is doubled compared to 4:1:1, but the vertical resolution is halved. The data rate stays the same.
5. 4:1:0 This ratio uses half of the vertical and one-fourth the horizontal colour resolutions, with only one-eighth of the bandwidth of the maximum colour resolutions used.

Many more information about chroma subsampling can be found in these papers: [23] [24] [25]

1.3 Compression formats

For the assessment 5 lossy image formats were chosen. As a traditional and well known representative JPEG was chosen along with it's wavelet variation JPEG 2000. Two formats that are based on High Efficiency Video Compression (HEVC) are also presented, Bellard's Better portable graphics (BPG) and High efficency image file format (HEIC). The last used file format is WebP from Google. In the Table 1.1 are presented also other formats that are not presented on this work. The table does not represent every single lossy file format available. For example JPEG family has 10 formats and standards of image compression[5].

DCT Based	HEIF	BPG	JPEG/JFIF	WebP
Wavelet Based	DjVu	ICER	JPEG 2000	PGF

Table 1.1: Image lossy file formats

1.3.1 JPEG

JPEG is an acronym for the Joint Photographic Experts Group, which created the standard in 1992 [7]. As JPEG is also called the file format of JPEG/JFIF, whole compression method and standard. Since its introduction in 1992, JPEG has been the most widely used image compression standard in the world and the most widely used digital image format[5].

JPEG compression is based on the discrete cosine transform. The image is converted from RGB into YCbCr colour space and subsampled using one of the subsampling method described above. Than each channel is divided into block of 8×8 pixels and each block is converted by DCT and quantized [6].

1.3.2 JPEG 2000

Initially images have to be transformed from the RGB colour space to another colour space using Irreversible Colour Transform (ICT) resulting in YCrCr colour space or Reversible Colour Transform (RCT) with modified YUV colour space. Then the image is split into tiles, rectangular regions of the image that are transformed and encoded separately. Tiles can be any size. Once the size is chosen, all the tiles will have the same size. These tiles are then transformed using wavelet transformation, quantized and coded.

1.3.3 Better Portable Graphics

BPG is a lossy and lossless image compression format based on High Efficiency Video Compression format from Joint Collaborative Team on Video Coding (JCT-VC). As such, instead of macroblock it uses coding tree units (CTUs) which can use larger block structures of up to 64×64 samples and can better sub-partition the picture into variable sized structures. HEVC initially divides the picture into CTUs which can be 64×64, 32×32, or 16×16 with a larger pixel block size usually increasing the coding efficiency. The blocks are then transformed using derived form of discrete cosine transform (DCT). In addition, 4×4 luma transform blocks that belong to an intra coded region are transformed using an integer transform that is derived from discrete sine transform (DST). [13]

The source code of the bpgenc, bpgdec and bpgview command line utilities (for Linux) and the associated libbpg library, the source code of the Javascript decoder and binary distribution for Windows (64 bit only) are available online at [12]

1.3.4 WebP

Developed by Google[14], WebP was focused on reducing the size of images on websites and increasing their loading speed. Both lossless and lossy variants are available. WebP's lossy compression uses the same methodology as VP8 for predicting (video) frames. As such the algorithm divides the frame into smaller segments. within them, the encoder can predict redundant motion and colour information based on previously processed blocks. The redundant data can then be subtracted from the block. After being subject to a mathematically invertible Discrete Cosine Transform, the residuals typically contain many zero values, which can be compressed much more effectively. The result is then quantized and entropy-coded. Google provides number of resources, including precompiled WebP utilities and library for Linux, Windows and Mac OS X [15]

1.3.5 HEIF/HEIC

The High Efficiency Image Format (HEIF) is used by Apple's iPhone, often with .HEIC file extension. This image format is, like BPG, based on the High Efficiency Video Compression format.

1.4 Related research

This work is not the first research carried out on the matter of image compression performance evaluation.

Developers of WebP conducted their research to compare WebP and JPEG. In fact, they have carried out 2 experiments regarding those two formats. One in 2010[16] and the second one[17] later on to supersede the early study. The first study focused on measurement of compression achieved by those methods and analyzing the trade-offs between image sizes and compression. The second one studies the additional compression achieved by WebP at the same quality level of JPEG and analyzes SSIM vs bits per pixel (bpp) plots for WebP and JPEG.

Mozilla Corporation compared in 2014 several codecs[18]. Their study compares the compression performance JPEG, WebP, JPEG XR, and HEVC-MSP. In their study they used PSNR-HVS-M, Y-SSIM, MS-SSIM and RGB-SSIM metrics to compare the results of individual compressions.

Chapter 2

Compression and evaluation methodology

This chapter focuses on decribing the process of quality assessment. At first it describes how the proper images for testing are selected and what criteria are taken into account during the selection. Then it is explained how the compression of selected images is made and what metrics are used to describe quality of compressed images. Last part of this chapter shows the whole process implemented in Matlab.

2.1 Image selection

In order to properly test compression codecs, features and complexity of testing images should vary[39]. It is also more convenient when the source is in a file format that those codecs can read, because there is no need to transfer the source into different file format. For specific testing scenarios, the device which created the images should be also known. The source should also cover typical image data that one could find around them. A picture of almost infinite number of straight lines in various colour and angles between them can be found complex based on some points of view, however, such an image usually does not resemble anything that people create.

On the internet many image datasets can be found[37],[38],[39]. Those are primary used for other purposes than compression and, if not already compressed, which was not appropriate for this work, also require huge amount of free space, dozens or hundreds of gigabytes. That is also limiting. Datasets directly intended for image compression exist, however their content has often low resolution and is in gray-scale, which does not reflect typical picture data on the internet in the era this work is written in.

Therefore, it was decided to create a dataset specifically for this work. Three different devices were chosen to take testing images from, that is an older smartphone, newer smartphone and a professional digital camera. The selection was made with two different approaches, the first one based on image complexity of stored images in those devices, the second one based on typical content on the internet, specifically on social media.

2.2 Complexity

It is not an easy task to define image complexity, yet to measure one. An overview of complexity definitions from different point of view, like mathematical, aesthetic, verbal or based on amount of detail or intricacy, and different experiments already carried out in the past can be found in [29]. Also an interesting method of measurement is proposed there. However, those experiments required a subjective evaluation from participants, which is time consuming and difficult to organize. More objective and less time consuming method was required.

For selecting a proper image for the needs of this work, a method based on evaluation 3 different criteria was used. For the first criterion a mean value of entropy of each colour channel was computed. The first idea was to to see if there is a correlation between this value and the other two metrics, especially gray-scale entropy. After that the image was converted into a gray-scale domain. For the gray-scale version of the source image an entropy was calculated, which is the second value that was taken into account.

As a third value served the Spatial perceptual information. It's definition for N frames is represented in [30] as:

$$SI = max_{time}(std_{time}(Sobel(F_N))) \tag{2.1}$$

If we choose N=1, SI of a single frame is computed. The chosen testing image serves as this 1 frame. However, we lose the time domain over which the original SI is calculated. This calculation provided 1 number for the whole time series. Without the time domain, the calculation provided a vector of numbers. To substitute the loss of time domain, mean value of the vector was used instead.

A search algorithm then selected those photos which had one, two, or all of these criteria greater than a chosen value. Just one number could not be used, because higher colour entropy did not always mean higher gray scale entropy and vice versa and SI is not meant to be associated with entropy.

2.3 Typical internet content

Not every image that is posted on the internet meets the requirement of being complex. The parameters which matter in such choice could be author's subjective evaluation of the selected photo or it's purpose. And if the author is a professional with art education, photography enthusiast or an amateur, the final posted product can be different.

It was proven difficult to find a reliable scientific source that would summarize what can be seen on the internet, therefore a survey was conducted for the purpose of this work. People were asked to provide their opinion on the matter of typical internet images. They were to search through their digital albums and albums of their contacts on social media and give an overview of the content using as simple answers as they could. The most common descriptions were: "selfie", food, animals and nature. The answers provided 74 people aged from 16 to 25 years. If not presented, those types of images were added to source collection.

2.4 Reference Images


cmp (2).jpg


cmp (3).jpg


cmp (4).jpg


cmp (5).jpg


cmp (6).jpg


cmp (7).jpg


cmp (8).jpg

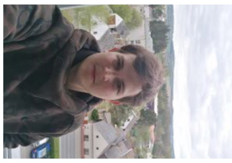

cmp (9).jpg

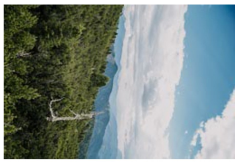

cmp (10).jpg

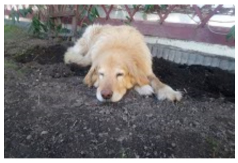

cmp (11).jpg


cmp (12).jpg


cmp (13).jpg


cmp (14).jpg

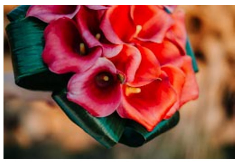

cmp (15).jpg


cmp (16).jpg


cmp (17).jpg


cmp (18).jpg


cmp (19).jpg


cmp (20).jpg


cmp (1).jpg

2.5 Compression testing

Using methods described above, 20 reference images were chosen. As for compression, in order to make testing fast it was important to find coders that could be executed via Windows command line. Apart from compression into HEIC format, all testing was made with Matlab code, which can be found as digital attachment of this work. Executable encoder for HEIC was not found, therefore images were compressed into this format with XnView MP[31], an image viewer with compression capabilities. This program supports other formats, including those used for this work, but using GUI is not convenient for used testing method[32]. WebP [15] and BPG [12] have official Windows distributions as executable files, thus there was no need for a third party program. For JPEG a program called ImageMagick [34], which uses *jpglib*, complete implementation comming directly from The Joint Photographic Experts Group [7]. Compression into JPEG 2000 was made using nconvert [33], which uses Open JPEG distribution of this format.

2.6 Compression process

First it was necessary to determine target bit rate of the compression. At first it was decided to take the largest image and compress it with the maximum and minimum quality setting used codecs were able to reach. However, this lead to inconclusive results for some images. On the other hand starting with extremely low bitrate would generate redundant data for larger images. Therefore, a compromise between those two options was made and the bitrate between 0.1 bpp to 1.5 bpp was chosen. Different steps between individual values were used, specifically from 0.1 bpp to 0.5 bpp with a step 0.02 bpp and then from 0.5 bpp to 1.5 bpp with a step 0.05. Google used in their second study 0.1 bpp to 1 bpp [17] and Mozilla used bitrate up to 3 bpp with starting bitrate ranging from 0.1 to 0.2 depending on the dataset used. As for BPG, it's binary distribution for Windows is able to compress images using quality settings. Images were compressed with every quality setting available, from 1 to 51 with a step of 1. Every other setting was not changed. Conversion into HEIC format was done with quality slider within XnView MP, from 1 to 100, using small step for lower quality values and bigger step for higher quality values. As in previous case every other setting stayed default.

We can see in the Table 2.1 that the only setting that was changed is the quality setting that ranges from 1 to 51, with 1 being the best quality and 51 being the worst. The other settings that could affect the final product of the compression remained unchanged.

BPG	Changed	Unchanged				
	Quality	Speed	Bit depth	Lossless	Quantizer	Color space
	1 to 51	Default	Default (8)	Default (No)	Default	Default

Table 2.1: BPG Settings

The only 2 settings that were available for HEIC codec can be seen in the Table 2.2. The quality of compression ranges from 1 being the worst and 100 being the best. Saving metadata remained unchanged, which means the metadata were saved if available.

HEIC	Changed	Unchanged
	Quality	Meta data
	1 to 100	

Table 2.2: HEIC Settings

The next three tables describe available settings for JPEG, JPEG 2000 and WebP. For all 3 codecs the only setting that was changed is the target filesize. Other 4 settings for JPEG and 3 settings for JPEG 2000 and WebP remained unchanged. All three codecs have also settings that modify the image but not the compression itself, like cropping, resolution changing or colour mixing, etc. None of those were used.

JPEG	Changed	Unchanged			
	Quality	DCT Method	Huffman table	Sampling	Quantizer
	Filesize	Default	Optimal	Unchanged	Unchanged

Table 2.3: JPEG Settings

JPEG 2000	Changed	Unchanged		
	Quality	Compress ratio	Colour space	Image modifiers
	Filesize			

Table 2.4: JPEG 2000 Settings

WebP	Changed	Unchanged		
	Quality	Alpha channel	Compression	Segments
	Filesize	Default = 100	Default = 4	Default = 4

Table 2.5: WebP Settings

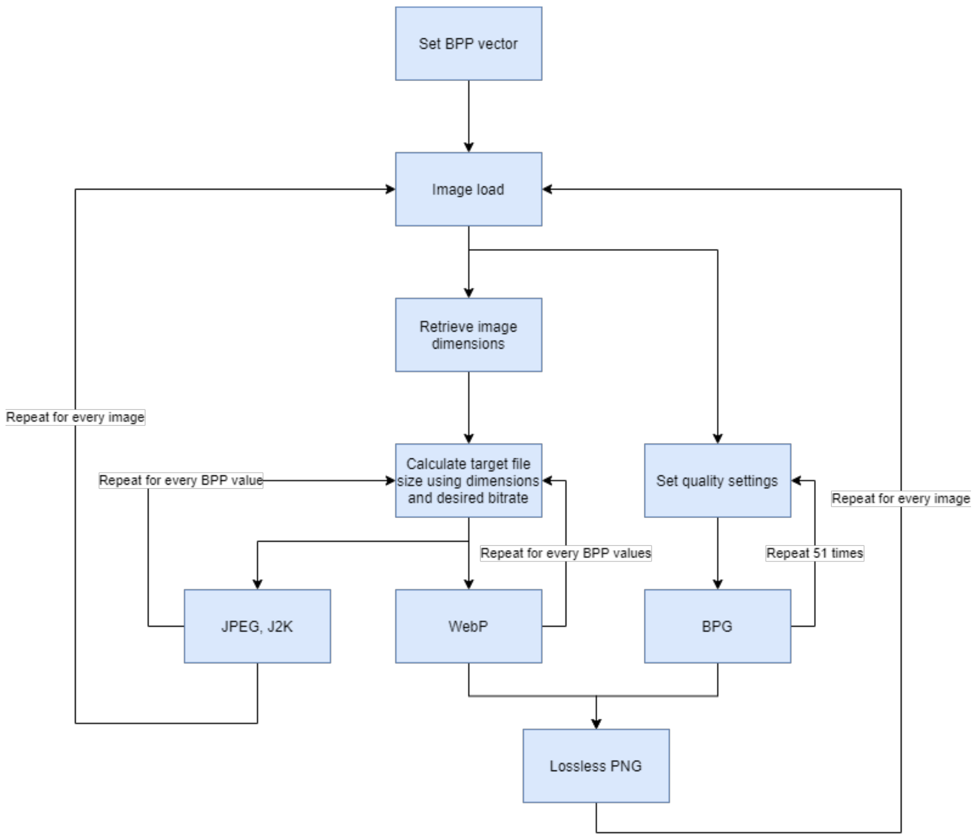


Figure 2.1: Compression diagram for WebP, BPG, J2K, JPEG

2.7 Quality Metrics

To determine effectiveness of compression, full-reference quality metrics were used and also Bjøntegaard's metric to compare the results for individual codecs and metrics. That is Structural similarity (SSIM) index, Multi-scale structural similarity (MS-SIM), Feature Similarity Index (FSIM), Peak signal to noice ratio (PSNR) and Mean-square error rate (MSE). Apart from FSIM and Bjøntegaard's metric Matlab has built-in function for image quality assessment[21]. As build-in functions, the exact algorithm Matlab uses to calculate these is not available. For FSIM and Bjøntegaard's metric external code was needed. SSIM, MSE and PSNR were introduced with Matlab versions 2014a and 2014b, while MS-SIM is available in version 2020a.

2.7.1 SSIM

Structural Similarity Index Method is a perception based model. In this metric, structures are patterns of pixel intensities, especially among neighboring pixels, after normalizing for luminance and contrast. Because the human visual system is good at perceiving structure, the SSIM quality metric agrees more closely with the subjective quality score[21]. It's metric combines local image structure, luminance, and contrast into a single local quality score[**?**].

2.7.2 FSIM and FSIMc

The phase congruency (PC), which is a dimensionless measure of the significance of a local structure, is used as the primary feature in FSIM. Considering that PC is contrast invariant while the contrast information does affect perception of image quality, the image gradient magnitude (GM) is employed as the secondary feature in FSIM. PC and GM play complementary roles in characterizing the image local quality. After obtaining the local similarity map, PC is used again as a weighting function to derive a single quality score [19]. The FSIMc is an extension of FSIM that incorporates chrominance information into the metric computation [20].

Authors of FSIM index published their official Matlab source code available for educational and research purposes[19]. This code acts as a typical Matlab function. Outputs of the functions are FSIM and FSIMc. The difference is that FSIM converts colourful images into gray-scale while FSIMc keeps the chrominance information and computes the index with it. With the algorithm colour images are converted into YIQ colour space and then the FSIM and/or FSIMc is computed as the authors describe in their work [20].

2.7.3 MS-SIM

Advanced version of SSIM called Multi Scale Structural Similarity Index Method (MS-SSIM) that evaluates various structural similarity images at different image scale[22]

Multi-scale structural similarity (MS-SSIM) index. The MS-SSIM metric expands on the SSIM index by combining luminance information at the highest resolution level with structure and contrast information at several downsampled resolutions, or scales. The MS-accounts for variability in the perception of image details caused by factors such as viewing distance from the image, distance from the scene to the sensor, and resolution of the image acquisition sensor[21].

2.7.4 Bjøntegaard's metric

Bjøntegaard's metric function was taken from Matlab file exchange[35]. This function is able to calculate Bjøntegaard's metric from more than 4 rate-distortion (RD) points, however only for 2 RD curves at once. Unlike the previous metrics, this metric does not serve as a distortion indicator, it only evaluates two different RD curves.

2.8 Matlab implementation

Every data computation and most of the compression was done using Matlab software. Source codes which carried out the instructions are available as digital attachments. Files containing those instructions are called *Complexity.m*, *Framework.m* and *testing.m*. Figure 2.2 shows how the individual parts of each file and each part of the code are connected. Every file is separated into different section, where every section is responsible for different part of the computation. I have written it in a way that every section should be executed manually. This allows the user to directly set different settings for every section of the code. For example changing target bit-rate or a step between bpp results in different dimensions of variable in which the computed data are saved. However the changes need to be done manually. *Testing.m* file was used to do additional operations for the work as increasing steps between individual bpp or changing the minimum bpp the compression starts at and then composing variables generated by both scripts into one that is used the *Framework.m*. For reproducing the results, the *testing.m* file is not needed, if the main code is modified in a way that does not require additional calculations. *Complexity.m* was used to select proper images from 3 different folders. The individual sections usually follow this pattern: First loop selects source image, second loop selects compressed image of that source image. The images are named in a way that allows to easily select them with loop variables. First loop variable that controls the source image selection is always k, second is usually i depending on what the individual section of the code is responsible for. The codes are available as a digital attachment in MAT folder.

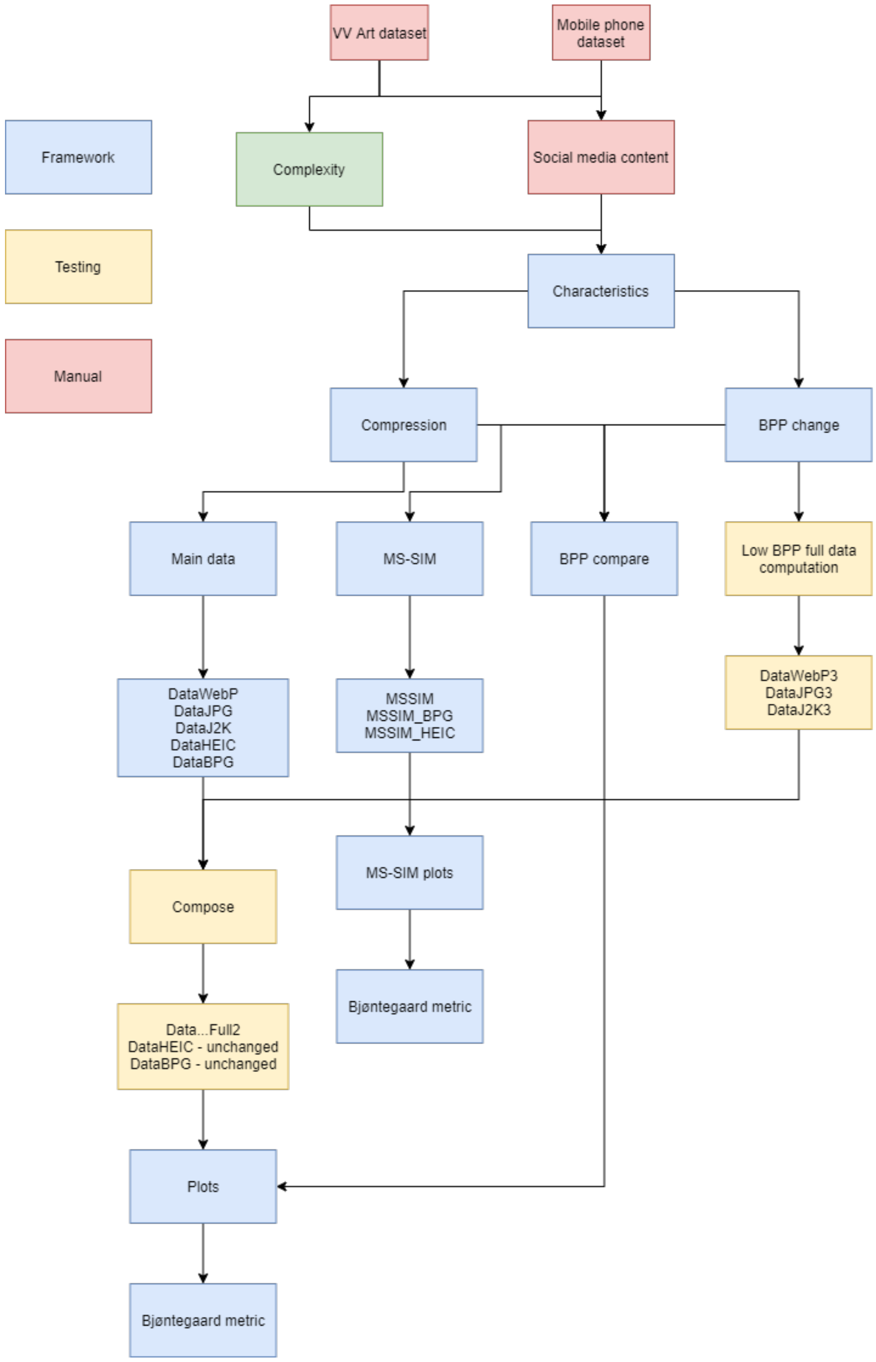


Figure 2.2: Matlab implementation diagram

2.8.1 Complexity based image selection

This code takes every image in a folder and computes complexity indicators which were described in section 3.2. Number of images in the folder is known and set in a variable. Instead of writing a static number, this could be changed to scan the folder for images. That way the code would be more resilient. First two sections of this file do the same calculations but for different folders.

At first there were 3 folders with images, however, one set of the images was protected by intellectual property law, therefore this set could not be used for this work. For every image RGB entropy, gray-scale entropy and spatial information is calculated and saved to an .xls file as described in 3.2. My implementation of the spatial information is somehow limited by a certain number where none of the 200 images reached higher value of SI than this number. Basically those images that hit this value were then chosen for compression. The file with all complexity values is scanned and those images that have not met certain criteria have their serial number deleted from the file. Those images that stayed were manually moved to a different folder and renamed. The whole processes is described in the Figure 2.3

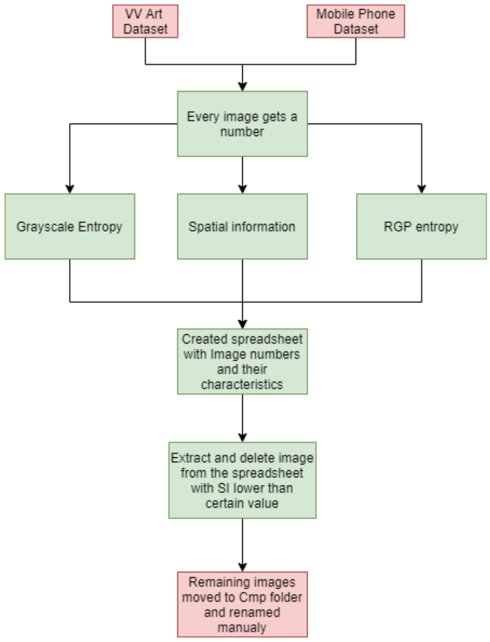


Figure 2.3: Image selection diagram

2.8.2 Characteristics

This part of the *Framework.m* file computes characteristics of the selected images. Those characteristics are YCbCr entropy, gray-scale entropy and spatial information. They are calculated in this section again because not every image was selected by the method described in 2.8.1.

2.8.3 Compression

Compression instructions can be found in *Image compression* part of the framework. At the beginning of this part desired bitrate is set in *bpp* variable. Instructions are carried out using three loops. At the beginning of the first loop dimensions of the image that is about to be compressed are read and based on their dimension and the chosen bitrate, target file value of compressed image is calculated. WebP and JPEG codecs accept the value in bits, while J2K codec accepts them in kilobytes. Also the first loop is responsible for choosing the base image, which is then compressed in the second loop for WebP, JPG, J2K and in the third loop for BPG. These two loops are nested inside the first one. The second loop controls target value of the compression command which is put together in Matlab and then executed via windows command line. The third loop changes quality settings for BPG codec.

2.8.4 BPP change

The chosen bit-rate did not cover lower values of smaller images. It was corrected in this part. The only two things that distinguish this section from the previous one are the bit-rate and folders in which are the compressed images saved. The bit-rate correction was done once more, that time in *testing.m* file. If the bit-rate is changed in *Compression* section, this part can be skipped. This section has 5 different nested loops. The reason for this is that for WebP, JPEG and J2K 3 different sets of bit-rate were used and therefore the files are saved in 3 different places. Last 2 loops are for BPG and HEIC.

2.8.5 BPP compare

This part analyzes the compressed images and saves their real bit-rate. It takes the dimension information from a source image, size of a compressed image and calculates real bit-rate that this compressed image has. It also saves the variables with used bit-rates. *Name* and *Path* variables controls the path to the images, those should be changed if the user works with different folders.

2.8.6 Main Data

For every compressed image, here are the quality metrics calculated. The results are then saved as a .mat file and as .xls table for potential further work and as a back-up. In the *Framework* file only images created with the first bit-rate estimation are processed, the rest is done in *testing* file. In the Figure 2.2 it is the left branch that generates DataWebP, etc. Otherwise this section is straight-forward, because calculations are made with built-in and downloaded functions with exception of HEIC and BPG with their own loops to process their compressed images. Those processes take time, the more images we have, the more time it requires. The calculations of the set created with my settings took approximately 50 hours[1].

[1]Hardware: 16GB RAM, NVIDIA GeForce 1060 6GB, Intel Core i5-9300H (2.4GHz, TB 4.1GHz, HyperThreading)

2.8.7 MS SSIM and MSSIM Plots

Those two sections are written in a same manner as *Main data* and *Plots*. The only reason those two are separated is that MS SSIM has become available as a part of new Matlab version and I was able to get a licence for it, while the rest was already implemented in previous version. For some reason calling a path in a same way as in *Main data* did not work, therefore I choose using another function to do so.

2.8.8 Plots

In this part results calculated in *Main Data* are visualized. At the beginning of this part, previously obtained data are loaded. Also data for BPG are inverted for easier handling and sorted, because the codec's setting for quality had number 1 assigned for the best quality, therefore in the variable with corresponding data those data are sorted from best to worse quality, while for the rest of the variables it is vice versa. The part itself is then divided into 3 section. First section creates graphs showing results of all codecs for 1 image and 1 metrics. The second part creates plots for 1 codec and multiple images and the last part shows differences between desired bit-rate and a real bit-rate that the codecs were able to get.

2.8.9 Bjøntegaard

This part uses Bjøntegaard funcion to compare the results and support the results given by the plots, especially those with less detailed resolution. It operates in two modes: 'dsnr' for average metric difference and 'rate' for percentage of bitrate saving between two data sets. The results are compared to JPEG and then saved as an .xls file and .mat files for both modes.

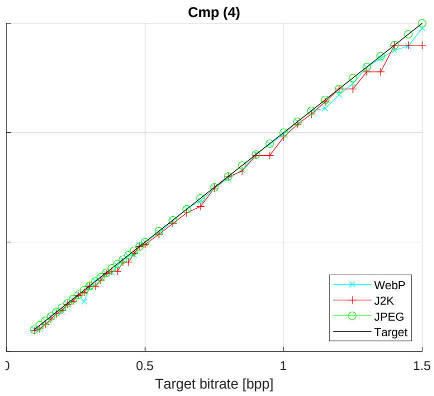
Chapter 3
Results

For the results I chose to demonstrate 2 images, numbered 4 and 14, see Reference images in Attachments, Image Cmp (4).jpg and Cmp (14).jpg. Their complexity parameters can be found in the Table 3.1 The reason for this choice is that number 4 represents a picture with which codecs had a little difficulty, while number 14 is the the opposite. Each image has a section presented in this chapter. Those sections have a general description of that image, a table with complexity values for the image and average values, a plot designed to compare bit rate between a desired and a real bitrate, plots of three metrics and a table of Bjontegaard metrics. As for the rest of the images, all the plots and tables can be found as a digital attachment in Plots and Tables folders. The next Table, Table 3.1 shows calculated complexity results for images

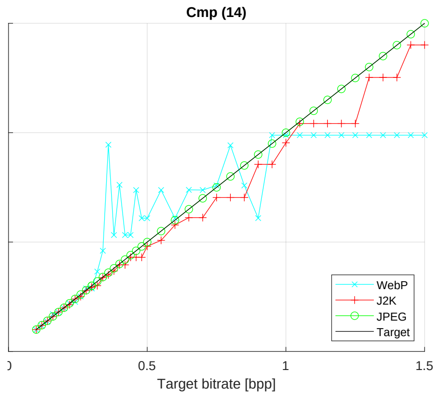
Image #	YCbCr Entropy	Grey Entropy	SI
1	6,61469	7,81376	0,52705
2	6,63443	7,91391	0,52705
3	6,51045	7,89597	0,52705
4	6,36768	7,22102	0,52655
5	6,85215	7,77581	0,52705
6	6,43910	7,82624	0,52705
7	6,36984	7,65049	0,52705
8	6,29704	7,81483	0,52704
9	6,60032	7,73479	0,52705
10	6,58425	7,65304	0,52705
11	6,78808	7,75108	0,52705
12	6,09207	7,69402	0,50105
13	6,66694	7,38118	0,40575
14	5,92230	7,59106	0,49121
15	6,51819	7,65701	0,52705
16	5,92138	7,65933	0,52702
17	6,84155	7,66911	0,50503
18	6,90433	7,71589	0,50473
19	5,75689	7,62818	0,44009
20	7,32225	7,46233	0,45101
Average	6,50020	7,67545	0,50750

Table 3.1: Complexity of reference images

Comparing bitrate is only related to the codecs that allow us to compress into a specific bitrate or filesize. In this case those codecs are WebP, J2K and JPEG, because compression involving HEIC and BPG formats were done with the quality option, not target file size, because that option is not available for these two codecs. The results are shown as a plot with bitrate values on both axis. The optimal outcome is a straight line with the same values on X and Y axis represented by a black line. The closer the codec could get to the optimal outcome the more precise that codec is and gives us more control over what we want. The offset from optimal line does not necessarily mean the codec is bad in general. It just means it had trouble compressing that specific image to that specific bitrate.



(a) : Bitrate comparison of Image 4



(b) : Bitrate comparison of Image 14

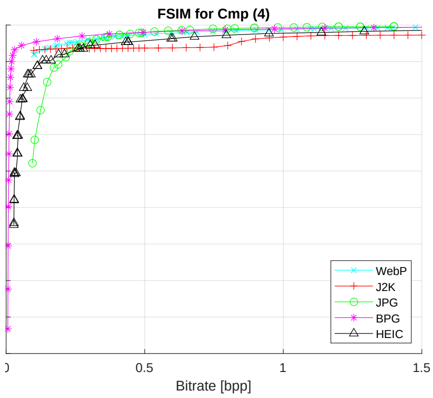
Figure 3.1: Bitrate comparison

3.1 Image 4

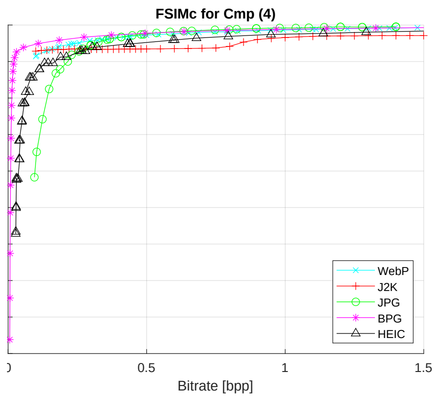
This image represents a typical internet content, in this case food, Cmp (4).jpg in the Reference images. It was taken with Canon EOS 60D. It's entropy values are below average, while SI is above average as can be seen in the table 3.1. From all of compressed images this is one of the complex ones even visually. From the bit-rate plot 3.1a we can tell that the codecs had a little difficulty compressing it into desired bit-rate. JPEG codec had no problem at all while the other two used a slightly less bpp that wanted.

Figure 3.2b shows FSIMc metric for this image. We can see that for this image, we could chose lower bpp to start with to see the starting trend BPG and HEIC provides. However, with another picture we would have more data then needed. From the point that all codecs start to have data, JPEG compression provides the lowest quality, even though the values of FSIMc are still quite high. However it quickly overcomes HEIC and J2K compressions almost at the same point, then for even higher bitrate it achieves higher FSIMc than WebP and BPG. HEIC starts with higher metric then JPEG but lower then the rest. It only overcomes J2K, which apart from the beginning reaches the lowest FSIMc. WebP starts on lower values that J2K, however, that quickly changes. For higher bitrate it reaches the similar values to BPG, but does not reach higher. As for BPG codec, it starts with the highest metric and keeps at it. As mentioned above, the only codec that reaches higher FSIMc for higher bitrate is JPEG. When we compare the FSIMc and

FSIM plots, we can notice that FSIMc values are slightly lower then those on FSIM plot. The main reason for this difference could be sub-sampling or overall handling of colour channels during the process of FSIMc calculation.



(a) : FSIM of Image 4



(b) : FSIMc of Image 4

Figure 3.2: FSIM and FSIMc plots for Image 4

Multi-scale SSIM can be seen in Figure 3.3a. Apart from Figure 3.2b the curves show more gradual increase of the metric values. Another difference is that JPEG achieves much lower values than the rest and J2K has second highest values. HEIC and WebP both achieve similar values with HEIC having slightly higher for the same bitrate. As in previous figure, MS SIM metric is also highest for BPG with only J2K getting almost the same values for medium bitrate.

The last of the presented metrics for this image is SSIM. This plot resembles the overall trend in Figure 3.3a with sligtly lower maximum values. Another difference is that the lowest values does not have JPEG, but J2K, which has second highest values of MS-SIM, and JPEG having second lowest values instead. Also WebP only reaches same or lower, but never higher than HEIC.

The tables 3.2 and 3.3 show the results of Bjøntegaard's metric calculations. All of the results are compared to JPEG codec. The first table shows average bitrate saving for equivalent quality on the range of quality levels that are presented in both compared RD-curves. It takes one quality value and compares the bitrate needed by compared codecs to achieve this quality value. We can see that BPG can reach lower bitrate for equivalent that the rest. Strangely the result of J2K codec according to MS-SSIM metric is opposite of what we can see in the other metrics. However, the rate metric is inconclusive for BPG and HEIC, since the compression into those formats was done by changing quality factor over all possible qualities available while for the rest BPP/Filesize setting was changed. Moreover, according to [36] the BD-rate metric does not necessarily represent average coding efficiency for all test points involved in the actual test and the test should be designed in a way that the distortion overlap between the two tested codecs covers a range of qualities of interest for specific application. In the used method, the quality settings cover the whole bitrate the codecs are able to compress into, while the filesize setting cover only a specific bitrate. This problem is apparently not treated in the Bjøntegaard's metric Matlab code, therefore the function bitrate and quality metrics inputs should be those that overlap with each other.

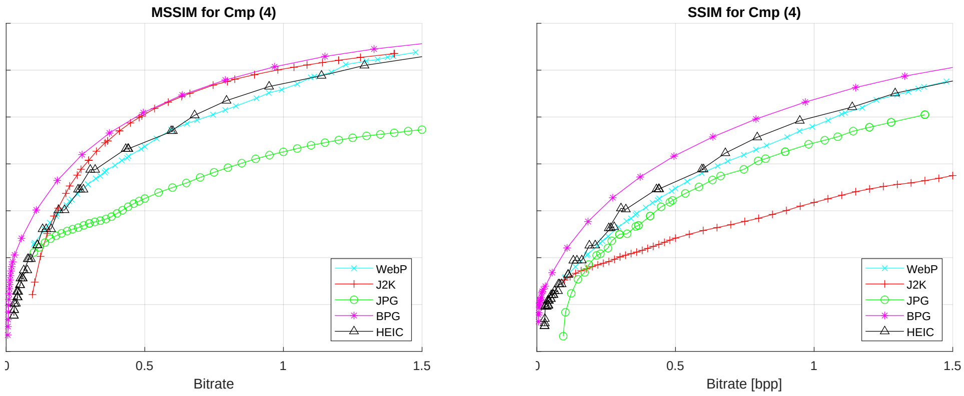
The second table shows quality improvement for equivalent bitrate. This metric takes a bitrate value and compares the quality achieved by the codecs on that bitrate. According to this table, the overall improvement in quality is better for every codec, apart from SSIM, MSE and PSNR metrics for J2K codec. Even though we can see in the Figure 3.2b that JPEG reaches higher values that the rest, the difference for higher bitrate is less significant then the difference for lower bitrate.

	FSIM [%]	FSIMc [%]	MS-SSIM [%]	SSIM [%]	MSE [%]	PSNR [%]
WebP	-19,46	-23,99	-49,33	-16,92	-22	-20,3
J2K	127,53	108	-51,38	67,26	58,59	65,10
HEIC	-45,99	-47,31	-50,53	-37,69	-43,19	-36,17
BPG	3624,75	3699,88	998,08	782,67	1268,97	632,10

Table 3.2: Bjontegaard's bitrate saving

	FSIM [-]	FSIMc [-]	MS-SSIM [-]	SSIM [-]	MSE [-]	PSNR [-]
WebP	0,000714	0,00085	0,018	0,007	-5,14	0,58
J2K	0,0003	0,00045	0,021	-0,011	3,99	-0,55
HEIC	0,00056	0,00066	0,019	0,011	-7,39	0,93
BPG	0,0029	0,0031	0,079	0,076	-41,9	9,12

Table 3.3: Bjontegaard's metric improvement



(a) : MS-SIM plot

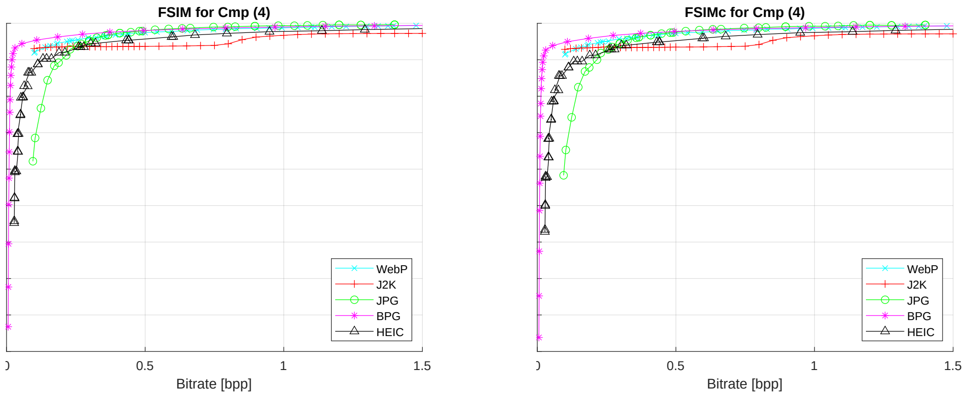
(b) : SSIM plot

Figure 3.3: MS-SIM and SSIM plots

3.2 Image 14

This image represents a selfie type of internet content, see Reference images, Image Cmp (14).jpg. It was taken with Huawei P30 lite mobile phone. It's statistics are bellow average according to Table 3.1. Despite the values in the table, WebP and J2K codecs have significant problem compressing the image to desired bitrate. For the lowest bitrates this problem does not occur. While J2K codec is getting the same or higher bitrate, same thing does not apply for WebP codec, where the values look more or less random. Because the real bitrate on the X axis was used to create those plot, this randomness shows it's effect in those plots. The problem could be caused by bitrate control mechanism of the codec. This could be a potential subject to further research.

From the FSIMc Figure 3.4b we can only see details for the low bitrate values. At the point where all curves are starting to be presented we can tell that BPG reaches the highest values, WebP has the second highest, then it is HEIC, then J2K and then JPEG with the lowest values at that point. For the higher bitrate there is not much we can see, because overall FSIMc values are very high for all codecs.



(a) : FSIM of Image 14 **(b) :** FSIMc of Image 14

Figure 3.4: FSIM and FSIMc plots for Image 14

In the Figures 3.5a and 3.5b we can see that the values differ for JPEG codec, for MS-SIM the JPEG as the lowest values, while the same thing goes for J2K for SSIM metric. Also the consequence of WebP codec having problem with specific compression is more visible on those plots that on the previous one. Instead of going on to high bitrate, the curve stops at some point and starts going back to lower bitrate, making the curve thicker in the area. This occurs in the other plots that are not shown here.

Strangely what we can see in the Figures is not reflected by Bjøtnegaart's metrics in Tables 3.4 and 3.5. Comparing those tables with FSIMc metric does not lead to any conclusion because of the high metric values and the curves overlaping with each other. However, according to the tables, better than JPEG in bitrate saving are J2K and BPG codecs and better in metric improvement are WebP and J2K codecs only. But still the differences between the individual values in the the table 3.5 are minimal compared to what we can see in the Table 3.3. The other two plots show more details between the curves.

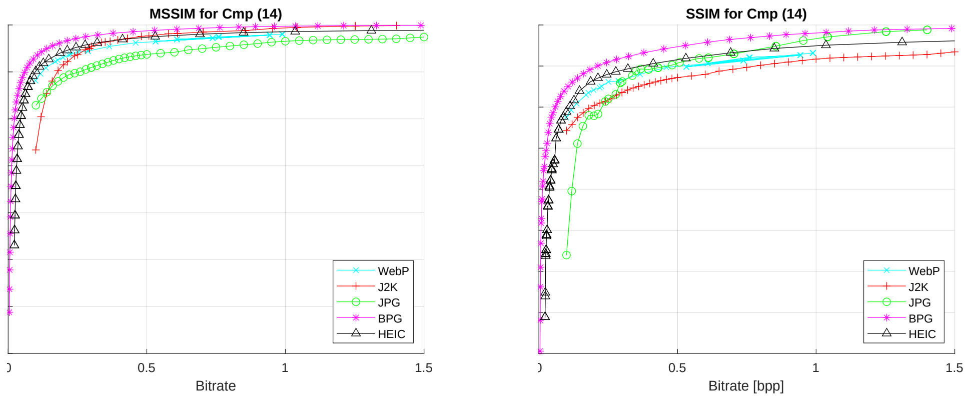
The main thing that is not reflected with Bjøntegaard's metric is that BPG reaches higher MS-SIM and SSIM metrics with lower bitrate while in the tables the values are negative, which indicates the opposite of what we can see.

	FSIM [%]	FSIMc [%]	MS-SSIM [%]	SSIM [%]	MSE [%]	PSNR [%]
WebP	-4,02	-37,61	-49,69	-16,55	-17,56	-8,42
J2K	23,94	10,26	-31,78	16,73	22,49	39,06
HEIC	-59,8	-61,59	-54,65	-55,12	-47,97	-25,52
BPG	103,39	141,35	-29,51	71,65	75,01	-83,45

Table 3.4: Bjøntegaard's bitrate improvement

	FSIM	FSIMc	MS-SSIM	SSIM	MSE	PSNR
WebP	0,00019	0,00024	0,0034	0,0044	-0,83	0,41
J2K	8,626E-05	0,00014	0,0019	0,00049	0,15	-1,73
HEIC	-4,73E-05	-1,29E-05	0,004	0,005	-0,42	-0,73
BPG	-0,00051	-0,00054	-0,0079	-0,01	2,84	-4,6

Table 3.5: Bjøntegaard's metric improvement



(a) : MS-SIM plot

(b) : SSIM plot

Figure 3.5: MS-SIM and SSIM plots

3.3 Overall results

What we can see in the previous sections, especially in Section 4.2, generally occurs with the other images. The individual results differs based on which metric was used, however, there are some similarities among them. The difference between MS-SIM and the other metrics wherever JPEG or J2K has the lowest values persist with every image. The next thing is that the curve representing HEIC is always uneven, especially on lower bitrate, and the curves have similar trend to WebP.

3.3.1 JPEG

From the tables of metric improvement in the attachments we can tell that overall most metrics have positive values, meaning the metric improvement is higher than JPEG codec. There are some cases for which this is not true.

For 10 specific images J2K provides lower quality than JPEG according to SSIM metric and for 3 cases for FSIM metrics. There is also 1 case in which J2K has negative value for FSIMc metric, image number 13. For other metrics we can find less negative values. For 1 image there are 2 codecs, BPG and HEIC, with negative results, for image number 14 and every metrics for BPG and FSIM and FSIMc for HIEC. There is only 1 case in which only BPG has negative SSIM.

For MS-SIM metric improvement, there are only 5 cases where one of the codecs has negative values, meaning the improvement of that codec is worse than improvement of JPEG. Three times it is for J2K codec for images 10, 12, and 15 and twice for BPG codec for images 14 and 19. Other than that, we can find 8 images where every single metric reaches higher quality improvement than JPEG.

Overall the course of the curves for JPEG starts with the lowest metric value on it's starting bitrate, however for higher bitrate it overcomes J2K for SSIM, FSIM, FSIMc metrics, in some cases even HEIC and reaches similar quality as WebP. For the highest bitrate values the curves become indistinguishable from WebP and BPG for FSIM and FSIMc metrics. As far as MS-SIM goes, according to this metric, JPEG starts with higher values than J2K, but is quickly overcome by it, or starts at far lower quality then the rest of the codecs and does not reach the rest.

The used codec achieved zero difference between desired bitrate and real bitrate for every single reference and compressed image.

3.3.2 WebP

From the three codecs that were able to compress into a set bitrate, WebP starts with higher metrics values of those three. The curves on the plots also start most of the time with similar value to HEIC and for some images those two have almost identical courses of the curves. Depending on the metric the curve can be seen lower than HEIC, but most of the time it is higher or similar to it. It also gets close to or even higher then BPG, but overall according to the tables it gets mostly lower than that.

For MS-SIM metric in 12 cases the quality improvement is higher then HEIC, in other 2 cases the values are nearly the same. However, for FSIMc metric it is higher in 10 cases, for FSIM in 10 and for SSIM for 8 cases only.

What happened with compression precision of Image 14 happened also with 4 other images, numbered 17, 18, 19 and 20 impacting the measurement of Bjontegaard's metric as well. But according to comparison plots, overall this codec can achieve less difference between desired bitrate and real bitrate the compressed images actually have.

3.3.3 J2K

Depending on the metrics, the curves for this codec start between WebP and JPEG or bellow JPEG. According to MS-SIM metric it has the lowest improvement of all other formats, with exception image 4, where it has second highest values. In 3 cases, the MS-SIM metric goes into negative numbers, which means the quality improvement is worse than JPEG.

The other metrics differs with values for J2K. In 10 cases the SSIM metric for J2K has negative numbers but only in 3 cases for FSIM and in 1 case for FSIMc metrics. On the other hand, it reached second highest FSIM metric in 4 cases and in 3 cases it achieved the same with FSIMc.

As for the compression precision, the thing that happened to WebP also occured to J2K, however, with less significant impact on the actual plot. Overall it is the least precise of the three codecs.

3.3.4 HEIC

Even when the compression was done with increasing quality setting with each step, we can see on the plots that the bitrate decreases at some points, hovever, the quality metric increases between the steps, leaving the curves with teeth like, but overall ascending course.

For the quality metrics, HEIC achieves seconds highest improvement of FSIM and FSIMc for Image 17, on the other hand, negative values for Image 14, however, it has the highest SSIM improvement for said image. In 12 cases the SSIM values are higher then WebP has. For 10 images the FSIM and FSIMc metrics have better improvement then WebP. For MS-SIM metric, there are 6 cases with HEIC having higher values than WebP and 2 cases with similar values. Comparing the metrics with J2K, we can find only 1 case where HEIC has worse SSIM result than J2K, for 15 images it is better in FSIM metrics, but only in 9 cases it has better FSIMc metric.

Used codec does not allow us to compress into specific bitrate or filesize, therefore there is no need to evaluate how close the codec can compress reference images to desired filesize.

3.3.5 BPG

There are only 3 cases where the BPG does not have the highest MS-SIM values and 1 case for the other metrics. Overall on the plots the curves hold themselves above the others. On the bitrate where other codecs start, it has the highest metric, most of the time unchallenged and if so, then only by WebP or J2K for some images and some bitrate or because the bitrate is so high that the curves are indistinguishable one from another. So according to tables and plots, this format gives the best quality for low bitrate.

As in previous case, this codec does not allow us to compress into specific bitrate or filesize, therefore there is no need to evaluate how close the codec can compress reference images to that filesize

Chapter 4
Conclusion

In this work I have studied how lossy image formats and publicly available implementations of related codec on a dataset of high quality images. The images in the dataset were taken either by a professional camera or a mobile phone to represent a different media content, one from a professional area, the other one from social medias.

Basic principles of those formats were described in Section 1.3 The dataset was then reduced to 20 reference representative images to decrease computational demands. To do so spatial information was calculated and based on this some of the reference images were chosen. The reference images were then filled with images typical for social media. More details on how the dataset composing and images analysis is described in the first 3 sections of chapter 2.

Then it was necessary to compress the reference images with the use of available codecs that were chosen to perform the compression. The compression was carried out with a Matlab script for 4 formats and a GUI program for one format. How the script works is described by Section 2.8.

Based on the prior art reliable objective image quality assessment metrics were selected and used to compare the quality of a compressed image with the reference. The resulting quality metrics were then plotted into graphs in a form of rate-distortion curves. To get more detailed results RD curves were then analysed with Bjøntegaard's metric also implemented in Matlab. The last chapter discusses results for two specific reference images as well as the overall differences achieved among the formats that were subjects to the performance analysis.

For further research Bjøndegaard's metric could be calculated for sections of the RD curves. Also finding a correlation between complexity metrics and average objective image quality assessment might be useful if there is any.

Bibliography

[1] BAIRAGI, V. K., A. M. SAPKAL a M. S. GAIKWAD, 2013. *The Role of Transforms in Image Compression.* Journal of The Institution of Engineers (India): Series B [online]. 94(2), 135-140 [cit. 2020-08-05]. DOI: 10.1007/s40031-013-0049-9. ISSN 2250-2106. Available at: http://link.springer.com/10.1007/s40031-013-0049-9

[2] AGARWAL, N. 2014. *Application of DCT in image processing.*[online] Available at: https://www.researchgate.net/publication/294580216

[3] TOUFIK, Bouden a Nibouche MOKHTAR, 2012. *The Wavelet Transform for Image Processing Applications.* In: Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology [online]. InTech, 2012-04-04 [cit. 2020-08-05]. DOI: 10.5772/35982. ISBN 978-953-51-0494-0. Available at: http://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/the-wavelet-transform-for-image-processing-applications

[4] AKANSU, A.N. SERDIJN, W.A. SELESNICK, I.W. *Wavelet Transforms in Signal Processing: A Review of Emerging Applications.* Physical Communication, Elsevier, vol. 3, issue 1, pp. 1–18, March 2010. Cit. 14.2. 2020. Available at: https://web.njit.edu/ akansu/PAPERS/ANA-IWS-WAS-ELSEVIER%20PHYSCOM%202010.pdf

[5] CAPLAN, P., 2013.*What Is a JPEG? The Invisible Object You See Every Day* [Online]. SEPTEMBER 24, 2013. Cit 14.2. 2020. Available at: https://www.theatlantic.com/technology/archive/2013/09/what-is-a-jpeg-the-invisible-object-you-see-every-day/279954/

[6] TRINKWALDER, A. 2016. *JPEG: 25 Jahre und kein bisschen alt*[online]. 07.10. 2016. Cit. 14.2. 2020. Available at: https://www.heise.de/newsticker/meldung/JPEG-25-Jahre-und-kein-bisschen-alt-3342519.html

[7] *Overview of JPEG*[online]. Available at: https://jpeg.org/jpeg/

[8] *Overview of JPEG 2000*[online]. Available at: https://jpeg.org/jpeg2000/index.html

[9] Van der KNIJFF, Johan. *JPEG 2000 for Long-term Preservation: JP2 as a Preservation Format*[online]. OI:10.1045/may2011-vanderknijff. Available at: http://www.dlib.org/dlib/may11/vanderknijff/05vanderknijff.html

[10] DUFAUX, Frederic, Gary J. SULLIVAN a Touradj EBRAHIMI, 2009. The JPEG XR image coding standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine* [online]. 26(6), 195-204 [cit. 2020-08-05]. DOI: 10.1109/MSP.2009.934187. ISSN 1053-5888. Available at: http://ieeexplore.ieee.org/document/5230820/

[11] International telecommunication union. *T.832 : Information technology - JPEG XR image coding system - Image coding specification*[online]. Available at: https://www.itu.int/rec/T-REC-T.832-201906-I/en

[12] Bellard F.,2018 *BPG Image format* [online]. Cit. 14.2. 2020, Available at: https://bellard.org/bpg/bpg_spec.txt

[13] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand. *Overview of the High Efficiency Video Coding (HEVC) Standard*". In IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012. DOI: 10.1109/TCSVT.2012.2221191. Cit. 14.2. 2020. Available at: https://ieeexplore.ieee.org/document/6316136

[14] *A new image format for the Web* [online]. Google Developers. Cit. 14.2. 2020. Available at: https://developers.google.com/speed/webp/docs/compression

[15] *Downloading and Installing WebP* [online]. Google Developers. Cit. 14.2. 2020. Available at: https://developers.google.com/speed/webp/download

[16] *Comparative Study* [online], 2010. Dublin: Google [cit. 2020-08-06]. Available at: https://developers.google.com/speed/webp/docs/c_study

[17] *WebP Compression Study* [online]. Dublin: Google [cit. 2020-08-06]. Available at: https://developers.google.com/speed/webp/docs/webp_studyy

[18] Lossy Compressed Image Formats Study [online], 2014. Berlin: Mozilla Corporation [cit. 2020-08-06]. Available at: https://web.archive.org/web/20160311235158/ http://people.mozilla.org/ josh/lossy_compressed_image_study_july_2014/

[19] XUANQIN, Mou. ZHANG, David. ZHANG, Lei. ZHANG, Lin. *FSIM: A Feature SIMilarity Index for Image Quality Assessme* [online]. In IEEE Trans. Image Processing, vol. 20, no. 8, pp. 2378-2386, 2011. Cit. 9.6. 2020. Available at: https://www4.comp.polyu.edu.hk/ cslzhang/IQA/FSIM/FSIM.htm

[20] LIN ZHANG, LEI ZHANG, XUANQIN MOU a D. ZHANG, 2011. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE*

Transactions on Image Processing [online]. 20(8), 2378-2386 [cit. 2020-06-09]. DOI: 10.1109/TIP.2011.2109730. ISSN 1057-7149. Available at: http://ieeexplore.ieee.org/document/5705575/

[21] *Image Quality Metrics* [online]. The MathWorks, Inc: © 1994-2020. Cit. 9.6. 2020. Available at: https://www.mathworks.com/help/images/image-quality-metrics.html

[22] WANG, Z., E.P. SIMONCELLI, A.C. BOVIK a E.P. SIMONCELLI. Multiscale structural similarity for image quality assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing [online]. IEEE, 2003, 2010, 13(4), 1398-1402 [cit. 2020-08-05]. DOI: 10.1109/ACSSC.2003.1292216. ISBN 0-7803-8104-1. ISSN 1057-7149. Available at: http://ieeexplore.ieee.org/document/1292216/

[23] KERR, Douglas A. *Chrominance Subsampling in Digital Images* [online]. Issue 3. January 19, 2012. Cit. 14.2.2020. Available at: http://dougkerr.net/Pumpkin/articles/Subsampling.pdf

[24] *Digital Color Coding. Using the YUV model to reduce pixel depth* [online]. Video services Forum. January 7, 2014. Cit. 14.2. 2020 Available at: https://web.archive.org/web/20140107171831 /http://www.telairity.com/assets/downloads/Digital%20Color%20Coding.pdf

[25] POYTON, C. *Chroma subsampling notation* [online]. Cit. 14.2. 2020. Available at: http://poynton.ca/PDFs/Chroma_subsampling_notation.pdf

[26] MANSUROV, N. *What is ISO? The Complete Guide for Beginners* [online]. August 14, 2019. Cit. 14.2. 2020. Available at: https://photographylife.com/what-is-iso-in-photography

[27] *WebP Compression Study* [online]. Google Developers. Cit. 14.2. 2020. Available at: https://developers.google.com/speed/webp/docs/webp_study

[28] *Lossy Compressed Image Formats Study*[online]. Mozilla Corporation. July 2014. Cit. 14.2. 2020. Available at https://web.archive.org/web/20160311235158/http://people.mozilla.org /j̃osh/lossy_compressed_image_study_july_2014/

[29] CORCHS, Silvia Elena, Gianluigi CIOCCA, Emanuela BRICOLO, Francesca GASPARINI a Feng SHAO, 2016. Predicting Complexity Perception of Real World Images. *PLOS ONE* [online]. 11(6) [cit. 2020-04-28]. DOI: 10.1371/journal.pone.0157986. ISSN 1932-6203. Available at: https://dx.plos.org/10.1371/journal.pone.0157986

[30] ITU-T P.910, 2008. *Subjective video quality assessment methods for multimedia applications.* International Telecommunication Union.

[31] *XnView MP* [online], c2013-2020. France: XnSoft [cit. 2020-05-06]. Available at: https://www.xnview.com/en/xnviewmp/

[32] *XnView MP Formats* [online], c2013-2020. France: XnSoft. [cit. 2020-08-06]. Available at: https://www.xnview.com/en/xnviewmp/#formats

[33] *Nconvert* [online], c2013-2020. France: XnSoft [cit. 2020-05-06]. Available at: https://www.xnview.com/en/nconvert/

[34] *ImageMagick* [online], c2013-2020. ImageMagick Studio [cit. 2020-05-06]. Available at: https://imagemagick.org/script/download.php

[35] Serge (2020). MATLAB Central File Exchange. *Bjontegaard metric calculation (BD-PSNR)*[online]. Cit. 10.6.2020. Available at: https://www.mathworks.com/matlabcentral/fileexchange/41749-bjontegaard-metric-calculation-bd-psnr.

[36] TAN, Thiow Keng, Rajitha WEERAKKODY, Marta MRAK, Naeem RAMZAN, Vittorio BARONCINI, Jens-Rainer OHM a Gary J. SULLIVAN, 2016. Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.* 26(1), 76-90. DOI: 10.1109/TCSVT.2015.2477916. ISSN 1051-8215. Available at: http://ieeexplore.ieee.org/document/7254155/

[37] *Kodak Lossless True Color Image Suite* [online]. Rich Franzen [cit. 2020-08-05]. Available at: http://r0k.us/graphics/kodak/

[38] *The new test images* [online], c2008-2015. New Delhi: Rawzor - Lossless compression software for camera raw images [cit. 2020-08-05]. Available at: https://imagecompression.info/test_images/

[39] SCHAEFER, Gerald, Gary J. SULLIVAN a Touradj EBRAHIMI. An uncompressed benchmark image dataset for colour imaging. 2010 IEEE International Conference on Image Processing [online]. IEEE, 2010, 2010, 26(6), 3537-3540 [cit. 2020-08-05]. DOI: 10.1109/ICIP.2010.5651245. ISBN 978-1-4244-7992-4. ISSN 1053-5888. Available at: http://ieeexplore.ieee.org/document/5651245/

[40] WANG, Z., A.C. BOVIK, H.R. SHEIKH a E.P. SIMONCELLI. *Image Quality Assessment: From Error Visibility to Structural Similarity.* IEEE Transactions on Image Processing [online]. IEEE, 2004, 2010, 13(4), 600-612 [cit. 2020-08-05]. DOI: 10.1109/TIP.2003.819861. ISBN 978-1-4244-7992-4. ISSN 1057-7149. Available at: http://ieeexplore.ieee.org/document/1284395/

Attachments

Tables

The following tables show calculated Bjøntegaard metrics of objective quality assessment metrics for every reference image. Each section presents 1 metric and 2 tables for every calculation mode: "dsnr" and "rate". Every table contains results for 10 reference images

MS-SIM

Image	1	2	3	4	5	6	7	8	9	10
WebP	0,079	0,088	0,062	0,018	0,069	0,074	0,058	0,057	0,061	0,021
J2K	0,075	0,075	0,048	0,021	0,047	0,047	0,038	0,023	0,031	-0,0026
HEIC	0,075	0,076	0,054	0,019	0,057	0,058	0,053	0,052	0,056	0,022
BPG	0,15	0,16	0,11	0,079	0,13	0,13	0,11	0,11	0,12	0,055

Table 4.1: MS-SIM dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	0,028	0,015	0,014	0,0034	0,0079	0,046	0,0077	0,0031	0,0032	0,0039
J2K	0,014	-0,0021	0,014	0,0019	-0,0034	0,023	0,0046	0,00067	0,0014	0,0058
HEIC	0,026	0,014	0,015	0,004	0,0078	0,04	0,0071	0,0043	0,0038	0,007
BPG	0,056	0,034	0,025	-0,0079	0,028	0,10	0,013	0,0032	-0,0022	0,023

Table 4.2: MS-SIM dsnr Part 2

Image	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10 [%]
WebP	-62,56	-64,61	-65,88	-49,33	-68,04	-67,55	-62,39	-65,47	-65,78	-41,09
J2K	-48,25	-45,19	-39,27	-51,38	-38,15	-35,12	-31,22	-15,53	-23,33	8,75
HEIC	-60,22	-60,47	-61,14	-50,53	-61,34	-59,17	-60,49	-62,55	-64,57	-45,42
BPG	858,73	928,81	858,66	998,08	1032,19	992,84	1105,62	1132,80	1279,4	1541,87

Table 4.3: MS-SIM rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15 [%]	16 [%]	17 [%]	18 [%]	19 [%]	20 [%]
WebP	-57,24	-57,16	-49,69	-34,742	-63,77	-51,8	-43,2	-55,00	-30,01	-30,01
J2K	-14,08	-48,9	-31,78	-3,22	-28,92	-38,02	-24,08	-34,66	-42,83	-42,83
HEIC	-48,25	-59,08	-54,65	-31,87	-54,69	-45,79	-40,71	-55,06	-47,18	-47,18
BPG	1061,8	358,51	-29,51	1487,2	1358,3	159,8	43,55	-35,22	384,76	384,76

Table 4.4: MS-SIM rate Part 2

SSIM

Image	1	2	3	4	5	6	7	8	9	10
WebP	0,022	0,034	0,046	0,007	0,039	0,052	0,044	0,054	0,05	0,044
J2K	-0,094	-0,074	-0,061	-0,011	-0,018	-0,033	-0,012	0,0035	0,00014	0,03
HEIC	0,023	0,025	0,031	0,011	0,031	0,03	0,038	0,047	0,05	0,048
BPG	0,18	0,2	0,19	0,076	0,16	0,21	0,18	0,18	0,18	0,14

Table 4.5: SSIM dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	0,022	0,041	0,0023	0,0044	0,023	0,040	0,0034	0,0053	0,0067	0,0034
J2K	0,00064	0,016	-0,0025	0,00049	0,014	-0,024	0,0017	0,0032	0,0012	-0,00088
HEIC	0,023	0,047	0,0027	0,0050	0,023	0,043	0,0028	0,0055	0,0075	0,004
BPG	0,059	0,096	0,0061	-0,010	0,065	0,19	0,0051	0,0041	-0,0038	0,013

Table 4.6: SSIM dsnr Part 2

Image	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10
WebP	-13,84	-19,41	-28,95	-16,92	-29,66	-30,10	-29,75	-35,73	-31,94	-34,63
J2K	65,71	33,76	36,23	67,26	23,85	25,20	22,50	14,23	24,03	-12,98
HEIC	-14,00	-12,99	-17,62	-37,69	-19,74	-14,74	-21,48	-27,72	-30,00	-34,08
BPG	1244,61	1204,71	1052,28	782,68	1098,26	1014,38	1161,77	835,85	986,34	940,82

Table 4.7: SSIM rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15 [%]	16 [%]	17 [%]	18 [%]	19 [%]	20 [%]
WebP	-33,25	-34,70	-22,21	-16,55	-36,41	-28,37	-33,37	-27,22	-22,03	-28,29
J2K	34,29	10,32	77,47	16,73	-17,78	37,45	4,04	-0,84	23,78	48,42
HEIC	-31,83	-42,51	-43,06	-55,12	-36,53	-20,60	-45,90	-51,75	-55,51	-46,31
BPG	776,71	688,97	271,67	71,65	844,00	1049,79	441,51	449,02	138,05	622,28

Table 4.8: SSIM rate Part 2

FSIM

Image	1	2	3	4	5	6	7	8	9	10
WebP	0,0013	0,0019	0,0021	0,00071	0,003	0,0028	0,002	0,003	0,0081	0,0035
J2K	-0,0011	0,0001	0,0014	0,00030	0,0023	0,0022	0,0015	0,0028	0,0069	0,0034
HEIC	0,00052	0,00053	0,0014	0,00056	0,0021	0,0013	0,0017	0,0026	0,0071	0,0037
BPG	0,0058	0,0073	0,0062	0,0029	0,0084	0,0082	0,0061	0,0086	0,017	0,0076

Table 4.9: FSIM dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	0,0026	0,0024	0,00023	0,00019	0,0047	0,0038	0,00093	0,00054	0,00012	0,00044
J2K	0,0017	0,0014	-0,00015	8,62	0,0039	-0,00014	0,00038	0,00024	0,000002	0,00016
HEIC	0,0031	0,003	0,00046	-4,73	0,006	0,0063	0,0015	0,0011	0,00028	0,00062
BPG	0,005	0,0045	0,0035	-0,00051	0,0083	0,014	0,0034	0,0022	0,00046	0,002

Table 4.10: FSIM dsnr Part 2

Image	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10 [%]
WebP	-17,3	-23,98	-36,77	-19,46	-36,57	-32,78	-17,96	-11,03	-47,92	-8,82
J2K	-14,80	-34,36	-33,95	127,53	-11,31	-20,68	-27,22	3,15	-12,03	-4,49
HEIC	5,08	3,14	-16,64	-45,99	-20,42	-10,84	-20,267	-18,42	-41,43	-31,89
BPG	1492,8	1522,71	1338,31	3624,8	1538,49	1493,99	1533,0	1878,68	1409,59	1917,39

Table 4.11: FSIM rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15 [%]	16 [%]	17 [%]	18 [%]	19[%]	20 [%]
WebP	-51,22	-46,52	-19,17	-4,02	-59,34	-41,62	-20,82	-11,13	-0,99	-23,29
J2K	5,15	8,01	13,28	23,94	-42,95	50,02	23,65	31,81	41,94	33,65
HEIC	-30,53	-29,05	26,87	-59,80	-39,81	-8,34	-59,74	-54,21	-55,18	-57,27
BPG	1524,25	2488,50	25,94	103,39	2241,15	2637,91	1252,32	396,77	13,06	1516,26

Table 4.12: FSIM rate Part 2

FSIMc

Image	1	2	3	4	5	6	7	8	9	10
WebP	0,0027	0,0031	0,0034	0,00085	0,0039	0,0039	0,0031	0,0039	0,0095	0,00461
J2K	0,0019	0,0027	0,0035	0,00045	0,0041	0,0040	0,0039	0,0041	0,0089	0,0048
HEIC	0,0017	0,0015	0,0026	0,00066	0,003	0,0021	0,0028	0,0034	0,0084	0,0048
BPG	0,0092	0,010	0,0087	0,0031	0,011	0,010	0,0088	0,010	0,019	0,0095

Table 4.13: FSIMc dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	0,003	0,0031	0,00028	0,00024	0,0051	0,0046	0,0010	0,00061	0,00025	0,00047
J2K	0,0022	0,0020	-0,0001	0,00014	0,0043	0,00077	0,00045	0,00033	0,00008	0,0002
HEIC	0,0036	0,0038	0,00050	-0,00001	0,0063	0,0074	0,0016	0,0012	0,00033	0,00064
BPG	0,0056	0,0055	0,0036	-0,00054	0,0088	0,016	0,0034	0,0023	0,00048	0,0021

Table 4.14: FSIMc dsnr Part 2

Image	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10 [%]
WebP	-29,09	-33,89	-44,92	-23,99	-42,05	-40,87	-31,53	-25,56	-51,33	-27,09
J2K	-31,01	-28,02	-30,72	108,00	-14,22	-20,26	-22,61	0,99	-19,00	-15,15
HEIC	-8,57	-8,38	-25,83	-47,31	-25,93	-18,84	-27,42	-26,38	-43,83	-36,52
BPG	1354,91	1382,86	1151,13	3699,88	1404,10	1319,60	1374,53	1653,44	1308,03	1702,62

Table 4.15: FSIMc rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15 [%]	16 [%]	17 [%]	18 [%]	19 [%]	20 [%]
WebP	-52,19	-50,49	-22,11	-37,61	-58,96	-45,07	-22,94	-15,17	-9,31	-25,28
J2K	-1,45	-9,39	8,33	10,26	-44,58	33,88	17,79	23,33	28,71	26,32
HEIC	-31,62	-35,14	17,23	-61,59	-39,59	-15,65	-59,66	-55,46	-56,64	-57,08
BPG	1594,82	2372,34	38,96	141,35	2151,42	2412,55	1285,05	442,59	36,13	1603,26

Table 4.16: FSIMc rate Part 2

MSE

Image	1	2	3	4	5	6	7	8	9	10
WebP	-2,51	-49,63	-52,34	-5,14	-48,15	-47,23	-48,57	-36,76	-39,22	-30,80
J2K	-1,07	226,58	179,42	3,99	105,65	117,17	77,02	58,56	45,71	-11,79
HEIC	-3,60	-56,36	-50,04	-7,39	-48,91	-39,96	-47,88	-46,62	-51,42	-37,92
BPG	-12,47	-347,82	-268,03	-41,90	-233,80	-237,02	-211,71	-161,17	-170,34	-109,07

Table 4.17: MSE dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	-14,47	-14,66	-1,80	-0,84	-12,40	-19,49	-2,56	-2,01	-1,09	-2,51
J2K	6,86	-0,75	2,03	0,15	-6,49	20,29	-0,44	-1,08	-0,03	-1,07
HEIC	-17,64	-18,84	-2,04	-0,42	-14,66	-24,92	-2,58	-2,26	-0,97	-3,60
BPG	-44,80	-37,77	-5,49	2,84	-39,25	-91,99	-5,66	-2,47	0,36	-12,47

Table 4.18: MSE dsnr Part 2

Image	1 [%]	2 [%]	3[%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10 [%]
WebP	-11,03	-16,30	-22,47	-22,00	-23,70	-22,86	-26,04	-24,06	-23,93	-26,14
J2K	83,65	82,50	76,63	58,59	47,37	58,82	34,71	38,66	42,65	-4,22
HEIC	-21,19	-20,16	-21,78	-43,19	-24,69	-20,92	-24,58	-32,28	-34,14	-31,72
BPG	1299,93	1324,80	1250,83	1268,98	1385,15	1330,93	1369,84	1350,48	1423,56	1333,72

Table 4.19: MSE rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15 [%]	16 [%]	17 [%]	18 [%]	19 [%]	20 [%]
WebP	-11,03	-16,30	-22,47	-22,00	-23,70	-22,86	-26,04	-24,06	-23,93	-26,14
J2K	83,65	82,50	76,63	58,59	47,37	58,82	34,71	38,66	42,65	-4,22
HEIC	-21,19	-20,16	-21,78	-43,19	-24,69	-20,92	-24,58	-32,28	-34,14	-31,72
BPG	1299,93	1324,80	1250,83	1268,98	1385,15	1330,93	1369,84	1350,48	1423,56	1333,72

Table 4.20: MSE rate Part 2

PSNR

Image	1	2	3	4	5	6	7	8	9	10
WebP	0,48	0,99	1,41	0,57	1,48	1,44	1,63	1,58	1,47	1,44
J2K	0,12	-2,84	-2,91	-0,55	-2,41	-2,54	-2,25	-2,15	-2,12	0,04
HEIC	0,71	1,10	1,22	0,93	1,37	1,20	1,36	2,00	2,08	1,66
BPG	3,16	24,16	22,82	9,12	20,41	22,38	20,57	18,79	19,61	12,13

Table 4.21: PSNR dsnr Part 1

Image	11	12	13	14	15	16	17	18	19	20
WebP	1,74	1,82	0,65	0,41	1,46	1,61	0,90	1,08	1,01	0,48
J2K	-1,47	-0,69	-0,90	-1,73	0,23	-1,73	-0,19	0,23	-0,52	0,12
HEIC	1,60	2,13	0,66	-0,73	1,40	1,11	0,65	0,84	0,60	0,71
BPG	10,74	9,24	2,37	-4,60	9,03	17,43	1,97	0,08	-1,63	3,16

Table 4.22: PSNR dsnr Part 2

Image	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	6 [%]	7 [%]	8 [%]	9 [%]	10 [%]
WebP	-9,29	-16,20	-21,92	-20,30	-23,89	-22,96	-26,14	-25,72	-24,50	-26,80
J2K	95,78	97,05	92,23	65,10	73,95	81,08	62,12	64,42	70,08	1,03
HEIC	-18,75	-18,42	-19,45	-36,17	-22,78	-19,68	-21,64	-31,91	-33,34	-30,66
BPG	850,40	837,92	752,11	632,11	784,37	750,11	773,14	676,35	689,01	637,76

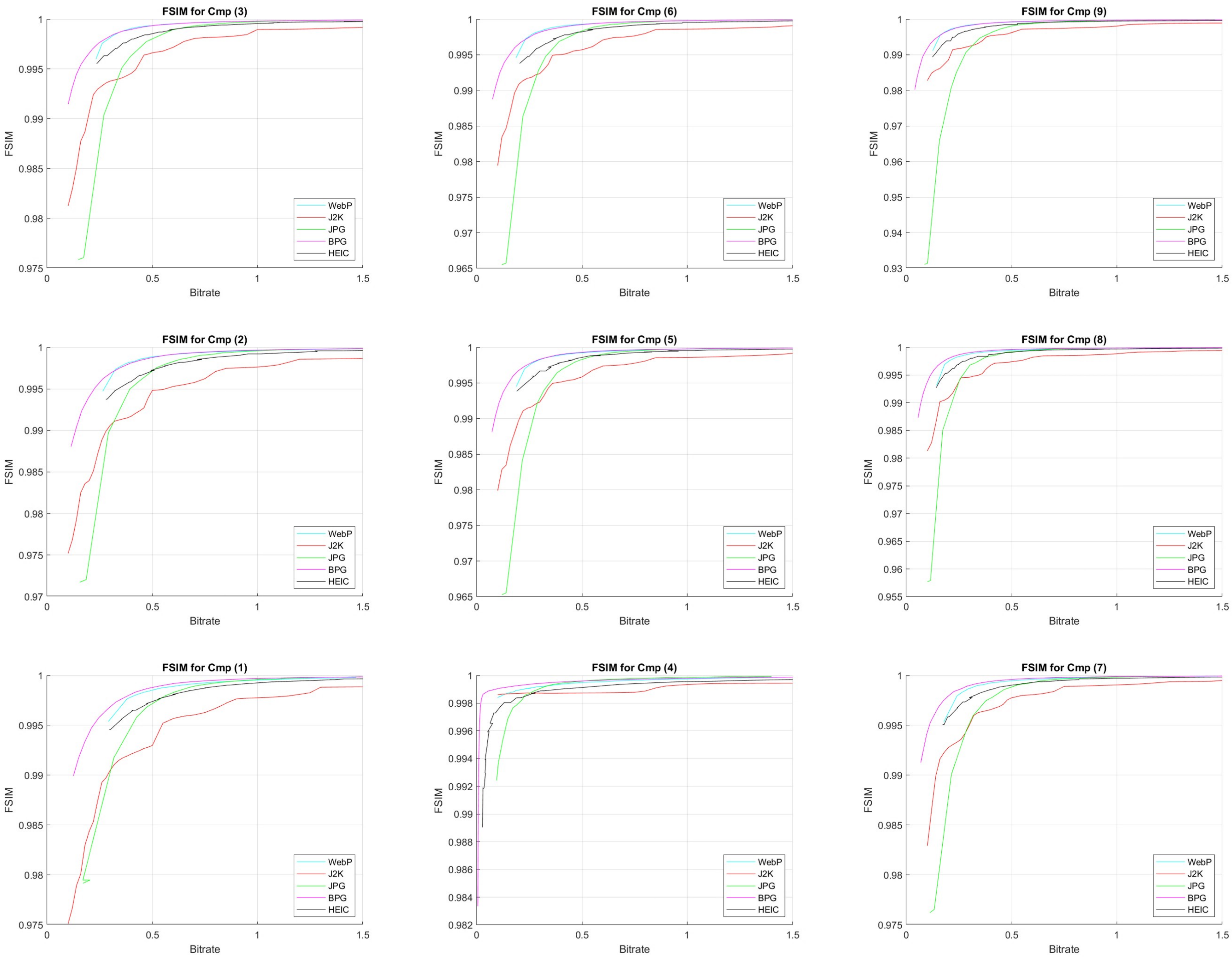
Table 4.23: PSNR rate Part 1

Image	11 [%]	12 [%]	13 [%]	14 [%]	15[%]	16 [%]	17 [%]	18 [%]	19 [%]	20 [%]
WebP	-30,12	-33,57	-20,78	-8,42	-29,64	-28,95	-25,03	-29,35	-21,71	-22,62
J2K	48,62	25,80	73,59	39,06	-1,57	59,08	22,28	0,21	31,73	2,36
HEIC	-31,08	-41,93	-36,23	-25,52	-31,80	-20,26	-34,17	-34,34	-30,04	-41,49
BPG	501,69	490,24	99,54	-83,45	520,08	671,72	146,27	72,94	-40,56	301,76

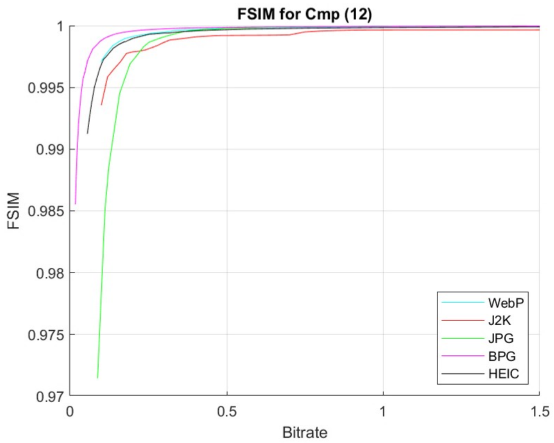
Table 4.24: PSNR rate Part 2

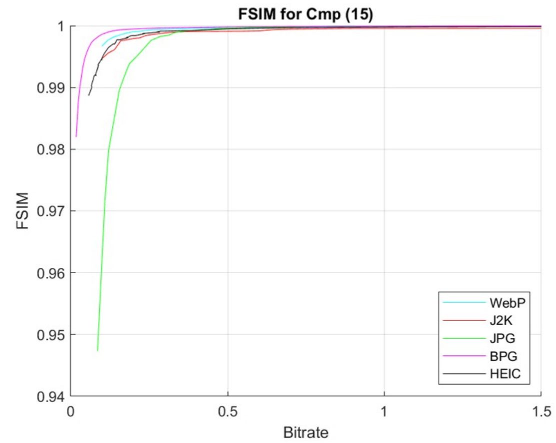
Plots

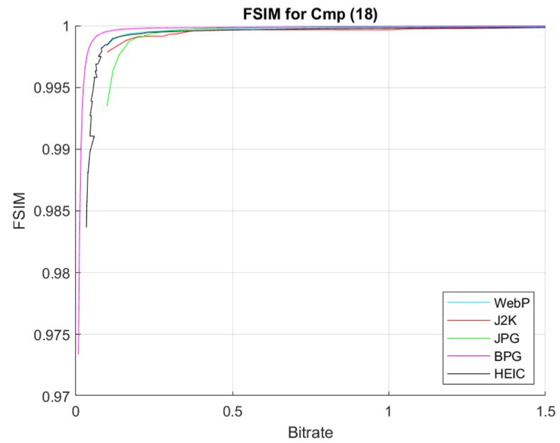
The following plots show every objective quality assessment metrics for every reference image. The order of the metrics is: FSIM, FSIMc, MSE, MS-SSIM, PSNR and SSIM.

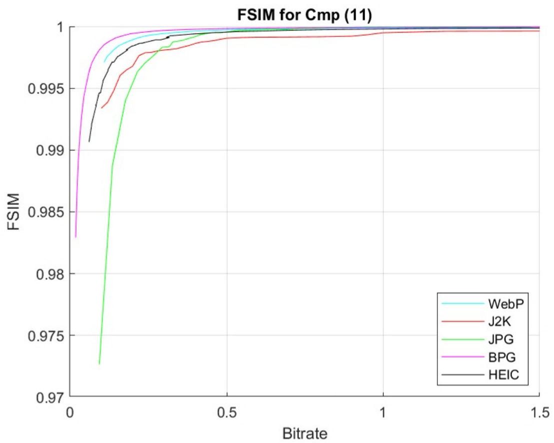


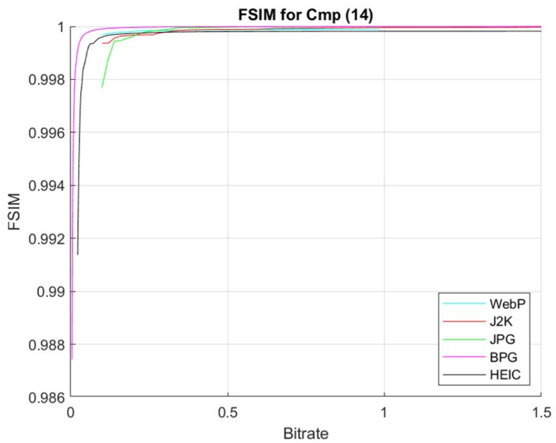
FSIM for Cmp (3)
FSIM for Cmp (6)
FSIM for Cmp (9)
FSIM for Cmp (2)
FSIM for Cmp (5)
FSIM for Cmp (8)
FSIM for Cmp (1)
FSIM for Cmp (4)
FSIM for Cmp (7)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

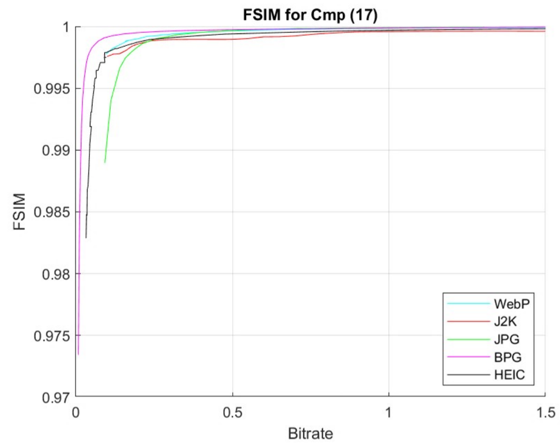

FSIM for Cmp (12)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

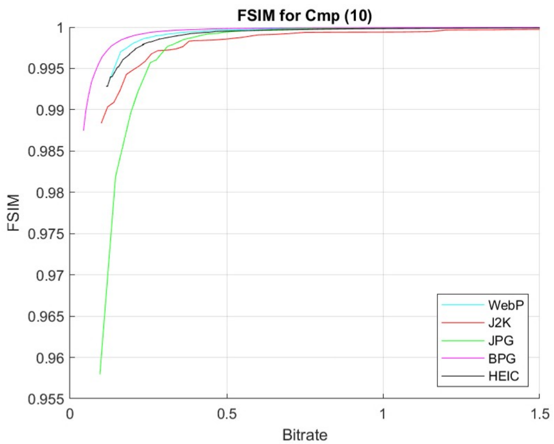

FSIM for Cmp (15)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

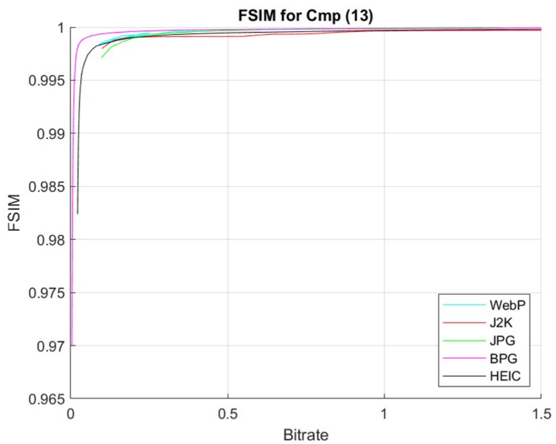

FSIM for Cmp (18)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

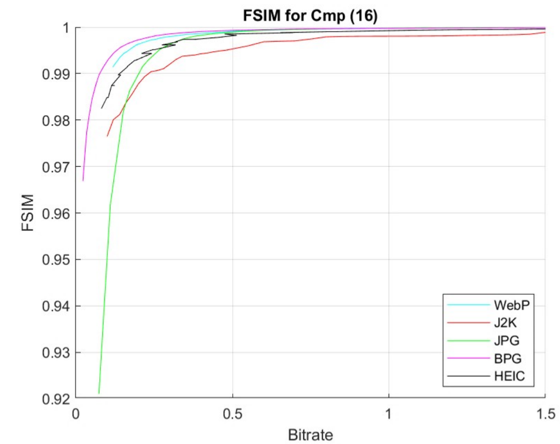

FSIM for Cmp (11)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

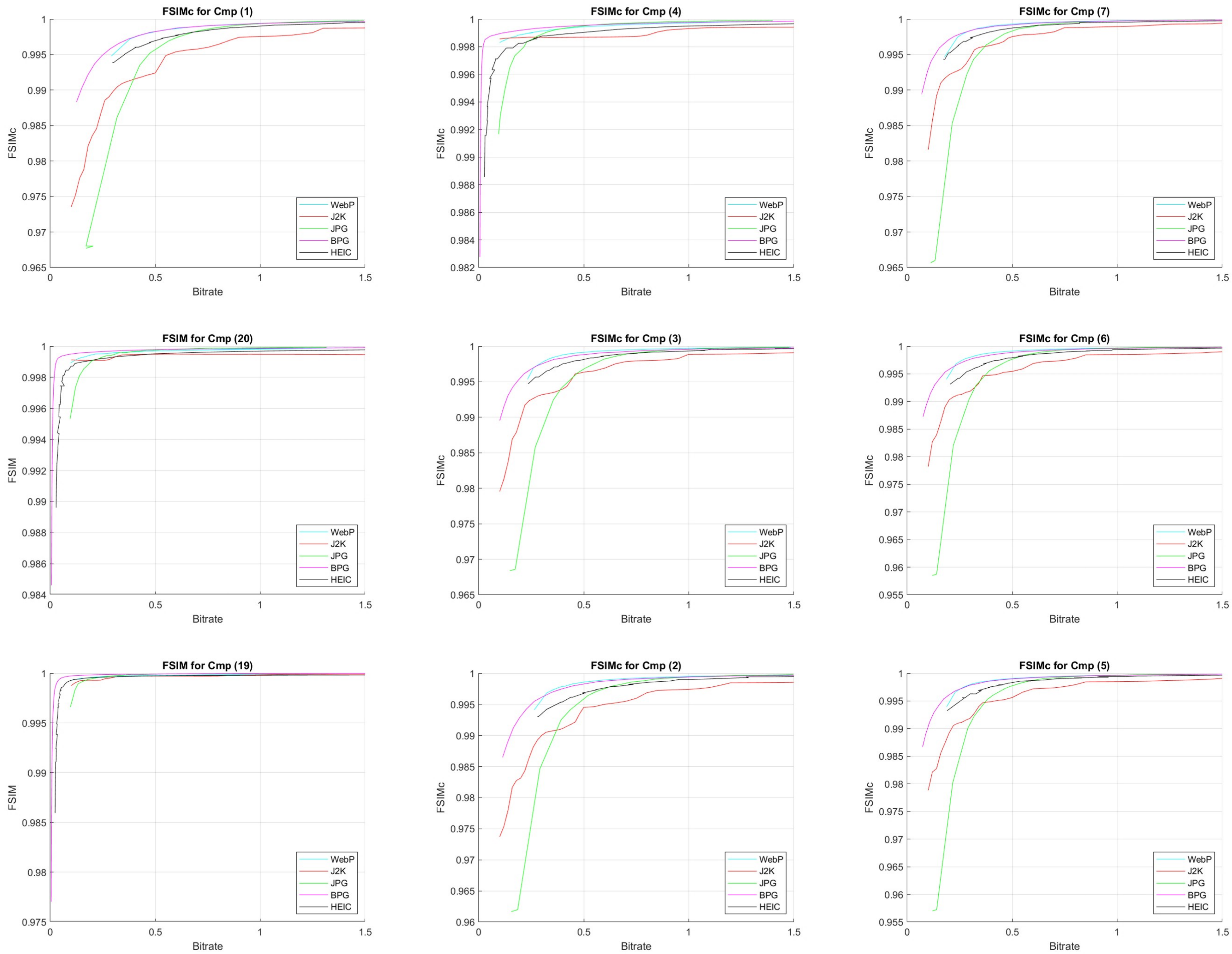

FSIM for Cmp (14)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

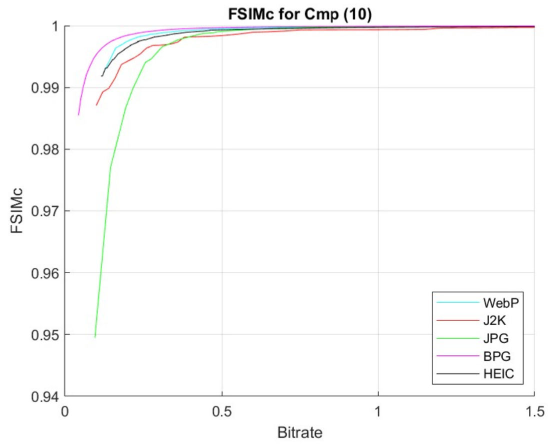

FSIM for Cmp (17)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

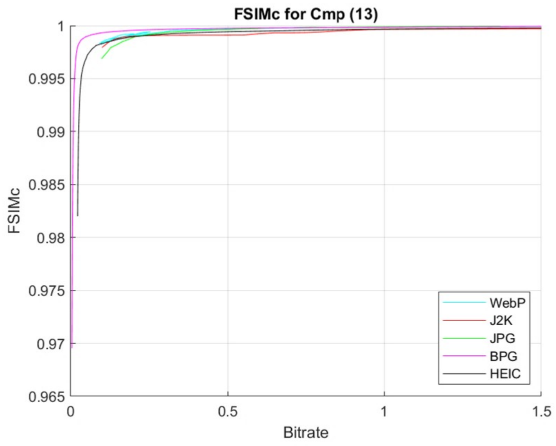

FSIM for Cmp (10)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

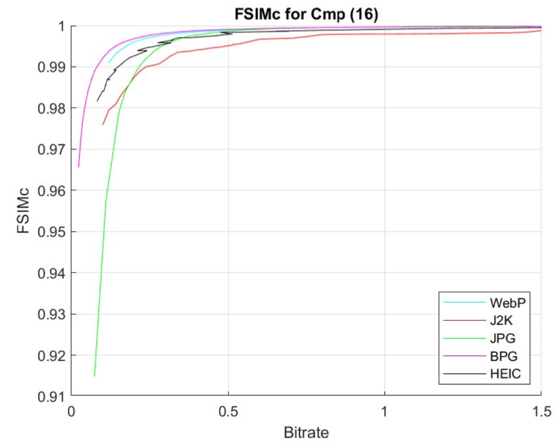

FSIM for Cmp (13)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

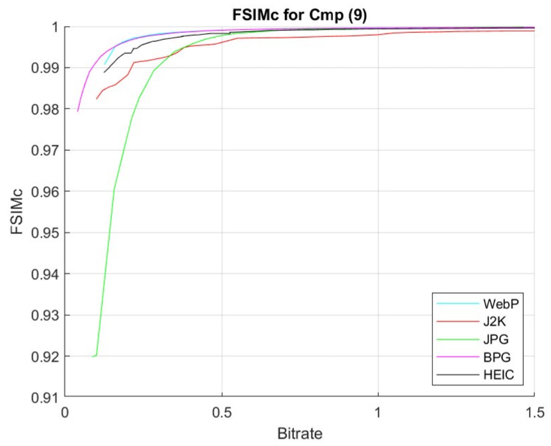

FSIM for Cmp (16)
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

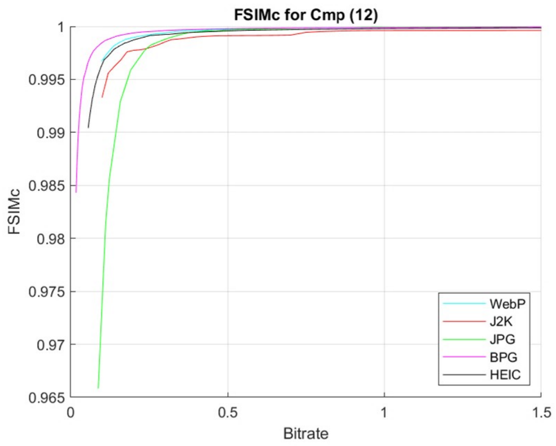

FSIMc for Cmp (1)
FSIMc for Cmp (4)
FSIMc for Cmp (7)
FSIM for Cmp (20)
FSIMc for Cmp (3)
FSIMc for Cmp (6)
FSIM for Cmp (19)
FSIMc for Cmp (2)
FSIMc for Cmp (5)
FSIMc
FSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

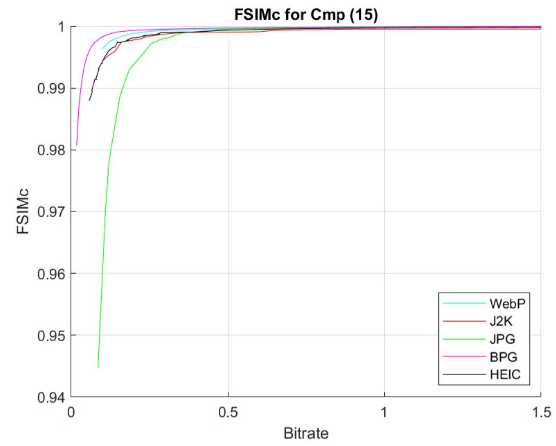

FSIMc for Cmp (10)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

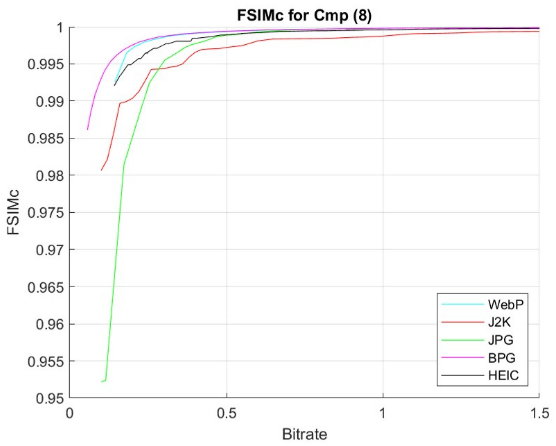

FSIMc for Cmp (13)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

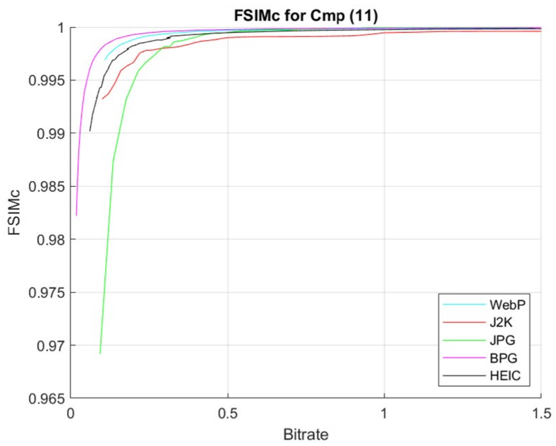

FSIMc for Cmp (16)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

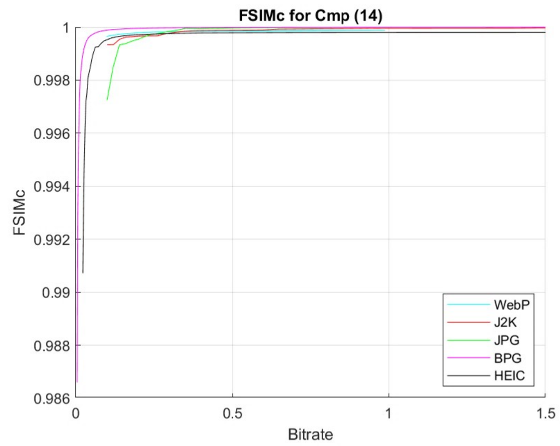

FSIMc for Cmp (9)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

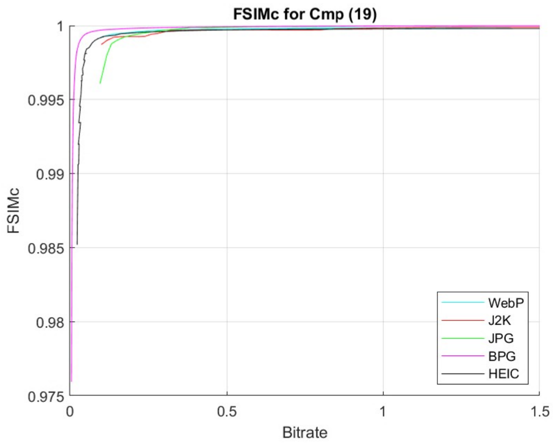

FSIMc for Cmp (12)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

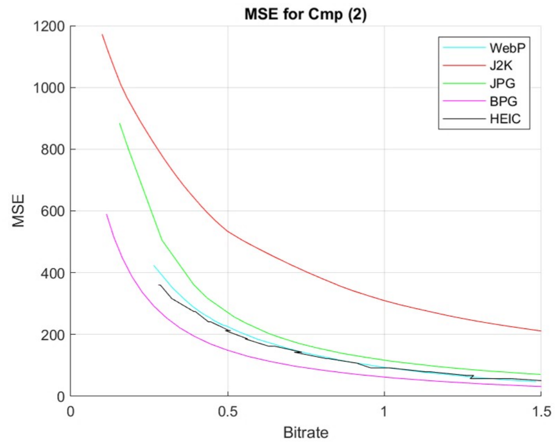

FSIMc for Cmp (15)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

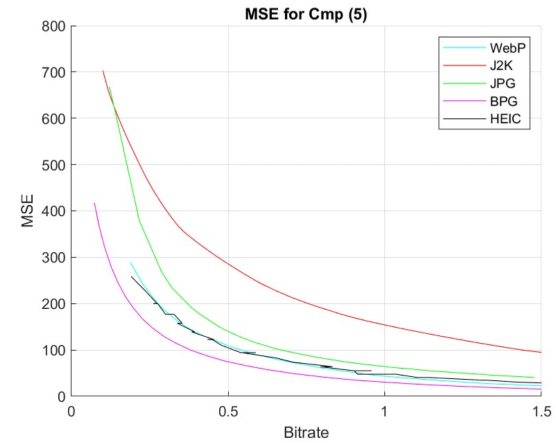

FSIMc for Cmp (8)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

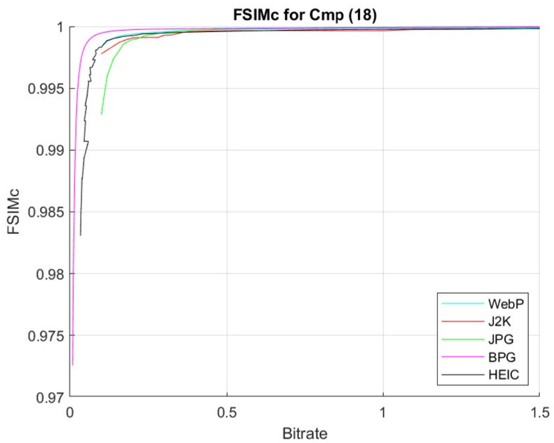

FSIMc for Cmp (11)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

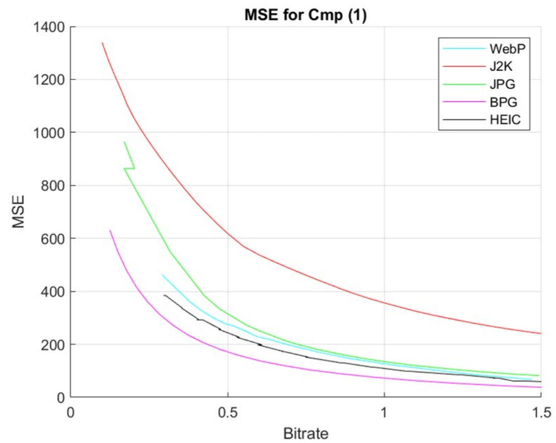

FSIMc for Cmp (14)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

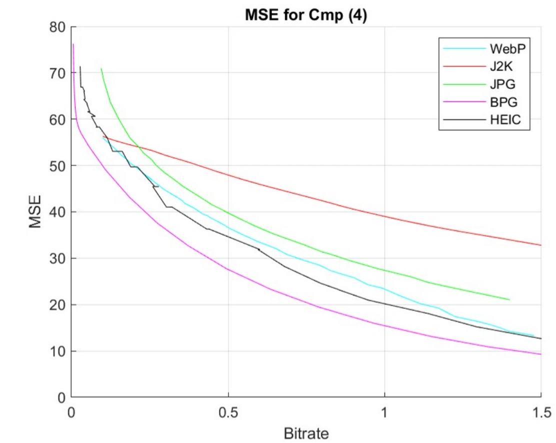

FSIMc for Cmp (19)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

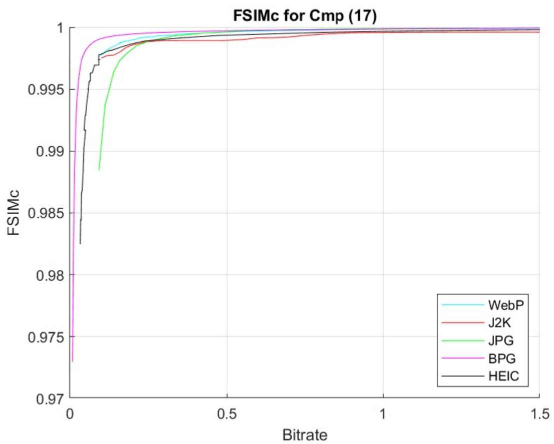

MSE for Cmp (2)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

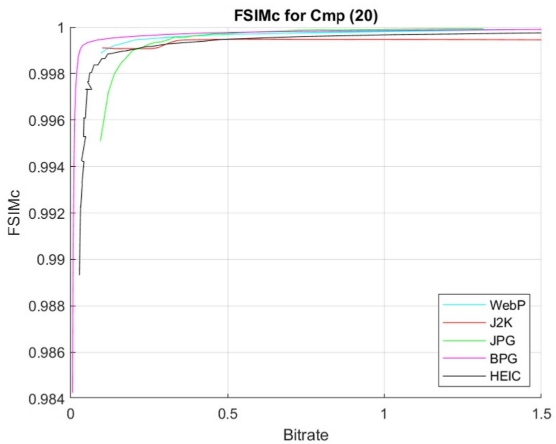

MSE for Cmp (5)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

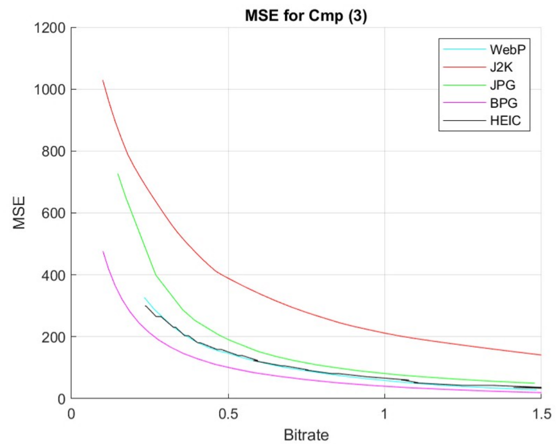

FSIMc for Cmp (18)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

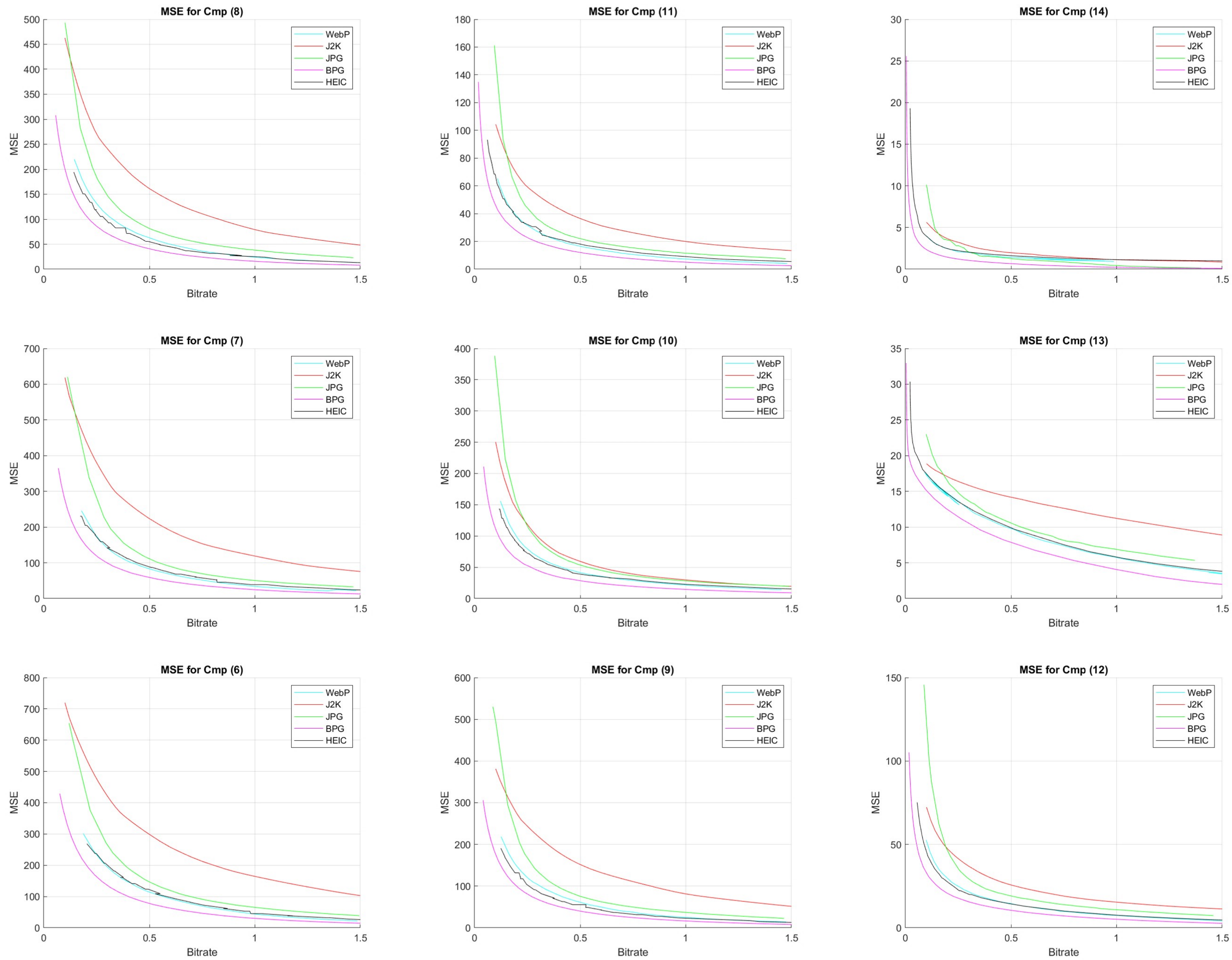

MSE for Cmp (1)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

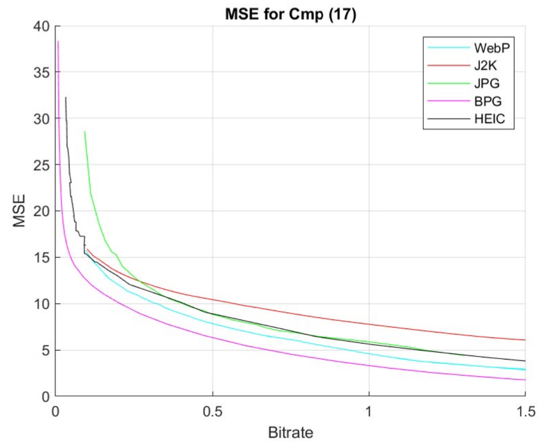

MSE for Cmp (4)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

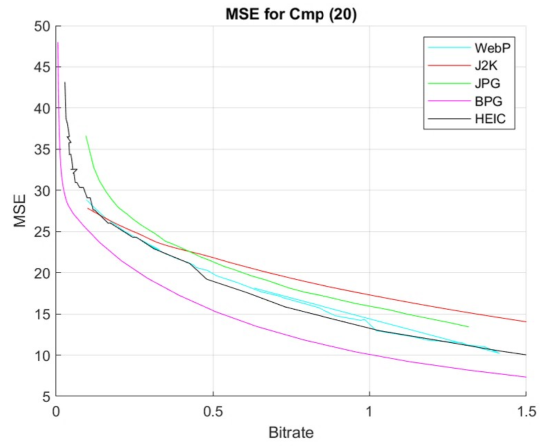

FSIMc for Cmp (17)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

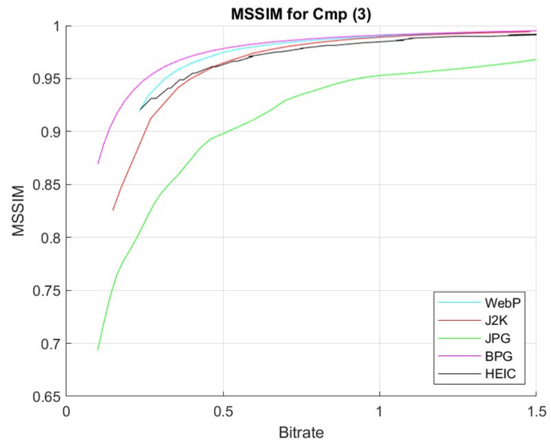

FSIMc for Cmp (20)
FSIMc
Bitrate
WebP
J2K
JPG
BPG
HEIC

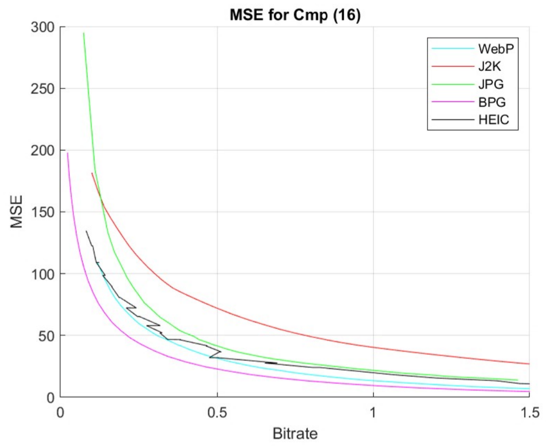

MSE for Cmp (3)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

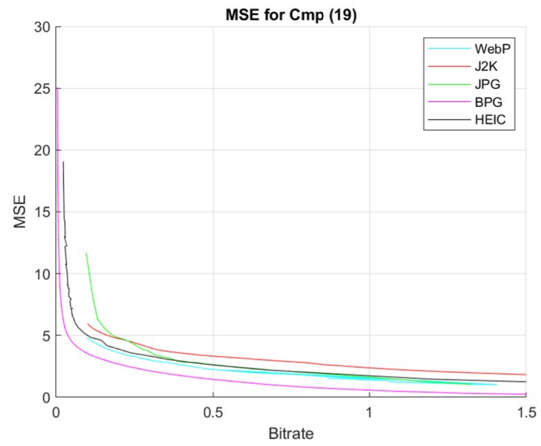

MSE for Cmp (8)
MSE for Cmp (11)
MSE for Cmp (14)
MSE for Cmp (7)
MSE for Cmp (10)
MSE for Cmp (13)
MSE for Cmp (6)
MSE for Cmp (9)
MSE for Cmp (12)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

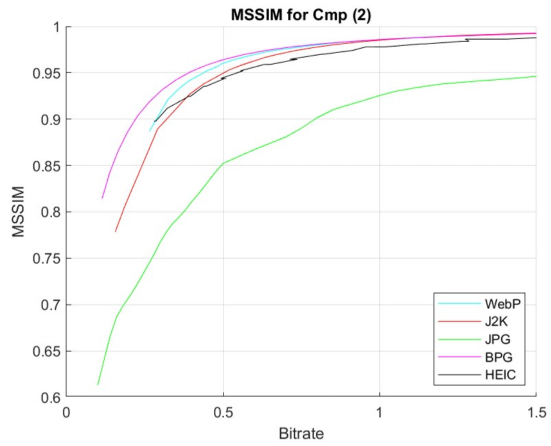

MSE for Cmp (17)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

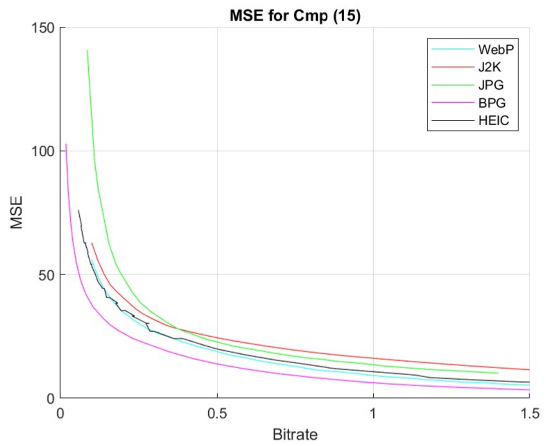

MSE for Cmp (20)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

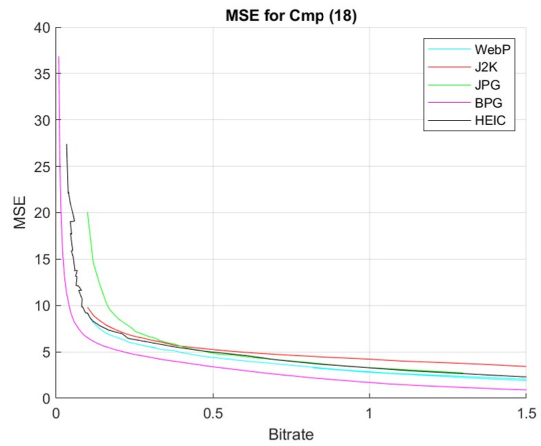

MSSIM for Cmp (3)
MSSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

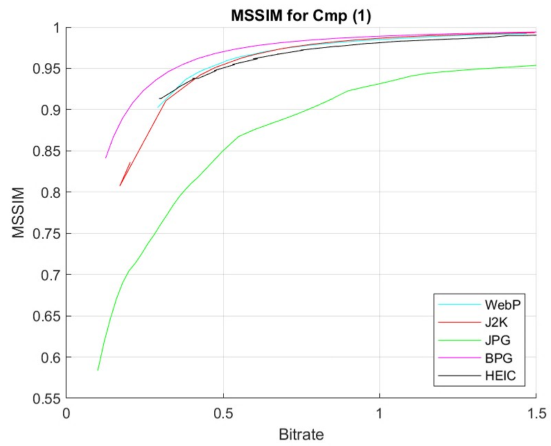

MSE for Cmp (16)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

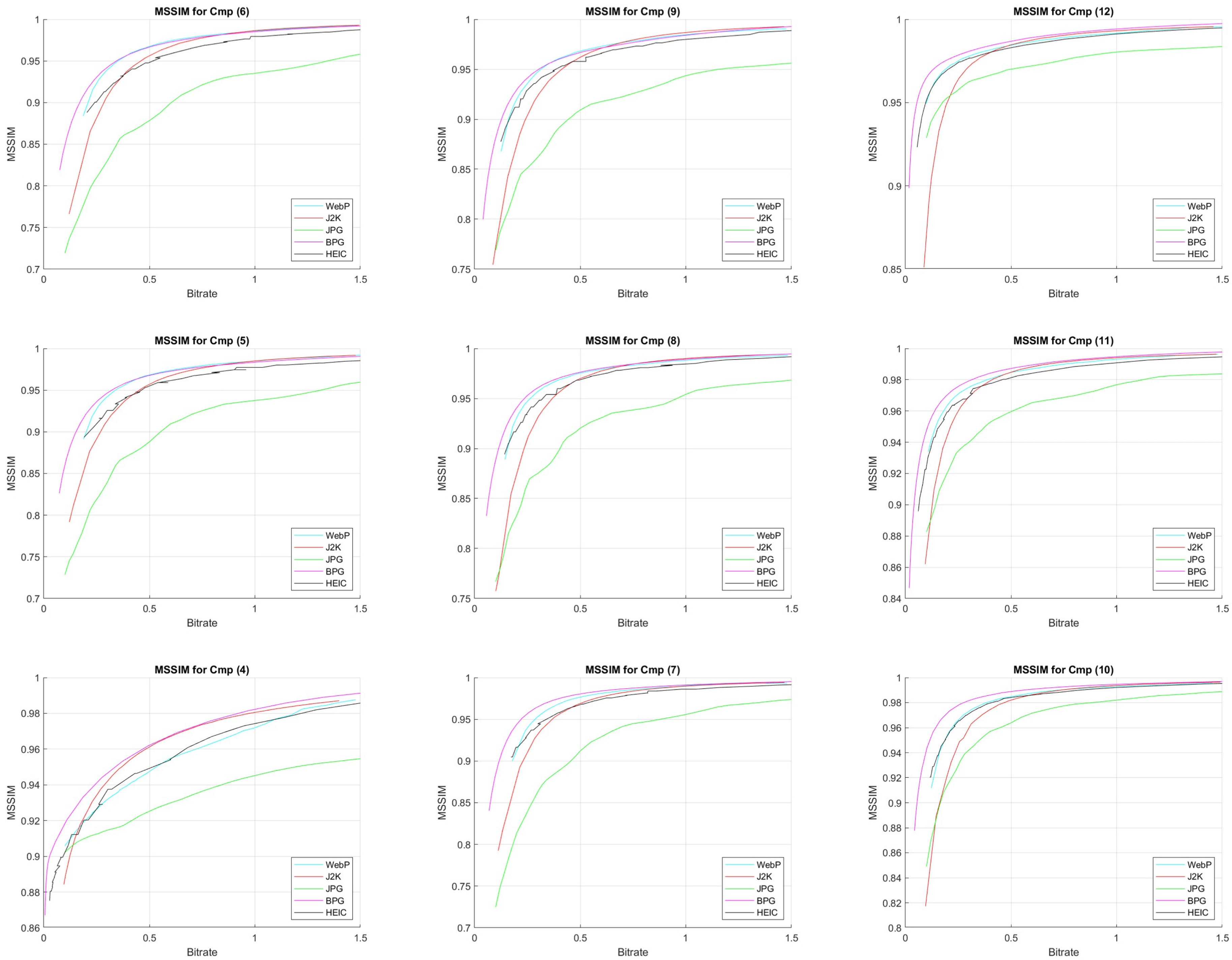

MSE for Cmp (19)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

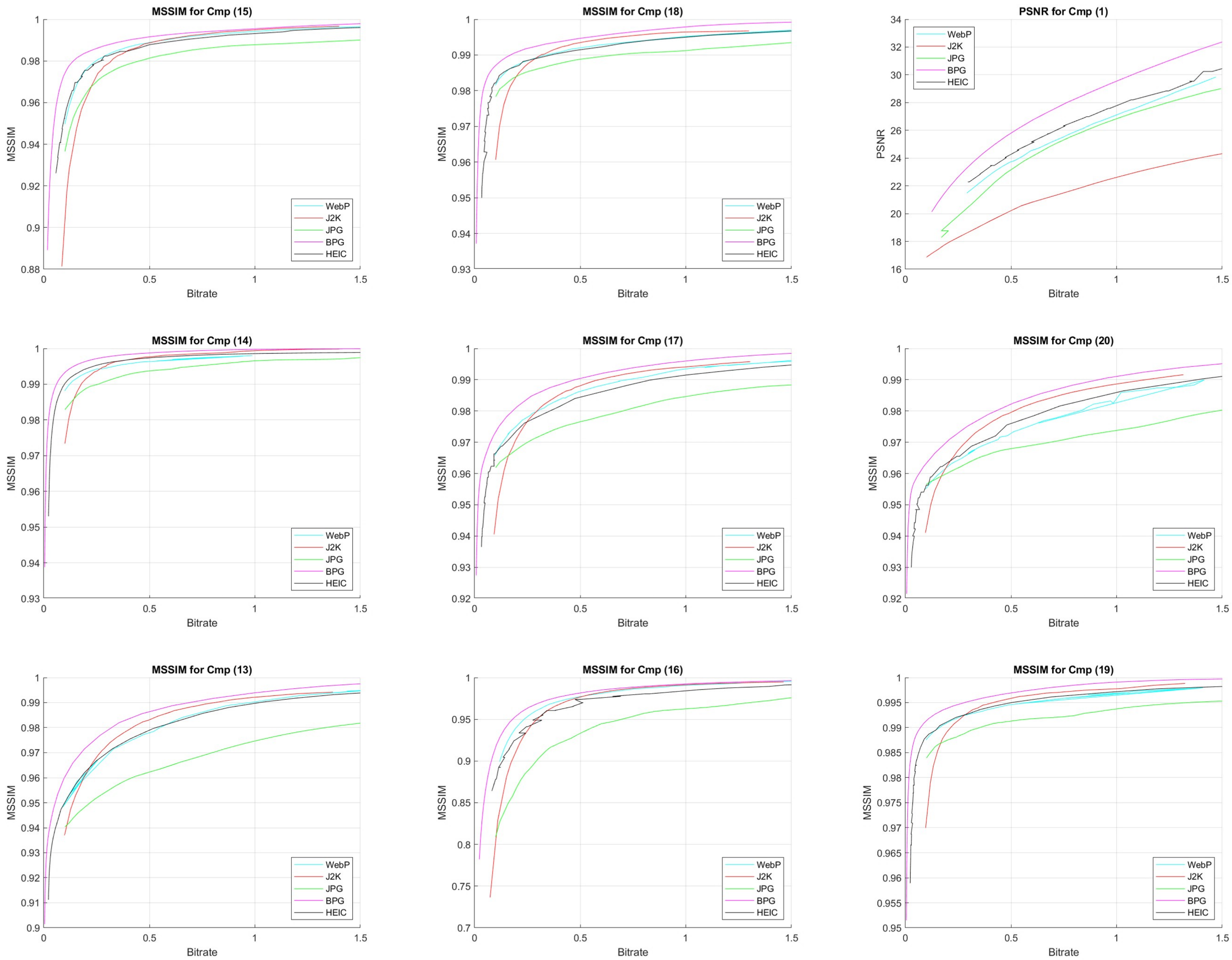

MSSIM for Cmp (2)
MSSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

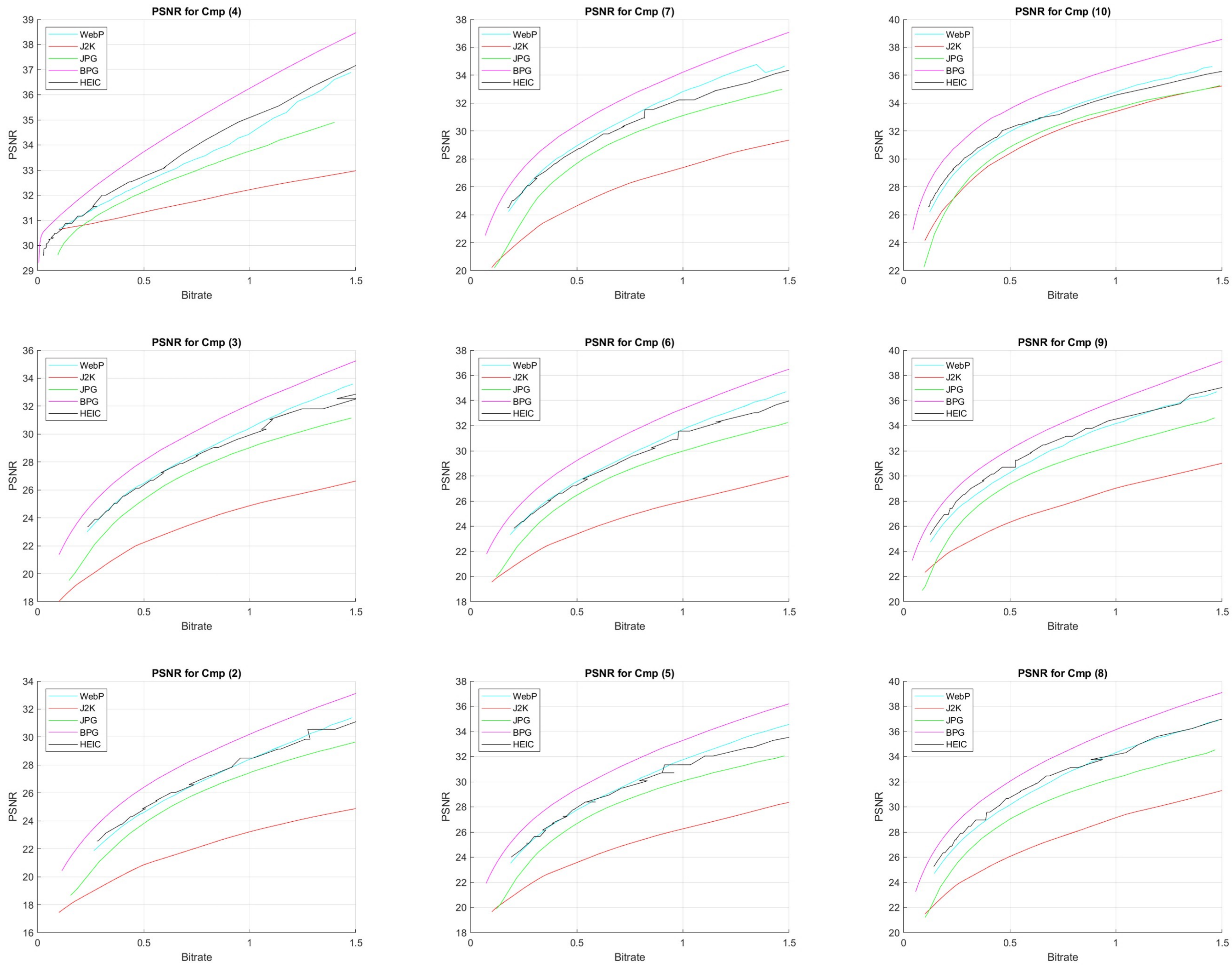

MSE for Cmp (15)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

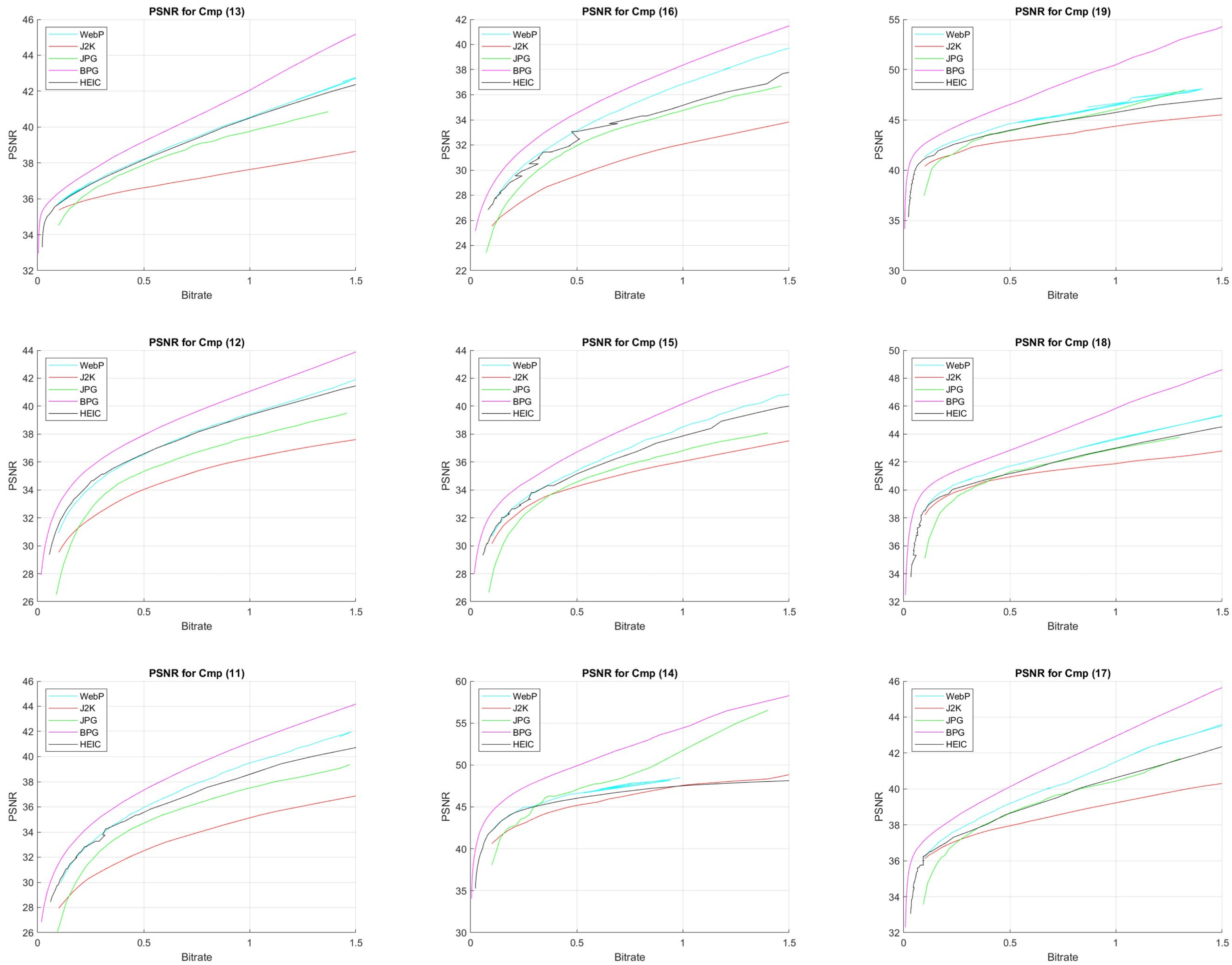

MSE for Cmp (18)
MSE
Bitrate
WebP
J2K
JPG
BPG
HEIC

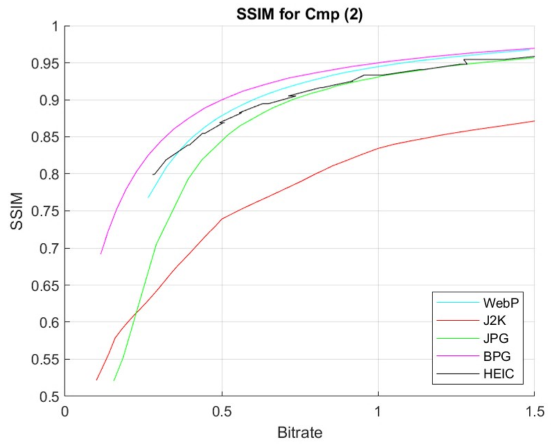

MSSIM for Cmp (1)
MSSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

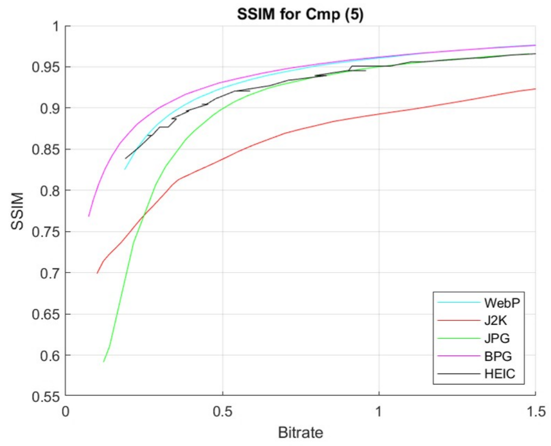

MSSIM for Cmp (6)
MSSIM for Cmp (9)
MSSIM for Cmp (12)
MSSIM for Cmp (5)
MSSIM for Cmp (8)
MSSIM for Cmp (11)
MSSIM for Cmp (4)
MSSIM for Cmp (7)
MSSIM for Cmp (10)
MSSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

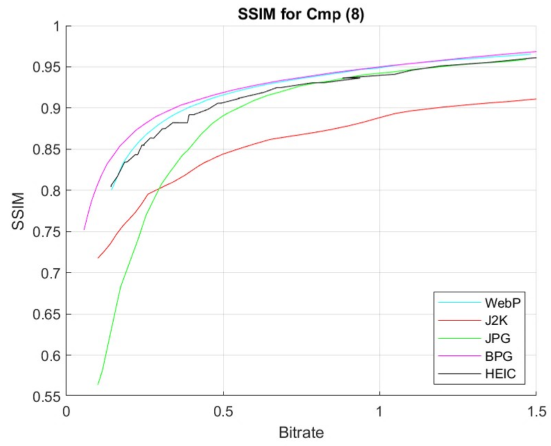

MSSIM for Cmp (15)
MSSIM for Cmp (18)
PSNR for Cmp (1)
MSSIM for Cmp (14)
MSSIM for Cmp (17)
MSSIM for Cmp (20)
MSSIM for Cmp (13)
MSSIM for Cmp (16)
MSSIM for Cmp (19)
MSSIM
PSNR
Bitrate
WebP
J2K
JPG
BPG
HEIC

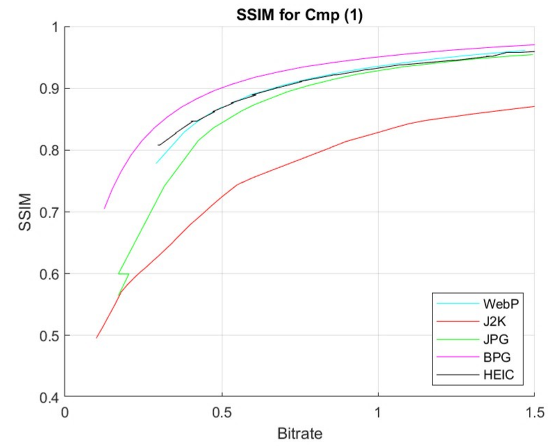

PSNR for Cmp (4)
PSNR for Cmp (7)
PSNR for Cmp (10)
PSNR for Cmp (3)
PSNR for Cmp (6)
PSNR for Cmp (9)
PSNR for Cmp (2)
PSNR for Cmp (5)
PSNR for Cmp (8)
WebP
J2K
JPG
BPG
HEIC
PSNR
Bitrate
0
0.5
1
1.5

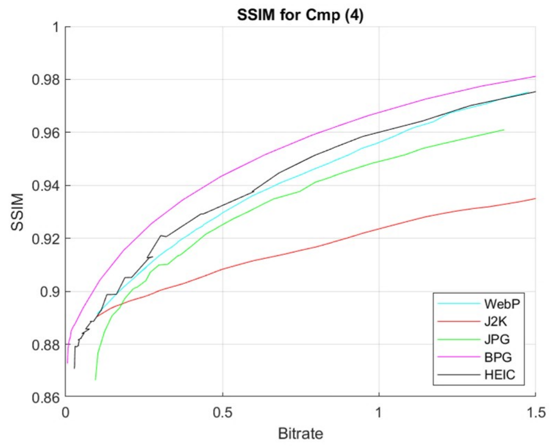

PSNR for Cmp (13)
PSNR for Cmp (16)
PSNR for Cmp (19)
PSNR for Cmp (12)
PSNR for Cmp (15)
PSNR for Cmp (18)
PSNR for Cmp (11)
PSNR for Cmp (14)
PSNR for Cmp (17)
PSNR
Bitrate
0
0.5
1
1.5
WebP
J2K
JPG
BPG
HEIC

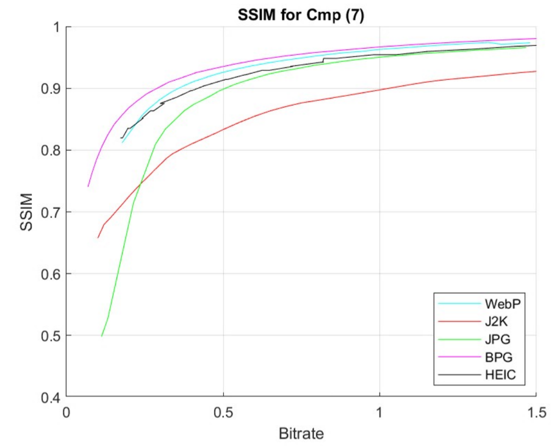

SSIM for Cmp (2)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

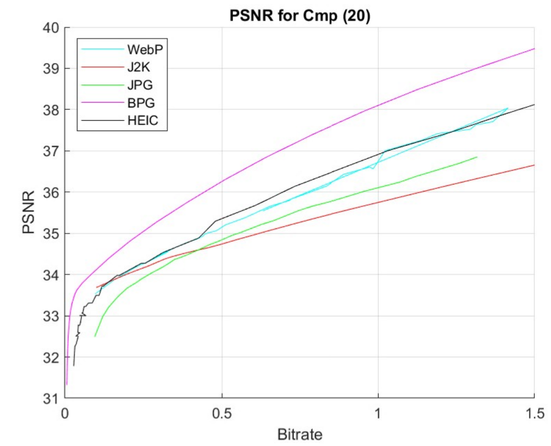

SSIM for Cmp (5)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

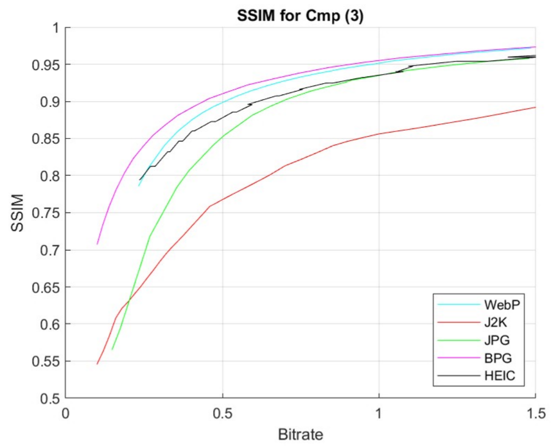

SSIM for Cmp (8)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

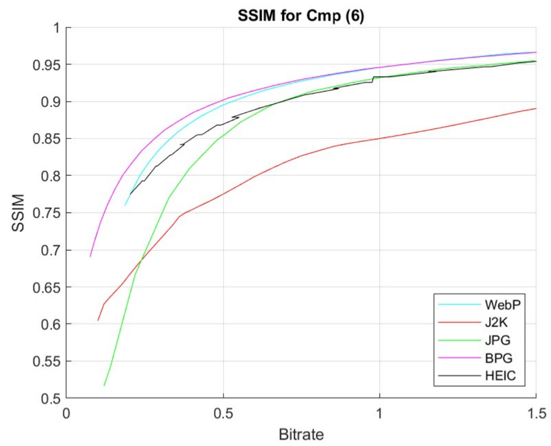

SSIM for Cmp (1)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

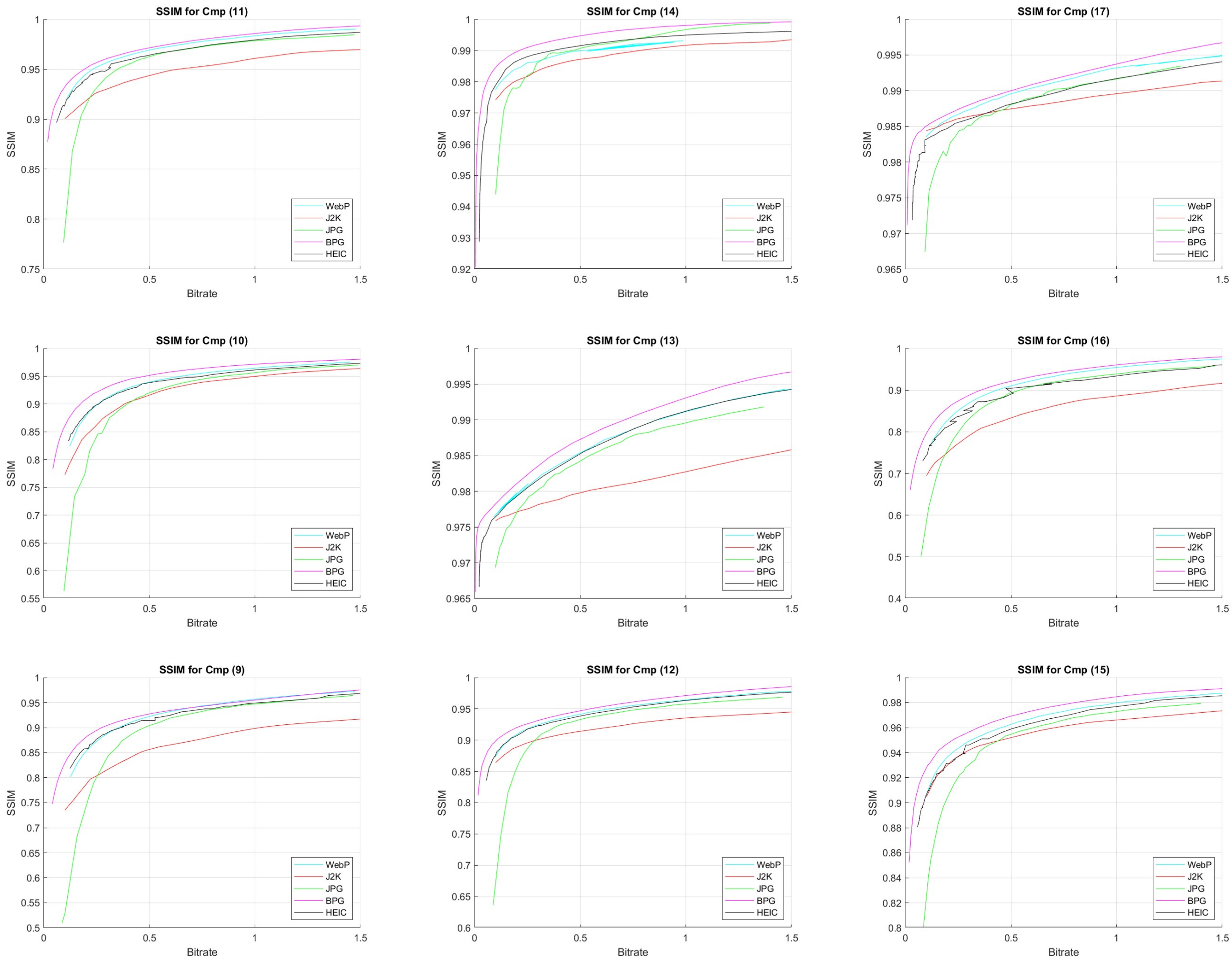

SSIM for Cmp (4)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

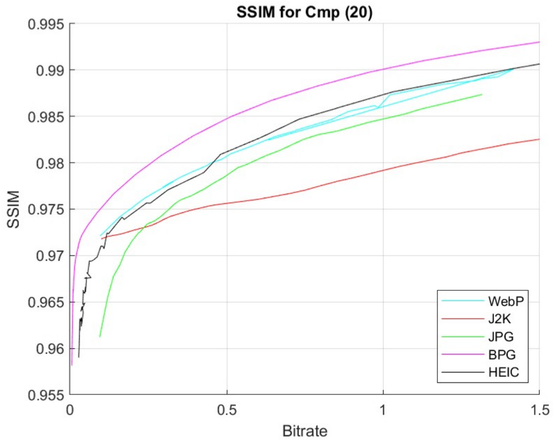

SSIM for Cmp (7)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC

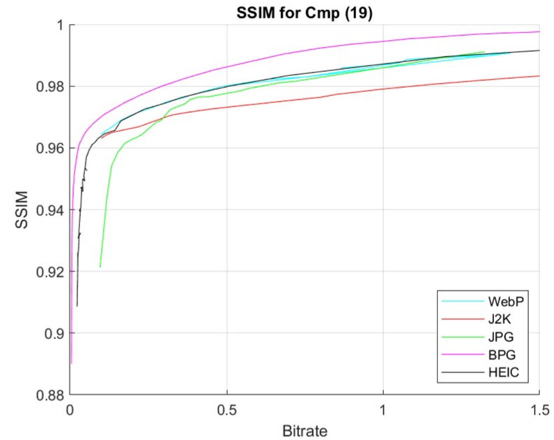

PSNR for Cmp (20)
PSNR
Bitrate
WebP
J2K
JPG
BPG
HEIC

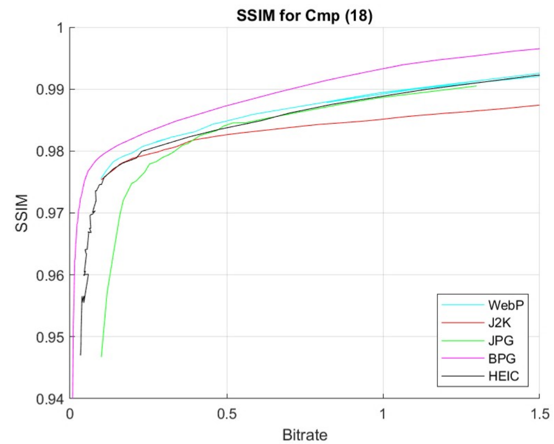

SSIM for Cmp (3)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC


SSIM for Cmp (6)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC


SSIM for Cmp (11)
SSIM for Cmp (14)
SSIM for Cmp (17)
SSIM for Cmp (10)
SSIM for Cmp (13)
SSIM for Cmp (16)
SSIM for Cmp (9)
SSIM for Cmp (12)
SSIM for Cmp (15)
SSIM
Bitrate
WebP
J2K
JPG
BPG
HEIC


SSIM for Cmp (20)
0.995
0.99
0.985
0.98
0.975
0.97
0.965
0.96
0.955
SSIM
0
0.5
1
1.5
Bitrate
WebP
J2K
JPG
BPG
HEIC


SSIM for Cmp (19)
1
0.98
0.96
0.94
0.92
0.9
0.88
SSIM
0
0.5
1
1.5
Bitrate
WebP
J2K
JPG
BPG
HEIC


SSIM for Cmp (18)
1
0.99
0.98
0.97
0.96
0.95
0.94
SSIM
0
0.5
1
1.5
Bitrate
WebP
J2K
JPG
BPG
HEIC

Electronic attachments

The following list describes what can be found in the individual folders of the electronic attachments

1. Bjontegaard: Function for calculating Bjontegaard's metric
2. BPP: Used and calculated bitrate of compressed images
3. MAT: Outputs from Framework.m and testing.m
4. Plots: Plots calculated by Framework.m
5. Tables: Tables created by Framework.m
6. Testing.m
7. Framework.m
8. CMP: Reference images - Available at external link in README
9. README - Contains this table + external link to Reference images