hochschule darmstadt

member of EUROPEAN UNIVERSITY OF TECHNOLOGY

Darmstadt University of Applied Sciences

– Faculty of Computer Science –

Assessing Lossy Image Compression for Face Recognition

Submitted in partial fulfilment of the requirements for the degree of

Master of Science (M. Sc.)

by

Paul Andreas

Matriculation number: 770607

First Examiner	:	Prof. Dr. Christoph Busch
Second Examiner	:	Torsten Schlett

DECLARATION

I hereby certify that I have written the present work independently and have not used any sources other than those listed in the bibliography.

All passages that have been taken literally or in substance from published or unpublished sources are identified as such.

The drawings or illustrations in this work were created by me or are provided with an appropriate source citation.

This thesis has not been submitted, in the same or a similar form, to any other examination authority.

Darmstadt, March 11, 2026

Paul Andreas

ZUSAMMENFASSUNG

Um das Gesicht für die biometrische Identifizierung zu verwenden, muss ein Probebild mit einem gespeichertem Referenzbild verglichen werden. Im Fall von ICAO-konformen maschinenlesbaren Reisedokumenten liegen diese Referenzbilder JPEG- oder JPEG 2000-komprimiert vor. Damit temporäre Ausweisdokumente nicht mit teuren RFID Chips zur maschinellen Lesbarkeit ausgestattet werden müssen, kann das enthaltene Referenzbild in Form eines 2D Barcodes eingebunden werden. Dieses Vorgehen spart zwar Ressourcen und beschleunigt zusätzlich den Ausstellungsprozess, kommt aber mit der Herausforderung, dass 2D Barcodes deutlich weniger Speicherkapazität bieten als RFID Chips. Aus diesem Grund muss das Referenzbild auf eine Größe von 1024 Bytes reduziert werden.

Diese Arbeit untersucht mögliche Vorverarbeitungsschritte und Kompressionseinstellungen, welche genutzt werden können, um diese Zielgröße zu erreichen, ohne dabei die Gesichtserkennung maßgeblich zu beeinträchtigen. Hierfür werden die Kompressionsalgorithmen JPEG, JPEG 2000, JPEG XL, JPEG AI, HEIF, AVIF und WebP verglichen. Während das Referenzbild im Ausweisdokument immer den Vorgaben der ICAO entsprechen muss, können die vor Ort gemachten Probebilder – abhängig vom Anwendungsszenario – diesen ebenfalls entsprechen oder von diesen abweichen. Aus diesem Grund werden in dieser Arbeit die besten Einstellungen für beide genannten Anwendungsfälle evaluiert.

Es zeigt sich, dass in beiden betrachteten Szenarien JPEG AI-komprimierte Referenzbilder – in Kombination mit den optimierten Einstellungen – die beste Gesichtserkennung ermöglichen, wobei auch AVIF und WebP nah an dessen Performance herankommen. Die durch die Kompression verursachte verringerte Leistung der Gesichtserkennung fällt dabei nicht stark aus. Für einen Vergleich von ausschließlich ICAO-konformen Lichtbildern stellt sich eine Graustufenkonvertierung als gut geeignete Vorverarbeitung heraus. Für den Vergleich mit weniger gut geeigneten Gesichtsbildern hingegen ist der Erhalt der Farbe wichtig. Eine weitere Erkenntnis ist, dass das Glätten und Skalieren der Bilder vorab ebenfalls einen positiven Einfluss auf das Ergebnis hat.

ABSTRACT

Biometric face verification requires that a biometric probe can be compared against a reference sample, which in the case of ICAO compliant MRTDs is stored as a JPEG or JPEG 2000 image. In order to avoid equipping temporary ID documents with expensive RFID chips for machine readability, the reference sample should be encoded in conventional 2D barcodes. This saves resources and speeds up the issuing process, but comes with the challenge of storing the face images at significantly smaller storage capacities. For this reason, it is important to reduce the file size of these images to a maximum of 1024 bytes.

This study examines preprocessing steps and compression configurations that can be used to achieve this target size while minimizing the impact on the performance of face recognition algorithms. Therefore seven compression algorithms are examined, namely JPEG, JPEG 2000, JPEG XL, JPEG AI, HEIF, AVIF, and WebP. While the reference sample must always comply with ICAO specifications, the individual samples may or may not meet these requirements, depending on the application. This work identifies the optimal compression steps for both of these scenarios.

It is shown that in both scenarios, JPEG AI, when using optimized settings, provides the best face recognition performance, closely followed by AVIF and WebP. The losses caused by the strong lossy compression are comparatively small. For the comparison of ICAO-compliant face images only, converting the images to grayscale proves to be an important preprocessing step, whereas for comparisons involving less suitable samples, preserving color is essential. In addition, smoothing and resizing the images beforehand also turns out to be beneficial.

ACKNOWLEDGMENTS

Special thanks to my supervisors Prof. Dr. Christoph Busch and Torsten Schlett for their close supervision and for always having an open ear for my questions and problems throughout the entire development period, and for supporting me in the best possible way through important stimuli for thought and critical inquiries.

I am also grateful to my parents, Nils and Christine, who have always supported me throughout my studies and provided me with a place of retreat even during stressful periods.

My thanks also go to my roommates Chantal and Johann, who supported me during the writing phase and, through their help in everyday life, gave me the necessary freedom to work.

Finally, I would like to thank my employer for providing the flexible environment that made it possible to pursue this degree alongside my professional responsibilities.

Portions of the research in this thesis use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

CONTENTS

II Appendix

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABC	Automated Border Control
BSI	Federal Office for Information Security
CLI	Command Line Interface
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EDC	Error versus Discard Characteristic
FIQA	Face Image Quality Assessment
FIQAT	Face Image Quality Assessment Toolkit
FMR	False Match Rate
FNMR	False Non-Match Rate
ICAO	International Civil Aviation Organization
KDE	Kernel Density Estimate
MRTD	Machine Readable Travel Document
OFIQ	Open Face Image Quality
RFID	Radio Frequency Identification
SCRFD	Sample and Computation Redistribution for Efficient Face Detection
VarDCT	Variable-sized Discrete Cosine Transform

Part I

THESIS

1 INTRODUCTION

Biometrics are nowadays a common way to identify or authorize people. Most smartphones support unlocking by finger print or face recognition, ID cards contain portraits of the owner, and even cars are able to recognize the current driver to adjust settings accordingly. The advantages of biometrics are obvious: In contrast to classical keys or passwords one cannot forget or lose a biometric trait and additionally, it cannot be shared with unauthorized third parties. A particularly convenient biometric characteristic is the face. Humans are naturally good at recognizing faces, so it is widely accepted, in addition, it can be captured from a distance with cheap devices (namely cameras) which increases user acceptance. As such, with advancing digitalization it has great potential to be included in more processes especially in public places where identification is needed. One example for such a place would be an airport. Here, thousands of people must be identified every day, to ensure safety. One step towards more digitalized processes are Automated Border Control (ABC) gates.

As their name already suggests, they automate the control process at border crossings. In order to successfully prove your identity here, you need an identity document with a Radio Frequency Identification (RFID) chip that contains a biometric, standard-compliant facial image. This can be read by the ABC gate and compared with a sample probe image. The gate is designed in such a way that the captured sample image also represents the face clearly and visibly. This process enables gates to identify passengers in a matter of seconds by using facial recognition and therefore speeding up the border control process [San+16].

Although this approach offers many advantages, it is nevertheless essential that every user has a Machine Readable Travel Document (MRTD) [Ica]. If one considers, for example, temporary travel documents, these must either be equipped with an RFID chip—which is not an economical option due to their short period of use—or another option must be implemented for integrating a compliant photograph into the document so that it is still machine-readable. A common and already standardized solution for this are 2D barcodes [3124b][3124a][3124c]. These consist of black and white tiles and can be easily captured optically by machines. To enable such an approach, one needs to encode the biometrically compliant photograph in a correspondingly large 2D barcode.

1.1 GOALS OF THIS WORK

In order to embed facial images into 2D barcodes, the images have to have a file size of 1024 bytes or less. Compared to current file sizes of 12 to 15 kB

[Int21] this is a massive decrease. On the other hand they also must provide a quality high enough to be used for identification, which means the features of the face image which are used by the algorithms to calculate the similarity between sample and probe must be preserved. Therefore the main goal of this study is to reduce the file sizes of face images, by using several file size reduction strategies and determine which preprocessings can be beneficial to increase the amount of biometric data preserved in the reduced file.

Three ways to reduce the file size of a face image are used in this work: deleting less important information, reducing resolution, and applying compression algorithms. Each of these steps can decrease the file size by its own, but they also influence each other when used in combination. One goal is to optimize this processing as a process chain so that the results of each step deliver the best outcome for the subsequent steps.

Another important aspect is to prove, that the entire procedure of reducing the file size does not significantly harm the recognition performance. Therefore, self-similarity scores, mated similarity scores, and non-mated similarity scores are calculated and their distributions are compared against the distributions of similarity scores calculated from non-compressed samples. In addition, the changes in face recognition performance are examined.

Last, the impact of the size reduction on face image quality scores is inspected.

1.2 STRUCTURE

The structure of this work is roughly divided into three parts. The first part presents the basic knowledge required to carry out the corresponding investigation. This includes, among other things, standards, options for image compression, as well as processes related to face recognition and findings from previous studies that are relevant to this thesis. This is done in Chapters 2 and 3.

Once these foundations have been laid, Chapters 4 and 5 describe the setup of the experiments conducted for this thesis and the resulting findings. The ideas behind the approach are explained in more detail, the limitations of this approach are discussed, and various metrics are evaluated which, in combination, provide an overview of the results achieved and allow them to be assessed.

The last part of the thesis deals with the discussion of the results obtained. For this purpose, conspicuous observations and their possible causes are explained in Chapter 6. Chapter 7 provides a summary of the preceding findings and Chapter 8 addresses suggestions for improvements and possible follow-up studies.

2 FUNDAMENTALS

This work aims to reduce facial image file sizes, so they can be embedded in 2D barcodes which can then be used in biometric verification systems. Therefore, the images must be compressed without influencing the performance of such identification or authorization processes. This chapter deals with the prerequisites, definitions, standards, and fundamental technical solutions that are required to better understand the following chapters of this thesis.

2.1 2D BARCODES

2D barcodes are a common way to visually embed data into documents, so it is still easily machine readable. With the rise of smartphones, 2D barcodes have entered nearly every aspect of our daily life, for example, product identification, electronic stamps or the menu-card at a restaurant.

2D barcodes consist of multiple units, which are visually distinguishable because of their high contrast (most common are black and white). Together, these units form a binary representation of the data contained. A key advantage of 2D barcodes is that data can be encoded both horizontally and vertically. This ensures a higher data density and, at the same time, improved error tolerance. Thus, for example, it is possible to retrieve the data even if the code is slightly damaged.

The aim of this work is to compress facial images so that they can be encoded in 2D barcodes. Two conditions are important for this. First, the barcode type used must offer sufficient storage capacity and, second, should be standardized. The 2D barcodes that meet these criteria are presented below.

2.1.1 *QR Code*

QR Code is an abbreviation for Quick Response Code [U.S26]. This 2D barcode was developed by the company Denso Wave and has since been standardized in ISO/IEC 18004:2024 [3124b], with the name itself protected as a trademark [DEN26]. QR Codes are widely used due to their robustness and fast readability. A maximum-sized QR Code is shown in Figure 2.1, a second one with exactly 1024 bytes of data is shown in Figure 2.2 for reference. A QR Code essentially consists of black and white modules (squares) that encode the data in binary form. In addition, it contains information on the version (i.e., resolution, marked yellow in Figure 2.1), the format used (marked purple in Figure 2.1), as well as three finder patterns (marked green in Figure 2.1), at least one alignment pattern (only one marked red in Figure 2.1 as an example), and two timing patterns (marked blue in Figure 2.1).

Furthermore, a quiet zone must exist around the QR code to distinguish it from the background. The maximum resolution of a QR Code is Version 40 with 177×177 modules, which provides a storage capacity of 2953 bytes.

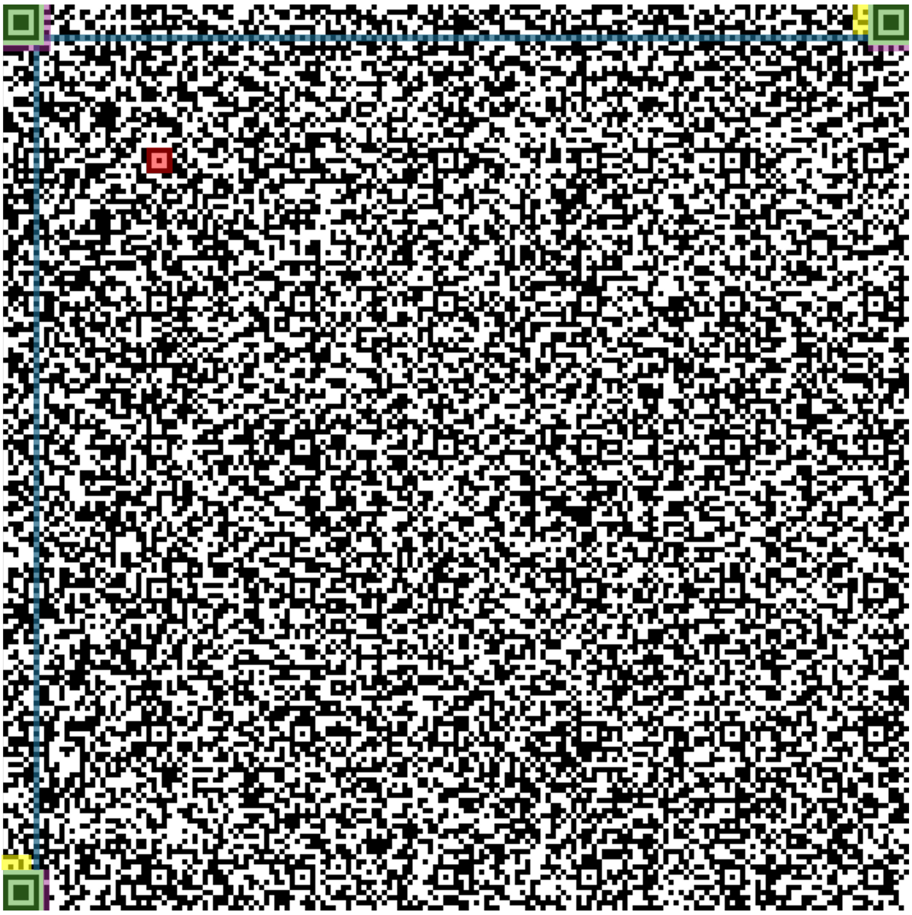


Figure 2.1: This is an example of a maximum-sized QR Code. It consists of 177×177 modules. The colored marking is for illustrative purposes only and does not comply with ISO standards. Only one alignment pattern has been marked red.

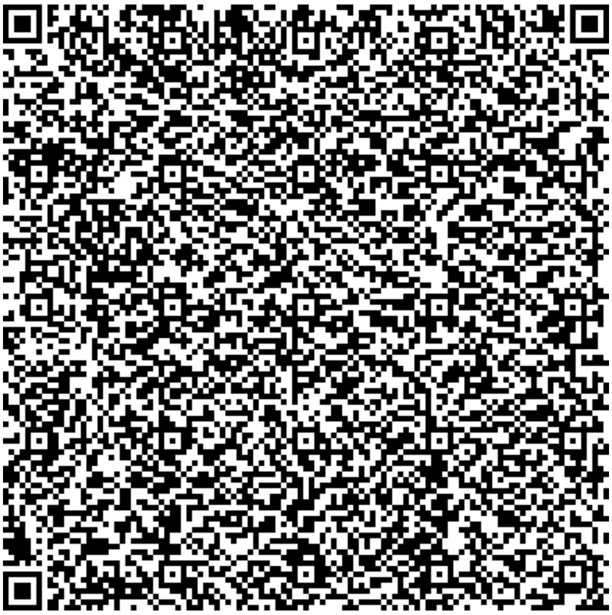


Figure 2.2: This figure shows a QR Code representing exactly 1024 bytes of data.

2.1.2 *Data Matrix Code*

Data Matrix Codes are standardized by ISO/IEC 16022:2024 [3124a]. They consist of dark and light squares. They can contain up to 1555 bytes with 28.5% of the bytes used for error correction. An example of a Data Matrix Code of the maximum size is shown in Figure 2.3 and an example for exactly 1024 bytes of data encoded into a Data Matrix Code is shown in Figure 2.4. The figure illustrates that one Data Matrix Code can contain up to 36 units. Each unit is marked by the finder pattern (marked red in Figure 2.3). This acts as a perimeter to the data region. Its lower and left sides create an L shape consisting of a single color, contrary to the background the Data Matrix Code is printed on. The upper and right sides consist of alternating dark and light squares. A Data Matrix Code includes at least one unit (marked green in Figure 2.3) and at most 36 units. The smallest codes have a resolution of 10×10 squares, the maximum resolution is 144×144 squares. For each size, a strict amount of storage capacity is reserved for error correction.

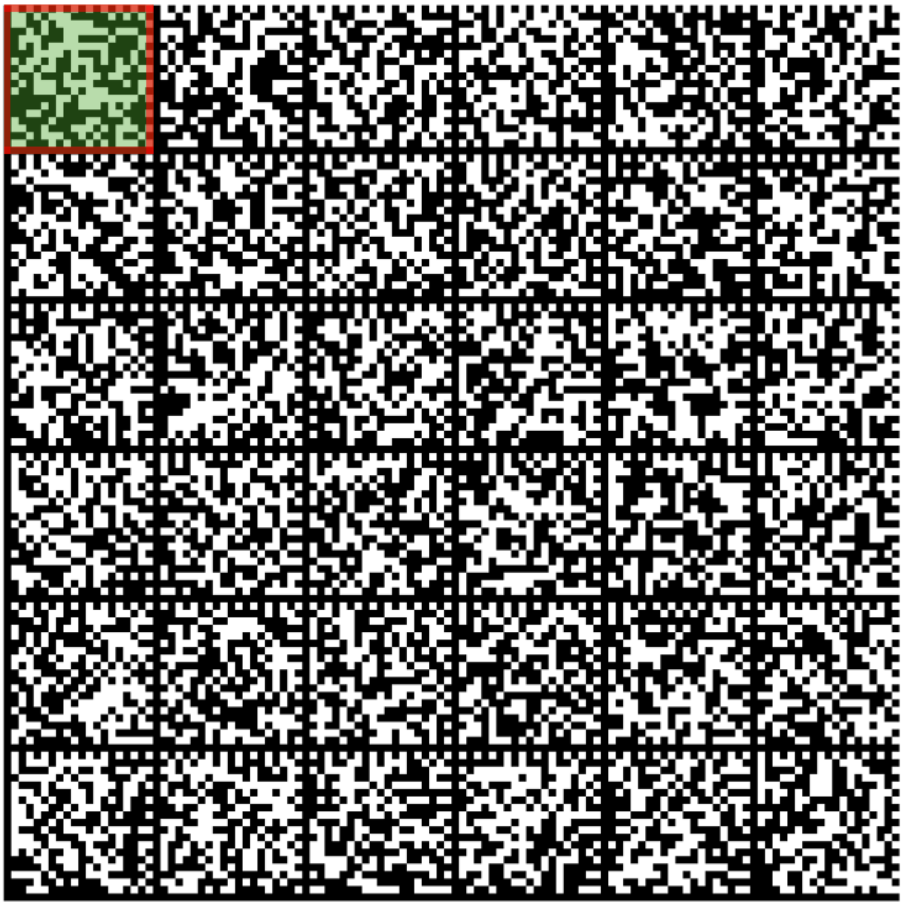


Figure 2.3: This is an example of a maximum Data Matrix Code. It consists of 36 units in total and has a resolution of 144×144 squares (24×24 per unit). The colored marking on the first unit is for illustrative purposes only and does not comply with ISO standards.

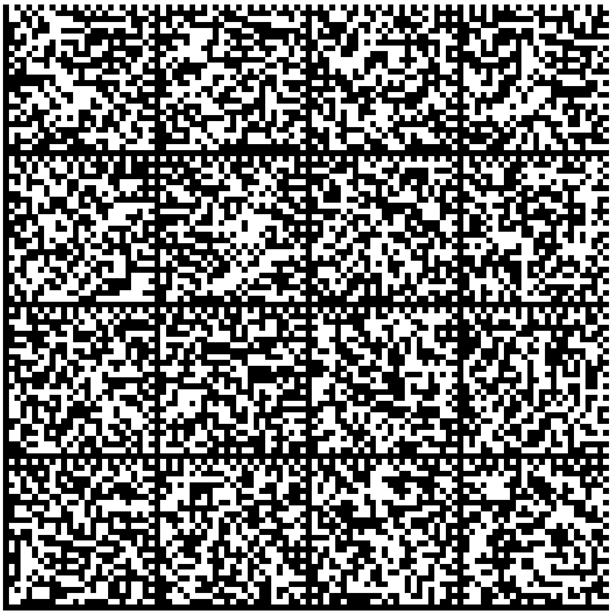


Figure 2.4: This figure shows a Data Matrix Code representing exactly 1024 bytes of data.

2.1.3 *Aztec Code*

The Aztec Code is standardized in ISO/IEC 24778:2024 [3124c]. An example of a maximum-size Aztec Code is shown in Figure 2.5. Figure 2.6 shows an Aztec Code of exactly 1024 bytes of data. It consists of a centrally located fixed structure (marked green in Figure 2.5), which includes the finder pattern, as well as information about orientation and additional metadata, and, surrounding this, annular data layers. For large Aztec Codes, additional alternating structures are incorporated into the outer data layers as a reference grid (marked red in Figure 2.5). The maximum capacity of an Aztec Code is reached at a resolution of 151×151 modules with 1914 bytes. The error correction level for Aztec Codes is always 23%.

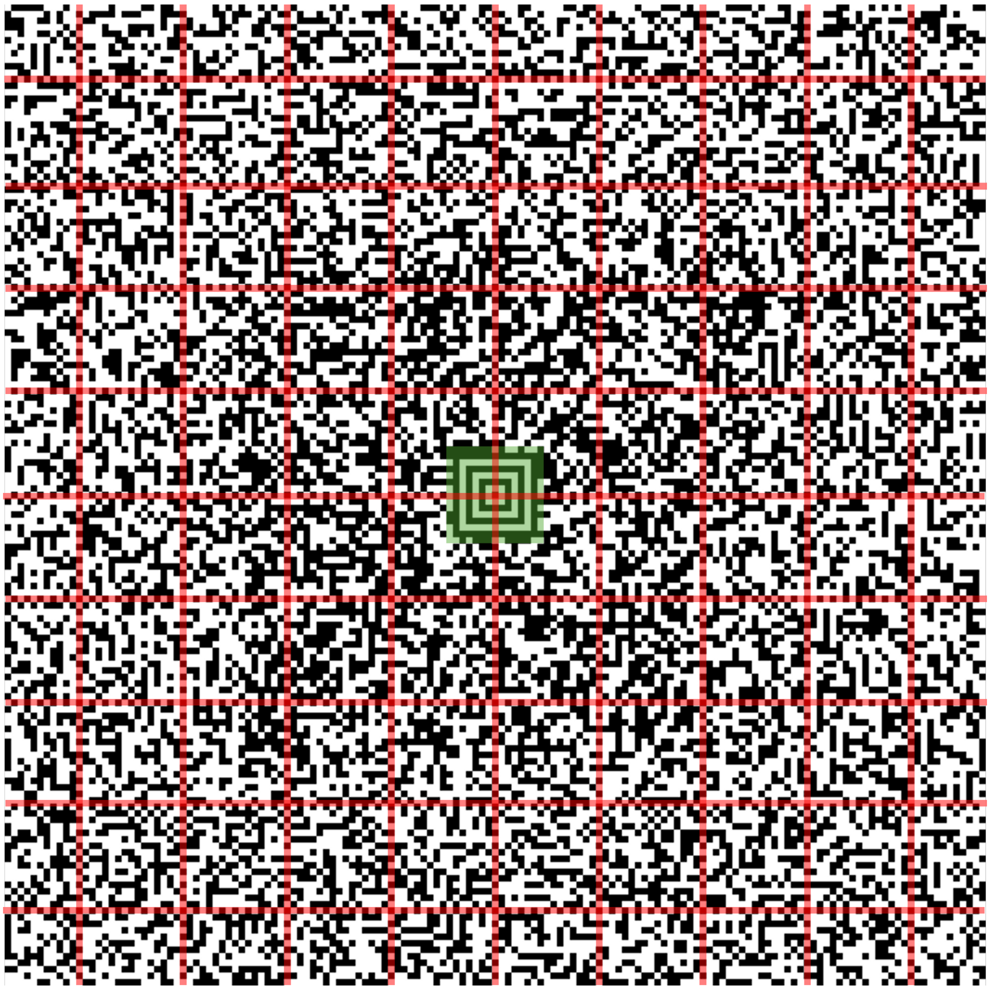


Figure 2.5: This is an example of a maximum size Aztec Code (151×151 squares). The colored marking is for illustrative purposes only and does not comply with ISO standards.

Figure 2.6: This figure shows an Aztec Code representing exactly 1024 bytes of data.

2.2 IMAGE COMPRESSION

Digital images consist of many pixels. In a simple format, each value of a pixel has to be specified individually. The best-known format for describing a pixel is RGB, which is an abbreviation for red, green, blue, and assigns each pixel three values between 0 and 255 that describe the respective proportion of red, green, and blue, which in turn determines the color of the pixel as a whole. This means that each pixel requires 3 bytes of memory. In addition, metadata about the image must be stored. As a consequence a target size of 1024 bytes allows only about 330 pixels to be stored in total. For a square image, which is required by face recognition algorithms, this resolution would only be 18×18 pixels, which would be far too low to be able to store meaningful information about a face at all.

The problem of digital images requiring large amounts of storage capacity is quite old, which is why over the years various methods and formats have been developed to reduce the required storage capacity. These different mechanisms for more efficient storage of images can be distinguished between lossless compression and lossy compression.

Lossless compression searches for other formats in order to store the image data more efficiently without losing information. To achieve this, it can, for example, be useful to no longer describe every pixel individually. In contrast, lossy compression allows information to be discarded. In doing so, properties of human perception are usually exploited, such as the fact that humans are quite sensitive to differences in contrast but do not perceive small details in images particularly well. The color perception of the human eye is also quite unbalanced. For example, humans can distinguish shades of green much better than shades of red [Str25].

The following sections present some compression methods in more detail, as well as the compression algorithms used in this work, which generally combine several of these compression methods.

2.2.1 *Entropy coding*

Entropy coding (also entropy encoding) is part of information theory and aims to losslessly compress information by minimizing redundancy [Str25]. This is limited to a fixed ratio, stated by Shannon's source coding theorem, which states that information can never be lossless compressed to less than its entropy [Goe15][Str25]. This is because symbols with lower appearing-probability contain more information than symbols occurring more frequently. Two very common entropy coding algorithms are explained in the following sections.

2.2.1.1 *Huffman coding*

Huffman coding is a relatively simple form of entropy encoding. It was invented by David A. Huffman in 1952 [Huf52]. The idea is to represent symbols by different amounts of bits based on their probability. This is done by following five steps:

1. Calculate the probability of each symbol and sort them by descending probability. "Sustain a situation as a state." This sentence consists of 10 different symbols (Table 2.1).

Letter	a	s	t	␣	i	u	n	o	e	.
Probability	0.194	0.161	0.161	0.161	0.097	0.065	0.065	0.032	0.032	0.032

Table 2.1: Rounded character probabilities in descending order.

2. Split the symbols into two groups, so that each group has roughly the same probability.
3. Assign the value 1 to the first group and the value 0 to the second one (Table 2.2).

Group Value	Letters	Group Probability
1	a, s, t	0.516
0	␣, i, u, n, o, e, .	0.484

Table 2.2: First Huffman split with near-equal group probabilities.

4. Repeat steps 2 and 3 for each group, until every group contains only one symbol (Table 2.3).
5. For each symbol concatenate the values of the groups it was assigned to, beginning with the group created first (Table 2.4).

Letter	Split 1	Split 2	Split 3	Split 4	Split 5
a	1	1			
s	1	0	1		
t	1	0	0		
␣	0	1	1		
i	0	1	0		
u	0	0	1	1	
n	0	0	1	0	
o	0	0	0	1	
e	0	0	0	0	1
.	0	0	0	0	0

Table 2.3: Huffman split visualization (columns show assigned value at each split).

Letter	Bit Pattern
a	11
s	101
t	100
␣	011
i	010
u	0011
n	0010
o	0001
e	00001
.	00000

Table 2.4: Final Huffman encoding for “Sustain a situation as a state.”

This result can also be represented as a binary tree, called the Huffman tree. When using the new bit patterns for representing this sentence, one only needs 96 bits instead of 124 bits as before, resulting in a compression of around 30%.

2.2.1.2 *Arithmetic Coding*

Arithmetic coding is a more complex algorithm for entropy encoding, which usually yields a better compression rate than Huffman coding [VD22]. The basic principle behind this encoding variant is the representation of a given order of symbols by a floating-point number in the range of 0 to 1 [Pak18]. The encoding is performed in six steps:

1. Get the alphabet of the given sequence and calculate the appearing-probability of each symbol. For "cocoa" this would be the alphabet a, c, o with the following probabilities (Table 2.5):

Letter	a	c	o
Probability	0.2	0.4	0.4

Table 2.5: Probabilities for the word "cocoa".

2. Set the interval $[0, 1]$ as the "current interval".

3. Divide the "current interval" into segments so that each symbol is assigned a segment whose size, as a percentage of the "current interval", corresponds to its probability of occurrence (Table 2.6).

0.0	a	c	c	o	o	**1.0**

Table 2.6: Interval $[0, 1]$ split into three segments, with each segments length representing the probability of its symbol.

4. Select the interval of the next symbol appearing as the "current interval".

5. Repeat the steps 3 and 4 until no more symbols are left in the string that is encoded (Table 2.7).

6. Choose a point in the last interval and convert it to binary. This floating-point represents the arithmetic encoded input. For this case, a valid binary encoding would be 100000110. This is equal to 262 in decimal form and $\frac{262}{512} = 0.51171875$, which is in the resulting interval $[0.5104, 0.51552]$.

Initial interval	**0.0**	a	**c**	**c**	o	o	**1.0**
Interval of c	**0.2**	a	c	c	**o**	**o**	**0.6**
Interval of co	**0.44**	a	**c**	**c**	o	o	**0.6**
Interval of coc	**0.472**	a	c	c	**o**	**o**	**0.536**
Interval of coco	**0.5104**	**a**	c	c	o	o	**0.536**
Interval of cocoa	**0.5104**	a	c	c	o	o	**0.51552**

Table 2.7: Each line shows another iteration of the procedure. In the process, the interval that represents the string up to that point keeps shrinking, until ultimately only the interval $[0.5104, 0.51552]$ remains.

2.2.2 *Dictionary-based compression*

Another method of lossless compression is dictionary-based compression. In this approach, recurring patterns in the original data are identified and stored in a kind of dictionary. For compression, these patterns are replaced by references to the corresponding record in the dictionary [Str25].
Run-length coding can be considered a special form of dictionary-based compression. In simplified terms, it stores, how many times a symbol occurs consecutively. Especially in images with large areas of uniform color, this approach is very efficient [Str25].

2.2.3 *Transformation-based compression*

While graphics such as logos or text usually have very sharp edges, where the values of two neighboring pixels can differ significantly, this is hardly the case for photographic images such as face images. Here, neighboring pixels are often similar and transitions are smooth from a computer vision perspective. For such images, transformation-based compression is suitable. The two most important methods in this category are Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT).

2.2.3.1 *Discrete Cosine Transform*

The DCT is derived from the Discrete Fourier Transform and aims to represent a row or a column as a signal that is defined solely by coefficients [Str25]. The DCT is applied first row-wise and then column-wise; as a result, a 2D coefficient matrix can be created. For compression, the coefficients can then be quantized, which is lossy but has little to no impact on human perception.

2.2.3.2 *Discrete Wavelet Transform*

Like DCT, DWT is based on signal processing. Two filters are applied to the image, each along rows and columns. One filter detects contrasts based on high frequencies (high-pass); the other detects smooth regions based on low frequencies (low-pass). Based on these filters, four bands are created: an approximation of the image (LL) and three bands for details in horizontal

(LH), vertical (HL), and diagonal (HH) directions [2924a]. The advantage is that this step can now be applied recursively to the approximation. The DWT is invertible and can be made lossy through the choice of wavelet filters and quantization.

2.2.4 *Color space conversion*

As already mentioned above, human perception is better suited for detecting contrasts and differences in brightness than detailed color differences. There are also differences between individual colors, presumably for evolutionary reasons. Shades of green and yellow can be distinguished better than shades of red or blue. However, this fact is not taken into account by the RGB color system. Other color systems such as YCbCr take this into consideration and are therefore sometimes better suited to represent images with less storage space without having to accept visual losses [Str25].

Alternatively, it is also possible to convert to other color systems with loss. Instead of describing each pixel with three values as in RGB, a grayscale image can be created that only has to store a single gray value per pixel. Since in this case it is primarily contrasts and brightness that are preserved, this is often sufficiently detailed for human perception.

2.2.5 *Reduction of resolution*

Another way to reduce the file size of an image is to lower its resolution. Every image has a fixed number of vertical and horizontal pixels. If this number is reduced, fewer pixels need to be stored overall, and the storage requirement decreases. However, this process deletes information. If the resolution is greatly reduced, the textures in images become coarser as a result.

2.2.6 *Deleting unnecessary information*

Of course, other forms of file size reduction also involve the deletion of information. In particular, with methods such as quantization based on DCT, information is discarded that is barely perceptible to the human eye [Str25]. However, when it comes to the task of compressing facial images, clearly perceptible information can also be deleted. For example, other parts of the body visible in the images or objects in the background are completely irrelevant in this context and can therefore simply be removed.

2.2.7 *Compression algorithms*

In the preceding sections, several common methods that can be used for image compression were presented. The following sections will now introduce compression algorithms that are used in this thesis and make at least partial use of the methods described.

2.2.7.1 *JPEG*

JPEG (also JPEG 1) is the oldest image compression algorithm used in this study. It was standardized in 1992 in ISO/IEC 10918-1 [2994] and is an acronym for Joint Photographic Experts Group, which founded the standard.

JPEG compression is performed in several steps: First, the original image is converted to YCbCr and divided into blocks of 8×8 pixels. For each block, a DCT is then applied, followed by quantization of the coefficients. After that, the data is further compressed using entropy coding (most common here is Huffman coding).

The biggest problem with JPEG is the independent compression of the blocks. This leads to visible edges at the boundaries between blocks when the compression is stronger. This phenomenon is also referred to as block artifacts.

2.2.7.2 *JPEG 2000*

JPEG 2000 was published in year 2000 by the Joint Photographic Experts Group as ISO/IEC 15444-1:2024 [2924a] and is a successor of JPEG. Instead of DCT it uses DWT. It has several advantages in comparison to JPEG, including better compression with equal image quality, lossless image compression, as well as more supported color channels, more supported bits per channel, and larger image resolutions.

In its basic structure, JPEG 2000 works similarly to JPEG. First, the image is converted into a color space such as YCbCr or YUV. Then, optionally, the entire image can be divided into different rectangular tiles, which are subsequently processed independently of one another. After these preprocessing steps, the DWT is performed, followed by quantization. After that, arithmetic coding takes place [Str25].

2.2.7.3 *JPEG XL*

In 2021 the Joint Photographic Experts Group released another successor of JPEG called JPEG XL, which is standardized by ISO/IEC 18181-1:2024 [2924b] and is planned to replace JPEG in the future. In contrast to JPEG 2000, JPEG XL is backwards compatible to JPEG. It also features two modes for compression. For lossy compression it uses Variable-sized Discrete Cosine Transform (VarDCT), which allows the application of a DCT on blocks from 2×2 up to 256×256. For lossless compression a modified Haar wavelet is used [Ala+23].

JPEG XL works according to the following scheme: preprocessing, i.e. conversion into a color system such as XYB and splitting the image into blocks of similar texture, transformation, depending on usage either VarDCT (lossy) or modular (lossless using modified Haar wavelet), followed by quantization and subsequent entropy encoding [Ala+23].

2.2.7.4 *JPEG AI*

JPEG AI was released in 2021 by the Joint Photographic Experts Group and realizes a new approach for image compression using neuronal networks. It is standardized in ISO/IEC 6048-1:2025 [2925c]. Instead of implementing mathematical compression functions, JPEG AI uses an autoencoder consisting of an learning-based image encoder and an learning-based image decoder. The learning-based image encoder is a neural network trained to extract features of an image in a latent space vector, which contains enough information to reconstruct the image with a counter network, namely the learning-based image decoder. This technology has been shown to achieve much better results than common handcrafted solutions [Bou+22]. This improvement comes with two major disadvantages. First, the processing time takes much longer than classical methods. Second, such systems lack explainability, since no one can exactly tell, what the neuronal networks do internally.

The structure of JPEG AI is as follows: First, the original image is converted into the YUV color space and transformed into the 4:2:0 YUV format. Then, two neural networks are used to separately extract the latent space vectors from the Y channel and the UV channels. After that, entropy encoding is performed using an asymmetric numeral system [ISO24].

2.2.7.5 *HEIF*

HEIF is the acronym for High Efficiency Image File Format and is standardized in ISO/IEC 23008-12:2025 by the Moving Picture Experts Group (MPEG) [2925a]. This means that HEIF is not a compression algorithm but rather a kind of fixed, well-defined container that can be used to store compressed image data. The standard compression algorithm used is High Efficiency Video Coding, which is the long term for HEVC (also known as H.265 or MPEG-H Part 2). It was developed by the Joint Collaborative Team on Video Coding (JCT-VC) and standardized in 2013 by ISO/IEC 23008-2 [2925b] and, as the name suggests, was designed for the compression of videos. HEVC-compressed images stored in HEIF are also referred to as HEIC. The major drawback of this technology is that the HEVC codec is subject to licensing [Via26]. All of the following pages use HEIF synonymously for HEVC-compressed images in the format defined by HEIF.

The procedure of this compression algorithm is as follows: The image is hierarchically divided into square blocks. Within these blocks, dependencies on already encoded pixels are recorded in order to reduce redundancy. Then a DCT-like transform is applied, followed by quantization. After this step, entropy coding is performed using Context-based Adaptive Binary Arithmetic Coding (CABAC). Finally, filters are applied to reduce blocking artifacts and enhance edges [Str25].

2.2.7.6 *AVIF*

AVIF stands for AV1 Image File Format and was developed by the Alliance for Open Media [All26b]. Like HEIF, AVIF is not a compression algorithm, but rather describes the use of the AV1 video codec in combination with the storage format defined in ISO/IEC 23008-12:2025 (HEIF) [2925a]. In contrast to HEVC, however, this compression method is an open standard. All of the following pages use AVIF synonymously for AV1-compressed images in the format defined by HEIF.
The AV1 codec divides an image into multiple superblocks, which in turn can be divided into sub-blocks. These do not have to be square and are generally much more flexible in size. Similarly to HEVC, redundancy is then reduced by describing blocks to be encoded on the basis of pixels that have already been coded. Depending on the size of the block, AV1 uses DCT or the Asymmetric Discrete Sine Transform. The resulting coefficients are then quantized, and context-adaptive arithmetic coding is performed. Finally, filters are applied to suppress blocking artifacts and enhance details [Han+21].

2.2.7.7 *WebP*

WebP is a compression algorithm that was developed by Google in 2016 [Goo26a]. The format of the compressed data is based on RIFF. WebP offers both lossless and lossy compression and therefore operates differently in each mode [Goo26b].
For lossy compression, the method is based on VP8, a video compression scheme that was released in 2008. For this purpose, the image is first converted to YCbCr and then divided into 16×16 blocks. These are further divided into either 4×4 or 16×16 luma blocks and 8×8 chroma blocks. As with AV1 and HEVC, the blocks are then described using already coded pixels. This is followed by DCT with subsequent quantization of the coefficients. After that, arithmetic coding is applied. Lossy WebP also uses adaptive block quantization, in which the image is divided into visually similar regions that can be encoded in a similar way [Goo26b].
Lossless WebP, on the other hand, operates in the RGB color space. First, a predictor transform is performed, which exploits the fact that neighboring pixels in photographs often correlate with each other. Through this operation, only the differences to already coded pixels are stored. WebP then performs a color transform and creates color palettes that can be reused for the encoding of other regions. This is followed first by dictionary-based encoding and then by Huffman coding [Goo26b].

2.3 STANDARDIZED TERMINOLOGY

Standardization is an important step to enable interoperability between systems. This also includes defining a uniform vocabulary that ensures all parties involved refer to the same definition for a given term. Otherwise, misunderstandings can quickly arise due to mutual misinterpretations. In the case of biometrics, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) created a foundation for communication with the ISO/IEC 2382-37:2022 standard [3722]. The following lists selected definitions from this standard that are important for understanding the subsequent pages. (Note: Since these are fixed definitions, they are adopted word by word from the standard. Otherwise, their intended meaning would be lost.)

- **biometric system:** system for the purpose of the biometric recognition of individuals based on their behavioural and biological characteristics
- **biometric data:** biometric sample or aggregation of biometric samples at any stage of processing
- **biometric data record:** data record containing biometric data
- **biometric feature:** number or label extracted from biometric samples and used for comparison
- **biometric probe, biometric query:** biometric sample or biometric feature set input to an algorithm for comparison to a biometric reference(s)
- **biometric sample:** analogue or digital representation of biometric characteristics prior to biometric feature extraction
- **biometric characteristic:** DEPRECATED biometric, biological and behavioural characteristic of an individual from which distinguishing, repeatable biometric features can be extracted for the purpose of biometric recognition
- **biometric feature extraction:** process applied to a biometric sample with the intent of isolating and outputting repeatable and distinctive numbers or labels which can be compared to those extracted from other biometric samples
- **comparison:** DEPRECATED match, DEPRECATED matching, estimation, calculation or measurement of similarity or dissimilarity between a biometric probe(s) and a biometric reference(s)
- **biometric template, reference biometric feature set:** set of stored biometric features comparable directly to a biometric probe
- **biometric reference:** one or more stored biometric samples, biometric templates or biometric models attributed to a biometric data subject and used as the object of biometric comparison
- **comparison score:** DEPRECATED matching score, numerical value (or set of values) resulting from a comparison
- **false match:** comparison decision of a match for a biometric probe and a biometric reference that are from different biometric capture subjects

- **false match rate, FMR:** proportion of the completed biometric non-mated comparison trials that result in a false match
- **false non-match:** comparison decision of non-match for a biometric probe and a biometric reference that are from the same biometric capture subject and of the same biometric characteristic
- **false non-match rate, FNMR:** proportion of the completed biometric mated comparison trials that result in a false non-match
- **quality:** degree to which a biometric sample meets the specified requirements for its targeted application
- **biometric character:** set of attributes associated with a biometric characteristic that cannot be controlled during the biometric acquisition process
- **authentication:** act of proving or showing to be of undisputed origin or veracity
- **biometric identification:** process of searching against a biometric enrolment database to find and return the biometric reference identifier(s) attributable to a single individual
- **biometric verification:** process of confirming a biometric claim through comparison
- **similarity score:** comparison score that increases with similarity
- **threshold:** noun, numerical value (or set of values) at which a decision boundary exists
- **non-mated:** adj, based on a paired biometric probe and biometric reference that are not from the same biometric characteristic of the same biometric data subject
- **mated:** adj, based on a paired biometric probe and biometric reference that are from the same biometric characteristic of the same biometric data subject
- **match:** noun, comparison decision stating that the biometric probe(s) and the biometric reference are from the same source
- **non-match:** noun, comparison decision stating that the biometric probe(s) and the biometric reference are not from the same source
- **biometric data subject:** individual whose individualized biometric data is within the biometric system
- **comparison decision:** determination of whether the biometric probe(s) and biometric reference(s) have the same biometric source, based on a comparison score(s), a decision policy(ies) including a threshold and possibly other inputs

- **biometric property:** descriptive attributes of the biometric data subject estimated or derived from the biometric sample by automated means
- **biometric enrolment:** DEPRECATED registration, act of creating and storing a biometric enrolment data record in accordance with an enrolment policy
- **quality score:** quantitative value of the fitness of a biometric sample to accomplish or fulfil the comparison decision

2.4 THE FACE AS A BIOMETRIC CHARACTERISTIC

The face is a biometric characteristic. Compared to other biometric characteristics, it offers several advantages. Humans are fairly good at face recognition, which makes biometric recognition systems using the face very convenient and keeps the human in the loop, as an algorithm decision can be confirmed by a human expert. In addition, faces are one of the few biometric characteristics that can be easily captured from a distance, further increasing the convenience.
For this reason, the use of this biometric characteristic has already been integrated into many aspects of everyday life, whether for unlocking one's own smartphone or for verifying identity using an identification document, which contains a photograph in accordance with International Civil Aviation Organization (ICAO) standards [Ica].
The following sections provide an overview of how the technical processing of the face for use in a biometric context works, which standards must be met in this regard, and which software is used for this purpose in the following chapters.

2.4.1 *Face images*

In order to be able to use the face for a biometric recognition system, a biometric sample must first be taken. This is usually a photograph in which the face is clearly visible. In the following, such an image is referred to as a facial image or face image.
Since this work essentially focuses on the use of facial images in the context of MRTDs, the ICAO requirements for these facial images are important. For a facial image to be ICAO-compliant, it must have been taken within the last 6 months, have a width of 35 mm and a height of 45 mm, and show the entire head. The head must be positioned in the center of the image and occupy between 70% and 80% of the total image area. The photo must be sharp and of high quality, meaning that the colors must represent reality accurately and balanced contrasts should be depicted. The lighting of the image must be uniform; no shadows or reflections may be visible in the picture. The background of the image must be a single color and must contrast with the face and hair. The person's position must be frontal to the camera. The head must be straight and not tilted in any direction. The eyes must be open and

the mouth closed. In addition, the facial expression must be neutral. The eyes must look directly into the camera, and the face must not be obscured by hair, clothing, or other accessories [Int21].

In addition to the properties of the facial image, the ICAO also defines the structure of MRTDs. These must contain an RFID chip with a storage capacity of 32 kB. Of this capacity, 12 kB must be used for the facial image [Int21]. Thus, the storage capacity of the RFID chip is significantly higher than that of a 2D barcode.

2.4.2 *Face recognition*

In order to use face recognition, a corresponding face recognition system is required. During the enrolment process, this system creates a biometric reference based on one or more biometric samples, in this case facial images. It is important that optimal conditions prevail during enrolment and that the process ensures no false information can be smuggled into the system.

Once the biometric reference has been stored in the system, it can be used for biometric verification. For this, a biometric probe is taken from a subject (a photo of the face at the time of the request) and a biometric feature extraction is performed so that the biometric features of the biometric sample can be compared with those of the biometric reference. In the case of a face recognition system, the similarity of the biometric features is determined, which is expressed by a similarity score.

Since this thesis aims to investigate how facial images can be reduced to a file size of 1024 bytes and how their use can be designed for biometric verification systems, the effects on performance – that is, the fulfillment of the functional and non-functional requirements of a face recognition system – must also be examined. Various software is used for these steps.

2.4.2.1 *Face detection*

In the first processing step, the given image must be examined for the faces contained. A toolkit that provides this functionality is Face Image Quality Assessment Toolkit (FIQAT) [Sch24]. This toolkit comes with all the functions for FIQA, including all steps of face recognition.

Specifically, here the Sample and Computation Redistribution for Efficient Face Detection (SCRFD) algorithm [Guo+21] from the InsightFace project [Ins26] is used. Among other things, for each face detected, five landmarks are returned that mark the centers of the eyes, the tip of the nose, and the two corners of the mouth. Since ICAO-compliant images may contain only one face, this information is sufficient to crop the face and prepare it in such a way that it can be used for the subsequent steps.

In addition to FIQAT, the OFIQ software library [BSI26][Mer+24], which is maintained by the Federal Office for Information Security (BSI) is also used. It likewise pursues the goal of FIQA and provides additional options for detecting a face in an image. These include, on the one hand, an algorithm for extracting 99 landmarks that delimit the relevant region of the face and, on the other hand, a segmentation algorithm that can separate the individual parts of the face from the background. While OFIQ is not used in this work in the process of biometric feature extraction, the algorithm for extracting the 98 landmarks is used in part of the preprocessing for compression.

2.4.2.2 *Feature extraction*

After processing the facial image accordingly (detecting, aligning, cropping, rescaling) the biometric feature extraction can be performed. While both FIQAT and OFIQ offer corresponding algorithms, this work uses a more state-of-the-art algorithm. This algorithm is an openly available model from the CVLface library [Kim24] using a ViT-KPRPE architecture [Kim+24] trained with AdaFace [KJL23] loss on WebFace12M, and will be referred to as "AdaFace" herein.
The application of this algorithm produces, for an input image, a feature vector of 512 dimensions, where each dimension represents a biometric feature by means of a floating-point number. This vector is the result of the biometric feature extraction and can be used for comparison.

2.4.2.3 *Comparison*

To determine how similar two faces are, the cosine similarity of the respective feature vectors is calculated. The formula for this is:

$$\textit{cosine similarity} = \frac{\vec{x} \times \vec{y}}{\|\vec{x}\| \, \|\vec{y}\|}$$

The result of this formula is the similarity score, which lies in the interval $[-1, 1]$. In this thesis, three different types of similarity scores are distinguished:

- Self-similarity scores: This term is used to describe similarity scores resulting from comparing a facial image with itself. These scores are obtained by comparing the original face image with its compressed counterpart. Because the face content is exactly the same, the value is expected to be close to 1. A lower self-similarity score reflects stronger compression artifacts, as it represents the difference between the original and the compressed image.

- Mated similarity scores: Mated similarity scores are similarity scores obtained by comparing two different face images of the same individual.

- Non-mated similarity scores: Every similarity score received by comparing two faces of different individuals is a non-mated similarity score.

In order to make a comparison decision based on similarity scores, a threshold must also be defined. In a perfect face recognition system, all mated and self-similarity scores lie above this threshold and all non-mated similarity scores below it. For this to be the case, the respective similarity score distributions would have to be positioned in such a way that there is no overlap between them. Since there are no perfect systems in reality, this leads to incorrect decisions in some cases. As a consequence, false matches and false non-matches occur. Analyzing the frequency of these cases is one option, to examine what influence the reduction of facial images to a storage size of 1024 bytes has on the face recognition system.

2.4.3 *Face image quality*

In addition to evaluating face recognition performance, this work also examines the impact of reducing face images to 1024 bytes on face image quality. The quality measures for face images are defined in ISO/IEC 29794-5:2025 [3725]. For the evaluation in this work, two FIQA algorithms are used. These algorithms take a face image as input and provide an assessment of its quality. This assessment can refer to specific aspects or represent an overall evaluation of the image. In the latter case, the corresponding FIQA algorithm returns a scalar that expresses the quality. This is the unified quality score. The range of values that quality scores can take depends on the respective FIQA algorithm.
The two FIQA algorithms used are OFIQ Unified Quality Score model (also OFIQ UQS), which is based on MagFace [Men+21], and ViT-FIQA (C) [ABD25], a novel model based on Vision Transformer architectures that have predominantly been used for face recognition, but have also been proven to be reliable tools for FIQA.

3 RELATED WORK

This chapter provides an overview of existing work that addresses research questions or objectives similar to those pursued in this study. As in this study, the effects of lossy compression on the performance of biometric systems were investigated. Up to now, the main focus has been on the different performance levels of various compression algorithms. One aspect of the investigation that distinguishes this work from existing studies is its focus on the preprocessing of the data to be compressed.

3.1 COMPRESSION TO A FIXED TARGET SIZE

A basic prerequisite that is essential in this work is the target size of 1024 bytes for the compressed face images. Many related works focus on how the error rates develop during increasing compression ratio using different compression algorithms. The actual compression ratio used is not a relevant aspect in this work, because the one that is always chosen is the one that reduces the original image to a file size as close as possible to, but still equal to or below, the given target file size of 1024 bytes.

A study that also followed this approach is "NIST SP 500-343" by Grother et al. [GNH25]. The fact that their study follows the same approach as this thesis is no coincidence, since its research question is likewise aimed at encoding facial images from an identity document into 2D barcodes. NIST SP 500-343 uses QR Codes as storage medium. For this purpose, the authors examined target file sizes of 600, 800, 960, 1040, and 1200 bytes. To reduce the images to such small file sizes, they undergo a preprocessing pipeline that first detects the face, crops this region with a 25% expansion, and then reduces the resolution to fixed aspect ratios. They found applying a blur before this resizing step to be beneficial in order to avoid aliasing artifacts.

The authors investigate the suitability of the compression algorithms JPEG, JPEG 2000, JPEG LI, JPEG XL, HEIC, AVIF, and WebP. They found that JPEG and JPEG LI are unsuitable for such a purpose, as they are often unable to achieve the specified target sizes while preserving sufficient detail to allow facial recognition on the compressed images.

Subsequently, Grother et al. examined the error rates of the compressed images for the different combinations of resolution, target size, and compression algorithm and found that from a target size of 960 bytes upward, error rates similar to those of uncompressed images become achievable. The compression algorithms found to be the most suitable for this were WebP, AVIF, HEIC, and JPEG 2000. Overall, the authors observed that the mated similarity scores decrease on average due to the compression of the images. Another finding of the authors is that reducing the resolution has less negative impact on face recognition performance than lossy compression, which is why a low resolution of the images is recommended.

Another study that deals with the face recognition performance of images compressed to predefined target file sizes is "Effect of Lossy Compression Algorithms on Face Image Quality and Recognition" by Schlett et al. [Sch+23]. In this work, the authors examined the compression algorithms JPEG, JPEG 2000, and JPEG XL, as well as the reduction of the file size of PNG images by rescaling them to target sizes of 5 kB, 4.5 kB, 4 kB, 3.5 kB, 3 kB, 2.5 kB, and 2.2 kB. To achieve these file sizes, the authors first cropped the faces at their original size, then compressed or resized them to the target size, and subsequently analyzed both the face recognition performance of the compressed images relative to one another and the evolution of face image quality.

It turned out that out of the methods examined, JPEG XL is the most suitable algorithm for compressing the images without causing severe losses in face recognition performance. Moreover, only JPEG was found to be unsuitable for particularly small target sizes, which is consistent with the findings of Grother et al. [GNH25]. With regard to face image quality, the authors observed a slight reduction, which was comparatively minor.

3.2 GENERAL IMPACT OF COMPRESSION ON BIOMETRIC SYSTEMS

This section deals with literature that does not focus on specific quantitative image metrics like the file size, but rather on the general investigation of how suitable different compression algorithms are for usage in biometric recognition systems. In doing so, it covers not only the literature that examines face recognition.

In “Impact of Conventional and AI-based Image Coding on AI-based Face Recognition Performance” by Bousnina et al. [Bou+22], the authors examine the differences in the impact of conventional compression algorithms compared to AI-based compression algorithms on the face recognition performance of ArcFace. To this end, the authors compare the effects of JPEG, JPEG 2000, and JPEG XL with those of Balle2018, Cheng2020, and the HiFiC codec. They evaluated the peak signal-to-noise ratios, the multi-scale structural similarity index measure, as well as False Match Rate (FMR) and FNMR. It turns out that the AI-based compression algorithms provide both slightly better image quality and improved face recognition performance at the same compression rate compared to the conventional algorithms. Another finding of the authors is that, although face recognition performance is generally negatively affected by compression, in some cases an improvement occurs. They attribute this to the noise-filtering properties of the compression algorithms, which, at medium to high compression rates, may have the effect that the compressed images are less noisy, which in turn would benefit face recognition performance.

A study that does not deal with the effects of compression on facial images, but rather on iris images, is “Iris Image Compression Using Deep Convolutional Neural Networks” by Jalilian et al. [JHU22]. The aim of this investigation was, on the one hand, to evaluate DSSLIC with regard to its suitability for the compression of iris images and, on the other hand, to compare this compression algorithm with the traditional algorithms JPEG, JPEG 2000, BPG, HEVC, VCC, and AV1.

The results of the study show that the deep-learning-based compression algorithm DSSLIC, which is an acronym for Deep Semantic Segmentation-based Layered Image Compression, performs as well as, or better than the conventional compression algorithms for all metrics examined. With increasing compression ratio, recognition performance decreased; however, the better compression performance of DSSLIC had a directly positive effect on recognition performance in this context. It was also found that compression made the images overall more dissimilar, which is why the non-mated similarity scores were only slightly affected, while only the mated similarity scores deteriorated overall, which accounts for the reduced recognition performance. As a conclusion of the study, the deep-learning-based DSSLIC approach was identified as a promising method for the compression of iris images. Note: This algorithm was not included in this thesis because the time constraints would not have permitted it and, in addition, no information about its licensing conditions could be found, which would have posed a problem for the underlying application scenario.

3.3 IMAGE MANIPULATION TO SUPPORT COMPRESSION

The approach of this thesis, namely taking measures already before compression that improve its results and thus reduce the overall negative impact on face recognition performance, is by no means new. A study that applies this approach in the context of finger vein images is “Finger Vein Image Compression with Uniform Background” by Maser et al. [Mas+19]. The aim of the authors was to replace the background of the images with a uniform gray and to investigate the resulting effects on compression for lossy and lossless algorithms, as well as on recognition accuracy in the case of lossy compression. For lossless compression, lossless JPEG, JPEG XR, GIF, PNG, JPEG LS and ZIP were used; for lossy compression, JPEG, JPEG 2000, JPEG XR and BPG were examined.

The results of the study show that replacing the background with a uniform gray improves the compression performance of lossless algorithms. However, this effect cannot be transferred to lossy compression. In this case, recognition performance was often negatively affected by this manipulation of the original image. The authors’ explanation for this phenomenon lies in the strong edges between background and foreground that were introduced by this operation. Along these edges, lossy compression led to the formation of artifacts, which in turn had a negative impact on recognition accuracy.

4 EXPERIMENTAL SETUP

This chapter describes the steps required to find the optimized parameters. For this purpose, two sub-datasets must be prepared and a collection of possible parameters must be created that can be used for optimization. In addition, a process must be designed that enables the comparison of the influence the parameters have on the compressed results. At the end of the chapter, the limitations of the chosen approach are discussed.

4.1 IMAGE SELECTION AND PREPARATION

This section describes how the given ColorFERET dataset [Phi+98][Phi+00] is structured and which subsets are created from it in order to carry out the different stages of this study. Afterwards, the preprocessing steps performed are described, making the available images compatible with parameter optimization, compression, and the assessment of face recognition performance later on.

4.1.1 *The ColorFERET dataset*

The ColorFERET dataset consists of a total of 11,128 uncompressed facial images belonging to 966 individuals. Images of each individual can be included from different perspectives. These can include frontal views, slight rotation, side profiles, with neutral and other facial expressions, different clothing, different backgrounds, different hairstyles, as well as different eye levels. The ColorFERET database thus reflects the perspectives of an application scenario where images are taken more or less in the wild. For reasons of better readability and standardization, this entire dataset will be referred to as the "full dataset" in the following chapters.

Since images used for compression must later on meet the requirements for an identification document, it is not feasible to use all images of the dataset for compression evaluation. These must be taken under certain conditions, as stated in international standards such as ISO/IEC 19794-5 [3711] and ISO/IEC 39794-5 [3719] and demanded by ICAO [Ica]. Consequently, only a subset of the original data set containing exclusively frontal images is used to evaluate the effects of compression on face recognition. This subset will be referred to as the "frontal dataset" in the rest of this study. It comprises 2,638 images of 966 individuals. Not all of the used images show strictly neutral facial expressions, since the provided ColorFERET data lacked labels to easily enable such a selection.

Since the search for optimal parameters and further preprocessing steps is iterative, the frontal dataset must be reduced further. Because the various adjustment parameters can be combined with each other, there are too many possibilities to process all frontal face images per algorithm each time. For this reason, another subset is created for the search, which contains only 10 images of 10 individuals (one image per individual), which are manually chosen to be as diverse as possible. This subset will be referred to as the "parameter search dataset". Based on the parameter search dataset, all effects of the settings are examined in the following and afterwards only the best 3 settings are validated on the frontal dataset.

4.1.2 *Image preparation process*

In order to reduce the face image file size to 1024 bytes or less, different preparation steps are taken. First, the relevant face area of each image is cropped out and aligned. This is always a necessary step, when using face recognition algorithms and already reduces the file size by a significant amount. Afterwards, a grayscale version of the images is created. This should further reduce the file sizes to only a third. After these two steps there are now two versions of each dataset (color and grayscale), which will be evaluated separately.

4.1.2.1 *Alignment and cropping*

Reducing the file size of an image to such small numbers, comes with loss of information. Since most of the information in the original images from ColorFERET is not needed for face recognition, it is first removed. The FIQAT [Sch24] is used for this purpose. The specific facial region is cut out in three steps. First, all possible faces in an image are detected using SCRFD [Guo+21]. Then, the face that the algorithm considers to be the most reliable is selected, which in the case of ColorFERET is always the only face in the image. In the next step, this face is first aligned along five landmarks (corners of the mouth, center of the eyes, and tip of the nose) extracted by SCRFD [Guo+21], and then the face is cut out. In the final step, the resolution of the cut-out face is adjusted to 112×112 pixels without changing the geometry of the face. The effects of this image processing are shown in Figure 4.1. AdaFace requires this resolution of 112×112 pixels. Since in later processing steps the compressed images have to be compared with the image that would be used in a comparison without any file size limitation, this version is needed anyway and is stored as the original for all future image variants. In addition, for the parameter optimization part, further resolutions greater than 112×112 pixels are created. The adaptation to different resolutions is carried out by the native function of the FIQAT.

4.1.2.2 *Grayscale conversion*

After cropping out the faces, a grayscale conversion is performed for each image. The idea behind this step is that original images are stored with color, where each pixel is described by 24 bits, while after a grayscale conversion

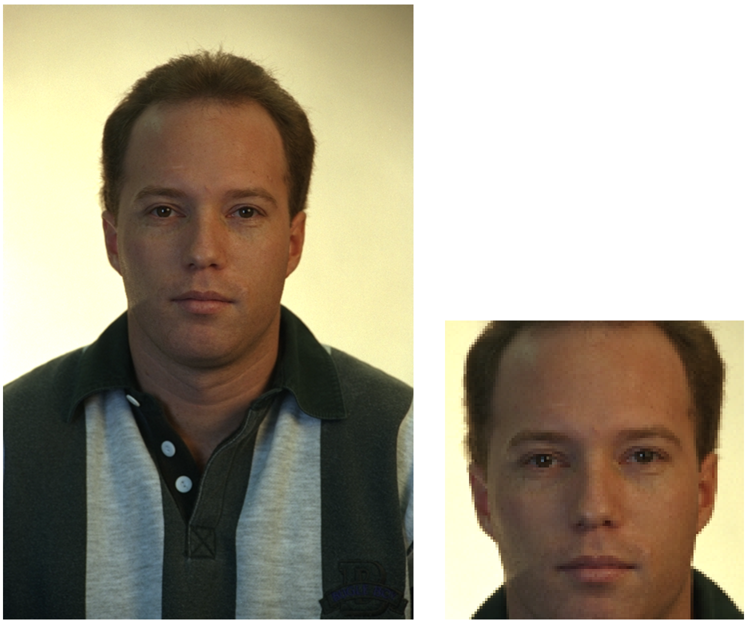


Figure 4.1: This figure shows both the original image included in ColorFERET (left) and the resulting image obtained from it after cropping and alignment (right). Note: The resolutions of these images do not match the real ones, because the image from ColorFERET has a very high resolution.

each pixel is only described by 8 bits, reducing the file size by a total of 6̃6%. Of course, this also results in a loss of information, which is why all following steps are performed on both the colored dataset as well as the grayscale dataset. The comparison of the images before and after conversion is shown in Figure 4.2.

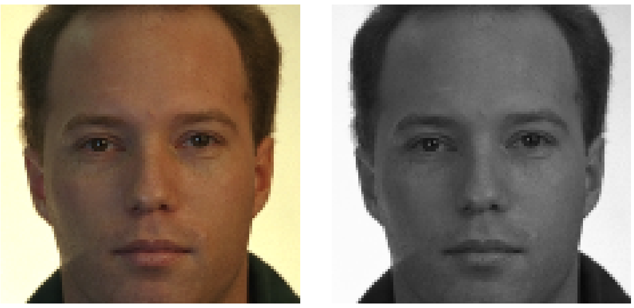


Figure 4.2: This Figure shows the impact of the grayscale conversion.

4.2 COLLECTION OF AVAILABLE PARAMETERS

Now that all datasets have been prepared in such a way that they are suitable for face recognition and have additionally been reduced by a non-negligible size, the possible parameters must be determined for each compression algorithm that could lead to an optimization of the compression and/or the subsequent face recognition. The most common methods for reducing the file size of images have already been discussed in 2.2. For the following steps, three adjustment levers are still relevant: First, the various parameters provided by the Command Line Interface (CLI) tools of each individual compression algorithm. Second, the resolution of the images that are to be compressed. While the images processed by the AdaFace face recognition algorithm must always have a resolution of 112×112 pixels, this resolution can also be achieved after compression by means of scaling. Third, there are additional image manipulation strategies that can support the performance of the compression algorithms due to their designs. All available parameters are presented individually in the following sections.

4.2.1 *Compression algorithm parameters*

For compression seven different algorithms are used in this study: JPEG, JPEG 2000, JPEG XL, JPEG AI, HEIF using HEVC codec, AVIF using AV1 codec, and WebP. The algorithms can be applied using Python libraries like Pillow or by installing the CLI tools from the official source code repositories. Since Python libraries come with fewer options, the CLI tools are chosen. As a consequence the following version are used: JPEG-Turbo v3.1.1 [lib25], Open-JPEG v2.5.4 [ucl25], libjxl v0.11.1 [lib26], JPEG-AI Reference Software [WG124], libheif v1.21.0 [str26] with H.265 v3.5 [FFm26], libavif v1.3.0 [AOM25] with AOM v3.13.1 [All26a], and libwebp v1.6.0 [Web25]. Each algorithm comes with its own CLI tool parameters, which are described in the following sections.

4.2.1.1 *JPEG*

With JPEG Turbo [lib25] comes the CLI tool cjpeg. The parameters used for evaluation are “grayscale”, which specifies to create a monochrome result, “rgb” for colored results, “optimize” for better suited Huffman tables used for Huffman coding, “arithmetic” for switching from Huffman coding to arithmetic coding, and “progressive” resulting in multiple quality passes, instead of one baseline encoding.

4.2.1.2 *JPEG 2000*

OpenJPEG [ucl25] installs opj_compress for CLI usage. This comes with several options. For the following pages seven of them are used for evaluation, being “ratio”, determining the compression ratio, “q” for quality giving the peak signal to noise ratio, “n” for the number of resolutions, which corresponds to the number of discrete wavelet transforms decompositions, “b” for block width and block height, defining the code-block sizes used, “t” for tile sizes used, “s” for subsampling, and “p” for the progression order. The rest of the offered parameters defines more metadata information, which is of no use for this study.

4.2.1.3 *JPEG XL*

Libjxl [lib26] comes with cjxl as CLI tool, which offers "quality" for quality but also "distance" for visual distance as options for the ratios of compression applied. Further options are "effort" allowing more computations while consuming more resources, "progressive" enabling more responsive decoding, "compress_boxes" and "brotli_effort" giving the opportunity to compress format information, "override_bit_depth" for changing the amount of bits used per color channel to either 8, 10, 12, or 16, and "resampling" for forcing the encoder to use downsampling.

4.2.1.4 *JPEG AI*

JPEG AI [WG124] is still in an early stage of development and does not come with a CLI tool. Instead, Python modules have to be used for compression. Unlike the other compression algorithms JPEG AI is not offering a quality parameter, instead the Python module takes a bits used per pixel value ("set_target_bpp"), which can be used to achieve a compressed file size close to 1024 bytes.

4.2.1.5 *HEIF*

HEIF is used via libheif's [str26] CLI tool heif-enc. This offers the following parameters: "no-alpha" meaning the alpha channel information is dropped, "bit-depth", allowing to change the bits used per pixel per channel, "chroma-downsampling" opening the possibility to force the mode of the chroma downsampling algorithm to either "average", "nearest-neighbor", or "sharp-yuv", and the parameter "cut-tiles", which cuts the original image into square tiles before compression, in order to compress each tile separately. Since HEIF is not a compression codec and only defines the file format of the output, the compression codec H.265 is used by the library.

4.2.1.6 *AVIF*

Similar to HEIF, libavif [AOM25] contains the source code of AVIF and comes with its own CLI tool avifenc and a separate compression codec called AV1. The CLI tool provides settings "qcolor" and "qalpha" for quality definitions, which are normally used iteratively to obtain the target file size of 1024 bytes, but also "target-size" taking the desired size directly and adjusting the quality accordingly. (Note: This setting was not used, because it turned out to be unreliable. Sometimes the images extended the target size by one or two bytes using this parameter.) Further, "yuv" lets the user decide between 444, 422, 420, and 400 as YUV-format, "depth" allows to set the bits used per pixel per channel to either 8, 10 or 12, "premultiply" for premultiplying color channels with alpha channel, tiling like in heif-enc is also provided ("tilecolslog2", "tilerowslog2"), as well as "speed" for the computational effort, "range" for YUV range can be selected between "full" and “limited”, and “progressive” supports progressive rendering for simple layered images.

4.2.1.7 *WebP*

WebP’s [Web25] CLI tool cwebp comes with the following options: “preset” provides different standard settings, regarding the content of the image, for example “icon”, “text”, or “photo”. The parameter “size” takes the target size for the compressed image, “m” lets the user chose between seven compression methods which differ in processing speed, “segments” splits the original image into areas of different properties, ranging from 1 to 4 areas, which may improve speed and quality, “psnr” sets the peak signal to noise ratio, “f” sets the filter strength, smoothing block artifacts, “sharpness” defines the level of edge preserving, “pass” defines the amount of passes performed for compression, and “sns”, which is spatial noise shaping, offers the possibility to define how much detail can be lost in order to improve compression.

4.2.2 *Resolution*

After evaluating the different parameters for each compression algorithm, the optimal parameters are selected for further optimizations. In their study, Grother et al. [GNH25] have shown that compression alone does not produce the best results, as it sometimes creates compression artifacts that ultimately distort faces. Instead, a combination of reduced resolution and image compression is better suited for reducing a facial image to small file sizes with as little loss of information as possible. The assumption here is as follows: The higher the resolution of an image, the more the compression algorithm has to compress. This inevitably leads to stronger compression artifacts. Reducing the resolution always deletes information but does not create compression artifacts. There should be an optimal ratio in which a compression algorithm causes less damage to the target image than a further reduction in resolution. The aim here is to find this ratio. Since running through all possible resolutions would be too computationally intensive, only rough steps in the reduction of resolutions are taken in the following. These include 112, which is the starting size because this resolution is required by the face recognition algorithm, as well as 56, 64, 75, 80, 96, 128, 160, 180, 200, and 224. Resolutions above 112 are used both directly from the original dataset and upscaled based on the required target size of 112, while resolutions below 112×112 pixels are created by downscaling the 112×112 images. FIQAT is always used for this adjustment. A special feature of this step is the processing by JPEG AI, as this compression algorithm can only work with images with a resolution of 160×160 pixels or higher.

4.2.3 *Further image manipulation*

As described earlier in this work, compression algorithms use different encodings to reduce file sizes. While they differ in detail, they also share some similarities, mainly because they are crafted in order to create results, which are especially suitable for human recognition. This includes that these algorithms try to preserve contrasts and use more bits to describe them than, for example, fading areas.

A good example can be visualized with a small experiment. In order to save bits, one can reduce the colors available. In a classic color image, every pixel is represented by 24 bits, allowing for a total of 16.777.216 different colors. To reduce the file size one can reduce the available colors, for example, by reducing all three color channel values per pixel to a multiple of sixteen. The effects of such an operation can be seen in Figure 4.3. The image on the left shows the original image, while the one on the right shows the change when all color channel values have been reduced to the next lower multiple of 16. Now fewer colors are used in general. As a consequence stripes start appearing since harder contrasts are introduced. In theory a better encoding for the right image should be possible, since less colors are used in total. When compressing both images with JPEG and a quality level of 70%, the file size of the left image is reduced from 1,920,015 bytes to 16,968 bytes. For the same parameters, the right image is reduced to 21,741 bytes, which is about one quarter more in file size. This means that JPEG does not benefit from using fewer colors; on the contrary, it actually has more trouble compressing the image, because the compression algorithm is optimized to preserve sharper edges more accurately.

In contrast, this experiment shows that certain operations like smoothing edges or replacing less important areas with uniform color might improve the quality of the compression results. The following section therefore presents several operations that are applied with the aim of manipulating the images in such a way that the compression algorithms achieve results that are as optimal as possible.

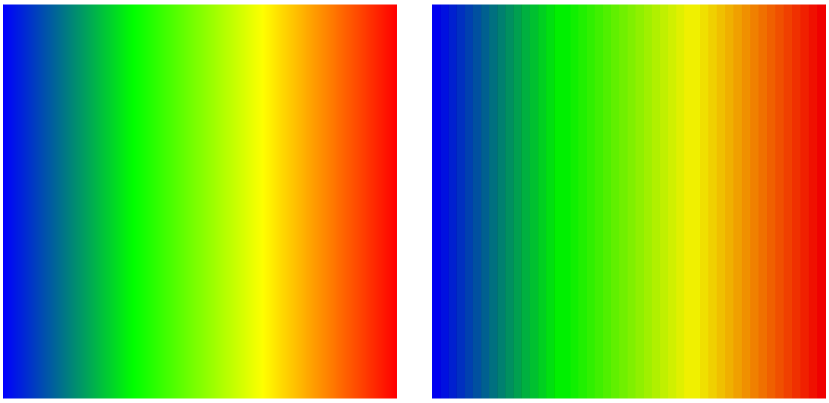


Figure 4.3: The left image shows a color fade, where all values between 0 and 255 for each color channel are used, the right image results from the left image, when setting each RGB-value to the next lower multiple of 16.

4.2.3.1 *Defining fore- and background*

Before the actual image manipulation operations used in the optimized parameter search process are presented here, two fundamental methods for determining the foreground and background of the face images must first be defined. This is important because almost all operations are intended to adjust only the background of the images, without distorting the regions of the face images that are particularly important for face recognition.
The first method of differentiation is relatively simple and is shown visually in Figure 4.4. For this purpose, five landmarks are first detected using SCRFD [Guo+21]. These include the left and right eye, the tip of the nose, and the left and right corners of the mouth. Subsequently, from the image coordinates of these five points, the highest and lowest x and y values are selected. Then, the smallest x and y coordinates are each reduced by 20% of the total resolution, while the two largest values are each increased by 20% of the total resolution. If a value falls below 0 or above the total resolution, it is capped at these values. As a result, a rectangle is spanned around the landmarks, whose interior is assumed to be the foreground and whose exterior is assumed to be the background.

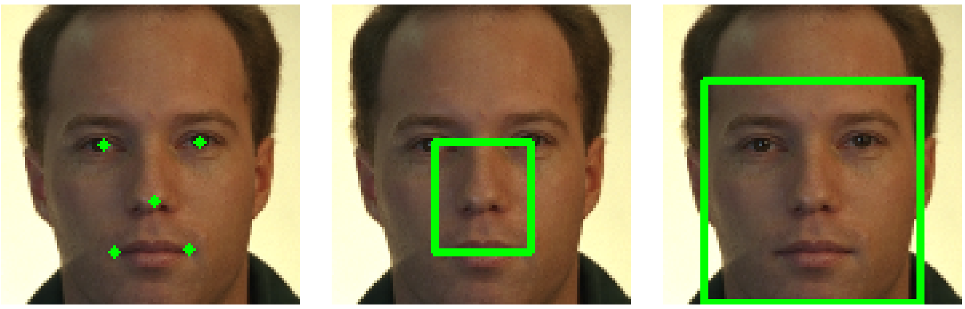


Figure 4.4: This figure shows the different processing steps for a somewhat simpler approach to dividing the facial image into foreground and background. On the left, the five landmarks are marked; in the middle, the rectangle around them can be seen; on the right, the final expanded rectangle is shown.

The second method to define foreground and background uses the OFIQ landmarks. These comprise 98 landmarks that outline the facial region around the mouth, nose, and eyes. In the subsequent process, the area inside these landmarks is used as the foreground and the area outside as the background. A depiction of this OFIQ landmark-based region can be found in Figure 4.5.

4.2.3.2 *Blacken Background*

As mentioned in 4.2.3, it could benefit compression algorithms performance replacing less important areas of the facial image with a single color. In this case, the separation of foreground and background is achieved using the rectangle described above, and the area outside it is uniformly replaced by black

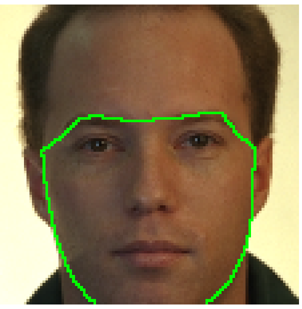


Figure 4.5: This figure shows the region that is selected as the foreground using OFIQ landmarks, outlined in green.

pixels. Although this method of image manipulation renders large parts of the image uniformly colored, it produces strongly pronounced edges along the transition from background to foreground. The result of this operation is shown in Figure 4.6.

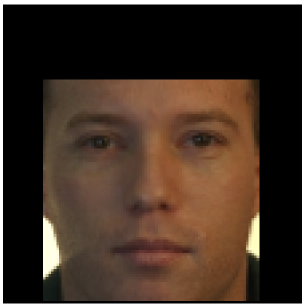


Figure 4.6: This figure shows a face image after the background was blacked out.

4.2.3.3 *Whiten Background*

Similar to 4.2.3.2 a version of the dataset is created where the background, defined by the five landmarks, is replaced with white pixels instead of black ones. An example is shown in Figure 4.7.

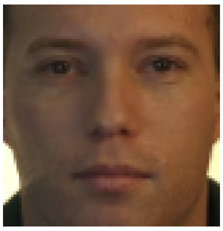


Figure 4.7: This figure shows a face image after the background was whitened out.

4.2.3.4 *Blur by overlay-rectangle*

Replacing the pixels outside of the rectangle around the facial landmarks results in a massive loss of information, and, in addition, up to four extremely sharp edges are introduced, because the transition from background to foreground is now an abrupt change in texture. While the information lost might not be very important for face recognition algorithms, it can still decrease the similarity. Moreover, face recognition based on the images becomes much harder for humans in the loop which in some cases must also have a look at them, since information about hairstyle and other information are completely deleted in the process. Another approach for optimizing compression is to blur the less important areas of the image. This removes edges, which normally require more storage space, but at least some of the background information is retained. For this reason, in this manipulation step, the background (i.e., the areas outside the rectangle) is only smoothed with a mean kernel. The result is shown in Figure 4.8

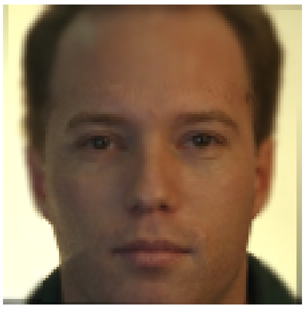


Figure 4.8: This figure shows a face image after the background, defined by the five landmarks, was blurred.

4.2.3.5 *Blur full image*

Another idea is to smooth the entire image before compression. To do this, a mean kernel is applied to the entire image. Of course, this also affects details in the face, but high compression already causes the resulting images to appear washed out, which is why this preprocessing could still have a positive effect on the overall result. For an example of this image manipulation method, see Figure 4.9.

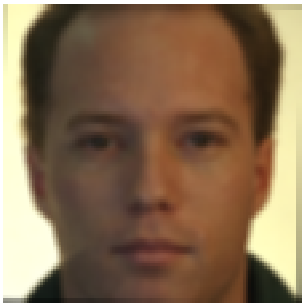


Figure 4.9: This figure shows a fully blurred face image.

4.2.3.6 *Blur by OFIQ landmark-based region*

The idea behind this image processing is the same as in section 4.2.3.4. However, now the OFIQ landmark-based region is used for distinction between foreground and background. The result of this operation is shown in Figure 4.10.

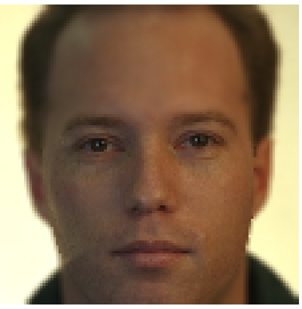


Figure 4.10: This figure shows a face image after the background, defined by OFIQ landmarks, was blurred using a mean kernel.

4.2.3.7 *Apply low-pass filter by overlay-rectangle*

This preprocessing is only implemented for grayscale images. The image is transformed into the frequency domain using Fourier transformation. Then the upper 20% of the frequencies are deleted. This is referred to as a low-pass filter. As a result, contrasts become lower. Similar operations are also used in JPEG. The frequency space is then converted back and the image area outside the rectangle described above is replaced by the new image. The entire process is similar to smoothing the image, but creates artifacts in monochrome image areas due to the deleted frequencies. An example of this operation is shown in Figure 4.11.

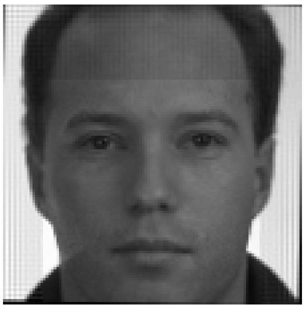


Figure 4.11: This figure shows a face image after a low-pass filter was applied to the background defined by the five landmarks. As a side effect, vertical and horizontal stripes are now visible in the image.

4.2.3.8 *Apply low-pass filter on full image*

Again, the actions performed on the defined background of the original image in section 4.2.3.7 are also performed on the full image. The result is displayed in Figure 4.12.

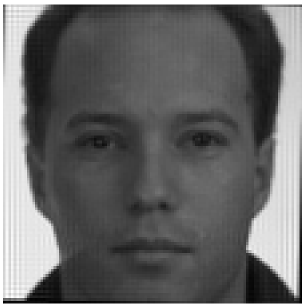


Figure 4.12: This figure shows a face image after a low-pass filter was applied to the full image. Again, the stripe-artifacts are now visible in the image.

4.3 PARAMETER OPTIMIZATION

Now the various parameters are known from which the optimal combination is to be determined. Next, it must be clarified what this determination should look like. An initial overview of this is provided by Figure 4.13.

In Figure 4.13, it can be seen that the evaluation of the different parameters and parameter combinations is carried out using the self-similarity score. The basic idea behind this approach is that the self-similarity score indicates how similar the images are before and after being modified according to the parameters. If every pixel remains the same, the self-similarity score is 1; the more dissimilar the images become, the lower the self-similarity score. The combination of parameters that achieves the highest self-similarity score should therefore result in the smallest change in face recognition performance. The process shown in Figure 4.13 describes the theoretical procedure of the parameter search in a simplified way for the sake of clarity. In reality, the selected parameter combinations are not selected randomly; instead, all parameters for the CLI tools are first tested one by one. For each parameter, it is checked whether it has a positive, negative, or no effect on the resulting self-similarity score. For parameters that require a specific set of arguments, this is examined for each argument. For example, if the required argument is a number between 0 and 100, values are tested in steps of 10. For the best range, each value is then evaluated again in steps of 1. In a second step, all parameters without a positive effect are discarded. Finally, combinations of the remaining parameters are tested. The Python library Optuna [Aki+19] is used for this purpose. It is designed for parameter optimization and offers the possibility of determining the best combination by performing a grid search on a defined set of parameters. For this case, the grid search is programmed to test the influence of each parameter on the resulting self-similarity score.

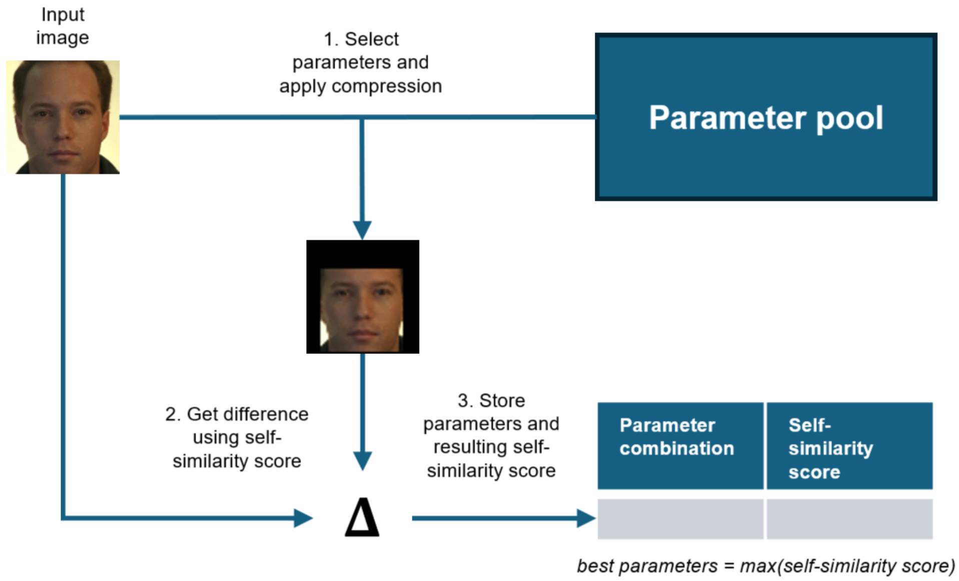

Input image
1. Select parameters and apply compression
Parameter pool
2. Get difference using self-similarity score
3. Store parameters and resulting self-similarity score
Δ
Parameter combination
Self-similarity score
best parameters = max(self-similarity score)

Figure 4.13: This figure illustrates the process of parameter optimization. For this purpose, different combinations of the available parameters are applied to the input image, and the resulting self-similarity score for each combination is stored. The combination that yields the highest self-similarity score is considered the best choice. Note: This figure is greatly simplified for the sake of clarity. The selection of parameters is more complex, and an averaged self-similarity score based on multiple images is stored.

After the CLI tool parameters have been optimized, the best combination is selected to determine the most suitable resolution. This is again done using an Optuna grid search. Afterwards, the best CLI parameters together with the best resolution are used to evaluate the various image manipulations. As a result, for each algorithm and each color space, one obtains a combination of the parameters of the CLI tool, the resolution, and the image manipulation that produces the highest self-similarity score.

This approach was chosen because trying all available parameter combinations would not have been feasible within the given time frame using the resources available.

In addition, there are a few more details about the general procedure: The effects of the parameters on the self-similarity score during optimization are not measured using a single image, but on the entire parameter search dataset. Consequently, the result is always an averaged self-similarity score, which is being used for determining the parameters benefit. The self-similarity score is calculated by first using SCRFD [Guo+21] for detecting the landmarks in both images that are compared, and yield these landmarks to AdaFace [KJL23] for extracting the feature vector for the faces contained. Afterwards the cosine similarity between the vectors of the input image and the compressed image is computed. This procedure secures, that face recognition is still possible for the compressed images. If the selected parameters include a different resolution or an image manipulation, the

resolution is changed first, then the image manipulation is applied, and only then is the image compressed. Since very few of the CLI tools for the compression algorithms used here allow specifying a target file size (and in the case of AVIF this function does not work correctly), the compression strength is usually determined iteratively using a binary search, finding the highest quality setting, that still leads to a file size of ≤ 1024 bytes. The only algorithm that did not need this procedure was WebP because it offers a working parameter that lets one choose the target file size.
All code used for the optimization is available at `https://github.com/dasec/1kB-FaceImage`.

4.4 LIMITATIONS

Nevertheless, the approach described here comes with some disadvantages. By first evaluating each CLI tool parameter individually, there is a possibility that parameters will already be discarded at this stage which, although they offer no improvement on their own, would represent an optimization in combination with other parameters.
Another risk arising from the design of the optimization process is the step-by-step handling of the different parameters regarding compression algorithm, resolution, and image manipulation. Because more and more parameters are fixed over the course of the process, and further parameters are then optimized based on those fixed ones, there is, on the one hand, the possibility that an alternative parameter combination is never tested even though it would actually deliver better results (for example, a lower initial resolution might favor different compression algorithm parameters). On the other hand, this approach ensures that any errors made at an early stage affect all subsequent results, which could, in the worst case, render the overall result invalid. To at least partially counteract this, the respective top 3 parameter combinations are re-evaluated in the final assessment. Overall, evaluating all possible combinations would of course be the safest approach, but this is not feasible due to the given constraints.
Another weak point is the processing of compressed images by the Python code. Before processing by AdaFace [KJL23], the images must be decompressed again. This is done partly automatically in the code; in other cases, the corresponding CLI tool must be used again with the decoder.
Another point worth discussing is the use of the self-similarity score as the sole metric for assessing whether a parameter has a positive effect on the compression result. This score only assesses how similar the images are before and after processing. However, it does not provide any information about whether, for example, the similarity of the faces of two different people has increased. In such a case, the overall result would be less useful because it would make the entire face recognition system less reliable, which would be the opposite of the desired result. This aspect in particular emphasizes the importance of examining the face image quality and sensitivity index more closely after optimizing the parameters.

Another point of criticism is the number of images on which the parameter search is performed. This amounts to only 10 images, which puts the results and their transferability to larger datasets into question. For this reason, the following chapter also examines how the self-similarity scores of the parameter search dataset change in comparison to the frontal dataset. Like many of the previous points of criticism, this circumstance is due to the framework conditions of this study.

5 RESULTS

This chapter is divided into three main sections. In the first section, the results of the preceding parameter optimization for the individual compression algorithms are presented, both for color images and for grayscale images. In addition, some images before and after compression are also shown and discussed as examples.

Based on the optimized parameters, all images of the frontal dataset are subsequently compressed to a target size of less than or equal to 1024 bytes. For this purpose, three parameter combinations are tested which, based on the parameter search dataset, promise the best results. Afterwards, the mated, non-mated, and self-similarity scores are computed. For the evaluation, two different scenarios are considered. The scenario that is more important for real-world applications is the comparison of the compressed facial images with the images from the frontal dataset. A system that could use compressed face images in the form of 2D barcodes would be ABC gates. These are designed in such a way that the probe images captured on site are, at least predominantly, frontal shots. A comparison of the compressed reference image with an uncompressed side profile would not be a relevant use case in such an application scenario, which is why these images are not taken into account in the first part of the evaluation. In the second part of the evaluation, the effects of comparing the compressed face images with the full dataset are investigated.

Based on the different similarity score distributions, three different metrics are further analyzed in the second section of this chapter. These include the impact on individual images, the similarities between images, and the effects on the overall system. Therefore, first, the average self-similarity score is analyzed, as this provides insight into the extent to which the results based on the parameter search dataset can be transferred to larger datasets. Second, the effects of image compression on the similarity score distributions are examined. For this purpose, the sensitivity index is used and several histograms are analyzed in more detail as examples. The last metrics evaluated are the FMR and the FNMR. These are arguably the metrics of most interest, since they show the impact of the compression on application scenarios, as these indicate how many errors the system now produces.

The third and final section then deals with FIQA. For this, the FIQA algorithms OFIQ Unified Quality Score and ViT-FIQA (C) are used to calculate the EDC curves for the results of the individual compression algorithms. Afterwards, it is examined to what extent these conventional methods for FIQA can also be applied to such heavily compressed data.

5.1 Optimized Parameters

This section lists the final parameters used for compression. These optimized parameters achieve the best results in terms of face recognition performance out of three parameter combinations per algorithm per color/grayscale tested on the frontal dataset. However, it should be mentioned at this point that these are only very minor improvements over the respective other top-3 candidates. In most cases, the number of error cases for an algorithm within the same color differs by only one or two errors. As mentioned at the beginning of this chapter, a distinction between two scenarios is conducted. These include comparison against frontal face images and against the images of the full dataset. As a result, for every algorithm up to four different optimized parameter combinations can be possible. One for color images, when compared against frontal images, one for grayscale images, when compared against frontal images, one for color images, when compared against the full dataset, and finally one for grayscale images, when compared against the full dataset. In many cases, however, these distinctions have no or only very minor effects on the resulting choice of parameters.

5.1.1 *Final parameters for color images (frontal comparison)*

Table 5.1: Final optimal parameter sets for comparing compressed color images against color frontal dataset

Compression algorithm	CLI arguments and values	Resolution	Image manipulation
JPEG	n.a.	n.a.	n.a.
JPEG 2000	ratio number of resolutions = 3	56×56	rectangular blur
JPEG XL	quality effort = 6	64×64	rectangular blur
JPEG AI	n.a.	180×180	OFIQ landmark-based blur

AVIF	YUV format = 420
	speed = 3
	depth = 12
	range = f
		56×56	rectangular blur

HEIF	chroma downsampling = average	96×96	rectangular blur

WebP	method = 6
	sns = 40
		64×64	rectangular blur

Table 5.1 shows the final parameter choice per algorithm, when compressing color images and comparing them against the frontal dataset. It becomes apparent that blurring as an image manipulation always has a positive effect.

In most cases, the simpler rectangular blur from 4.2.3.4 is the most effective. For the sake of completeness, it should be mentioned that, in the course of optimization, various blurring strategies were tested. These include mean blur, Gaussian blur, and median blur. In addition, kernel sizes of 3×3, 5×5, and 7×7 were tested. The mean blur with a kernel size of 3×3 was the only one that produced positive results and was therefore used for every final blur. A trend can also be observed with respect to resolution, showing that lowering the resolution has a positive effect. The only exception here is JPEG AI, since this compression algorithm is currently only able to process images with a resolution of 160×160 pixels or higher, so this deviation is not particularly surprising. For the compression of color images, only JPEG is not able to reliably achieve the target size of 1024 bytes. Therefore, no parameters are listed for this compression algorithm. JPEG AI does not yet offer any further optimization options via the existing CLI, which is why this cell is left empty.

5.1.2 *Final parameters for grayscale images (frontal comparison)*

Table 5.2 contains the final parameter choice per algorithm, when compressing grayscale images and comparing them against the frontal dataset (also converted to grayscale). For this type of images, JPEG can now also produce satisfying results. As already observed for the color images in this scenario, applying a blur proves to be an effective image manipulation to improve the results. However, the rectangular blur is now no longer as dominant. The resolution still remains, in all cases except JPEG AI, below the 112×112 pixels required by AdaFace [KJL23], but it does change for HEIF, JPEG AI, JPEG XL, and WebP. For AVIF and JPEG XL, the arguments for the CLI tool change.

Table 5.2: Final optimal parameter sets for comparing compressed grayscale images against grayscale frontal dataset

Compression algorithm	CLI arguments and values	Resolution	Image manipulation
JPEG	grayscale arithmetic smooth = 30	96×96	rectangular blur
JPEG 2000	ratio number of resolutions = 5	56×56	rectangular blur
JPEG XL	quality effort = 10	56×56	OFIQ landmark-based blur
JPEG AI	n.a.	200×200	full blur
AVIF	YUV format = 400 speed = 1	56×56	no manipulation
HEIF	n.a.	80×80	OFIQ landmark-based blur

WebP	method = 6
	sns = 40
		80×80	OFIQ landmark-based blur

5.1.3 *Final parameters for color images (full comparison)*

Table 5.3 shows the final parameter choice per algorithm, when compressing color images and comparing them against the full dataset. An interesting development is that, compared to the ones in Table 5.1, the choice of CLI arguments for AVIF and JPEG XL have changed in almost exactly the same way as in Table 5.2, with AVIF being the only deviation, since the YUV format 400 would result in grayscale images, which would not be a valid choice for this scenario. The selected image manipulation however, match exactly those from Table 5.1. Different resolutions were not tested.

Table 5.3: Final optimal parameter sets for comparing compressed color images against the full dataset

Compression algorithm	**CLI arguments and values**	**Resolution**	**Image manipulation**
JPEG	n.a.	n.a.	n.a.
JPEG 2000	ratio number of resolutions = 3	56×56	rectangular blur
JPEG XL	quality effort = 10	64×64	rectangular blur
JPEG AI	n.a.	180×180	OFIQ landmark-based blur

AVIF	yuv = 420
	speed = 1
		56×56	rectangular blur

HEIF	chroma downsampling = average	96×96	rectangular blur

WebP	method = 6
	sns = 40
		64×64	rectangular blur

5.1.4 *Final parameters for grayscale images (full comparison)*

Table 5.4 displays the final parameter choice per algorithm, when compressing grayscale images and comparing them against the grayscale version of full dataset. For this case, the chosen CLI arguments are exactly the same as those in Table 5.2; however, the selected image manipulations are much more similar to the results for the color images, because here only HEIF shows a change from rectangular blur to OFIQ landmark-based blur. As mentioned earlier, different resolutions have not been tested.

Table 5.4: Final optimal parameter sets for comparing compressed grayscale images against the grayscale full dataset

Compression algorithm	CLI arguments and values	Resolution	Image manipulation
JPEG	grayscale arithmetic smooth = 50	96×96	rectangular blur
JPEG 2000	ratio number of resolutions = 3	56×56	rectangular blur
JPEG XL	quality effort = 10	56×56	rectangular blur
JPEG AI	n.a.	200×200	OFIQ landmark-based blur
AVIF	YUV format = 400 speed = 1	56×56	no manipulation
HEIF	n.a.	80×80	OFIQ landmark-based blur

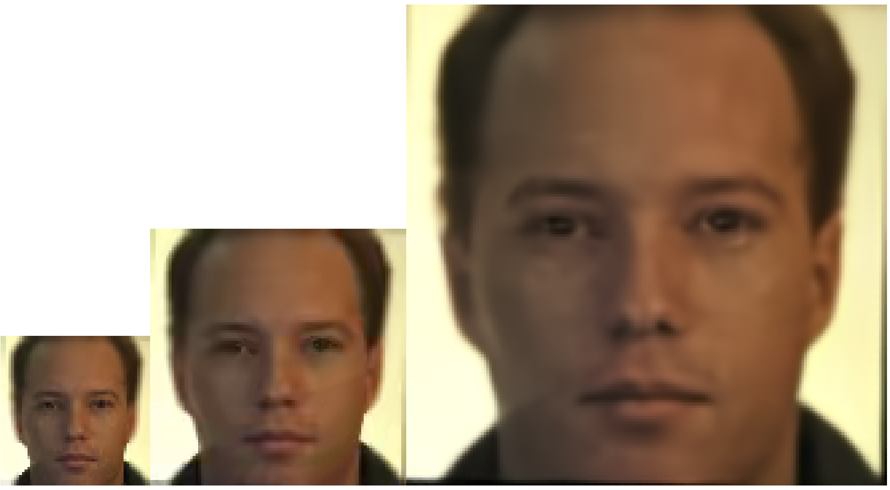
WebP	method = 6
	sns = 40
		80×80	rectangular blur

5.1.5 *Exemplary review of some compressed images*

Now that the final parameters for optimized compression for face recognition have been presented, a selected set of images is presented in order to gain a visual impression of the effects that lossy compression has on the facial images and of the differences between the individual compression algorithms.

Figure 5.1 shows the compressed color images for AVIF, HEIF, and JPEG AI using the optimized parameters for a comparison against the frontal dataset. The ratio of the resolutions to each other is preserved, so that it conveys what difference this setting has on the result of the compression, but the resolution is scaled up for printing purposes.

While this figure shows the final image that would be encoded into a 2D barcode, it must be made clear at this point that it does not correspond to the images that would be processed by the face recognition algorithm. AdaFace always requires a fixed resolution of 112×112 pixels, which is why the images must first be scaled to this uniform resolution before they can be compared. What the result of this operation looks like is shown in Figure 5.2. In addition the original image was added, so the differences caused by compression can be observed.

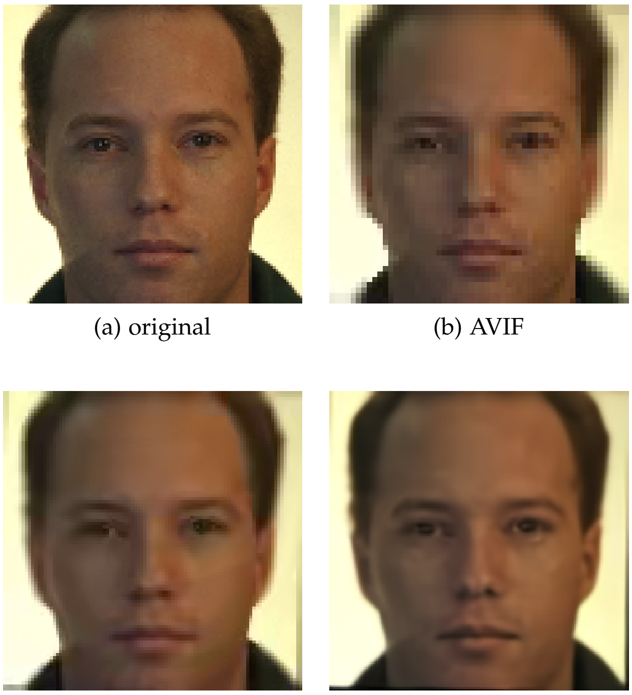


(a) AVIF

(b) HEIF

(c) JPEG AI

Figure 5.1: This figure shows the compressed results of AVIF, HEIF, and JPEG AI using color images. The images are scaled for print purposes, but the relative scale differences determined by the preprocessing are preserved.



(a) original

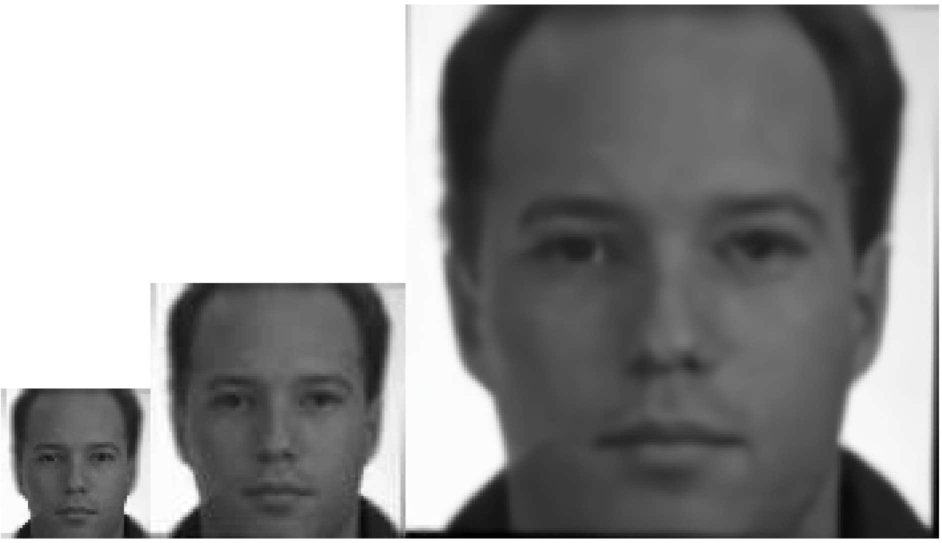
(b) AVIF

(c) HEIF

(d) JPEG AI

Figure 5.2: This figure shows the original image and the compressed results of AVIF, HEIF, and JPEG AI. The resolution is set to 112×112 pixels, which corresponds to the required resolution of the face recognition algorithm.

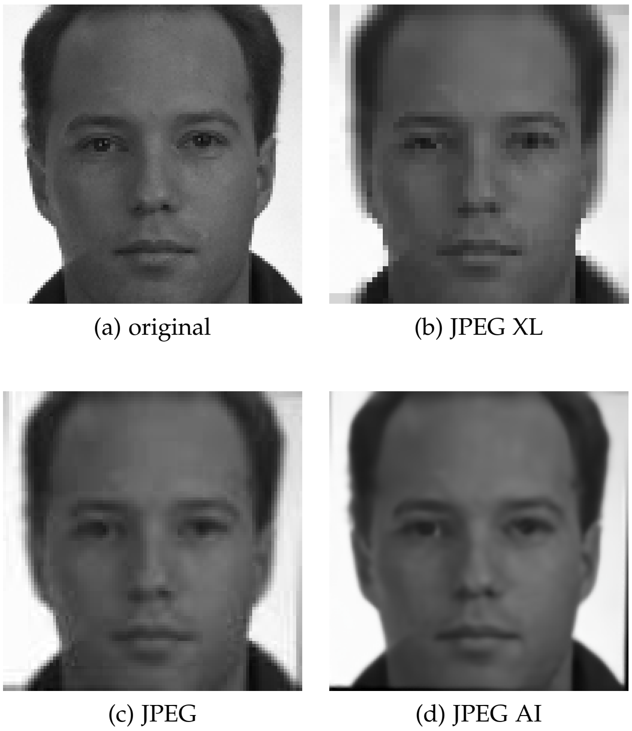
In Figure 5.2, some compression-related changes can now also be observed. As would be expected, the texture of the AVIF-compressed image is significantly coarser than the others. This is due to the low resolution. As a result, details are lost at the nose and in the eye on the right side in particular. One detail that is well preserved, however, is the individual colors of the original image. The situation is different for the HEIF image. Here, with a resolution of 96×96 pixels, it is already very close to the resolution of the original image. The textures are now heavily smoothed. This is also the case in the non-blurred area. While the nose and mouth have been preserved very well, the eyes have been damaged. The white parts have been almost completely replaced by the darker iris. In addition, from the nose across the right side, a conspicuous color transition has appeared that is not present in the original image. JPEG AI delivers the highest visual quality, which is certainly also due to the fact that the resolution has to be reduced here instead of increased. Overall, the color rendering of the JPEG AI image appears much more unnatural than in the other images. The eyes, mouth and nose have been visually well preserved; however, it seems that the proportions have been slightly altered—for example, the corners of the mouth point slightly downward, which is not the case in the original image.
In all compressed images, a loss of texture details and a smoothing of edges can be observed compared to the original image. This is, on the one hand, due to the preprocessing and, on the other hand, due to the lossy compression.



(a) JPEG XL (b) JPEG (c) JPEG AI

Figure 5.3: This figure shows the compressed results of JPEG XL, JPEG, and JPEG AI using grayscale images. The aspect ratio determined by the preprocessing is preserved.

Figure 5.3 includes the grayscale compressed images for JPEG XL, JPEG, and JPEG AI using the settings determined for the comparison against the frontal dataset. Again, the graphic preserves the aspect ratios of the compressed images to each other. The images rescaled to 112×112 used for face recognition are shown in Figure 5.4.



(a) original

(b) JPEG XL

(c) JPEG

(d) JPEG AI

Figure 5.4: This figure shows the original grayscale image and the compressed results of JPEG XL, JPEG, and JPEG AI. The resolution is set to 112×112 pixels, which corresponds to the required resolution of the face recognition algorithm.

As already seen in Figure 5.2, Figure 5.4 also shows that a lower resolution of the compressed image results in a coarser texture in the upscaled image used for comparison. All compressed grayscale images show more detail in the area of the eyes, the nose, and the mouth. For JPEG XL, the effect of the rectangle blur from the preprocessing can be seen upon closer inspection. In the blurred area, the textures are much more uniform than in the non-blurred area. In the JPEG-compressed image, the details have been well preserved. However, typical JPEG block artifacts can be observed. These are not yet so prominent that they are immediately noticeable, but clear edges can already be seen at the left eyebrow and the corners of the mouth. Unsurprisingly, JPEG AI once again offers the best visual quality. As with the corresponding color image, the texture is now much more unnatural. The fine textures of the skin and hair of the original image are, for example, no longer present in this image, and only smooth variations in brightness remain. The JPEG AI-compressed grayscale image is overall darker than the others.

As with the compressed color images, the compression here also results in a smoothing of edges and a general reduction of texture details.

5.2 METRICS EVALUATION

This section deals with the various metrics used to evaluate the compressed images. First, it is examined to what extent the individual images differ before and after compression. For this purpose, the averaged self-similarity score is used. These are obtained by extracting the biometric features from the original image and its compressed version using AdaFace and calculating the cosine similarity of the two resulting feature vectors, as described in 2.4.2.2 and 2.4.2.3. Then, a more detailed investigation is carried out into how the compression affects the relationships between the images, i.e. the mated and non-mated similarity scores. To this end, the general distributions of these scores are compared. This is done by using the sensitivity index, and by means of several selected histograms. In the final step, the impact on face recognition performance is examined. Here, the FNMRs at given FMRs are first considered, and subsequently the face recognition performance at the thresholds of the original dataset is analyzed.

5.2.1 *Self-similarity scores*

The expectation for the self-similarity scores is that they express the deviations of a compressed facial image from those of the original in a single scalar value. For this reason, this metric has already been used for parameter optimization.

Two questions now need to be answered: First, how high is the average self-similarity score? If the basic assumption is correct, they should remain close to 1, since 1 means two face images are exactly the same. Second, does the average self-similarity score determined for the parameter search dataset with the final parameters differ significantly from that of the equally compressed frontal dataset? This should not be the case, because otherwise one might suspect that optimization based on the parameter search dataset cannot be transferred to the larger frontal dataset. Tables 5.5 and 5.6 help answering these questions:

Both Table 5.5 and Table 5.6 are sorted in descending order after the first column (Color, Parameter search dataset). A comparison of the two tables already reveals a clear tendency as to which compression algorithms perform particularly well. Overall, the order changes only for JPEG XL and HEIF.

JPEG AI emerges as the top candidate for both the color images and the grayscale images. A surprising result is the performance of JPEG on the grayscale images. While this compression algorithm did not deliver sufficient results with the color images, it provides the second-best results for the grayscale images. Overall, the conversion to grayscale proves to be a highly positive measure for improving the self-similarity score. For all compression algorithms considered here, this measure results in an increase of this metric.

With regard to the two questions mentioned above, the following answers can also be noted: In general, the self-similarity scores can take on values

Table 5.5: Averaged self-similarity scores for the final settings of the comparison against the frontal dataset

Compression algorithm	Color		Grayscale	
	Parameter search dataset ↑	Frontal dataset ↑	Parameter search dataset ↑	Frontal dataset ↑
JPEG AI	0.9217	0.8938	0.9689	0.9484
AVIF	0.9075	0.8855	0.9348	0.9173
WebP	0.9040	0.8946	0.9383	0.9212
JPEG XL	0.8802	0.8582	0.9042	0.8648
HEIF	0.8789	0.8596	0.9064	0.8941
JPEG 2000	0.8502	0.8553	0.9065	0.8601
JPEG	n.a.	n.a.	0.9560	0.9199

Table 5.6: Averaged self-similarity scores for the final settings of the comparison against full dataset

Compression algorithm	Color		Grayscale	
	Parameter search dataset ↑	Full dataset ↑	Parameter search dataset ↑	Full dataset ↑
JPEG AI	0.9217	0.8938	0.9685	0.9484
AVIF	0.9076	0.8862	0.9220	0.9173
WebP	0.9040	0.8902	0.9482	0.9312
HEIF	0.8789	0.8596	0.9064	0.8941
JPEG XL	0.8735	0.8597	0.9042	0.8874
JPEG 2000	0.8502	0.8553	0.9065	0.8689
JPEG	n.a.	n.a.	0.9560	0.9190

between -1 and 1. All the average self-similarity scores determined here lie in the upper 10% of this interval. This suggests that the initial assumption about this metric is correct. Furthermore, the result tables also show that the deviations of the average self-similarity scores between the parameter search dataset and the frontal dataset are fairly small. This allows us to answer the second question in the affirmative as well. These values show that optimizations based on the parameter search dataset can be transferred to larger datasets.

When comparing the two tables, it also becomes apparent that even small changes in the parameters, such as changing the image manipulation from rectangular blur to OFIQ landmark-based blur, already result in measurable differences for face recognition, which underlines the meaningfulness of the project.

5.2.2 *Development of mated and non-mated similarity scores*

This section is about the compression impact on the mated and non-mated similarity scores. The goal for a face recognition system is always to make these two similarity score distributions as separable as possible. It now needs to be answered whether the strong compression of the images shifts or alters the individual score distributions. This is not inherently negative, as long as the distance between the distributions is not significantly reduced as a result.

5.2.2.1 *Mated score distributions*

Overall, for each compression algorithm there are four different mated score distributions. These arise from the combinations of color or grayscale images and comparison against either the frontal or the full dataset. Since the tendencies of the respective combinations are similar for all compression algorithms, the gain in insight from examining all of these distributions would not be significantly higher than the insight gained from examining only one algorithm as a representative for all of them. Therefore, in the following, only the distributions for JPEG AI will be examined as an example.

Figure 5.5 shows the development of mated similarity scores from color images when compressed with JPEG AI and compared against the frontal dataset. The graphic shows that compressing the images with JPEG AI causes a clear shift of the mated similarity scores to the left. The average mated similarity score decreases by a total of 0.0746, while the standard deviation increases slightly by 0.0008. A similar tendency can also be seen in Figure 5.6 for the comparison of the grayscale images against the frontal dataset. In contrast to the color images, however, the shift is significantly smaller here, with a decrease of the average mated similarity score by 0.0202 and even a decrease in standard deviation by 0.0008. The course of the Kernel Density Estimate (KDE) curve also remains much closer to the original than in the case of the color images.

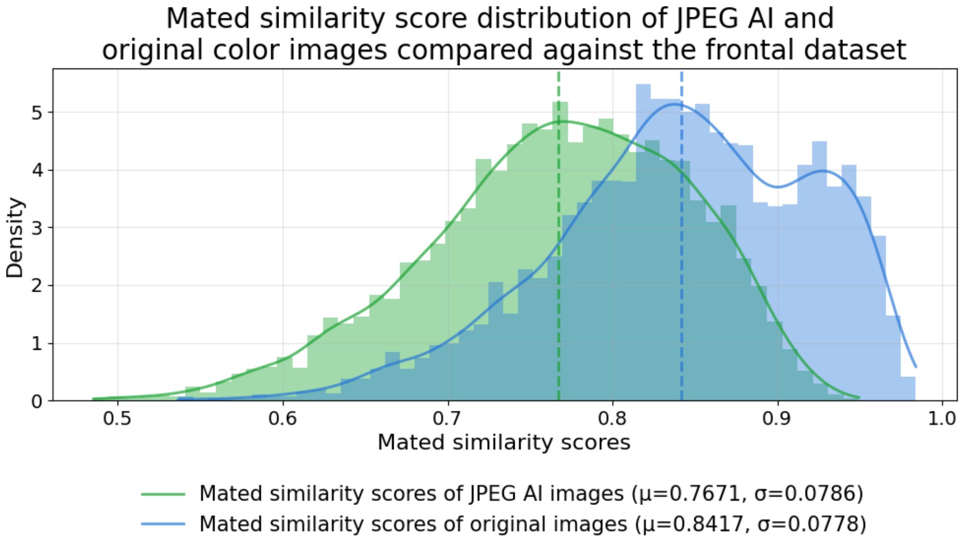


Figure 5.5: This figure shows the mated similarity score distribution produced by the comparison of JPEG AI compressed color face images against the frontal dataset (green) and the mated similarity score distribution produced by the comparison of the original color face images against the frontal dataset (blue). The distribution produced with the compressed images shows a significant shift to the left, which means that the mated similarity scores of the compressed images are overall lower. The dashed vertical lines each indicate the mean of the distribution of the same color.

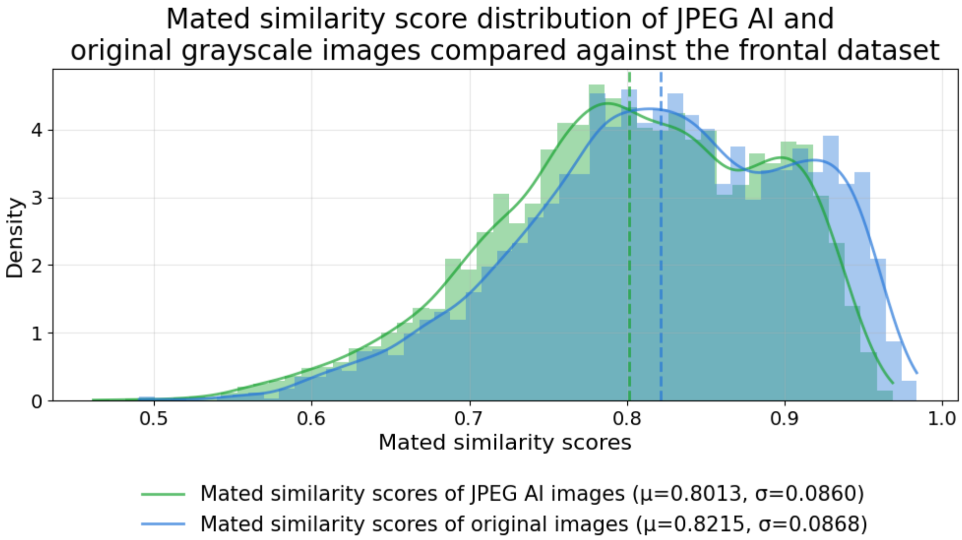


Figure 5.6: This figure shows the mated similarity score distribution produced by the comparison of JPEG AI compressed grayscale face images against the frontal dataset (green) and the mated similarity score distribution produced by the comparison of the original grayscale face images against the frontal dataset (blue). As already observed in the comparison of the color images, a shift to the left can be seen, although this is much smaller for the grayscale images. The dashed vertical lines each indicate the mean of the distribution of the same color.

The analyses of the mated similarity scores obtained from comparison with the full dataset show the same tendencies as those from the comparison with the frontal dataset. This can be seen in Figures 5.7 and 5.8. Since significantly more comparisons are carried out overall, and now the side profile face images with considerably lower face image quality are also included, the average mated similarity scores decrease noticeably and the distributions already start at much lower values. As before, the effects of compression on the grayscale images are less pronounced to the similarity score distributions than those on the color images.

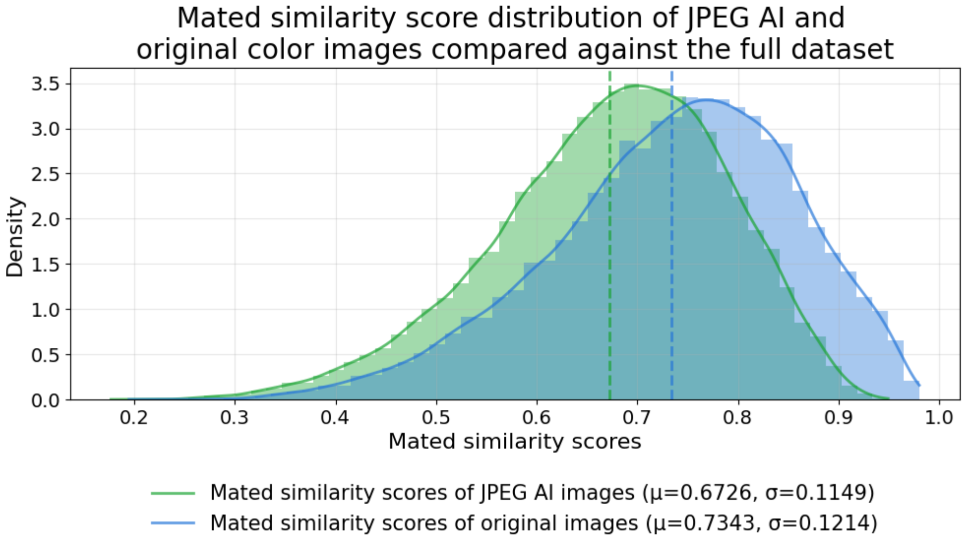


Figure 5.7: This figure shows the mated similarity score distribution produced by the comparison of JPEG AI compressed color face images against the full dataset (green) and the mated similarity score distribution produced by the comparison of the original color face images against the full dataset (blue). As already observed in the comparison with the frontal dataset, the mated similarity score distribution shifts significantly to the left. The dashed vertical lines each indicate the mean of the distribution of the same color.

In summary, regarding the effects of compression on the mated similarity scores, it can be stated that these become lower on average. This trend is more pronounced for color images than for grayscale images.

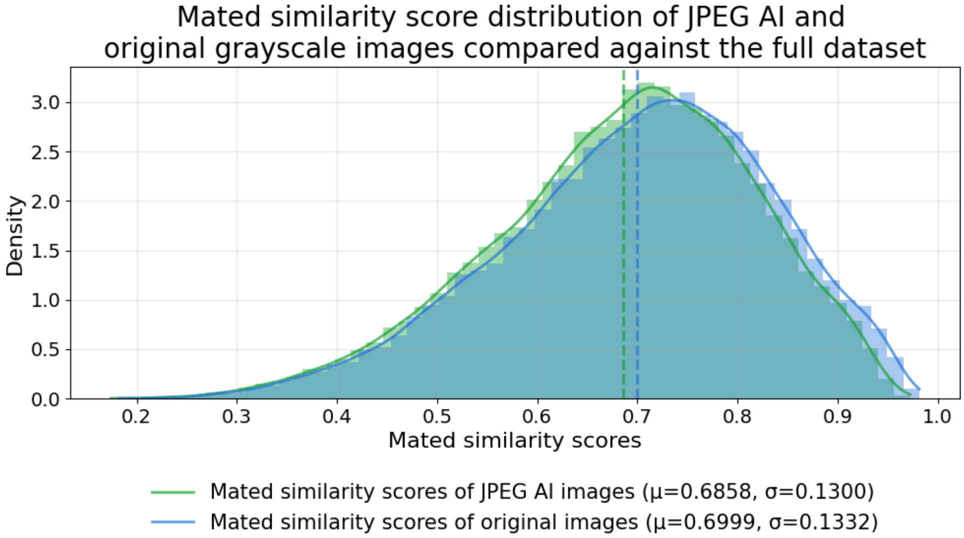


Figure 5.8: This figure shows the mated similarity score distribution produced by the comparison of JPEG AI compressed grayscale face images against the full dataset (green) and the mated similarity score distribution produced by the comparison of the original grayscale face images against the full dataset (blue). Here, too, the overall shift in values is lower. In particular, for lower values there is no significant difference between the values of the compressed and the original images. The dashed vertical lines each indicate the mean of the distribution of the same color.

5.2.2.2 *Non-mated similarity scores*

Just as in Section 5.2.2.1, in this section as well, only the evolution of the distributions is examined once, by way of example, using JPEG AI. As before, the experimental setup leads to four different non-mated similarity score distributions, which are analyzed here.

For the distributions of the non-mated similarity scores the effects observed for the distributions of the mated similarity scores are negligible. Figure 5.9 shows the resulting non-mated similarity scores from comparing JPEG AI and the original images against the frontal dataset. Once again, the scores of the compressed images shift to the left. This time, the average non-mated similarity score decreases by 0.0069, which is significantly less than the change observed for the mated similarity scores. The standard deviation also decreases for the color images, by 0.0005. This trend can again be observed for the grayscale images, as shown in Figure 5.10. Here, the shift is smaller, with the average non-mated similarity score for the compressed images decreasing by only 0.0018 and the standard deviation by 0.0002.

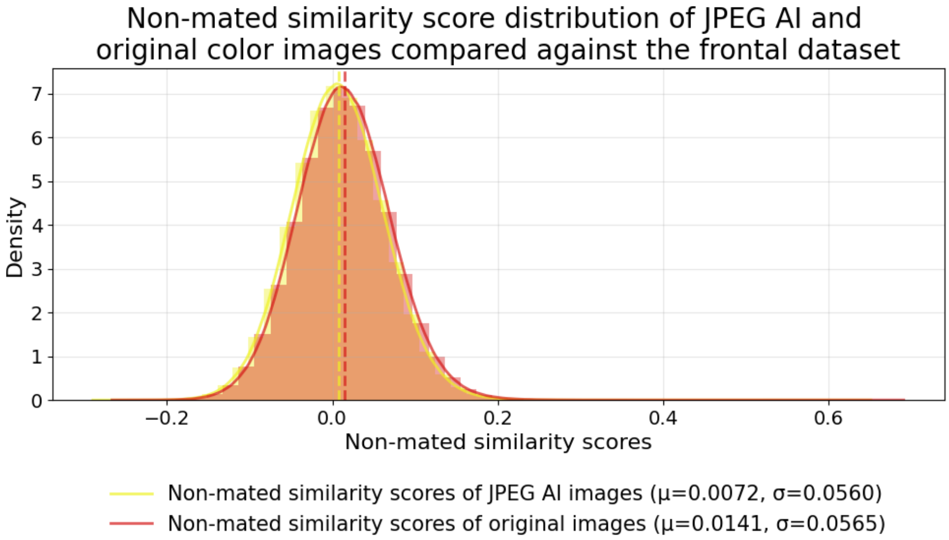


Figure 5.9: This figure shows the non-mated similarity score distribution produced by the comparison of JPEG AI compressed color face images against the frontal dataset (yellow) and the non-mated similarity score distribution produced by the comparison of the original color face images against the frontal dataset (red). The distribution produced with the compressed images shows a shift to the left, which means that the non-mated similarity scores of the compressed images, similar to the mated similarity scores, are overall lower. For this score type this is a positive development, since non-mated faces are now more dissimilar. The dashed vertical lines each indicate the mean of the distribution of the same color.

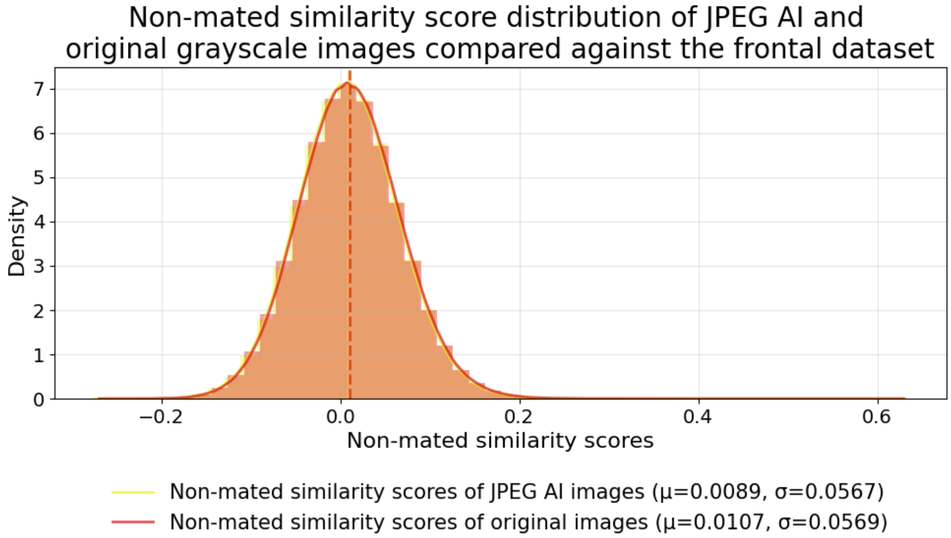


Figure 5.10: This figure shows the non-mated similarity score distribution produced by the comparison of JPEG AI compressed grayscale face images against the frontal dataset (yellow) and the non-mated similarity score distribution produced by the comparison of the original grayscale face images against the frontal dataset (red). As already observed in the comparison of the color images, a shift to the left can be seen, again this is smaller for the grayscale images. This means a less positive development compared to the color images, bit in exchange, the outliers at the top decrease in comparison, which is a more positive development. The dashed vertical lines each indicate the mean of the distribution of the same color.

The key difference from the mated similarity score distributions is that the shift of the non-mated similarity score distributions to the left has a positive effect on face recognition performance. This effect means that dissimilar faces become even more dissimilar due to the compression, which should positively influence the distinction between individuals.
Another difference is the number of available non-mated similarity scores. This is almost 1000 times larger than that of the mated similarity scores. As a result, larger outliers also occur, so that the scores, which predominantly lie between -0.2 and 0.2, can in isolated cases reach up to 0.65. This effect is even stronger for the color images.

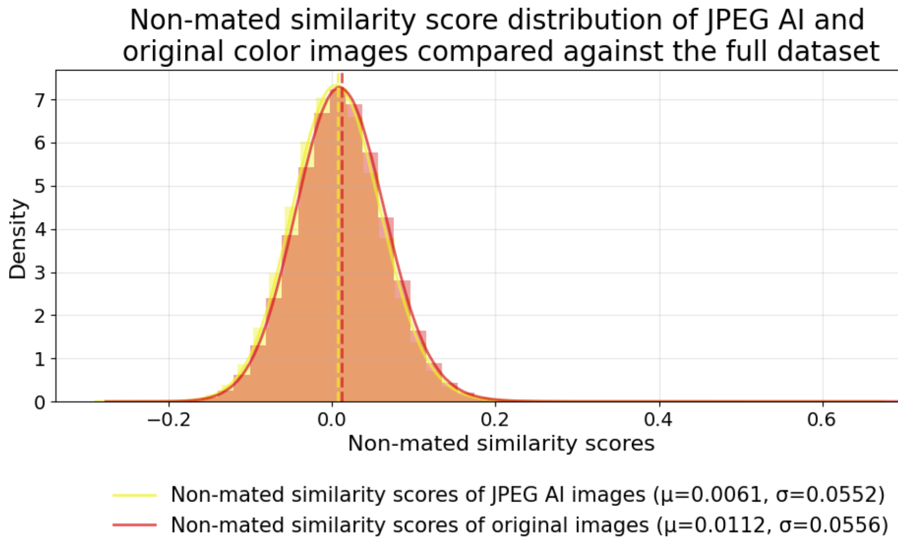


Figure 5.11: This figure shows the non-mated similarity score distribution produced by the comparison of JPEG AI compressed color face images against the full dataset (yellow) and the non-mated similarity score distribution produced by the comparison of the original color face images against the full dataset (red). As already observed for the other non-mated similarity score distributions, a slight shift to the left can be seen, which should have a positive effect on face recognition performance. The dashed vertical lines each indicate the mean of the distribution of the same color.

Figure 5.11 shows the non-mated score distributions for the color images, and Figure 5.12 shows those for the grayscale images, each resulting from comparisons against the full dataset. As in the comparison against the frontal dataset, slight leftward shifts caused by compression can be observed. These manifest themselves in a lower average non-mated similarity score compared to the original images. The score decreases by 0.0051 for the color images and by 0.0008 for the grayscale images. The standard deviation decreases by 0.0004 for the color images and remains unchanged for the grayscale images. Likewise, in the overall distribution of the non-mated similarity scores, there are no noteworthy differences between those from the comparison against the frontal dataset and those from the comparison against the full dataset.

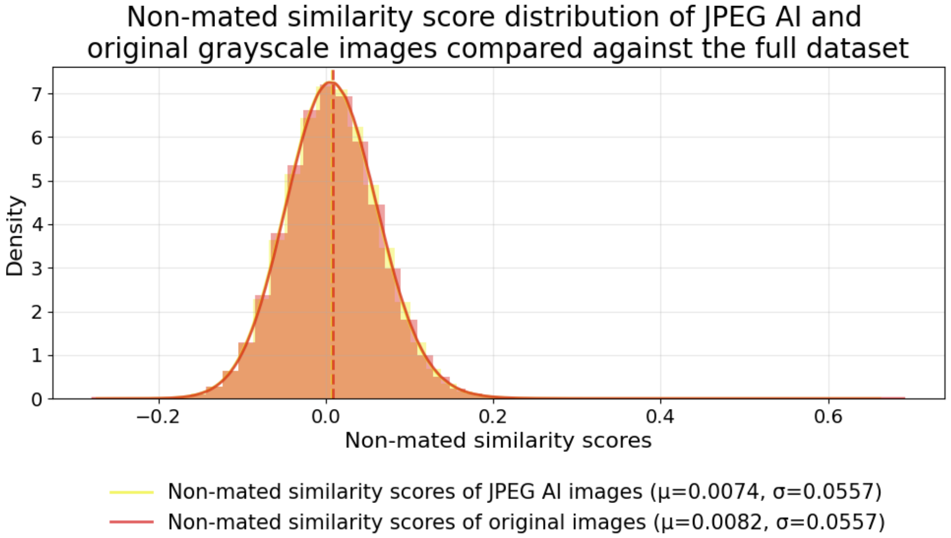


Figure 5.12: This figure shows the non-mated similarity score distribution produced by the comparison of JPEG AI compressed grayscale face images against the full dataset (yellow) and the non-mated similarity score distribution produced by the comparison of the original grayscale face images against the full dataset (red). Here too, the trend of a slight shift to the left can be seen; as with the other grayscale images, this effect is less pronounced than with the color images. The dashed vertical lines each indicate the mean of the distribution of the same color.

Finally, a slight improvement can be observed for all non-mated similarity scores. Overall, all scores are decreased slightly, which could benefit a face recognition system, as the chosen threshold for distinguishing between mated and non-mated faces can be set lower. This results from the on average lower non-mated similarity scores in combination with the hardly changed standard deviations.

5.2.2.3 *Sensitivity index*

The sensitivity index (also denoted as d') expresses how much two univariate distributions differ from each other [wik26]. To calculate it, the absolute value of the difference between the means of the distributions is divided by the mean standard deviation. The formula for the sensitivity index is:

$$d' = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}}$$

For this study, the goal is to compress the facial images in such a way that good face recognition performance can still be achieved with them. With regard to the sensitivity index, this means that it should be as high as possible, since in that case mated and non-mated similarity scores are well separable and a face recognition system would produce fewer errors.

Based on the previous analyzes of the different similarity score distributions, it must be assumed that the sensitivity indices derived from the mated and non-mated similarity score distributions of the compressed images each decrease. Although there is always a shift to the left, this shift is less pronounced for the non-mated similarity scores than for the mated similarity scores, causing the distributions to move closer together. This assumption is confirmed by Table 5.7.

Table 5.7: Sensitivity indices for the final settings

Compression algorithm	**Color**		**Grayscale**	
	Frontal dataset ↑	**Full dataset** ↑	**Frontal dataset** ↑	**Full dataset** ↑
no compression (baseline)	12.1717	7.6589	11.0518	6.7757
WebP	11.3483	7.5903	10.6089	6.7764
AVIF	11.2119	7.5987	10.5513	6.8026
JPEG AI	11.1395	7.3973	10.8834	6.7852
JPEG XL	10.8658	7.5400	10.0148	6.7492
JPEG 2000	10.8031	7.5513	9.8954	6.7032
HEIF	10.8002	7.4056	10.3680	6.7341
JPEG	n.a.	n.a.	10.4705	6.7362

Table 5.7 shows, for each compression algorithm used for examination, the resulting sensitivity indices of mated and non-mated similarity scores for color and grayscale images, in comparison with the frontal and the full dataset. The following points stand out: The sensitivity indices for the comparison with the frontal dataset are significantly higher than those for the comparison with the frontal dataset. This is not surprising, because in comparison with the complete dataset, many more images with poor face image quality are included, and therefore in particular the mated similarity scores cover a much wider range of values. As a consequence, the average mated similarity score decreases and, in turn, the standard deviation increases substantially, which in turn lowers the sensitivity index. This phenomenon can also be observed for the original data.

Another phenomenon that can be observed in all cases is the negative impact of grayscale conversion on the sensitivity index. This shows that the negative effects of grayscale conversion have a stronger adverse influence on the similarity score distributions than those of compression.

In addition, the order of the algorithms with the best results changes. However, the overall tendency that JPEG AI, AVIF, and WebP are better suited remains. Two further aspects of the specific values are particularly interesting. First, for the comparison of color images with the full dataset, JPEG AI shows the worst value of all compression algorithms, which contradicts the result of the average self-similarity scores and the sensitivity scores obtained with JPEG AI. Second, for the comparison of grayscale images with the full dataset, it can be seen that the sensitivity index for the algorithms JPEG AI, AVIF, and WebP is higher than for the comparison of the original images with full dataset.

Finally, with regard to the sensitivity indices, it must be noted that, although the degradations caused by compression and conversion to grayscale do have measurable effects, these are not dramatic—neither in relation to the results of the original images nor to the results of the individual compression algorithms compared with one another. All of them deliver satisfactory performance.

5.2.2.4 *Similarity score distribution developments*

The individual similarity score distributions have already been analyzed separately in the previous section using JPEG AI as an example. The focus was on the change caused by compression in relation to the original images (see Section 5.2.2.1 and 5.2.2.2). The sensitivity index offers a compact way to compare the developments between the algorithms, the differently colored images, and the two datasets compared against. While the sensitivity index expresses this development in a single scalar value, the actual relationships between these similarity score distributions are easier to understand when they are presented in a graphical form. This is once again examined exemplarily for JPEG AI.

Figure 5.13 includes the mated and non-mated similarity score distributions for color and grayscale images, when compared against the frontal dataset. Three aspects of this graphic are important: The non-mated similarity scores are only very slightly affected by the changes in color. This change primarily affects the distributions of mated similarity scores. The mated similarity score distribution of the grayscale images has a noticeably higher standard deviation than that of the color images, which results in a lower sensitivity index. However, it also becomes apparent that the mated similarity scores of the grayscale images are overall higher than those of the color images. In the critical region where mated and non-mated similarity scores have similar values (around 0.6), there are therefore more mated similarity scores from the color images. This, in turn, indicates that the face recognition performance of the grayscale images is better than that of the color images.

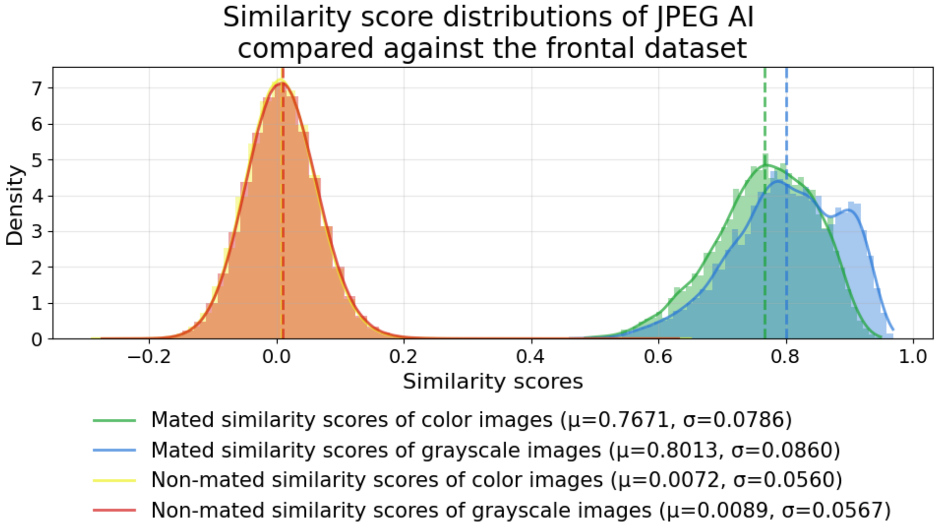


Figure 5.13: This figure shows mated and non-mated similarity score distributions for both color and grayscale JPEG AI-compressed images compared against the frontal dataset. It is noticeable that, in the comparison based on the frontal dataset, the mated similarity score distribution of the grayscale images lies clearly to the left of that of the color images. Since the non-mated similarity score distributions show this effect hardly at all. Because overall more mated similarity scores of the color images lie in the upper range of the non-mated similarity scores, it can be concluded that the face recognition performance is better for the grayscale images.

Compared to the similarity score distributions obtained from a comparison against the frontal dataset, Figure 5.14 shows a slight change in the mated similarity score distributions. These are now significantly more spread out. In addition, especially in the lower range of the mated similarity score distribution of the grayscale images, no difference to that of the color images can now be observed. As before, the non-mated similarity score distributions are hardly influenced by the choice of color space, and the mated similarity score distribution of the grayscale images is more dispersed than that of the color images. Since there is now hardly any difference between color and grayscale images in the critical region where the mated and non-mated score distributions overlap, it is no longer possible to predict which preprocessing method will yield better face recognition results.

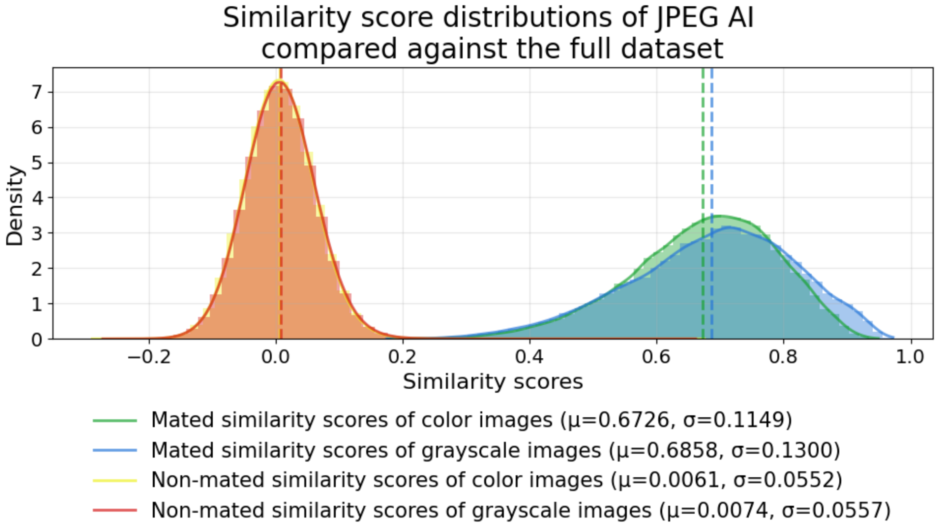


Figure 5.14: This figure shows mated and non-mated similarity score distributions for both color and grayscale JPEG AI-compressed images compared against the full dataset. While, compared to Figure 5.13, the non-mated similarity score distributions show hardly any change, it becomes apparent that the overlap with the mated similarity score distributions is now much higher. The latter are now significantly more widely dispersed.

Since the differences in the non-mated similarity score distributions are hardly noticeable even for the different comparisons against the frontal and the full dataset, the focus will only be on the mated score distributions. The respective distributions are shown in Figure 5.15. It highlights the influence the choice of images has on the results and how important it is to define the application scenario precisely. Overall, the tails of the distributions in comparison with the full dataset are significantly larger. Interestingly, in comparison the grayscale images perform slightly worse than the color images in the lower value range, even though this is not the case in the comparison with the frontal dataset.

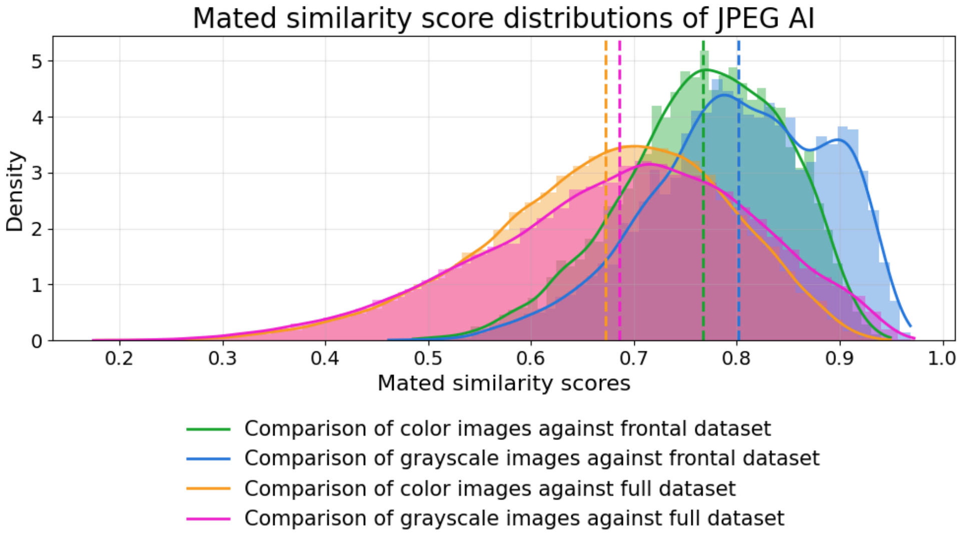


Figure 5.15: This figure shows the mated similarity score distributions obtained from the comparisons of the JPEG AI images. It becomes more apparent that the grayscale images deliver better results compared to the frontal dataset, while they produce worse results when compared to the full dataset.

5.2.3 *Impact on face recognition performance*

The last metrics that still need to be addressed are the FMR and the FNMR. These indicate how many errors a face recognition system would make when operating on the data. Here, much depends on the selected threshold. This determines what ratio of false matches and false non-matches occurs. For this purpose, FRONTEX, the European Border and Coast Guard Agency, defines the following guideline: a given system, at a FMR of 0.1%, should not exceed a FNMR of 5.0% [Eur16]. This is therefore the minimum requirement that the compressed images must meet.

In the following, for each comparison of color and grayscale images against the frontal dataset and the full dataset, the resulting FNMR values at a fixed FMR are examined.

The beginning is made by Table 5.8. It contains the FNMR of the color images that occurs at a FMR of 0.0001%. This threshold was chosen because at 0.1%, 0.01%, and 0.001% no false non-matches occurred, which would not have allowed a meaningful comparison of the algorithms' results. This threshold is already very low, because the cutoff is chosen in a way that only 1 out of 1,000,000 non-mated comparisons leads to a false match, which exceeds the FRONTEX benchmark by a factor of 1,000.

A closer look at Table 5.8 reveals a striking development in the FNMR. For all algorithms it is lower than for the original images. This is due to the fact that the non-mated similarity scores have generally decreased and therefore no longer extend as far into the critical region of the mated similarity scores. This becomes particularly clear when comparing the respective thresholds.

Table 5.8: Face recognition performance of each compression algorithm at a false match rate of 0.0001% comparing color images against the frontal dataset

Compression algorithm	FNMR in % at FMR = 0.0001% ↓	Threshold	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	2.4397	0.6677	170	8
HEIF	0.7032	0.5237	49	8
JPEG AI	0.8754	0.5651	61	8
WebP	1.1338	0.5734	79	8
JPEG XL	1.1625	0.5426	81	8
AVIF	1.9518	0.5841	136	8
JPEG 2000	2.0235	0.5611	141	8
JPEG	n.a.	n.a.	n.a.	n.a.

For the original images, the threshold of 0.6677 is significantly higher than those of the compressed images. A second aspect is which compression algorithms deliver the best results. In the previous metrics, HEIF was consistently in the lower range by comparison. AVIF, on the other hand, had usually delivered top tier results up to this point, but now is in the lower part of the table.

In Table 5.9, the FNMR values for the grayscale images are shown in comparison to the frontal dataset. As was already the case for the color images, the FMR must also be set to 0.0001% here in order to obtain any FNMR at all.

For the grayscale images, a development can now be observed that was already indicated in Section 5.2.2.4. Compared to the frontal dataset, grayscale images deliver better face recognition performance. This is due to the higher-lying mated similarity score distribution. Because the grayscale conversion also affects the non-mated similarity scores, the threshold for the original grayscale images is now clearly lower than that of the original color images, which has a positive effect on face recognition performance.

Table 5.9 shows that again some compressed images now deliver better results than the original images. In this case, these are JPEG AI, AVIF, JPEG, and WebP. While HEIF delivered the best result for the color images, the compression algorithm now falls to last place. The results of the various compression algorithms relative to each other are now once again consistent with the previous observations from the preceding sections.

After the face recognition performances for the comparison with the frontal dataset were evaluated, the results for the comparison with the full dataset are still missing.

Table 5.10 shows the data obtained for the color images at a FMR of 0.01%. This FMR is chosen because, at a FMR of 0.1%, the results for the individual algorithms in terms of FNMR do not differ sufficiently to allow for a mean-

Table 5.9: Face recognition performance of each compression algorithm at a false match rate of 0.0001% comparing grayscale images against the frontal dataset

Compression algorithm	FNMR in % at FMR = 0.0001% ↓	Threshold	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	0.4592	0.5864	32	8
JPEG AI	0.0718	0.5173	5	8
AVIF	0.2870	0.5238	20	8
JPEG	0.3157	0.5166	22	8
WebP	0.4162	0.5329	29	8
JPEG 2000	0.7319	0.4959	51	8
JPEG XL	0.8037	0.5041	56	8
HEIF	0.8754	0.5356	61	8

ingful assessment. Since a significantly higher FMR is now defined as the criterion, the threshold decreases significantly. As a result, the FNMR also drops considerably. Compared to the face recognition performances discussed so far, it is noticeable that, for this scenario, the original images yield the best result. Furthermore, the previously observed rankings of the compression algorithms are confirmed once again. The order (with the exception of JPEG) does not change compared to Table 5.9. It is also evident once again that, although there are differences in the respective suitability of the compression algorithms for this application, these differences are not particularly large. All results meet the requirements of FRONTEX.

Table 5.10: Face recognition performance of each compression algorithm at a false match rate of 0.01% comparing color images against the full dataset

Compression algorithm	FNMR in % at FMR = 0.01% ↓	Threshold	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	0.0100	0.2566	4	2932
WebP	0.0151	0.2528	6	2932
AVIF	0.0151	0.2523	6	2932
JPEG AI	0.0151	0.2493	6	2932
JPEG XL	0.0176	0.2531	7	2932
JPEG 2000	0.0251	0.2550	10	2932
HEIF	0.0276	0.2508	11	2932
JPEG	n.a.	n.a.	n.a.	n.a.

In Table 5.11, the results of the comparison of the grayscale images with the full dataset are listed. It becomes apparent that, for a comparison against images with lower face image quality, color information becomes more important. In contrast to the comparison against the frontal dataset, the grayscale images now achieve poorer face recognition performance. At a constant FMR, the FNMR quadruples. For this comparison, WebP, AVIF, and JPEG AI now deliver the best results. With regard to the performance of the individual compression algorithms, the previously observed general trend, namely that HEIF and JPEG 2000 are less suitable than the others, persists.

Table 5.11: Face recognition performance of each compression algorithm at a false match rate of 0.01% comparing grayscale images against the full dataset

Compression algorithm	**FNMR in % at FMR = 0.01%** ↓	**Threshold**	**False non-matches in total** ↓	**False matches in total** ↓
no compression (baseline)	0.0502	0.2551	20	2931
WebP	0.0602	0.2550	24	2931
JPEG AI	0.0753	0.2558	30	2931
AVIF	0.0803	0.2550	32	2931
JPEG	0.0853	0.2541	34	2931
JPEG XL	0.0979	0.2541	39	2931
HEIF	0.1104	0.2557	44	2931
JPEG 2000	0.1179	0.2555	47	2931

The previous tables all use dynamic thresholding. This is necessary because the FMR is defined as a fixed quantity. Another approach to examine the influence of strong compression on face recognition is to choose a fixed threshold. For this purpose, a single threshold is defined for all algorithms. For this evaluation, the threshold of the original images at a FMR of 0.01% is selected. The resulting figures are shown in Table 5.12.

Two things become apparent from the table: First, the impact on the similarity score distributions now becomes clear. Due to the general shift of these distributions to the left, using the compressed images without an adjusted threshold now results in significantly more errors than for the original data. Second, it also becomes evident that this type of data representation is much more difficult to interpret. The individual face recognition performances of the compression algorithms can no longer be easily compared with each other, because the ratio of FMR to FNMR differs between them. For example, JPEG-compressed images produce a FNMR that is in the range to that of the AVIF-compressed images. However, JPEG produces a FMR four times higher compared to AVIF.

Depending on the application scenario, a face recognition system may require either a particularly low FMR, a particularly low FNMR, or the most

Table 5.12: Face recognition performance of each compression algorithm at a threshold of 0.5864 comparing grayscale images against the frontal dataset

Compression algorithm	FNMR in % at threshold = 0.5864 ↓	FMR in % at threshold = 0.5864 ↓	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	0.4592	0.000115	32	8
JPEG AI	1.1194	0.000072	78	5
WebP	2.0379	0.000014	142	1
AVIF	2.1814	0.000014	152	1
JPEG	2.4684	0.000058	172	4
HEIF	3.3152	0.000043	231	3
JPEG XL	5.6401	0.000014	393	1
JPEG 2000	5.9127	0.000029	412	2

balanced possible ratio between the two error rates. The threshold should then be chosen on the basis of this requirement so that the system meets the specification. A threshold fixed in advance is therefore of little use. For this reason, the other images are not additionally evaluated according to this procedure at this point. However, the corresponding tables can be consulted in appendix C according to personal preference.

5.3 face image quality assessment

Finally, the following section examines the performance of FIQA algorithms for the compressed images. As in the previous parts of this work, not every compression algorithm is examined individually; instead, a representative example is presented. In this case, AVIF-compressed color and grayscale images were chosen for investigation (using the settings for the frontal dataset), because the results of this algorithm illustrate the general trends particularly well.

To evaluate the face image quality, the FIQA algorithms OFIQ Unified Quality Score and ViT-FIQA (C), introduced in 2.4.3, are used. The aim of this part of the study is to determine whether these two state-of-the-art FIQA algorithms are capable of providing an accurate assessment of face image quality even for heavily compressed face images.

For this purpose, the EDC curves that are generated based on the compressed images are examined. For a detailed description of the EDC evaluation methodology, see e.g. [Sch+24].

EDC curves make it possible to analyze the face images with respect to one of the two error rates (FMR or FNMR). For this study, FNMR is chosen. First, the quality score for each individual image is determined using a FIQA algorithm. Then all mated pairs are formed and their FNMR is calculated for a given similarity score threshold. Next, the images with the lowest quality scores are iteratively removed. Provided that a FIQA algorithm can handle the compressed images, the FNMR should decrease as a result, because images of higher quality lead to fewer errors. The EDC curve represents the ratio of the percentage of excluded images to the resulting FNMR. If this graph decreases, it can be concluded that the FIQA algorithm used assesses the images quality correctly.

In Figure 5.16, the EDC curves for the OFIQ Unified Quality Score and ViT-FIQA (C) using AVIF color images are shown exemplary. All results can be inspected in Appendix E. It can be seen that the OFIQ Unified Quality Score yields better results than ViT-FIQA (C) in the first half of the discarded comparisons. This suggests that the OFIQ Unified Quality Score is particularly good at correctly assessing images of low quality. From a value of 40% discarded comparisons onward, ViT-FIQA (C) is able to catch up with OFIQ Unified Quality Score and even surpass it in the second half. Here, the curve of the OFIQ Unified Quality Score initially flattens out before it starts to decline again from 70% discard. This shows that this FIQA algorithm has difficulties determining the correct quality scores in the range between 50% and 70% discard. From this point on, ViT-FIQA (C), in comparison, shows a clear drop in the graph, which indicates that the algorithm works very well in this quality range. From an exclusion rate of around 80%, the graph for ViT-FIQA (C) begins to rise, suggesting that the assessment of the top 20% quality can no longer maintain its correctness.

Overall, however, for both FIQA algorithms a decrease in the EDC curves can be observed, which means that they are indeed capable of evaluating the quality of the compressed color images.

Figure 5.17 shows the EDC curves of the two FIQA algorithms based on the AVIF-compressed grayscale images. As with the color images, both graphs also decline here, indicating that the combination of grayscale conversion and strong compression likewise does not cause a failure of the FIQA algorithms. In contrast to the color images, however, it becomes apparent that the OFIQ Unified Quality Score is now less capable overall of estimating quality than ViT-FIQA (C). On the other hand, ViT-FIQA (C) now shows improved performance in assessing images of lower quality. This trend suggests that ViT-FIQA (C) can handle grayscale images better, while the OFIQ Unified Quality Score should preferably be used with color images. Compression does not appear to pose any significant problems for either of the FIQA algorithms.

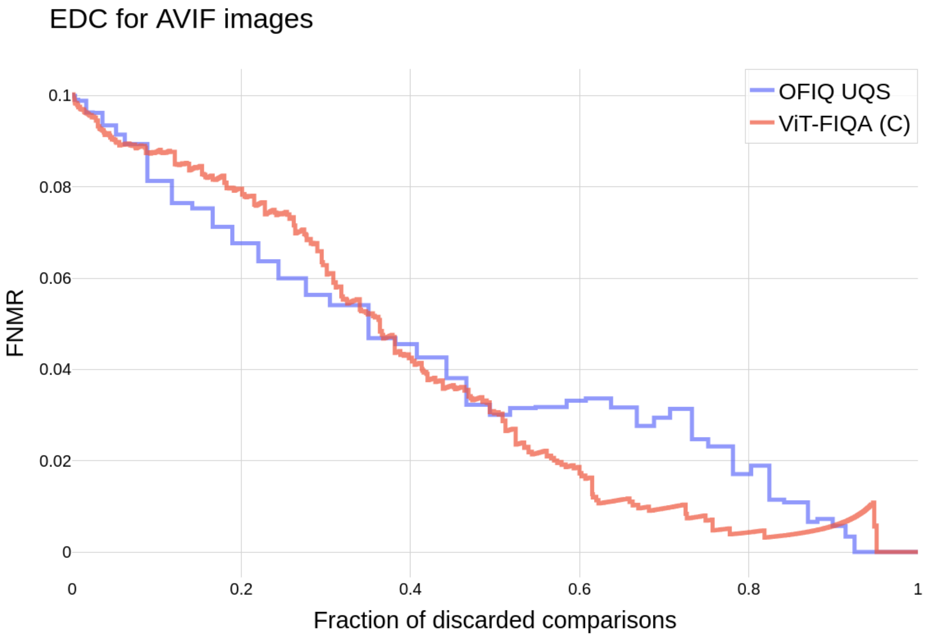


Figure 5.16: The figure shows the EDC curves for the FIQA algorithms OFIQ Unified Quality Score and ViT-FIQA (C) using AVIF-compressed color images. The similarity score threshold was selected so that the initial FNMR is 0.1. It can be seen that both algorithms decrease, which means that they are able to handle the heavily compressed images. While OFIQ Unified Quality Score delivers better results up to a discard of around 40%, ViT-FIQA (C) shows better performance above this threshold.

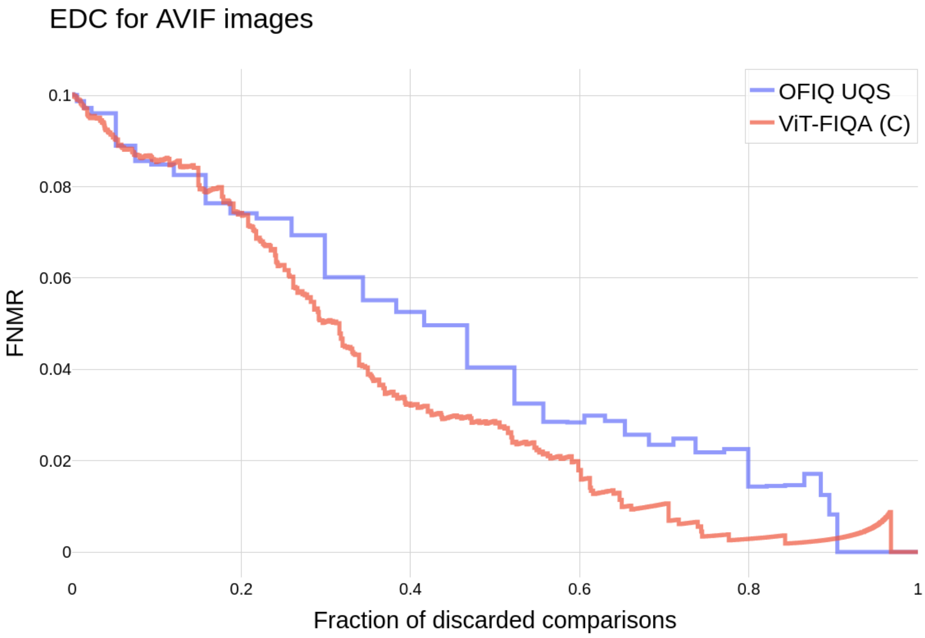


Figure 5.17: The figure shows the EDC curves for the FIQA algorithms OFIQ Unified Quality Score and ViT-FIQA (C) using AVIF-compressed grayscale images. The similarity score threshold was selected so that the initial FNMR is 0.1. It can be seen that both algorithms decrease, which means that they are able to handle the heavily compressed images. For the grayscale images, ViT-FIQA (C) shows the better performance of the two algorithms.

6 DISCUSSION

This chapter takes a more in-depth look at selected questions regarding the results from Chapter 5. Among other things, it aims to clarify which compression algorithm is ultimately best suited for compressing facial images to a file size of 1024 bytes, how the metrics used for optimization and evaluation should be viewed in retrospect, and which general trends were observed in the analysis of the data.

6.1 THE BEST COMPRESSION ALGORITHM

The results of the optimized compression of the facial images have shown that the question of which compression algorithm is best suited for this purpose cannot be answered in general terms.

In order to determine which compression algorithm should be chosen for a given application, this application must first be precisely defined so that it is possible to identify the specific conditions of the application and, in a second step, determine the optimal compression algorithm for these conditions, including optimized parameters.

As described in Chapter 1, the relevant application scenario for this thesis is the use of compressed facial images in MRTDs, which are used for identification at an ABC gate. In such an application scenario, it can be assumed that all compressed images comply with the biometric standards of ISO/IEC 19794-5 [3711] and ISO/IEC 39794-5 [3719] and ICAO [Ica]. At the same time, it can also be assumed that the probe images used for comparison, which are captured directly by the ABC gate, are frontal facial images. In addition, it is plausible to assume that multiple images may be taken, from which the one with the highest face image quality is used for the comparison, and that the individuals who wish to pass through the gate intend to provide the best possible probe image in order to keep the process time as short as possible. Furthermore, in such an application scenario, a human in the loop can be integrated easily so that uncertain cases can be filtered out and processed manually. These circumstances describe a scenario that is very similar to the comparison of the compressed images with the frontal dataset. Consequently, for recommendations in this scenario, the results of these comparisons can be used, and those of the comparison with the full dataset can be ignored.

For the application scenario ABC gate, JPEG AI achieves the best result in combination with grayscale images. The parameters used are listed in Table 6.1.

Table 6.1: Final optimized parameters for JPEG AI compression when compared against frontal dataset

Parameter	Value
Color or grayscale	Grayscale
Resolution	200×200
Image manipulation	Full blur

With a FMR of 0.0001%, JPEG AI-compressed facial images achieve a FNMR of 0.1435% compared to the frontal images. This means that if the threshold for distinguishing between match and non-match is chosen such that one person in 1,000,000 is incorrectly identified as someone else, then one person out of approximately 700 is incorrectly not identified as themselves. In the case of an ABC gate, this number is acceptable, because first, an additional probe image can be taken to still correctly identify the person, and second, in the worst case, identification can still be carried out by a human expert.

For this application scenario, in addition to JPEG AI, the algorithms AVIF, WebP, JPEG XL, and JPEG, in combination with the use of grayscale images, also produce results whose error rates are below or of the same order of magnitude as those of uncompressed images. This is consistent with the findings of Bousnina et al. [Bou+22]. Consequently, these algorithms would also not be a wrong choice.

The situation is different in an application scenario such as a supermarket, in which blacklisted individuals are to be detected through camera surveillance and automatically reported. Small images could reduce the overall costs of a corresponding system. Here, it is not always the case that the person looks directly into the camera, since cameras are usually positioned at higher locations in order to cover the largest possible area without blind spots and individuals are unlikely to cooperate. In such a scenario, it makes more sense to use the results of the comparison against the full dataset.

In this case, the JPEG AI, AVIF, and WebP algorithms all achieve the same error rate. JPEG XL can also achieve an only minimally worse rate. However, a key difference here is the choice of color images, because grayscale images produce significantly worse results in this scenario. A look at Table 6.2 also shows that the chosen parameters differ entirely from those of the other application scenario.

Table 6.2: Final parameters for JPEG AI compression when compared against the full dataset

Parameter	Value
Color or grayscale	Color
Resolution	180×180
Image manipulation	OFIQ landmark-region based blur

In summary, the question of the best algorithm can be answered as follows: The most important aspect is, first and foremost, to take the application scenario into account. In a second step, a compression algorithm and optimized parameters can be selected in compliance. For the scenarios considered here, the recommendation is JPEG AI, although the algorithms AVIF, WebP, and JPEG XL usually deliver results of comparable quality. This is also in line with the studies from 3, where deep-learning based algorithms usually perform equally good or better than traditional compression algorithms [Bou+22][JHU22]. Also a smoothing of the compressed images has in most cases proven to be an important preprocessing besides the necessary steps.
One metric that was not taken into account in this work is, for example, the runtime of the individual compression algorithms. In this respect, JPEG AI performs by far the worst. In a scenario where this factor were relevant, the recommendation would accordingly be different.

6.2 SUITABILITY OF USED METRICS

In 5.2.1, it was already noted that the self-similarity scores, as a metric for parameter optimization, appear to fulfill their intended purpose. Now that the results for the other metrics have also been presented, this can be revisited. Furthermore, this opportunity will be used to introduce additional insights regarding the metrics used.
Over the course of Chapter 5, it becomes apparent that the initially generated self-similarity scores already exhibit strong agreement with the later rankings of the respective compression algorithms. One notable observation regarding the self-similarity scores was the positive effect of grayscale conversion. In 5.2.2.4, it was also shown that grayscale conversion had a positive impact on the development of the mated similarity score distribution; for example, the average mated similarity score for the grayscale images is higher than for the color images. Another indication of the transferability of the self-similarity scores to the overall development of mated similarity scores is provided by Figure 6.1.

The figure shows that JPEG AI produces higher mated similarity scores compared to JPEG XL. Again, this graphic is only an example; however, the overall observation is that compression algorithms with higher average self-similarity scores also produce higher average mated similarity scores.
All of these aspects indicate that the developments of the self-similarity scores occur in parallel with those of the mated similarity scores. Since it was also found that the non-mated similarity scores are hardly affected by compression and preprocessing, which is consistent with Jalilian et al. [JHU22], it can be assumed that the self-similarity score fulfills its purpose.
The second metric that is discussed once again at this point is the sensitivity index. This metric also supports, in an overall view, the rankings of the individual compression algorithms relative to one another. However, what is not clearly evident from this metric is the effect of grayscale conversion. When merely looking at the corresponding table 5.7, it appears that the grayscale

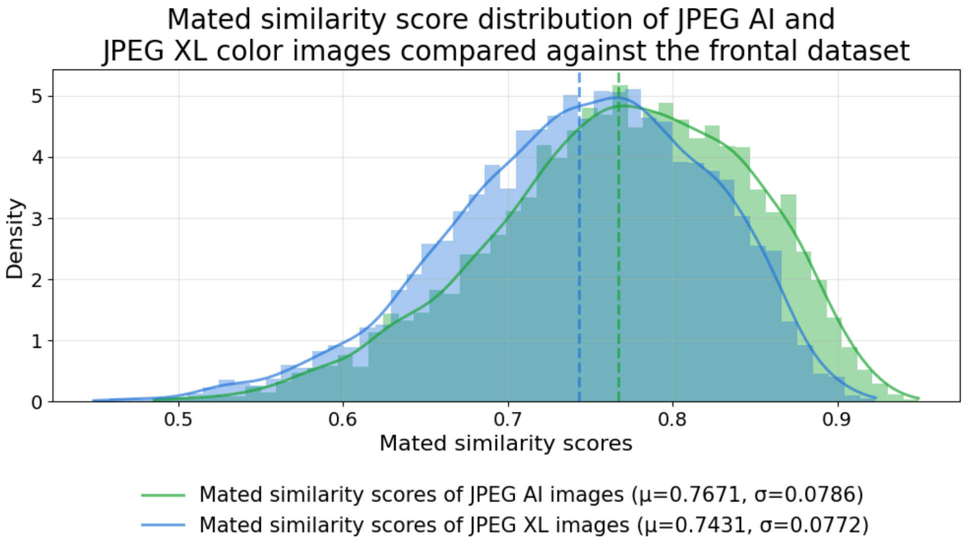


Figure 6.1: This Figure illustrates the comparison of mated similarity score distributions between JPEG AI and JPEG XL when comparing the color images against the frontal dataset. Both algorithm deliver good results, but as for the self-similarity scores, the mated similarity scores for JPEG XL are lower than the scores of JPEG AI on average.

conversion always has a negative effect on the similarity score distributions, which contradicts both the results of the self-similarity scores and the results of FMR and FNMR.
This is mainly because the distributions of the mated similarity scores for grayscale images exhibit a higher standard deviation than those for color images. The fact that the average mated similarity scores of the grayscale images are higher than those of the color images, and therefore, despite only slightly higher standard deviation, still lead to better face recognition performance, is not apparent here. For this reason, it is important to also examine the actual similarity score distributions, as is done in Chapter 5, in connection with this metric.

6.3 GENERAL TRENDS

In the course of the study, recurring patterns and trends were observed. These are explained and discussed in more detail in the following sections. These include, for example, the choice between color images and grayscale images, the positive effects of image blurring and the interplay between compression algorithms and different resolutions.

6.3.1 *Color versus grayscale images*

In the course of the results analysis in Chapter 5, it became apparent that converting the face images to grayscale can, depending on the images they are compared against, provide either an advantage or a disadvantage for the face recognition performance.

With regard to strong lossy compression, converting to grayscale offers the advantage that significantly fewer data are required per pixel, which in turn makes it possible to use more storage capacity to preserve details (lower compression rate). Naturally, this option comes with costs, which specifically affect the mated similarity score distribution. While Section 5.2.2.4 clearly showed that the lower compression rate has the effect that heavily compressed images produce, on average, higher mated similarity scores when compared with frontal images, the initial effect of grayscale conversion is a reduction of these scores, as can be seen in Figure 6.2.

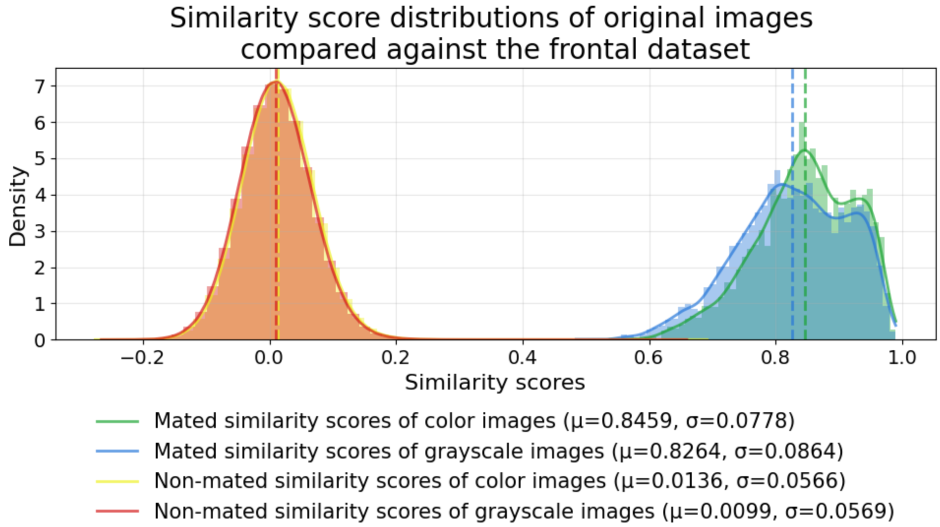


Figure 6.2: This Figure shows the mated and non-mated similarity score distributions for the original images when compared against the frontal dataset. It includes the distributions for both, color and grayscale images.

Figure 6.2 shows that an initial grayscale conversion, in addition to reducing the mated similarity scores, also results in an increased standard deviation. This effect was also observed for the compressed grayscale images.

In summary, it can be stated that the grayscale conversion of facial images can only become advantageous in the case of strong lossy compression, when the savings achieved by omitting the color information are proportionally less detrimental than the otherwise higher compression rate.

At the same time, the evaluation of the results also showed that the grayscale images only represent an improvement when compared exclusively with frontal images. In comparison with the full dataset, the color images were able to deliver better results. This is primarily due to the higher standard deviation of the mated similarity scores for the grayscale images. Since comparisons with images that tend to have lower face image quality also result in a higher standard deviation, the overall standard deviation for the grayscale images is so much higher that there is a greater overlap between the mated and non-mated similarity score distributions, which ultimately leads to a higher error rate.

The recommendation based on the findings of this work is that a conversion of color images to grayscale images should only be carried out if, on the one hand, an extremely strong reduction in file size is required and, on the other hand, high face image quality can be ensured for both the reference and probe images.

6.3.2 *Image smoothing*

A look at the optimized parameters for the individual compression algorithms quickly reveals that image manipulation in the form of a slight blur of the background is recommended in almost all cases. This finding is not particularly surprising considering the results of the experiment in 4.2.3. Blurring the background has little impact on the parts of the image that are crucial for face recognition and supports the compression algorithm, since it has to preserve fewer contrasts. On the other hand replacing the background completely with a uniform color did not favor the compression. This is in line with the observation of Maser et al. [Mas+19] that introducing hard edges prior to compression can have negative effects on compression performance.

What is much more interesting here is the choice of foreground and background. The two approaches for the separation between foreground and background are defined in 4.2.3.1. In one approach, a rectangle is placed around the most important content based on five landmarks, and in the other, a polygon is defined using 99 landmarks. The polygon adapts much more precisely to the shapes of the face. Intuitively, it therefore seems more plausible that this type of distinction between foreground and background, when applying blur, smooths a larger area and thus supports compression better. In fact, however, 5.1 shows a different picture. Across all constellations considered, the rectangle blur is clearly more often superior to the OFIQ landmark-based blur.

6.3.3 *Image resolution*

The reduction of the resolution was already presented by Grother et al. [GNH25] as an effective measure for reducing file size, which, in contrast to lossy compression, does not introduce artifacts. The results of this study also confirmed that reducing the resolution before compression has a positive effect. For almost all compression algorithms, a reduction of the resolution prior to compression is advantageous. The only exception here is JPEG AI. This algorithm occupies a special position in this context, because it requires a minimum resolution of 160×160.

The assumption regarding resolution formulated in 4.2.2 was that there must be a point at which the ratio between storage savings achieved by reducing the resolution and storage savings achieved by lossy compression is optimal, because at that point the introduced compression artifacts are less harmful than the deletion of pixels as a result of the reduced resolution. The expectation for the behavior of the self-similarity score in the course of parameter optimization of the resolution was that the value would first increase as the resolution decreases, until the damage caused by the lower resolution surpasses that caused by compression, which in turn should lead to a decrease in the self-similarity score. As a result, a downward-opening parabola was expected for the progression of the self-similarity scores.

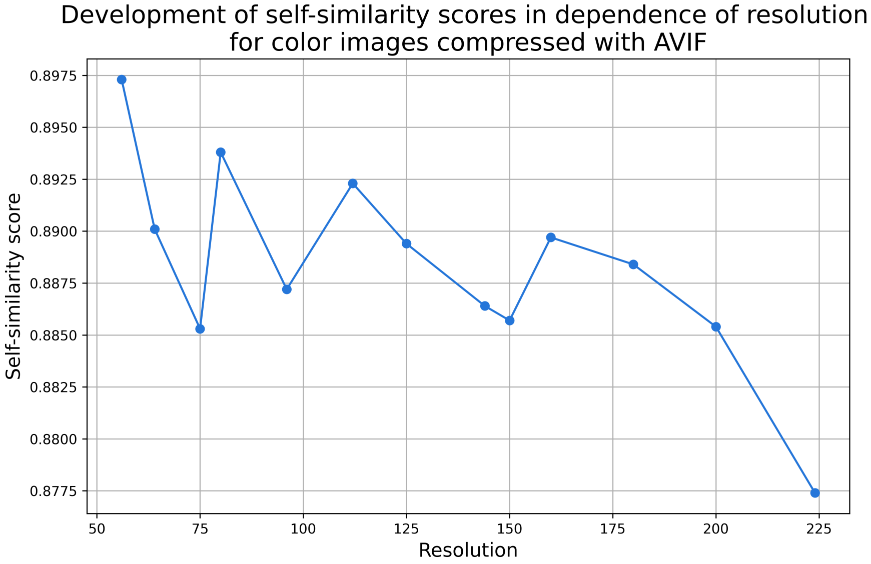


Figure 6.3: The figure shows the result of the resolution optimization for the AVIF-compressed color images.

Figure 6.3 now shows that although lower resolutions tend to lead to better self-similarity scores, they by no means produce a downward-opening parabola. The course of the curve is especially zigzagged for resolutions between 56×56 and 125×125. This behavior was observed not only for AVIF and suggests that certain resolutions are better suited than others. One resolution that, for example, turned out to be particularly unsuitable is 75×75. Others, by contrast, were sometimes better and sometimes worse depending on the chosen compression algorithm.

6.3.4 *OFIQ Unified Quality Score versus ViT-FIQA (C)*

In 5.3 it was already stated that the OFIQ Unified Quality Score yields better results for color images of low face image quality than ViT-FIQA (C), whereas the second FIQA algorithm, assesses images of higher quality better and, for grayscale images, delivers overall better performance than the OFIQ Unified Quality Score.

What has not yet been done in the evaluation of the results is a direct comparison of the EDC curves for grayscale images and color images. An example of this for AVIF-compressed images can be seen in Figure 6.4.

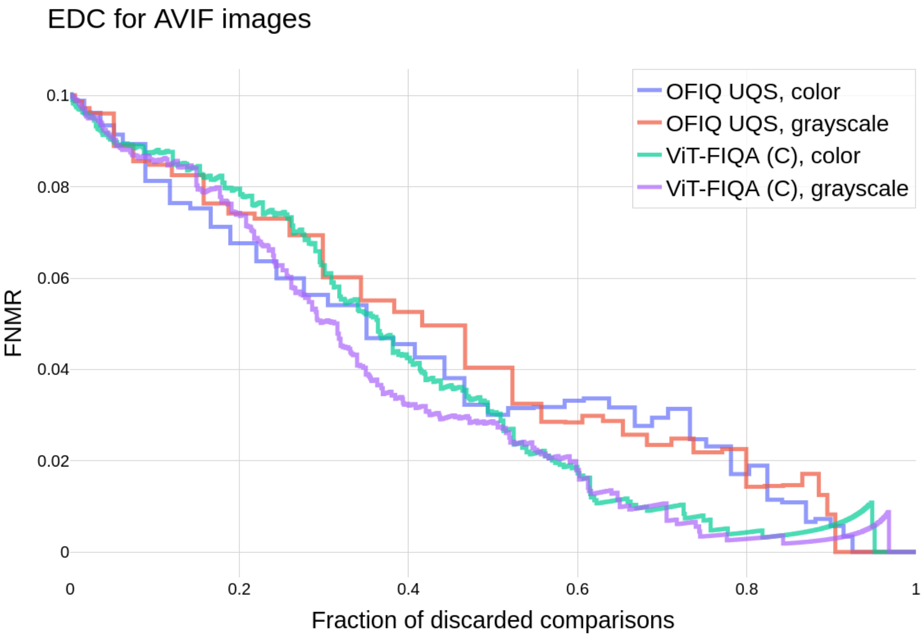


Figure 6.4: This figure shows the respective EDC curves of the AVIF-compressed color and grayscale images for the OFIQ Unified Quality Score and ViT-FIQA (C).

As already established in the previous chapter, both FIQA algorithms are fundamentally able to handle both the heavily compressed color images and the grayscale images. All EDC curves show a decline along the x-axis. However, Figure 6.4 now also clearly illustrates what impact the choice between color or grayscale images has on the FIQA algorithms.

In the comparison of the EDC curves of ViT-FIQA (C), it becomes apparent that this FIQA algorithm can generally handle the grayscale images better. Here, the corresponding graph runs almost consistently below that of the color images. This suggests that this algorithm incorporates color information only to a limited extent into its assessment of face image quality and therefore benefits from the somewhat better preserved details in the grayscale images.

The situation is exactly the opposite for the OFIQ Unified Quality Score, where the EDC curve of the color images runs almost consistently below that of the grayscale images. In this case, the preservation of color appears to be a more important criterion in the overall assessment of face image quality.

7 CONCLUSION

This chapter provides a summary of the preceding chapters. The stated objective of this thesis was to investigate the effects of strong lossy compression of facial images on face recognition performance. The concrete motivation for this was the idea of encoding facial images in 2D barcodes, which can be read reliably by machines and are more resource-efficient than conventional RFID chips, but offer significantly less storage capacity.

To achieve this goal, frontal facial images were prepared and various compression algorithms were used to reduce the images to a file size of 1024 bytes or smaller. For this process, different options such as scaling the resolution, manipulating the images, and fine-tuning the parameters of the compression algorithms were evaluated and optimized in combination. A distinction was made between two different application scenarios: In one case, only biometrically suitable images were compared with the compressed images, and in the other case, a comparison with general facial images was examined.

The results of these investigations showed that a strong reduction in file size is possible without degrading face recognition performance to the point where a corresponding system would become useless. In particular, the JPEG AI compression algorithm proved to be a suitable candidate, as it not only provides a very good compression rate but at the same time maintains a comparatively high resolution, which supports the human in the loop. However, the compression algorithms AVIF, and WebP were also able to deliver competitive results in both scenarios (see Table 7.1), which is consistent with the findings of previous studies.

Table 7.1: Face Recognition Performance the three most suitable compression algorithms in the respective scenarios.

Compression algorithm	**FNMR in % (FMR = 0.0001%, grayscale, frontal) ↓**	**Total false non-matches ↓**	**FNMR in % (FMR = 0.01%, color, full) ↓**	**Total false non-matches ↓**
no compression (baseline)	0.4592	32	0.0100	4
JPEG AI	0.0718	5	0.0151	6
AVIF	0.2870	20	0.0151	6
WebP	0.4162	29	0.0151	6

The results also showed that reducing the resolution and blurring the images before applying lossy compression has a positive impact on the quality of the outcomes, which again is in line with other literature.
A small surprise is provided by JPEG, the oldest compression algorithm considered here. After appropriate parameter optimization, it was able to deliver results for grayscale images that could keep up with the top candidates. The HEIF and JPEG 2000 algorithm, on the other hand, should not be used in contexts with strong lossy compression.
Less surprising is the finding that comparison of exclusively frontal images, even when using heavily compressed facial images, leads to significantly better face recognition performance than comparison against images without biometrically optimal conditions. It was shown that grayscale conversion is a well-suited measure to reduce file size when comparing images with higher face image quality, whereas for comparison against images of lower quality it leads to significantly worse results.
The investigation of the suitability of the FIQA algorithms OFIQ unified quality score and ViT-FIQA (C) for assessing the face image quality of strongly compressed color and grayscale images showed that both algorithms can handle the images, with ViT-FIQA (C) being the better choice for assessing grayscale images.

8 FUTURE WORK

This work has shown that strong image compression can also be applied to facial images without affecting the essential features needed for face recognition to an extent that it no longer works. Nevertheless, further optimization strategies could be investigated in the future and additional questions could be answered. A collection of possible follow-up questions is provided on the following page:

It was already pointed out in the course of this work that the search for the optimal parameters for preprocessing and compression was not carried out over all available combinations, because this was not possible due to the given time constraints. A modified approach here could lead to different results. In addition, a commonly used strategy for improving face recognition performance was not used. This consists of merging several images of the same individual. This can be employed both in the enrolment process and in the verification process in order to compensate for the weaknesses of a single facial image.

Another optimization of the previous approach would be the use of multiple face recognition algorithms. This could validate the results or, in the worst case, refute them for other algorithms, and would make a significant contribution to ensuring the reproducibility and methodological transparency of the results.

A further measure that would help to ensure the reproducibility of the results would be to transfer the findings to another dataset. In doing so, it could also be examined to what extent the operations presented in this thesis could be applied to images that are already (slightly) compressed. This is particularly interesting because many datasets are only available in already compressed formats due to storage limitations.

An investigation that would benefit the underlying application scenario would be an assessment of the quality of the compressed images by humans. Looking at the example of border control, in addition to pure functionality, the explainability of decisions is very important. For this, it would be helpful if the algorithms' results could also be confirmed by human experts.

Another experiment that could build on the results of this work would be a deeper investigation of face image quality. For example, it could be interesting to detect the compression of images algorithmically and, in the best case, even determine the compression algorithm used.

The suggestions mentioned are, of course, only a small selection of possibilities and do not claim to be exhaustive. It should now be clear, however, that even beyond the results of this thesis, many questions remain open and that further research in this area could be worthwhile.

Part II

APPENDIX

A

RESULTS OF THE PARAMETER OPTIMIZATION

Table A.1: Parameter optimization for JPEG (color)

Parameter(s)	Value(s)	Averaged self-similarity score
rgb	-	-
rgb smooth	- 0	-
rgb smooth	- 10	-
rgb smooth	- 20	-
rgb smooth	- 30	-
rgb smooth	- 40	-
rgb smooth	- 50	-
rgb smooth	- 60	-
rgb smooth	- 70	-
rgb smooth	- 80	-
rgb smooth	- 90	-
rgb smooth	- 100	-
rgb arithmetic	- -	-

rgb optimize	- -	-
rgb progressive	- -	-

Table A.2: Parameter optimization for JPEG (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
grayscale	-	0.9137
grayscale smooth	- 0	0.8931
grayscale smooth	- 10	0.9080
grayscale smooth	- 20	0.9088
grayscale smooth	- 30	0.9141
grayscale smooth	- 40	0.9125
grayscale smooth	- 50	0.9183
grayscale smooth	- 60	0.9139
grayscale smooth	- 70	0.9075
grayscale smooth	- 80	0.9031
grayscale smooth	- 90	0.9047
grayscale smooth	- 100	0.9062

grayscale smooth	- 45	0.9131
grayscale smooth	- 46	0.9135
grayscale smooth	- 47	0.9146
grayscale smooth	- 48	0.9155
grayscale smooth	- 49	0.9168
grayscale smooth	- 50	0.9183
grayscale smooth	- 51	0.9176
grayscale smooth	- 52	0.9129
grayscale smooth	- 53	0.9152
grayscale smooth	- 54	0.9112
grayscale smooth	- 55	0.9089
grayscale arithmetic	- -	0.9487
grayscale optimize	- -	0.9226
grayscale progressive	- -	0.9144
grayscale arithmetic smooth	- - 50	0.9479
grayscale arithmetic optimize	- - -	-

grayscale arithmetic progressive	- - -	-
grayscale optimize progressive	- - -	0.9218
grayscale optimize smooth	- - 50	0.9266
grayscale progressive smooth	- - 50	0.9183
grayscale optimize progressive smooth	- - - 50	0.9327
grayscale arithmetic smooth resolution	- - 50 56	0.9467
grayscale arithmetic smooth resolution	- - 50 64	0.9381
grayscale arithmetic smooth resolution	- - 50 75	0.8862
grayscale arithmetic smooth resolution	- - 50 80	0.9102
grayscale arithmetic smooth resolution	- - 50 96	0.9523

grayscale	-	0.9107
arithmetic	-	
smooth	50	
resolution	125	
grayscale	-	0.90287
arithmetic	-	
smooth	50	
resolution	144	
grayscale	-	0.8932
arithmetic	-	
smooth	50	
resolution	150	
grayscale	-	0.8757
arithmetic	-	
smooth	50	
resolution	160	
grayscale	-	-
arithmetic	-	
smooth	50	
resolution	180	
grayscale	-	-
arithmetic	-	
smooth	50	
resolution	200	
grayscale	-	-
arithmetic	-	
smooth	50	
resolution	224	
grayscale	-	0.9021
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	blackout	
grayscale	-	0.8837
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	whiteout	

grayscale	-	0.9414
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	full blur	
grayscale	-	0.9560
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	rectangular blur	
grayscale	-	0.9552
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	OFIQ landmark-region based blur	
grayscale	-	0.8915
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	full low-pass	
grayscale	-	0.8975
arithmetic	-	
smooth	50	
resolution	96	
image manipulation	rectangular low-pass	

Table A.3: Parameter optimization for JPEG 2000 (color)

Parameter(s)	Value(s)	Averaged self-similarity score
ratio	-	0.8031
psnr	-	0.7869
ratio	-	0.6169
number of resolutions	1	
ratio	-	0.8261
number of resolutions	2	
ratio	-	0.8382
number of resolutions	3	

ratio number of resolutions	- 4	0.7965
ratio number of resolutions	- 5	0.8294
ratio number of resolutions	- 6	0.7927
ratio number of resolutions	- 7	0.7882
ratio progression order	- LRCP	0.8031
ratio progression order	- RLCP	0.8031
ratio progression order	- PCLR	0.8031
ratio progression order	- CPRL	0.8031
ratio tiles	- 32×32	0.7883
ratio tiles	- 32×64	0.7629
ratio tiles	- 32×112	0.6635
ratio tiles	- 64×32	0.7848
ratio tiles	- 64×64	0.7621
ratio tiles	- 64×112	0.6892
ratio tiles	- 112×32	0.6486
ratio tiles	- 112×64	0.7389

ratio tiles	- 112×112	0.8031
ratio blocks	- 32×32	0.7856
ratio blocks	- 32×64	0.7761
ratio blocks	- 32×112	0.6169
ratio blocks	- 64×32	0.7163
ratio blocks	- 64×64	0.7747
ratio blocks	- 64×112	0.7539
ratio blocks	- 112×32	0.6387
ratio blocks	- 112×64	0.7856
ratio blocks	- 112×112	0.8031
ratio number of resolutions resolution	- 3 56	0.8416
ratio number of resolutions resolution	- 3 64	0.8227
ratio number of resolutions resolution	- 3 75	0.8061
ratio number of resolutions resolution	- 3 80	0.8188
ratio number of resolutions resolution	- 3 96	0.7996

ratio	-	0.8147
number of resolutions	3	
resolution	125	
ratio	-	0.8055
number of resolutions	3	
resolution	144	
ratio	-	0.8129
number of resolutions	3	
resolution	150	
ratio	-	0.8136
number of resolutions	3	
resolution	160	
ratio	-	0.7976
number of resolutions	3	
resolution	180	
ratio	-	0.8114
number of resolutions	3	
resolution	200	
ratio	-	-
number of resolutions	3	
resolution	224	
ratio	-	0.7486
number of resolutions	3	
resolution	56	
image manipulation	blackout	
ratio	-	0.7033
number of resolutions	3	
resolution	56	
image manipulation	whiteout	
ratio	-	0.7926
number of resolutions	3	
resolution	56	
image manipulation	full blur	
ratio	-	0.8502
number of resolutions	3	
resolution	56	
image manipulation	rectangular blur	

ratio number of resolutions resolution image manipulation	- 3 56 OFIQ landmark-region based blur	0.8492

Table A.4: Parameter optimization for JPEG 2000 (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
ratio	-	0.8686
psnr	-	0.8546
ratio number of resolutions	- 1	0.6241
ratio number of resolutions	- 2	0.8377
ratio number of resolutions	- 3	0.8836
ratio number of resolutions	- 4	0.8691
ratio number of resolutions	- 5	0.8722
ratio number of resolutions	- 6	0.8529
ratio number of resolutions	- 7	0.8439
ratio progression order	- LRCP	0.8686
ratio progression order	- RLCP	0.8686
ratio progression order	- PCLR	0.8686
ratio progression order	- CPRL	0.8686

ratio tiles	- 32×32	0.8261
ratio tiles	- 32×64	0.7936
ratio tiles	- 32×112	0.6972
ratio tiles	- 64×32	0.6893
ratio tiles	- 64×64	0.8221
ratio tiles	- 64×112	0.7326
ratio tiles	- 112×32	0.6572
ratio tiles	- 112×64	0.7544
ratio tiles	- 112×112	0.8686
ratio blocks	- 32×32	0.7659
ratio blocks	- 32×64	0.7491
ratio blocks	- 32×112	0.6448
ratio blocks	- 64×32	0.7281
ratio blocks	- 64×64	0.7963
ratio blocks	- 64×112	0.8084
ratio blocks	- 112×32	0.7556

ratio blocks	- 112×64	0.7989
ratio blocks	- 112×112	0.8686
ratio number of resolutions resolution	- 3 56	0.8942
ratio number of resolutions resolution	- 3 64	0.8381
ratio number of resolutions resolution	- 3 75	0.8199
ratio number of resolutions resolution	- 3 80	0.8678
ratio number of resolutions resolution	- 3 96	0.8593
ratio number of resolutions resolution	- 3 125	0.8639
ratio number of resolutions resolution	- 3 144	0.8712
ratio number of resolutions resolution	- 3 150	0.8479
ratio number of resolutions resolution	- 3 160	0.8136
ratio number of resolutions resolution	- 3 180	0.7976
ratio number of resolutions resolution	- 3 200	0.8114

ratio number of resolutions resolution	- 3 224	0.8371
ratio number of resolutions resolution image manipulation	- 3 56 blackout	0.8382
ratio number of resolutions resolution image manipulation	- 3 56 whiteout	0.8167
ratio number of resolutions resolution image manipulation	- 3 56 full blur	0.8851
ratio number of resolutions resolution image manipulation	- 3 56 rectangular blur	0.9065
ratio number of resolutions resolution image manipulation	- 3 56 OFIQ landmark-region based blur	0.9038
ratio number of resolutions resolution image manipulation	- 3 56 full low-pass	0.8487
ratio number of resolutions resolution image manipulation	- 3 56 rectangular low-pass	0.8631

Table A.5: Parameter optimization for JPEG XL (color)

Parameter(s)	Value(s)	Averaged self-similarity score
quality	-	0.8573
distance	-	0.8526

quality effort	- 1	0.8461
quality effort	- 2	0.8496
quality effort	- 3	0.8508
quality effort	- 4	0.8521
quality effort	- 5	0.8562
quality effort	- 6	0.8607
quality effort	- 7	0.8534
quality effort	- 8	0.8520
quality effort	- 9	0.8575
quality effort	- 10	0.8599
quality progressive	- -	0.8573
quality resampling	- -1	0.8366
quality resampling	- 1	0.8413
quality resampling	- 2	0.8074
quality resampling	- 4	0.8002
quality resampling	- 8	0.7839

quality compress boxes	- -	0.8573
quality override bit depth	- 8	0.8469
quality override bit depth	- 10	0.8541
quality override bit depth	- 12	0.8388
quality override bit depth	- 16	0.8573
quality brotli effort	- 0	0.8573
quality brotli effort	- 1	0.8573
quality brotli effort	- 2	0.8573
quality brotli effort	- 3	0.8573
quality brotli effort	- 4	0.8573
quality brotli effort	- 5	0.8573
quality brotli effort	- 6	0.8573
quality brotli effort	- 7	0.8573
quality brotli effort	- 8	0.8573
quality brotli effort	- 9	0.8573
quality brotli effort	- 10	0.8573

quality brotli effort	- 11	0.8573
quality effort resolution	- 6 56	0.8612
quality effort resolution	- 6 64	0.8751
quality effort resolution	- 6 75	0.8575
quality effort resolution	- 6 80	0.8659
quality effort resolution	- 6 96	0.8615
quality effort resolution	- 6 125	0.8489
quality effort resolution	- 6 144	0.8491
quality effort resolution	- 6 150	0.8233
quality effort resolution	- 6 160	0.8186
quality effort resolution	- 6 180	0.8067
quality effort resolution	- 6 200	0.8059
quality effort resolution	- 6 224	0.7992

quality	-	0.8361
effort	6	
resolution	64	
image manipulation	blackout	
quality	-	0.8021
effort	6	
resolution	64	
image manipulation	whiteout	
quality	-	0.8592
effort	6	
resolution	64	
image manipulation	full blur	
quality	-	0.8802
effort	6	
resolution	64	
image manipulation	rectangular blur	
quality	-	0.8764
effort	6	
resolution	64	
image manipulation	OFIQ landmark-region based blur	

Table A.6: Parameter optimization for JPEG XL (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
quality	-	0.8643
distance	-	0.8595
quality	-	0.8467
effort	1	
quality	-	0.8513
effort	2	
quality	-	0.8508
effort	3	
quality	-	0.8553
effort	4	

quality effort	- 5	0.8577
quality effort	- 6	0.8691
quality effort	- 7	0.8682
quality effort	- 8	0.8746
quality effort	- 9	0.8795
quality effort	- 10	0.8839
quality progressive	- -	0.8643
quality resampling	- -1	0.8593
quality resampling	- 1	0.8418
quality resampling	- 2	0.8489
quality resampling	- 4	0.8553
quality resampling	- 8	0.8461
quality compress boxes	- -	0.8643
quality override bit depth	- 8	0.8577
quality override bit depth	- 10	0.8529
quality override bit depth	- 12	0.8638

quality override bit depth	- 16	0.8643
quality brotli effort	- 0	0.8643
quality brotli effort	- 1	0.8643
quality brotli effort	- 2	0.8643
quality brotli effort	- 3	0.8643
quality brotli effort	- 4	0.8643
quality brotli effort	- 5	0.8643
quality brotli effort	- 6	0.8643
quality brotli effort	- 7	0.8643
quality brotli effort	- 8	0.8643
quality brotli effort	- 9	0.8643
quality brotli effort	- 10	0.8643
quality brotli effort	- 11	0.8643
quality effort resolution	- 10 56	0.8979
quality effort resolution	- 10 64	0.8848

quality	-	0.8692
effort	10	
resolution	75	
quality	-	0.8876
effort	10	
resolution	80	
quality	-	0.8745
effort	10	
resolution	96	
quality	-	0.8559
effort	10	
resolution	125	
quality	-	0.8631
effort	10	
resolution	144	
quality	-	0.8411
effort	10	
resolution	150	
quality	-	0.8369
effort	10	
resolution	160	
quality	-	0.8301
effort	10	
resolution	180	
quality	-	0.8079
effort	10	
resolution	200	
quality	-	0.7899
effort	10	
resolution	224	
quality	-	0.8741
effort	10	
resolution	56	
image manipulation	blackout	
quality	-	0.8537
effort	10	
resolution	56	
image manipulation	whiteout	

quality	-	0.8981
effort	10	
resolution	56	
image manipulation	full blur	
quality	-	0.9042
effort	10	
resolution	56	
image manipulation	rectangular blur	
quality	-	0.8995
effort	10	
resolution	56	
image manipulation	OFIQ landmark-region based blur	
quality	-	0.8268
effort	10	
resolution	56	
image manipulation	full low-pass	
quality	-	0.8455
effort	10	
resolution	56	
image manipulation	rectangular low-pass	

Table A.7: Parameter optimization for JPEG AI (color)

Parameter(s)	Value(s)	Averaged self-similarity score
resolution (rescaled to 160)	56	0.8951
resolution (rescaled to 160)	64	0.9013
resolution (rescaled to 160)	75	0.9002
resolution (rescaled to 160)	80	0.9026
resolution (rescaled to 160)	96	0.9039
resolution (rescaled to 160)	125	0.9021
resolution (rescaled to 160)	144	0.9132

resolution (rescaled to 160)	150	0.9098
resolution	160	0.9167
resolution	180	0.9183
resolution	200	0.9179
resolution	224	0.9134
resolution (rescaled to 160 directly scaled)	125	0.8955
resolution (rescaled to 160 directly scaled)	144	0.9063
resolution (rescaled to 160 directly scaled)	150	0.9101
resolution (directly scaled)	160	0.9074
resolution (directly scaled)	180	0.9112
resolution (directly scaled)	200	0.9031
resolution (directly scaled)	224	0.9083
resolution image manipulation	180 blackout	0.8973
resolution image manipulation	180 whiteout	0.8842
resolution image manipulation	180 full blur	0.9209

resolution image manipulation	180 rectangular blur	0.9197
resolution image manipulation	180 OFIQ landmark-region based blur	0.9217

Table A.8: Parameter optimization for JPEG AI (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
resolution (rescaled to 160)	56	0.8931
resolution (rescaled to 160)	64	0.9178
resolution (rescaled to 160)	75	0.9055
resolution (rescaled to 160)	80	0.9246
resolution (rescaled to 160)	96	0.9289
resolution (rescaled to 160)	125	0.9299
resolution (rescaled to 160)	144	0.9442
resolution (rescaled to 160)	150	0.9383
resolution	160	0.9463
resolution	180	0.9528
resolution	200	0.9557
resolution	224	0.9498

resolution (rescaled to 160 directly scaled)	125	0.9172
resolution (rescaled to 160 directly scaled)	144	0.9367
resolution (rescaled to 160 directly scaled)	150	0.9365
resolution (directly scaled)	160	0.9421
resolution (directly scaled)	180	0.9433
resolution (directly scaled)	200	0.9485
resolution (directly scaled)	224	0.9307
resolution image manipulation	200 blackout	0.9361
resolution image manipulation	200 whiteout	0.9178
resolution image manipulation	200 full blur	0.9689
resolution image manipulation	200 rectangular blur	0.9592
resolution image manipulation	200 OFIQ landmark-region based blur	0.9615
resolution image manipulation	200 full low-pass	0.9048
resolution image manipulation	200 rectangular low-pass	0.9144

Table A.9: Parameter optimization for HEIF (color)

Parameter(s)	Value(s)	Averaged self-similarity score
bit depth	8	0.8587
bit depth	10	0.8587
bit depth	12	0.8587
no alpha	-	0.8587
chroma downsampling	average	0.8652
chroma downsampling	Nearest-neighbor	0.8539
chroma downsampling	Sharp-yuv	0.8511
cut tiles	32	0.8073
cut tiles	64	0.5866
chroma downsampling resolution	average 56	0.8422
chroma downsampling resolution	average 64	0.8487
chroma downsampling resolution	average 75	0.8389
chroma downsampling resolution	average 80	0.8601
chroma downsampling resolution	average 96	0.8634
chroma downsampling resolution	average 125	0.8215

chroma downsampling resolution	average 144	0.8216
chroma downsampling resolution	average 150	0.7952
chroma downsampling resolution	average 160	0.8034
chroma downsampling resolution	average 180	0.8025
chroma downsampling resolution	average 200	0.7887
chroma downsampling resolution	average 224	0.7851
chroma downsampling resolution image manipulation	average 96 blackout	0.8316
chroma downsampling resolution image manipulation	average 96 whiteout	0.8277
chroma downsampling resolution image manipulation	average 96 full blur	0.8549
chroma downsampling resolution image manipulation	average 96 rectangular blur	0.8789
chroma downsampling resolution image manipulation	average 96 OFIQ landmark-region based blur	0.8659

Table A.10: Parameter optimization for HEIF (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
bit depth	8	0.8831
bit depth	10	0.8831

bit depth	12	0.8831
no alpha	-	0.8831
chroma downsampling	average	0.8831
chroma downsampling	Nearest-neighbor	0.8831
chroma downsampling	Sharp-yuv	0.8831
cut tiles	32	0.8579
cut tiles	64	0.7263
resolution	56	0.8752
resolution	64	0.8894
resolution	75	0.8716
resolution	80	0.9013
resolution	96	0.8899
resolution	125	0.8614
resolution	144	0.8817
resolution	150	0.8413
resolution	160	0.8418

resolution	180	0.8229
resolution	200	0.8156
resolution	224	0.8033
resolution image manipulation	80 blackout	0.8880
resolution image manipulation	80 whiteout	0.8793
resolution image manipulation	80 full blur	0.9018
resolution image manipulation	80 rectangular blur	0.9036
resolution image manipulation	80 OFIQ landmark-region based blur	0.9064
resolution image manipulation	80 full low-pass	0.8537
resolution image manipulation	80 rectangular low-pass	0.8761

Table A.11: Parameter optimization for AVIF (color)

Parameter(s)	Value(s)	Averaged self-similarity score
speed	0	0.8736
speed	1	0.8869
speed	2	0.8714
speed	3	0.8845
speed	4	0.8812

speed	5	0.8731
speed	6	0.8730
speed	7	0.8784
speed	8	0.8621
speed	9	0.8697
speed	10	0.8605
yuv	444	0.8472
yuv	422	0.8519
yuv	420	0.8656
depth	8	0.8392
depth	10	0.8411
depth	12	0.8414
range	limited	0.8316
range	full	0.8316
progressive	-	0.8017
yuv sharp yuv	420 -	0.8656

premultiply	-	0.8174
tile rows tile cols	0 0	0.8378
tile rows tile cols	0 1	0.8238
tile rows tile cols	0 2	0.8238
tile rows tile cols	0 3	0.8238
tile rows tile cols	0 4	0.8238
tile rows tile cols	0 5	0.8238
tile rows tile cols	0 6	0.8238
tile rows tile cols	1 0	0.8238
tile rows tile cols	1 1	0.8238
tile rows tile cols	1 2	0.8238
tile rows tile cols	1 3	0.8238

tile rows	1	0.8238
tile cols	4	
tile rows	1	0.8238
tile cols	5	
tile rows	1	0.8238
tile cols	6	
tile rows	2	0.8238
tile cols	0	
tile rows	2	0.8238
tile cols	1	
tile rows	2	0.8238
tile cols	2	
tile rows	2	0.8238
tile cols	3	
tile rows	2	0.8238
tile cols	4	
tile rows	2	0.8238
tile cols	5	
tile rows	2	0.8238
tile cols	6	
tile rows	3	0.8238
tile cols	0	
tile rows	3	0.8238
tile cols	1	

tile rows tile cols	3 2	0.8238
tile rows tile cols	3 3	0.8238
tile rows tile cols	3 4	0.8238
tile rows tile cols	3 5	0.8238
tile rows tile cols	3 6	0.8238
tile rows tile cols	4 0	0.8238
tile rows tile cols	4 1	0.8238
tile rows tile cols	4 2	0.8238
tile rows tile cols	4 3	0.8238
tile rows tile cols	4 4	0.8238
tile rows tile cols	4 5	0.8238
tile rows tile cols	4 6	0.8238

tile rows	5	0.8238
tile cols	0	
tile rows	5	0.8238
tile cols	1	
tile rows	5	0.8238
tile cols	2	
tile rows	5	0.8238
tile cols	3	
tile rows	5	0.8238
tile cols	4	
tile rows	5	0.8238
tile cols	5	
tile rows	5	0.8238
tile cols	6	
tile rows	6	0.8238
tile cols	0	
tile rows	6	0.8238
tile cols	1	
tile rows	6	0.8238
tile cols	2	
tile rows	6	0.8238
tile cols	3	
tile rows	6	0.8238
tile cols	4	

tile rows tile cols	6 5	0.8238
tile rows tile cols	6 6	0.8238
speed yuv	1 420	0.8954
speed range	1 full	0.8870
speed depth	1 12	0.8873
yuv range	420 full	0.8864
yuv depth	420 12	0.8701
range depth	full 12	0.8578
speed yuv range	1 420 full	0.8889
speed yuv depth	1 420 12	0.8836
speed range depth	1 full 12	0.8825

yuv range depth	420 full 12	0.8817
speed yuv range depth	1 420 full 12	0.8928
speed yuv resolution	1 420 56	0.8974
speed yuv resolution	1 420 64	0.8902
speed yuv resolution	1 420 75	0.8851
speed yuv resolution	1 420 80	0.8937
speed yuv resolution	1 420 96	0.8874
speed yuv resolution	1 420 125	0.8891
speed yuv resolution	1 420 144	0.8868
speed yuv resolution	1 420 150	0.8859
speed yuv resolution	1 420 160	0.8896
speed yuv resolution	1 420 180	0.8883

speed yuv resolution	1 420 200	0.8855
speed yuv resolution	1 420 224	0.8774
speed yuv resolution image manipulation	1 420 56 blackout	0.8828
speed yuv resolution image manipulation	1 420 56 whiteout	0.8751
speed yuv resolution image manipulation	1 420 56 full blur	0.8962
speed yuv resolution image manipulation	1 420 56 rectangular blur	0.9076
speed yuv resolution image manipulation	1 420 56 OFIQ landmark-region based blur	0.8975

Table A.12: Parameter optimization for AVIF (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
speed	0	0.9214
speed	1	0.9295
speed	2	0.9252
speed	3	0.9263

speed	4	0.9228
speed	5	0.9198
speed	6	0.9207
speed	7	0.9184
speed	8	0.9166
speed	9	0.9153
speed	10	0.9154
yuv	444	0.9218
yuv	422	0.9179
yuv	420	0.9234
yuv	400	0.9258
depth	8	0.9152
depth	10	0.9175
depth	12	0.9177
range	limited	0.9188
range	full	0.9193

progressive	-	0.9041
yuv	420	0.9234
sharp yuv	-	
premultiply	-	0.9062
tile rows	0	0.9143
tile cols	0	
tile rows	0	0.8952
tile cols	1	
tile rows	0	0.8952
tile cols	2	
tile rows	0	0.8952
tile cols	3	
tile rows	0	0.8952
tile cols	4	
tile rows	0	0.8952
tile cols	5	
tile rows	0	0.8952
tile cols	6	
tile rows	1	0.8952
tile cols	0	
tile rows	1	0.8952
tile cols	1	
tile rows	1	0.8952
tile cols	2	

tile rows	1	0.8952
tile cols	3	
tile rows	1	0.8952
tile cols	4	
tile rows	1	0.8952
tile cols	5	
tile rows	1	0.8952
tile cols	6	
tile rows	2	0.8952
tile cols	0	
tile rows	2	0.8952
tile cols	1	
tile rows	2	0.8952
tile cols	2	
tile rows	2	0.8952
tile cols	3	
tile rows	2	0.8952
tile cols	4	
tile rows	2	0.8952
tile cols	5	
tile rows	2	0.8952
tile cols	6	
tile rows	3	0.8952
tile cols	0	

tile rows	3	0.8952
tile cols	1	
tile rows	3	0.8952
tile cols	2	
tile rows	3	0.8952
tile cols	3	
tile rows	3	0.8952
tile cols	4	
tile rows	3	0.8952
tile cols	5	
tile rows	3	0.8952
tile cols	6	
tile rows	4	0.8952
tile cols	0	
tile rows	4	0.8952
tile cols	1	
tile rows	4	0.8952
tile cols	2	
tile rows	4	0.8952
tile cols	3	
tile rows	4	0.8952
tile cols	4	
tile rows	4	0.8952
tile cols	5	

```
    tile rows            4                  0.8952
    tile cols            6

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            0

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            1

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            2

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            3

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            4

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            5

--------------------------------------------------------------------------------

    tile rows            5                  0.8952
    tile cols            6

--------------------------------------------------------------------------------

    tile rows            6                  0.8952
    tile cols            0

--------------------------------------------------------------------------------

    tile rows            6                  0.8952
    tile cols            1

--------------------------------------------------------------------------------

    tile rows            6                  0.8952
    tile cols            2

--------------------------------------------------------------------------------

    tile rows            6                  0.8952
    tile cols            3

--------------------------------------------------------------------------------
```

tile rows tile cols	6 4	0.8952
tile rows tile cols	6 5	0.8952
tile rows tile cols	6 6	0.8952
speed yuv	1 400	0.9298
speed range	1 full	0.9173
speed depth	1 12	0.9169
yuv range	400 full	0.9201
yuv depth	400 12	0.9203
range depth	full 12	0.9148
speed yuv range	1 400 full	0.9274
speed yuv depth	1 400 12	0.9267
yuv range depth	400 full 12	0.9199

speed	1	0.9275
yuv	400	
range	full	
depth	12	
speed	1	0.9348
yuv	400	
resolution	56	
speed	1	0.9331
yuv	400	
resolution	64	
speed	1	0.9194
yuv	400	
resolution	75	
speed	1	0.9311
yuv	400	
resolution	80	
speed	1	0.9275
yuv	400	
resolution	96	
speed	1	0.9248
yuv	400	
resolution	125	
speed	1	0.9279
yuv	400	
resolution	144	
speed	1	0.9213
yuv	400	
resolution	150	
speed	1	0.9216
yuv	400	
resolution	160	
speed	1	0.9207
yuv	400	
resolution	180	
speed	1	0.9007
yuv	400	
resolution	200	

speed	1	0.9001
yuv	400	
resolution	224	
speed	1	0.9177
yuv	400	
resolution	56	
image manipulation	blackout	
speed	1	0.8954
yuv	400	
resolution	56	
image manipulation	whiteout	
speed	1	0.9162
yuv	400	
resolution	56	
image manipulation	full blur	
speed	1	0.9335
yuv	400	
resolution	56	
image manipulation	rectangular blur	
speed	1	0.9327
yuv	400	
resolution	56	
image manipulation	OFIQ landmark-region based blur	
speed	1	0.8820
yuv	400	
resolution	56	
image manipulation	full low-pass	
speed	1	0.8914
yuv	400	
resolution	56	
image manipulation	rectangular low-pass	

Table A.13: Parameter optimization for WebP (color)

Parameter(s)	Value(s)	Averaged self-similarity score
preset	default	0.8738
preset	photo	0.8736

preset	picture	0.8724
method	0	0.7185
method	1	0.7392
method	2	0.8393
method	3	0.8741
method	4	0.8828
method	5	0.8915
method	6	0.8978
segments	1	0.8724
segments	2	0.8724
segments	3	0.8724
segments	4	0.8724
psnr	10	0.8724
psnr	20	0.8724
psnr	30	0.8724
psnr	40	0.8724

psnr	50	0.8724
psnr	60	0.8724
filter strength	10	0.8724
filter strength	20	0.8724
filter strength	30	0.8724
filter strength	40	0.8724
filter strength	50	0.8724
filter strength	60	0.8724
sharpness	0	0.8724
sharpness	1	0.8724
sharpness	2	0.8724
sharpness	3	0.8724
sharpness	4	0.8724
sharpness	5	0.8724
sharpness	6	0.8724
sharpness	7	0.8724

no alpha	-	0.8724
sns	10	0.8531
sns	20	0.8710
sns	30	0.8961
sns	40	0.8965
sns	50	0.8948
sns	60	0.8826
sns	35	0.8880
sns	36	0.8872
sns	37	0.8962
sns	38	0.8947
sns	39	0.8969
sns	41	0.8956
sns	42	0.8922
sns	43	0.8894
sns	44	0.8905

sns	45	0.8896
pass	1	0.8724
pass	2	0.8724
pass	3	0.8724
pass	4	0.8724
pass	5	0.8724
pass	6	0.8724
pass	7	0.8724
pass	8	0.8724
pass	9	0.8724
pass	10	0.8724
method preset	6 default	0.8986
method sns	6 40	0.8995
preset sns	default 40	0.8915
method preset sns	6 default 40	0.8995

method sns resolution	6 40 56	0.8979
method sns resolution	6 40 64	0.9003
method sns resolution	6 40 75	0.8937
method sns resolution	6 40 80	0.9000
method sns resolution	6 40 96	0.8961
method sns resolution	6 40 125	0.8942
method sns resolution	6 40 144	0.8916
method sns resolution	6 40 150	0.8904
method sns resolution	6 40 160	0.8928
method sns resolution	6 40 180	0.8902
method sns resolution	6 40 200	0.8878
method sns resolution	6 40 224	0.8864

method sns resolution image manipulation	6 40 64 blackout	0.8992
method sns resolution image manipulation	6 40 64 whiteout	0.8869
method sns resolution image manipulation	6 40 64 full blur	0.9001
method sns resolution image manipulation	6 40 64 rectangular blur	0.9040
method sns resolution image manipulation	6 40 64 OFIQ landmark-region based blur	0.9034

Table A.14: Parameter optimization for WebP (grayscale)

Parameter(s)	Value(s)	Averaged self-similarity score
preset	default	0.9165
preset	photo	0.9160
preset	picture	0.9089
method	0	0.7826
method	1	0.8029
method	2	0.8613
method	3	0.8993

method	4	0.9165
method	5	0.9221
method	6	0.9396
segments	1	0.9152
segments	2	0.9152
segments	3	0.9152
segments	4	0.9152
psnr	10	0.9152
psnr	20	0.9152
psnr	30	0.9152
psnr	40	0.9152
psnr	50	0.9152
psnr	60	0.9152
filter strength	10	0.9152
filter strength	20	0.9152
filter strength	30	0.9152

filter strength	40	0.9152
filter strength	50	0.9152
filter strength	60	0.9152
sharpness	0	0.9152
sharpness	1	0.9152
sharpness	2	0.9152
sharpness	3	0.9152
sharpness	4	0.9152
sharpness	5	0.9152
sharpness	6	0.9152
sharpness	7	0.9152
no alpha	-	0.9152
sns	10	0.9052
sns	20	0.9210
sns	30	0.9337
sns	40	0.9341

sns	50	0.9318
sns	60	0.9295
sns	35	0.9273
sns	36	0.9334
sns	37	0.9313
sns	38	0.9261
sns	39	0.9307
sns	41	0.9289
sns	42	0.9325
sns	43	0.9286
sns	44	0.9271
sns	45	0.9301
pass	1	0.9152
pass	2	0.9152
pass	3	0.9152
pass	4	0.9152

pass	5	0.9152
pass	6	0.9152
pass	7	0.9152
pass	8	0.9152
pass	9	0.9152
pass	10	0.9152
method preset	6 default	0.9394
method sns	6 40	0.9403
preset sns	default 40	0.9389
method preset sns	6 default 40	0.9403
method sns resolution	6 40 56	0.9368
method sns resolution	6 40 64	0.9387
method sns resolution	6 40 75	0.9301
method sns resolution	6 40 80	0.9447

method	6	0.9421
sns	40	
resolution	96	
method	6	0.9382
sns	40	
resolution	125	
method	6	0.9361
sns	40	
resolution	144	
method	6	0.9324
sns	40	
resolution	150	
method	6	0.9364
sns	40	
resolution	160	
method	6	0.9255
sns	40	
resolution	180	
method	6	0.9249
sns	40	
resolution	200	
method	6	0.9243
sns	40	
resolution	224	
method	6	0.9222
sns	40	
resolution	80	
image manipulation	blackout	
method	6	0.9154
sns	40	
resolution	80	
image manipulation	whiteout	
method	6	0.9263
sns	40	
resolution	80	
image manipulation	full blur	

method	6	0.9482
sns	40	
resolution	80	
image manipulation	rectangular blur	
method	6	0.9469
sns	40	
resolution	80	
image manipulation	OFIQ landmark-region based blur	
method	6	0.9026
sns	40	
resolution	80	
image manipulation	full low-pass	
method	6	0.9067
sns	40	
resolution	80	
image manipulation	rectangular low-pass	

B RESULTS FOR TOP THREE SETTINGS

B.1 PARAMETERS USED

Table B.1: Top three settings for JPEG grayscale images

Number	**Parameters**	**Values**	**average self-similarity score ↑**
1	grayscale arithmetic smooth resolution image manipulation	- - 30 96 rectangle blur	0.9199
2	grayscale arithmetic smooth resolution image manipulation	- - 50 96 rectangle blur	0.9190
3	grayscale arithmetic smooth resolution image manipulation	- - 30 96 OFIQ landmark-region based blur	0.9148

Table B.2: Top three settings for JPEG 2000 grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	ratio number of resolutions resolution image manipulation	- 3 56 rectangle blur	0.8689
2	ratio number of resolutions resolution image manipulation	- 5 56 rectangle blur	0.8601
3	ratio number of resolutions resolution image manipulation	- 3 56 OFIQ landmark-region based blur	0.8475

Table B.3: Top three settings for JPEG 2000 color images

Number	Parameters	Values	average self-similarity score ↑
1	ratio number of resolutions resolution image manipulation	- 3 56 rectangle blur	0.8553
2	ratio number of resolutions resolution image manipulation	- 5 56 rectangle blur	0.8281
3	ratio number of resolutions resolution image manipulation	- 3 56 OFIQ landmark-region based blur	0.8383

Table B.4: Top three settings for JPEG XL grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	quality effort resolution image manipulation	- 10 56 rectangle blur	0.8874
2	distance effort resolution image manipulation	- 10 56 rectangle blur	0.8761
3	quality effort resolution image manipulation	- 10 56 OFIQ landmark-region based blur	0.8648

Table B.5: Top three settings for JPEG XL color images

Number	Parameters	Values	average self-similarity score ↑
1	quality effort resolution image manipulation	- 10 64 rectangle blur	0.8597
2	quality effort resolution image manipulation	- 6 64 rectangle blur	0.8582
3	quality effort resolution image manipulation	- 10 64 OFIQ landmark-region based blur	0.8495

Table B.6: Top three settings for JPEG AI grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	resolution image manipulation	200 full blur	0.9484
2	resolution image manipulation	200 OFIQ landmark-region based blur	0.9484
3	resolution image manipulation	200 rectangle blur	0.9486

Table B.7: Top three settings for JPEG AI color images

Number	Parameters	Values	average self-similarity score ↑
1	resolution image manipulation	180 rectangle blur	0.8939
2	resolution image manipulation	180 OFIQ landmark-region based blur	0.8938
3	resolution image manipulation	180 full blur	0.8963

Table B.8: Top three settings for AVIF grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	speed yuv resolution image manipulation	1 400 56 -	0.9173
2	speed yuv range depth resolution image manipulation	3 400 full 12 56 -	0.9156
3	speed yuv resolution image manipulation	1 400 56 rectangle blur	0.9088

Table B.9: Top three settings for AVIF color images

Number	Parameters	Values	average self-similarity score ↑
1	speed yuv resolution image manipulation	1 420 56 rectangle blur	0.8862
2	speed yuv range depth resolution image manipulation	3 420 full 12 56 rectangle blur	0.8855
3	speed yuv resolution image manipulation	1 420 56 OFIQ landmark-region based blur	0.8669

Table B.10: Top three settings for HEIF grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	resolution image manipulation	80 rectangle blur	0.9028
2	chroma downsampling resolution image manipulation	average 80 rectangle blur	0.9028
3	resolution image manipulation	80 OFIQ landmark-region based blur	0.8941

Table B.11: Top three settings for HEIF color images

Number	Parameters	Values	average self-similarity score ↑
1	chroma downsampling resolution image manipulation	average 96 rectangle blur	0.8596
2	resolution image manipulation	96 rectangle blur	0.8594
3	chroma downsampling resolution image manipulation	average 96 OFIQ landmark-region based blur	0.8576

Table B.12: Top three settings for WebP grayscale images

Number	Parameters	Values	average self-similarity score ↑
1	method sns preset resolution image manipulation	6 40 default 80 rectangle blur	0.9312
2	method sns resolution image manipulation	6 40 80 rectangle blur	0.9312
3	method sns resolution image manipulation	6 40 80 OFIQ landmark-region based blur	0.9212

Table B.13: Top three settings for WebP color images

Number	**Parameters**	**Values**	**average self-similarity score ↑**
1	method sns preset resolution image manipulation	6 default 40 64 -	0.8902
2	method sns resolution image manipulation	6 40 64 -	0.8902
3	method sns resolution image manipulation	6 40 64 rectangle blur	0.8946

B.2 FACE RECOGNITION PERFORMANCE

Table B.14: Face recognition performance for JPEG-compressed grayscale images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.4705	0.3157	0.5166	22
2	10.4386	0.8037	0.5471	56
3	10.4641	0.3588	0.5297	25

Table B.15: Face recognition performance for JPEG-compressed grayscale images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	6.7398	0.0903	0.2543	36
2	6.7362	0.0853	0.2541	34
3	6.7302	0.0878	0.2547	35

Table B.16: Face recognition performance for JPEG 2000-compressed grayscale images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.0235	1.1481	0.5215	80
2	9.8954	0.7319	0.4959	51
3	9.8545	1.2486	0.5110	87

Table B.17: Face recognition performance for JPEG 2000-compressed grayscale images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	6.7032	0.1179	0.2555	47
2	6.6710	0.1280	0.2556	51
3	6.6561	0.1631	0.2539	65

Table B.18: Face recognition performance for JPEG 2000-compressed color images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.8031	2.0235	0.5611	141
2	10.3392	3.3152	0.5576	231
3	10.5787	4.0327	0.5763	281

Table B.19: Face recognition performance for JPEG 2000-compressed color images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	7.5513	0.0251	0.2550	10
2	7.4212	0.0401	0.2534	16
3	7.4660	0.0401	0.2551	16

Table B.20: Face recognition performance for JPEG XL-compressed grayscale images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.2944	0.8324	0.5267	58
2	10.1993	0.9185	0.5207	64
3	10.0148	0.8037	0.5041	56

Table B.21: Face recognition performance for JPEG XL-compressed grayscale images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	6.7492	0.0979	0.2541	39
2	6.7346	0.1179	0.2549	47
3	6.6728	0.1380	0.2541	55

Table B.22: Face recognition performance for JPEG XL-compressed color images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.8960	1.3634	0.5506	95
2	10.8658	1.1625	0.5426	81
3	10.6835	2.2245	0.5572	155

Table B.23: Face recognition performance for JPEG XL-compressed color images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	7.5400	0.0176	0.2531	7
2	7.5309	0.0251	0.2524	10
3	7.4335	0.0251	0.2528	10

Table B.24: Face recognition performance for JPEG AI-compressed grayscale images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	10.8834	0.0718	0.5173	5
2	10.8987	0.1435	0.5294	10
3	10.9152	0.1292	0.5294	9

Table B.25: Face recognition performance for JPEG AI-compressed grayscale images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	6.7577	0.0903	0.2541	36
2	6.7852	0.0753	0.2558	30
3	6.7835	0.0753	0.2546	30

Table B.26: Face recognition performance for JPEG AI-compressed color images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	11.0780	1.1194	0.5694	78
2	11.1395	0.8754	0.5651	61
3	11.1162	1.0046	0.5682	70

Table B.27: Face recognition performance for JPEG AI-compressed color images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	7.3852	0.0201	0.2497	8
2	7.3973	0.0151	0.2493	6
3	7.3949	0.0201	0.2491	8

Table B.28: Face recognition performance for AVIF-compressed grayscale images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	10.5513	0.2870	0.5238	20
2	10.5190	0.3731	0.5296	26
3	10.5612	0.3588	0.5196	25

Table B.29: Face recognition performance for AVIF-compressed grayscale images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	6.8026	0.0803	0.2550	32
2	6.7815	0.0828	0.2548	33
3	6.7786	0.0853	0.2543	34

Table B.30: Face recognition performance for AVIF-compressed color images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	11.2686	2.1814	0.5933	152
2	11.2119	1.9518	0.5841	136
3	11.0657	2.1383	0.5762	149

Table B.31: Face recognition performance for AVIF-compressed color images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	7.5987	0.0151	0.2523	6
2	7.5416	0.0276	0.2509	11
3	7.4913	0.0151	0.2520	6

Table B.32: Face recognition performance for HEIF-compressed grayscale images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	10.4317	1.0046	0.5448	70
2	10.4317	1.0046	0.5448	70
3	10.3680	0.8754	0.5356	61

Table B.33: Face recognition performance for HEIF-compressed grayscale images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	6.7572	0.1104	0.2548	44
2	6.7572	0.1104	0.2548	44
3	6.7341	0.1104	0.2557	44

Table B.34: Face recognition performance for HEIF-compressed color images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	10.8002	0.7032	0.5237	49
2	10.8294	2.4110	0.5753	168
3	10.8323	1.4351	0.5525	100

Table B.35: Face recognition performance for HEIF-compressed color images compared against the full dataset

Number	Sensitivity index ↑	FNMR at FMR 0.01% in % ↓	Threshold	False non-matches in total ↓
1	7.4056	0.0276	0.2508	11
2	7.4507	0.0301	0.2510	12
3	7.4242	0.0276	0.2516	11

Table B.36: Face recognition performance for WebP-compressed grayscale images compared against the frontal dataset

Number	Sensitivity index ↑	FNMR at FMR 0.0001% in % ↓	Threshold	False non-matches in total ↓
1	10.7104	0.4305	0.5463	30
2	10.7104	0.4305	0.5463	30
3	10.6089	0.4162	0.5329	29

Table B.37: Face recognition performance for WebP-compressed grayscale images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	6.7764	0.0602	0.2550	24
2	6.7764	0.0602	0.2550	24
3	6.7460	0.1054	0.2552	42

Table B.38: Face recognition performance for WebP-compressed color images compared against the frontal dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.0001% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	11.2422	2.1096	0.5936	147
2	11.2422	2.1096	0.5936	147
3	11.3483	1.1338	0.5734	79

Table B.39: Face recognition performance for WebP-compressed color images compared against the full dataset

Number	**Sensitivity index** ↑	**FNMR at FMR 0.01% in %** ↓	**Threshold**	**False non-matches in total** ↓
1	7.6150	0.0176	0.2533	7
2	7.6150	0.0176	0.2533	7
3	7.5903	0.0151	0.2528	6

C

TABLES FOR FIXED THRESHOLDS

Table C.1: Face recognition performance of each compression algorithm at a threshold of 0.5864 comparing grayscale images against the frontal dataset

Compression algorithm	FNMR in % at threshold = 0.5864 ↓	FMR in % at threshold = 0.5864 ↓	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	0.4592	0.000115	32	8
JPEG AI	1.1194	0.000072	78	5
WebP	2.0379	0.000014	142	1
AVIF	2.1814	0.000014	152	1
JPEG	2.4684	0.000058	172	4
HEIF	3.3152	0.000043	231	3
JPEG XL	5.6401	0.000014	393	1
JPEG 2000	5.9127	0.000029	412	2

Table C.2: Face recognition performance of each compression algorithm at a threshold of 0.2551 comparing grayscale images against the full dataset

Compression algorithm	FNMR in % at threshold = 0.2551 ↓	FMR in % at threshold = 0.2551 ↓	False non-matches in total ↓	False matches in total ↓
no compression (baseline)	0.0502	0.010004	20	2931
WebP	0.0602	0.009970	24	2921
JPEG AI	0.0728	0.010332	29	3027
AVIF	0.0803	0.010001	32	2930
JPEG	0.0878	0.009598	35	2812
JPEG XL	0.0979	0.009605	39	2814
HEIF	0.1104	0.010294	44	3016
JPEG 2000	0.1179	0.010134	47	2969

Table C.3: Face recognition performance of each compression algorithm at a threshold of 0.6677 comparing color images against the frontal dataset

Compression algorithm	**FNMR in % at threshold = 0.6677** ↓	**FMR in % at threshold = 0.6677** ↓	**False non-matches in total** ↓	**False matches in total** ↓
no compression (baseline)	2.4397	0.000115	170	8
WebP	9.6584	0.000000	673	0
AVIF	11.1223	0.000000	775	0
JPEG AI	11.3088	0.000000	788	0
JPEG 2000	16.8054	0.000000	1171	0
HEIF	16.9202	0.000000	1179	0
JPEG XL	17.0781	0.000000	1190	0
JPEG	n.a.	n.a.	n.a.	n.a.

Table C.4: Face recognition performance of each compression algorithm at a threshold of 0.2566 comparing color images against the full dataset

Compression algorithm	**FNMR in % at threshold = 0.2566** ↓	**FMR in % at threshold = 0.2566** ↓	**False non-matches in total** ↓	**False matches in total** ↓
no compression (baseline)	0.0100	0.010006	4	2932
AVIF	0.0151	0.008323	6	2439
WebP	0.0176	0.008549	7	2505
JPEG AI	0.0251	0.007375	10	2161
JPEG XL	0.0301	0.008719	12	2555
JPEG 2000	0.0301	0.009453	12	2770
HEIF	0.0351	0.007719	14	2262
JPEG	n.a.	n.a.	n.a.	n.a.

D SIMILARITY SCORE DISTRIBUTIONS

D.1 MATED SIMILARITY SCORE DISTRIBUTIONS (COMPRESSED VERSUS ORIGINAL)

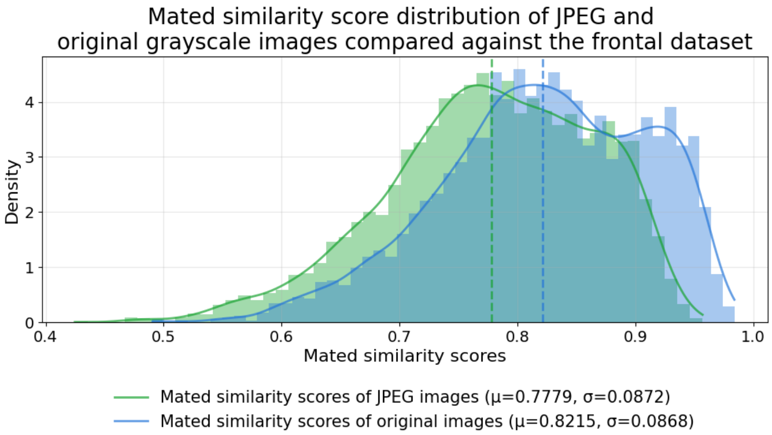


Figure D.1: Mated similarity score distributions of JPEG-compressed and original grayscale images compared against the frontal dataset.

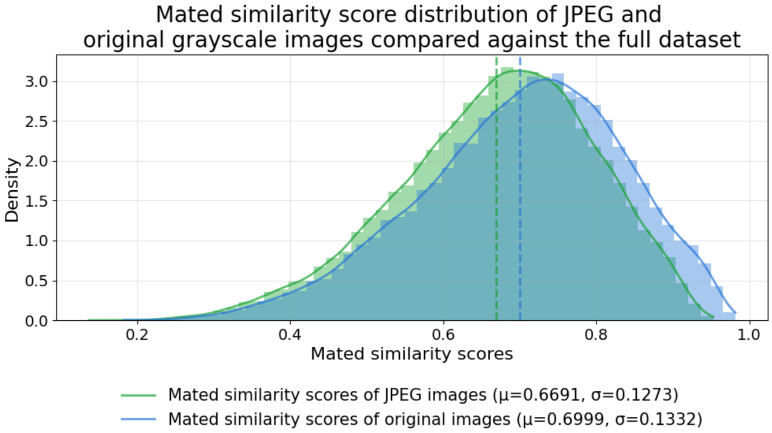


Figure D.2: Mated similarity score distributions of JPEG-compressed and original grayscale images compared against the full dataset.

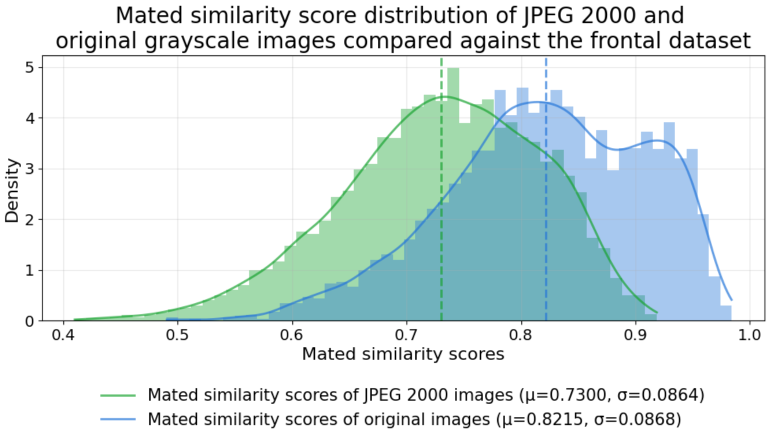


Figure D.3: Mated similarity score distributions of JPEG 2000-compressed and original grayscale images compared against the frontal dataset.

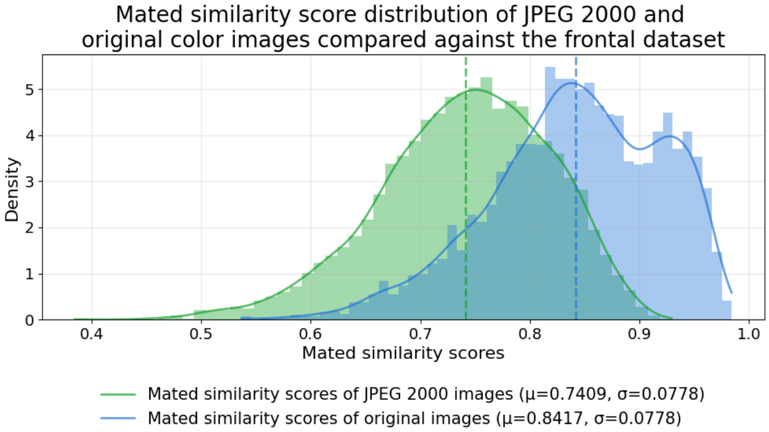


Figure D.4: Mated similarity score distributions of JPEG 2000-compressed and original color images compared against the frontal dataset.

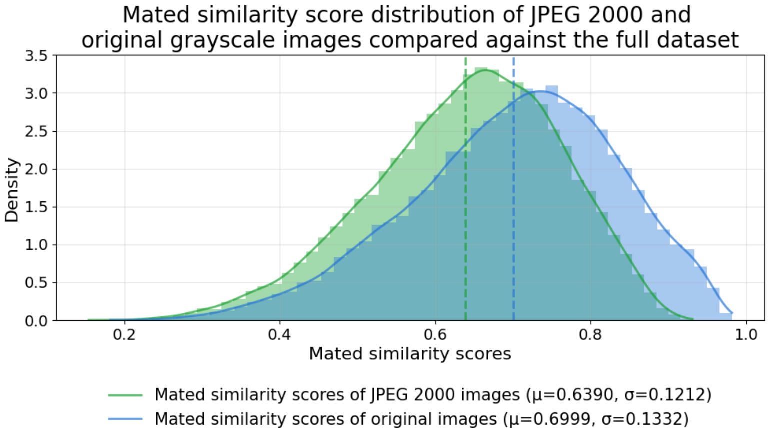


Figure D.5: Mated similarity score distributions of JPEG 2000-compressed and original grayscale images compared against the full dataset.

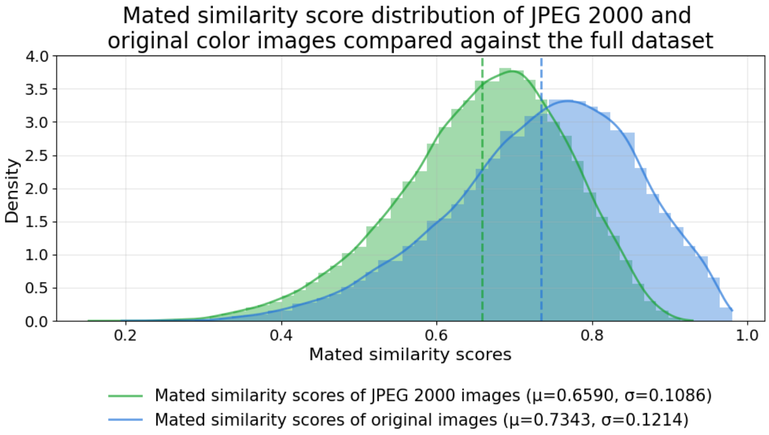


Figure D.6: Mated similarity score distributions of JPEG 2000-compressed and original color images compared against the full dataset.

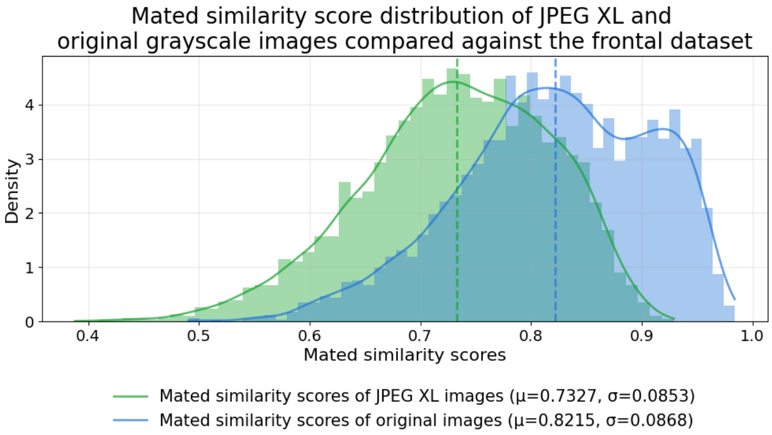


Figure D.7: Mated similarity score distributions of JPEG XL-compressed and original grayscale images compared against the frontal dataset.

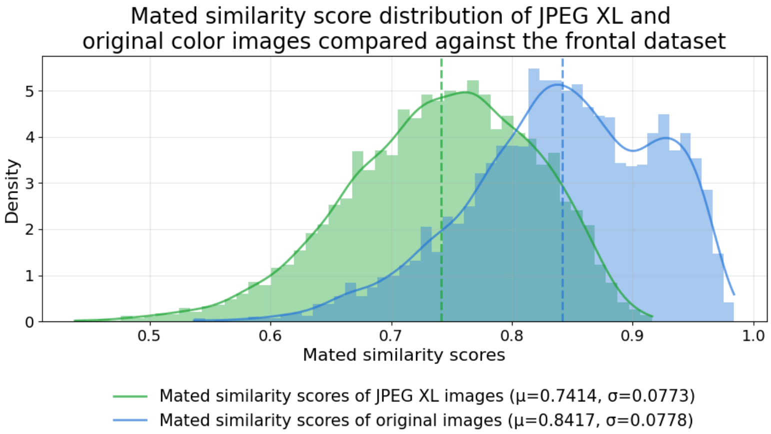


Figure D.8: Mated similarity score distributions of JPEG XL-compressed and original color images compared against the frontal dataset.

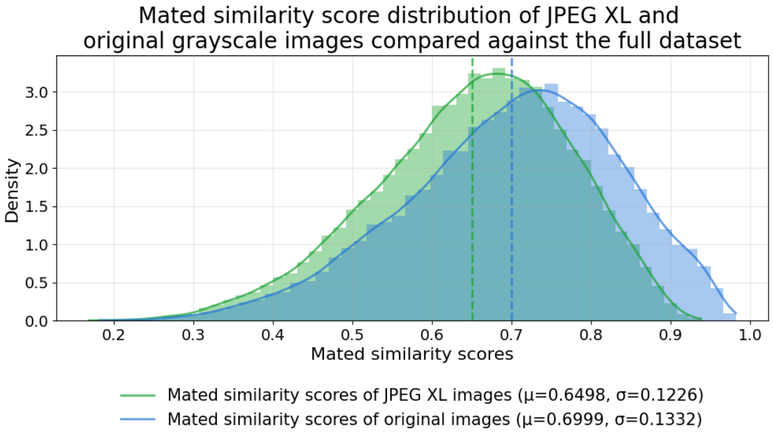


Figure D.9: Mated similarity score distributions of JPEG XL-compressed and original grayscale images compared against the full dataset.

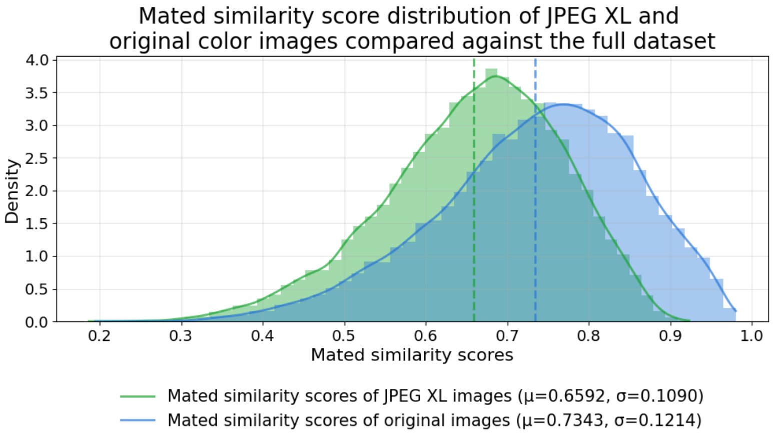


Figure D.10: Mated similarity score distributions of JPEG XL-compressed and original color images compared against the full dataset.

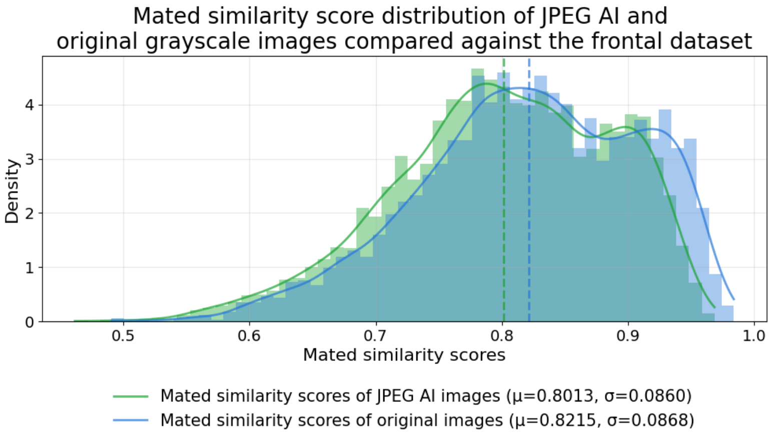


Figure D.11: Mated similarity score distributions of JPEG AI-compressed and original grayscale images compared against the frontal dataset.

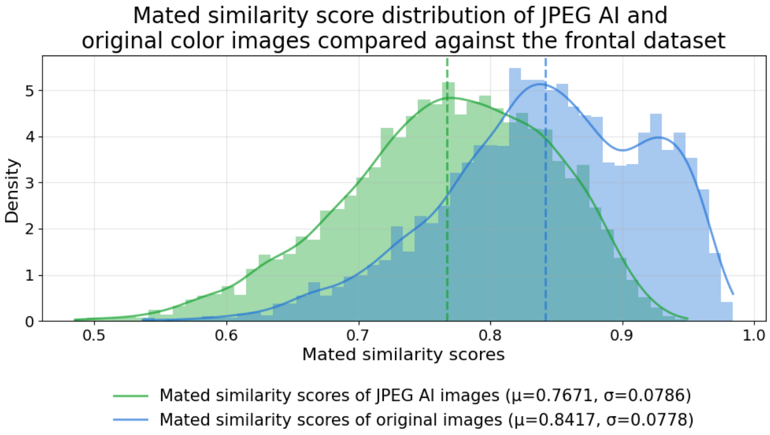


Figure D.12: Mated similarity score distributions of JPEG AI-compressed and original color images compared against the frontal dataset.

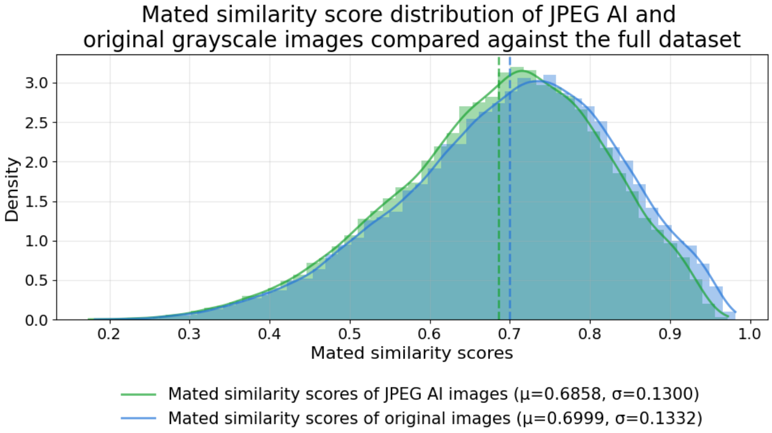


Figure D.13: Mated similarity score distributions of JPEG AI-compressed and original grayscale images compared against the full dataset.

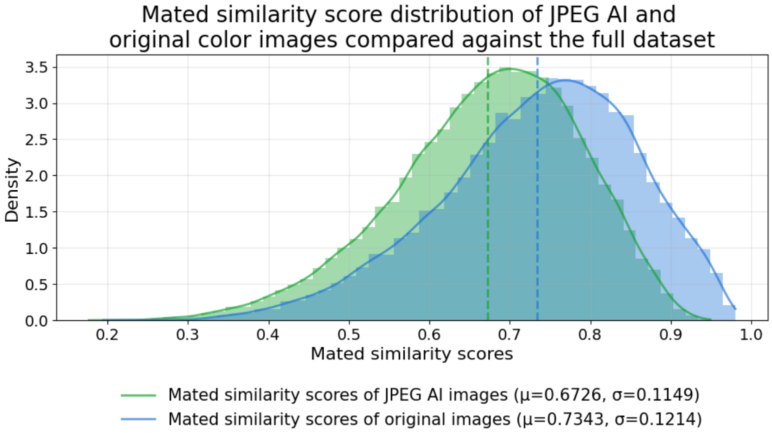


Figure D.14: Mated similarity score distributions of JPEG AI-compressed and original color images compared against the full dataset.

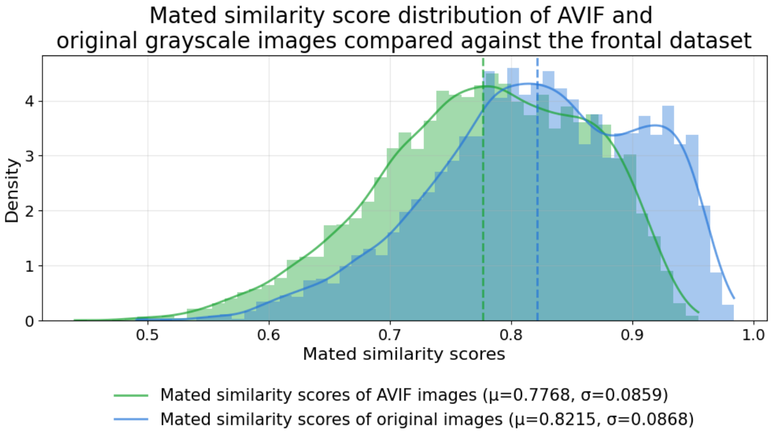


Figure D.15: Mated similarity score distributions of AVIF-compressed and original grayscale images compared against the frontal dataset.

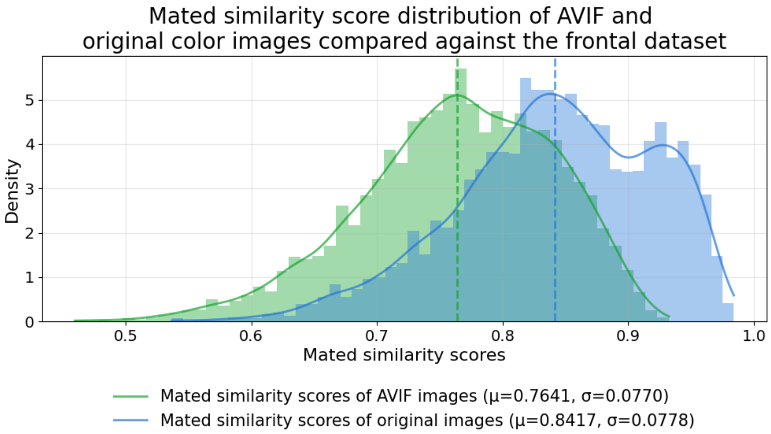


Figure D.16: Mated similarity score distributions of AVIF-compressed and original color images compared against the frontal dataset.

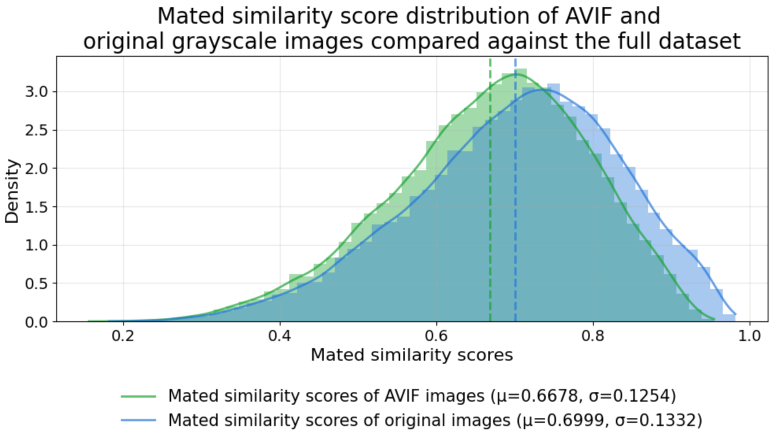


Figure D.17: Mated similarity score distributions of AVIF-compressed and original grayscale images compared against the full dataset.

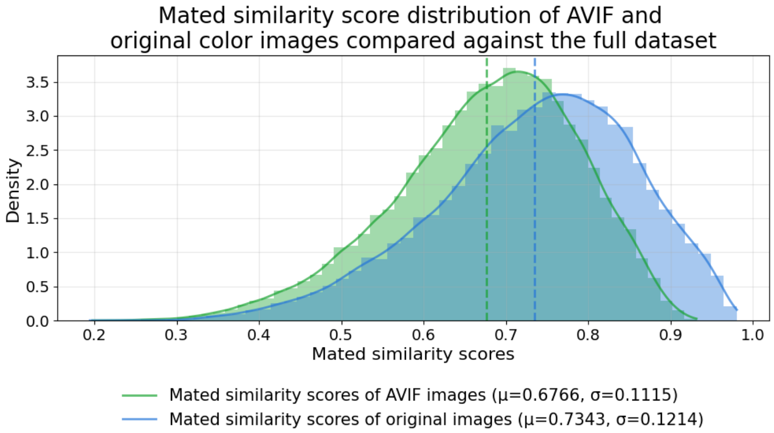


Figure D.18: Mated similarity score distributions of AVIF-compressed and original color images compared against the full dataset.

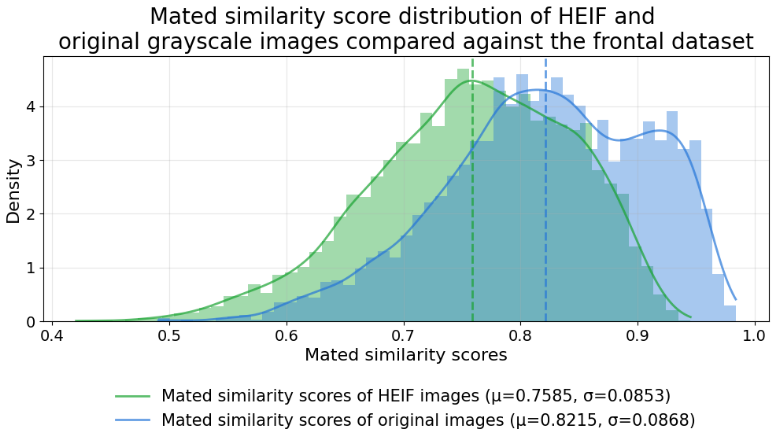


Figure D.19: Mated similarity score distributions of HEIF-compressed and original grayscale images compared against the frontal dataset.

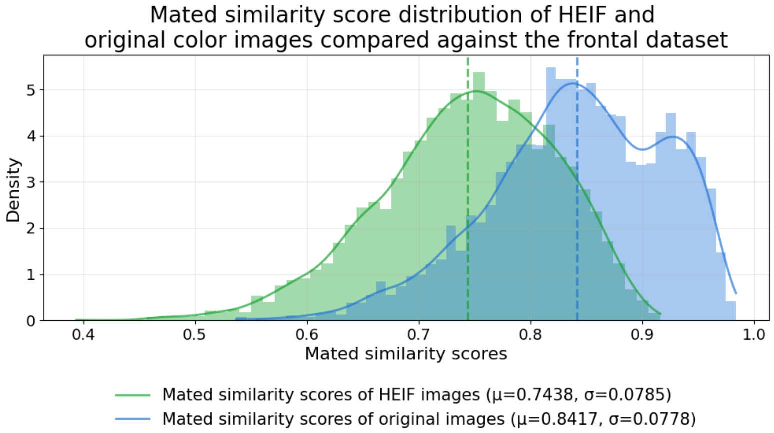


Figure D.20: Mated similarity score distributions of HEIF-compressed and original color images compared against the frontal dataset.

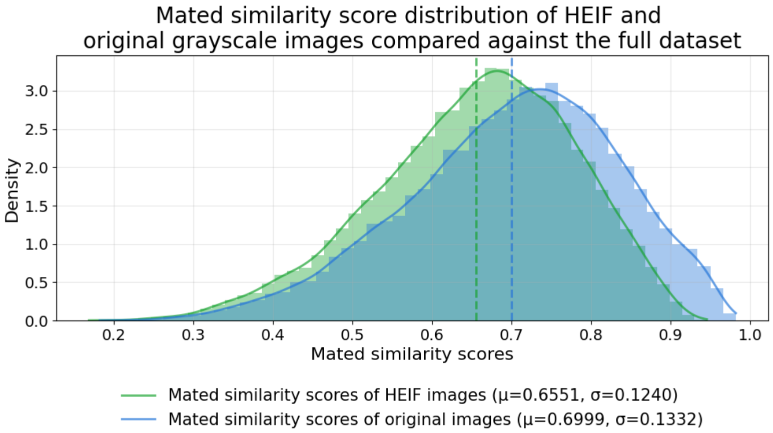


Figure D.21: Mated similarity score distributions of HEIF-compressed and original grayscale images compared against the full dataset.

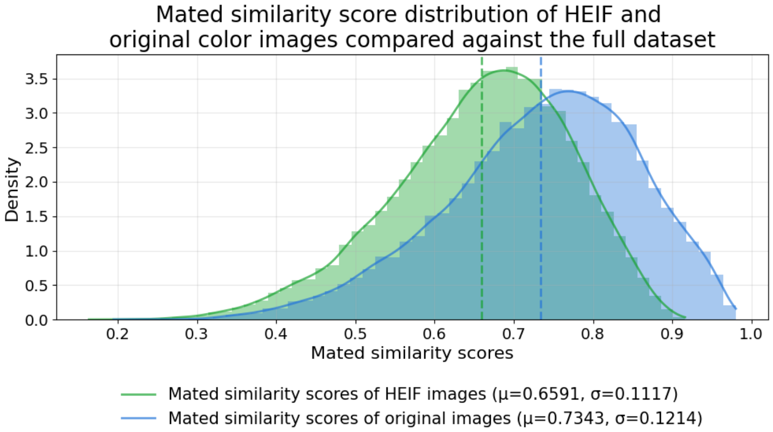


Figure D.22: Mated similarity score distributions of HEIF-compressed and original color images compared against the full dataset.

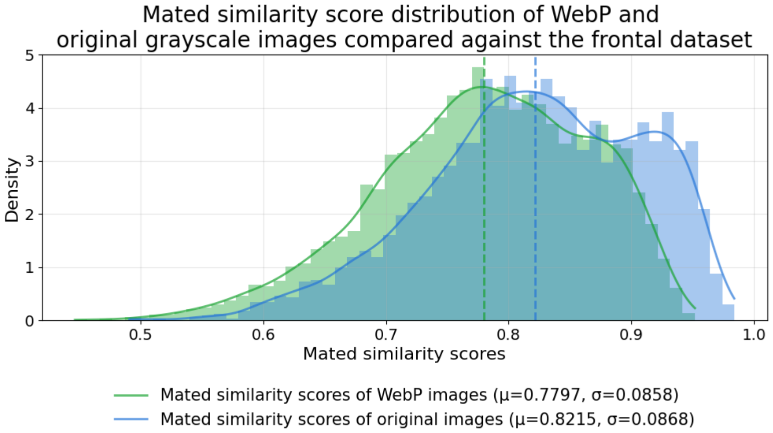


Figure D.23: Mated similarity score distributions of WebP-compressed and original grayscale images compared against the frontal dataset.

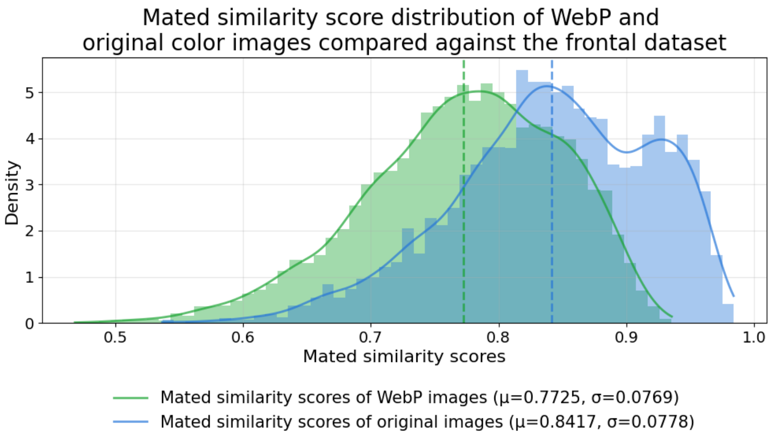


Figure D.24: Mated similarity score distributions of WebP-compressed and original color images compared against the frontal dataset.

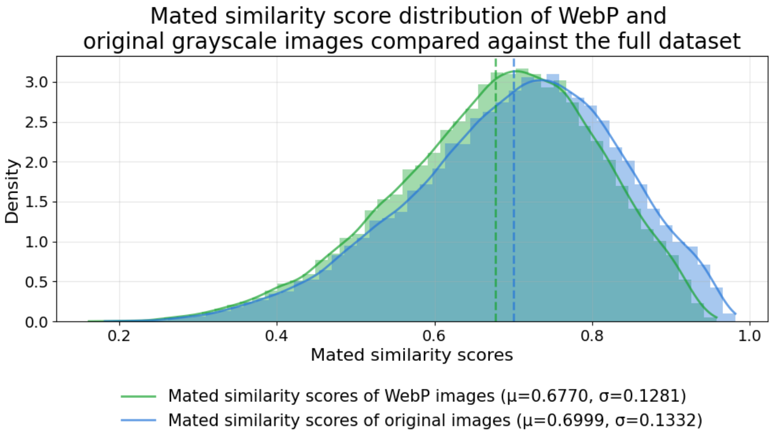


Figure D.25: Mated similarity score distributions of WebP-compressed and original grayscale images compared against the full dataset.

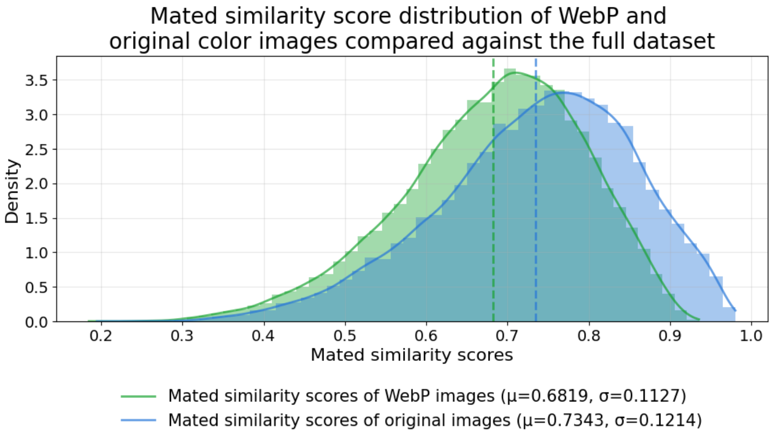


Figure D.26: Mated similarity score distributions of WebP-compressed and original color images compared against the full dataset.

D.2 NON-MATED SIMILARITY SCORE DISTRIBUTIONS (COMPRESSED VERSUS ORIGINAL)

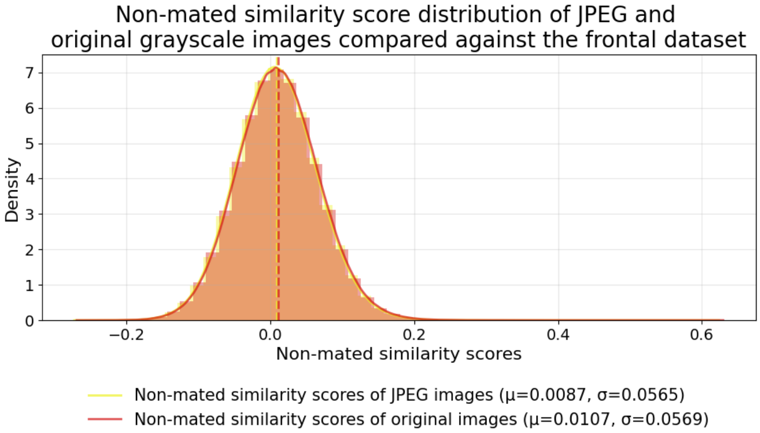


Figure D.27: Non-mated similarity score distributions of JPEG-compressed and original grayscale images compared against the frontal dataset.

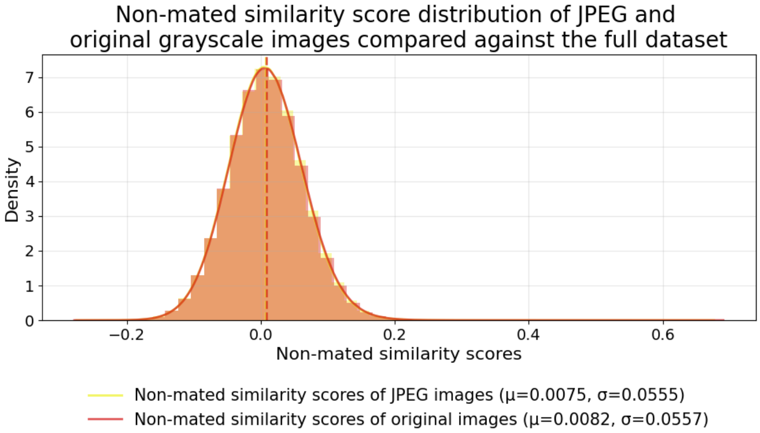


Figure D.28: Non-mated similarity score distributions of JPEG-compressed and original grayscale images compared against the full dataset.

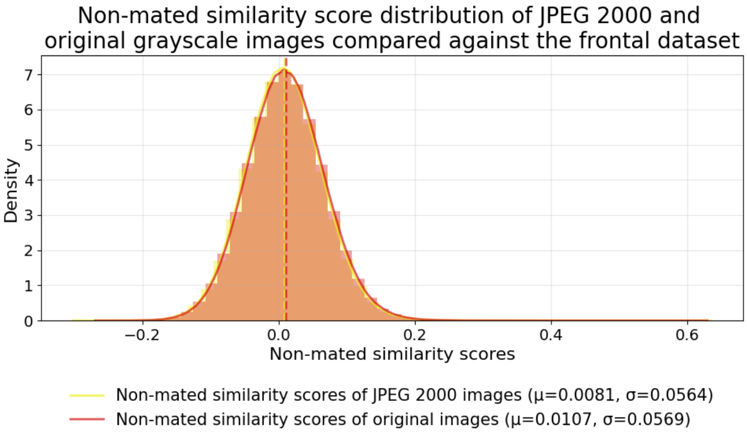


Figure D.29: Non-mated similarity score distributions of JPEG 2000-compressed and original grayscale images compared against the frontal dataset.

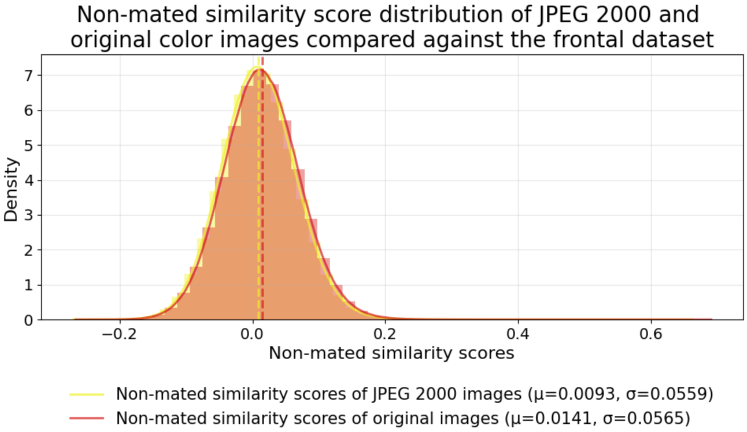


Figure D.30: Non-mated similarity score distributions of JPEG 2000-compressed and original color images compared against the frontal dataset.

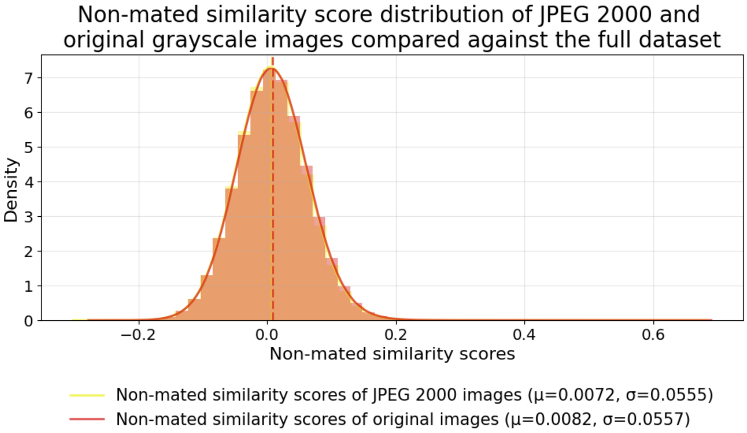


Figure D.31: Non-mated similarity score distributions of JPEG 2000-compressed and original grayscale images compared against the full dataset.

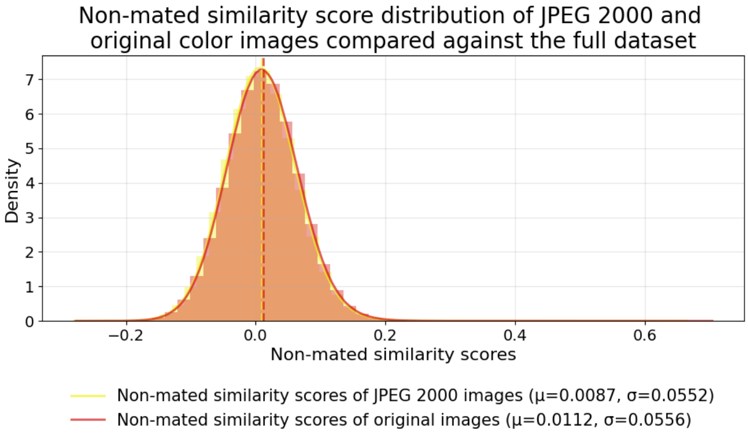


Figure D.32: Non-mated similarity score distributions of JPEG 2000-compressed and original color images compared against the full dataset.

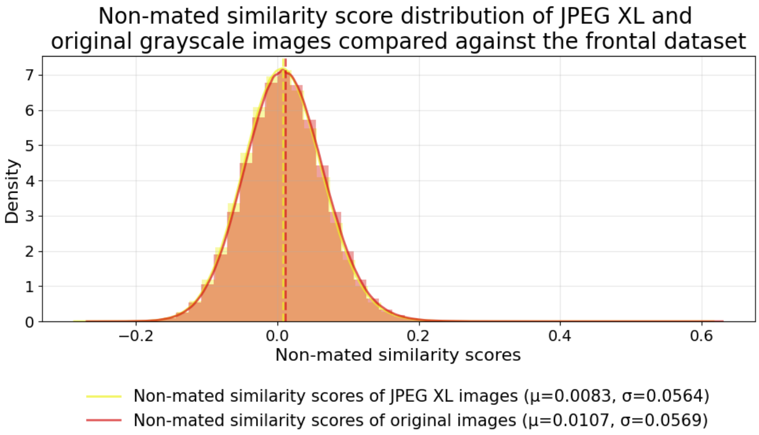


Figure D.33: Non-mated similarity score distributions of JPEG XL-compressed and original grayscale images compared against the frontal dataset.

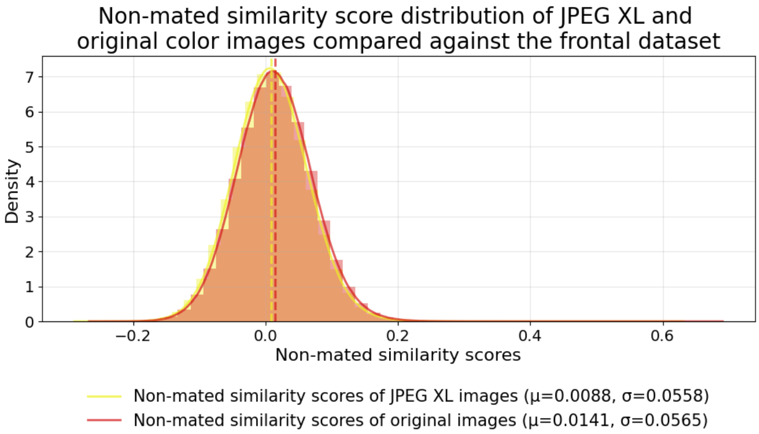


Figure D.34: Non-mated similarity score distributions of JPEG XL-compressed and original color images compared against the frontal dataset.

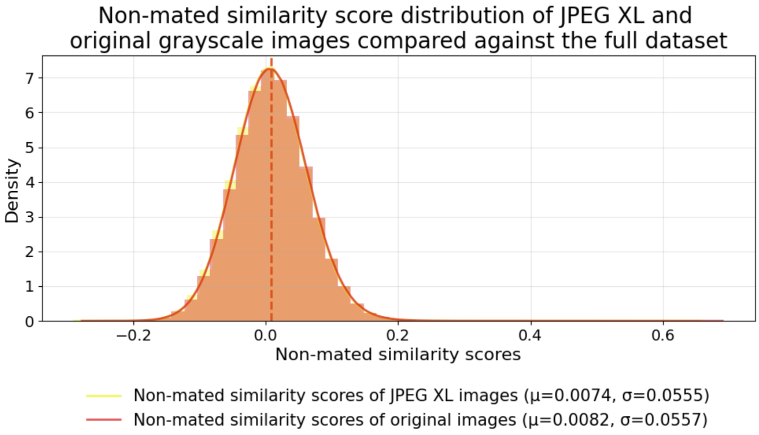


Figure D.35: Non-mated similarity score distributions of JPEG XL-compressed and original grayscale images compared against the full dataset.

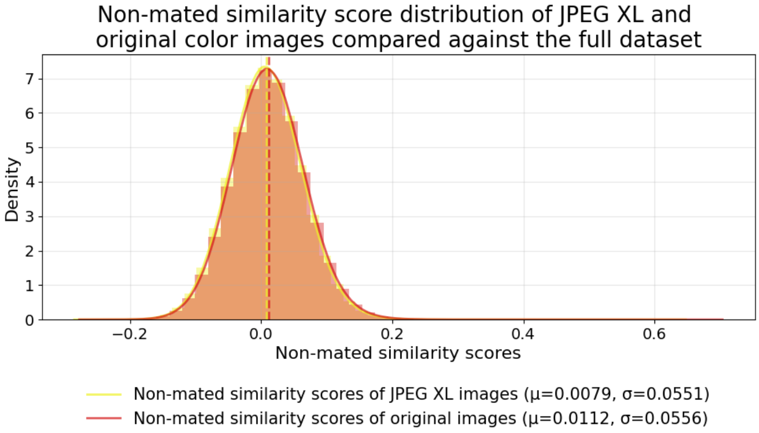


Figure D.36: Non-mated similarity score distributions of JPEG XL-compressed and original color images compared against the full dataset.

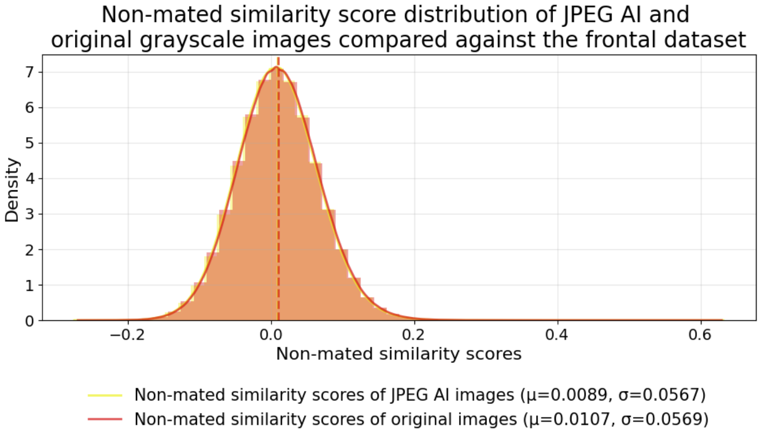


Figure D.37: Non-mated similarity score distributions of JPEG AI-compressed and original grayscale images compared against the frontal dataset.

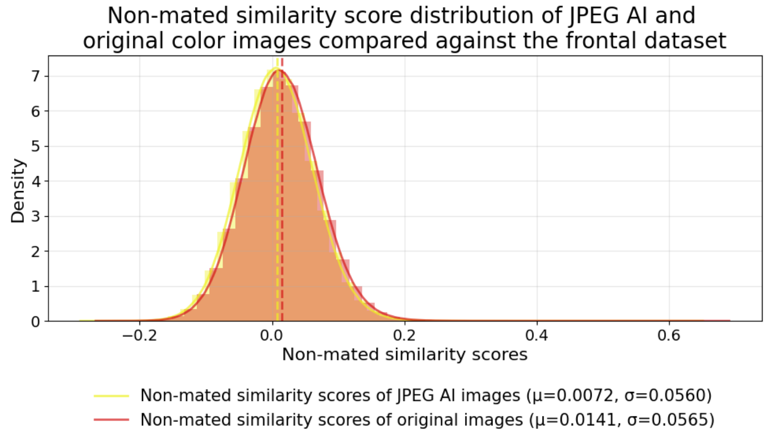


Figure D.38: Non-mated similarity score distributions of JPEG AI-compressed and original color images compared against the frontal dataset.

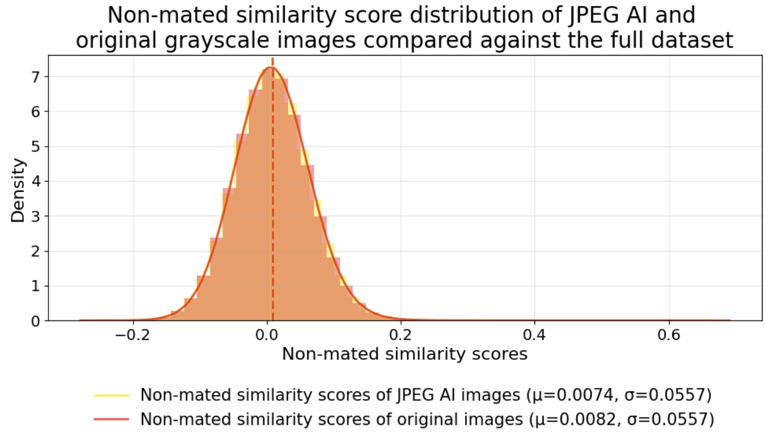


Figure D.39: Non-mated similarity score distributions of JPEG AI-compressed and original grayscale images compared against the full dataset.

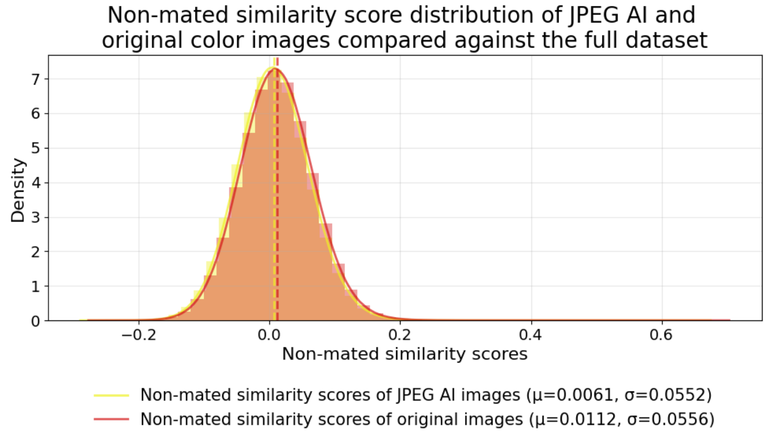


Figure D.40: Non-mated similarity score distributions of JPEG AI-compressed and original color images compared against the full dataset.

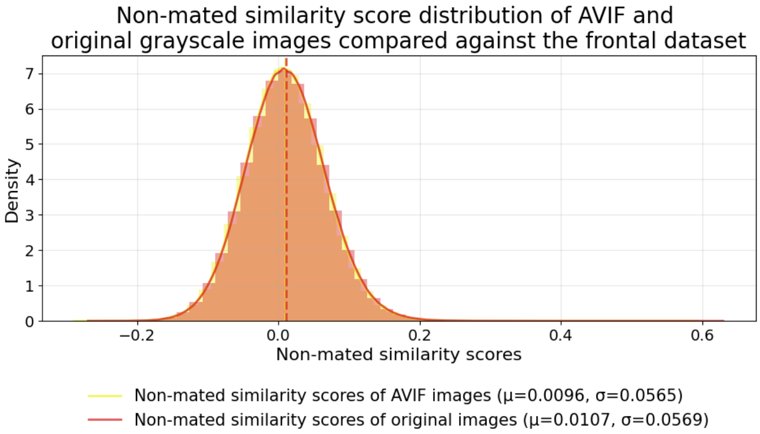


Figure D.41: Non-mated similarity score distributions of AVIF-compressed and original grayscale images compared against the frontal dataset.

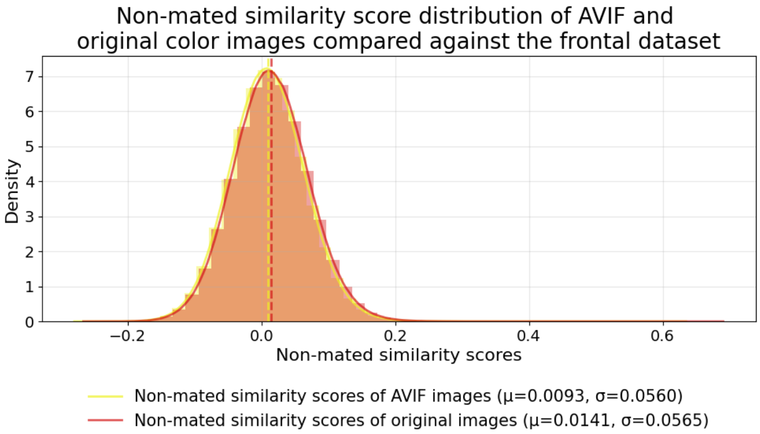


Figure D.42: Non-mated similarity score distributions of AVIF-compressed and original color images compared against the frontal dataset.

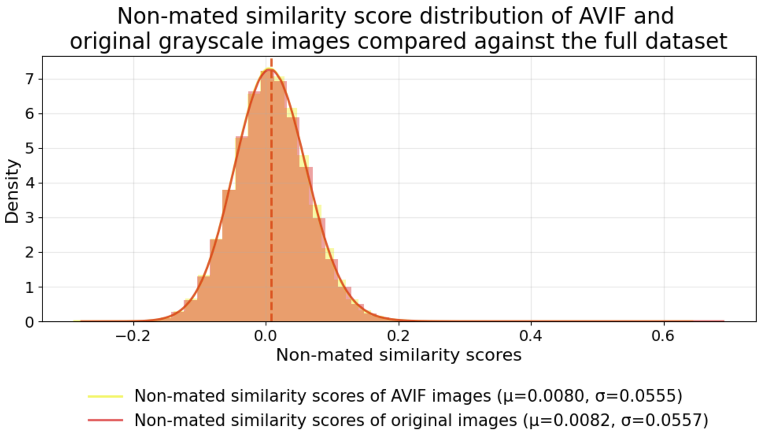


Figure D.43: Non-mated similarity score distributions of AVIF-compressed and original grayscale images compared against the full dataset.

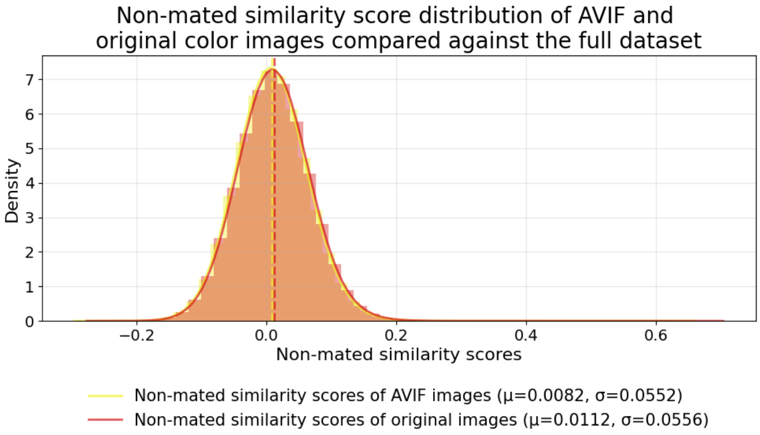


Figure D.44: Non-mated similarity score distributions of AVIF-compressed and original color images compared against the full dataset.

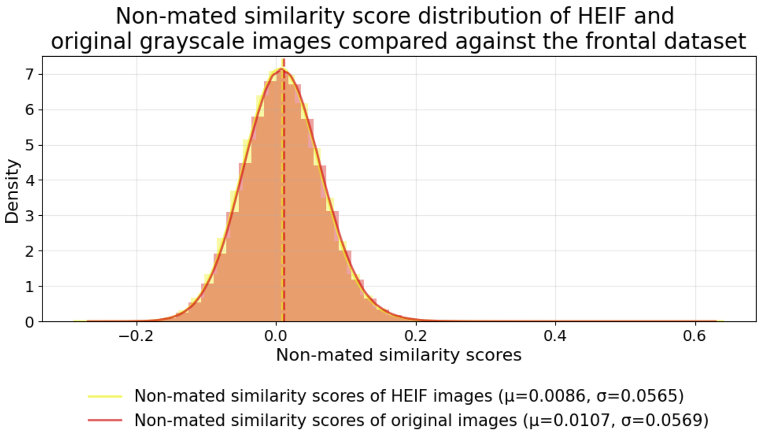


Figure D.45: Non-mated similarity score distributions of HEIF-compressed and original grayscale images compared against the frontal dataset.

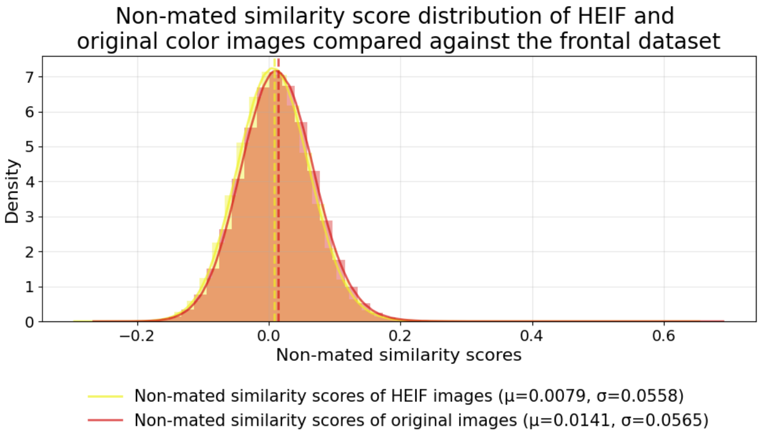


Figure D.46: Non-mated similarity score distributions of HEIF-compressed and original color images compared against the frontal dataset.

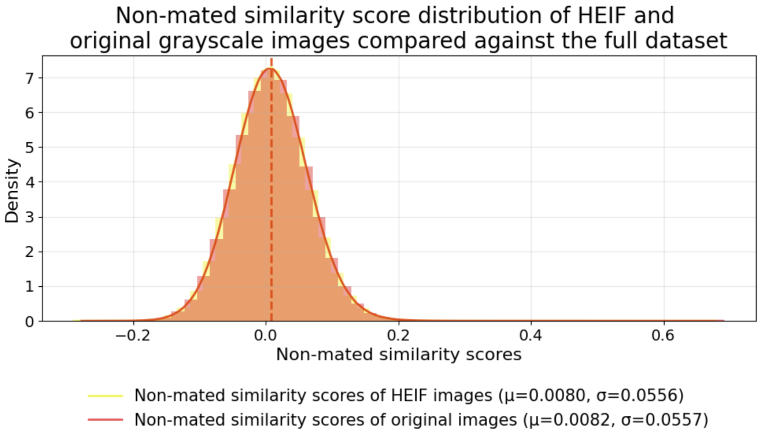


Figure D.47: Non-mated similarity score distributions of HEIF-compressed and original grayscale images compared against the full dataset.

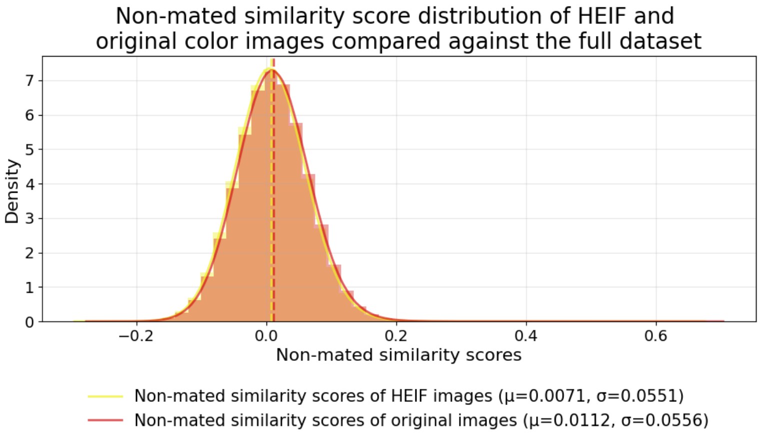


Figure D.48: Non-mated similarity score distributions of HEIF-compressed and original color images compared against the full dataset.

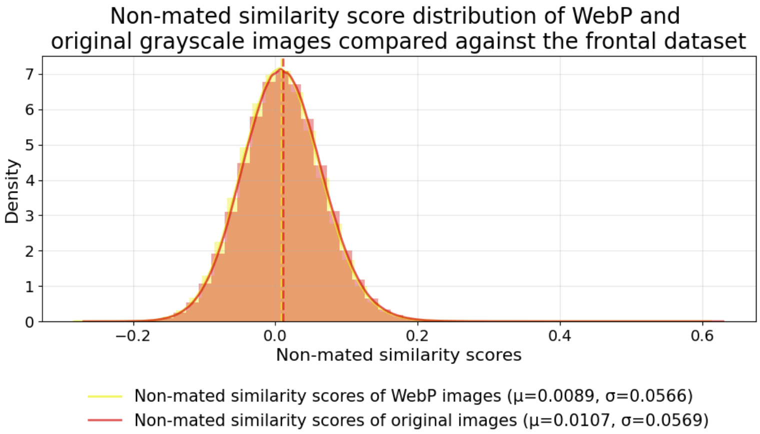


Figure D.49: Non-mated similarity score distributions of WebP-compressed and original grayscale images compared against the frontal dataset.

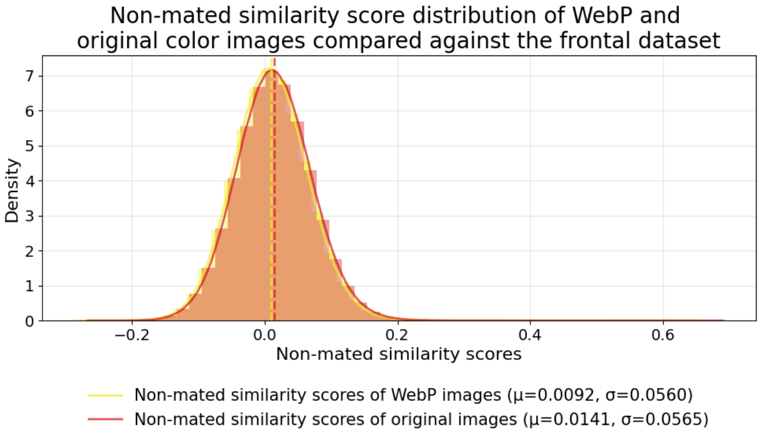


Figure D.50: Non-mated similarity score distributions of WebP-compressed and original color images compared against the frontal dataset.

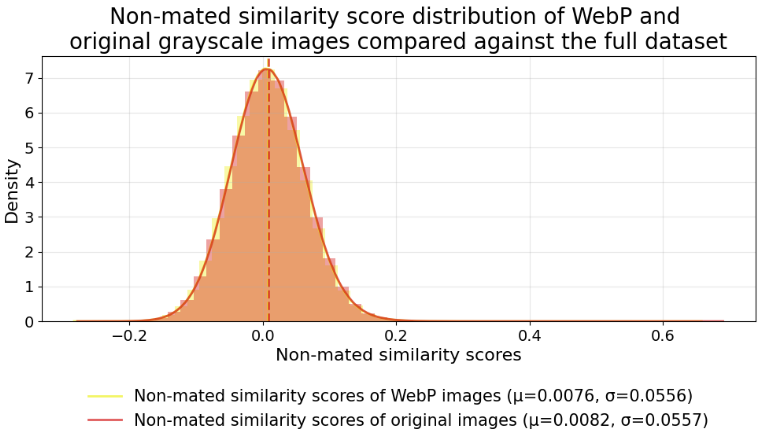


Figure D.51: Non-mated similarity score distributions of WebP-compressed and original grayscale images compared against the full dataset.

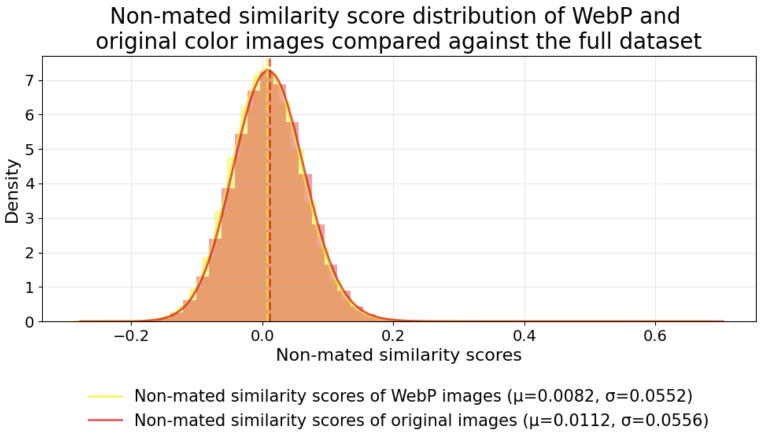


Figure D.52: Non-mated similarity score distributions of WebP-compressed and original color images compared against the full dataset.

D.3 SIMILARITY SCORE DISTRIBUTIONS (COLOR VERSUS GRAYSCALE)

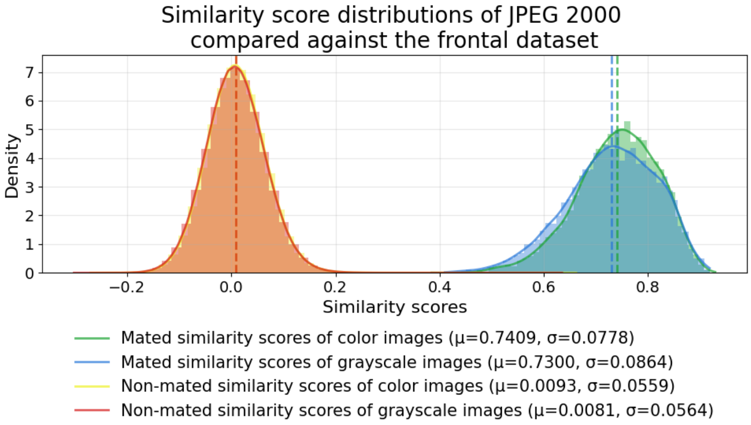


Figure D.53: Mated and non-mated similarity score distributions of JPEG 2000-compressed color and grayscale images compared against the frontal dataset.

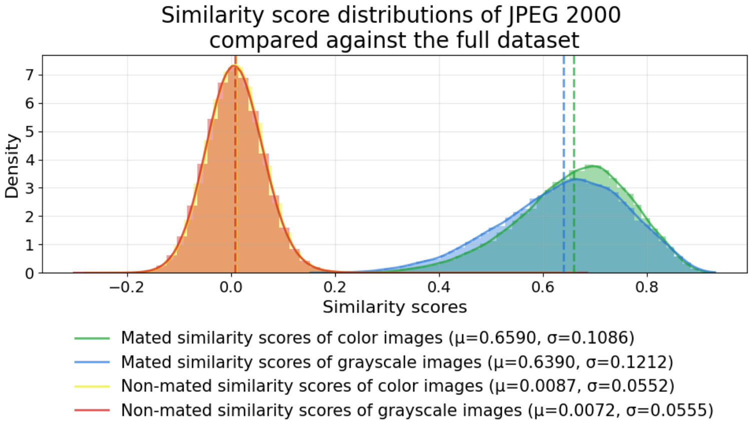


Figure D.54: Mated and non-mated similarity score distributions of JPEG 2000-compressed color and grayscale images compared against the full dataset.

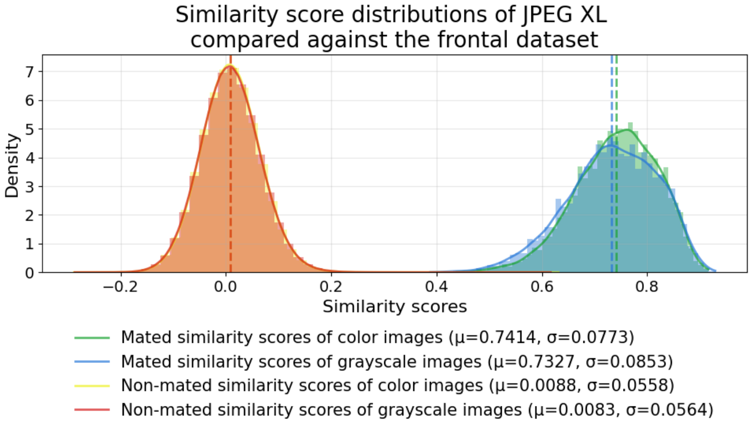


Figure D.55: Mated and non-mated similarity score distributions of JPEG XL-compressed color and grayscale images compared against the frontal dataset.

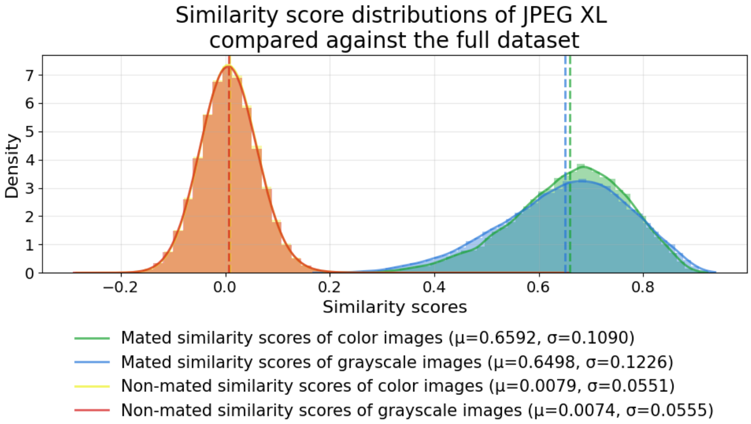


Figure D.56: Mated and non-mated similarity score distributions of JPEG XL-compressed color and grayscale images compared against the full dataset.

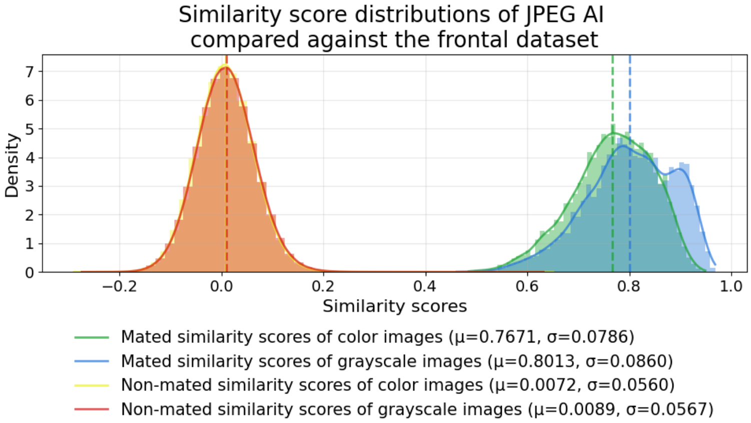


Figure D.57: Mated and non-mated similarity score distributions of JPEG AI-compressed color and grayscale images compared against the frontal dataset.

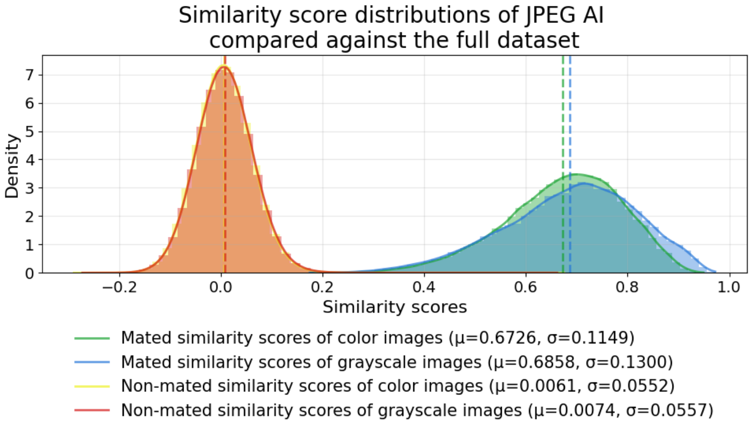


Figure D.58: Mated and non-mated similarity score distributions of JPEG AI-compressed color and grayscale images compared against the full dataset.

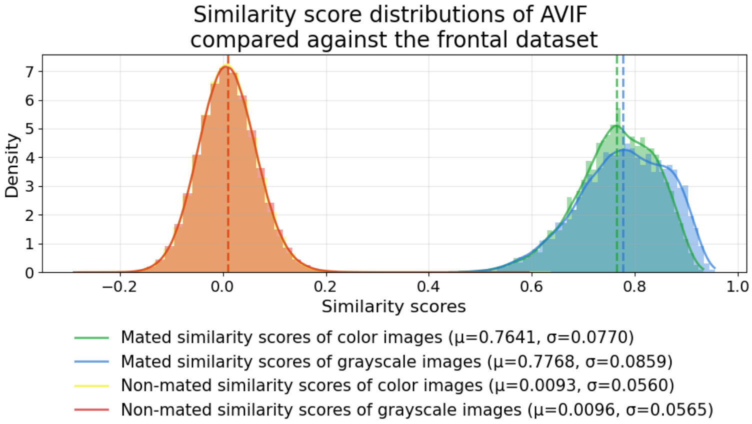


Figure D.59: Mated and non-mated similarity score distributions of AVIF-compressed color and grayscale images compared against the frontal dataset.

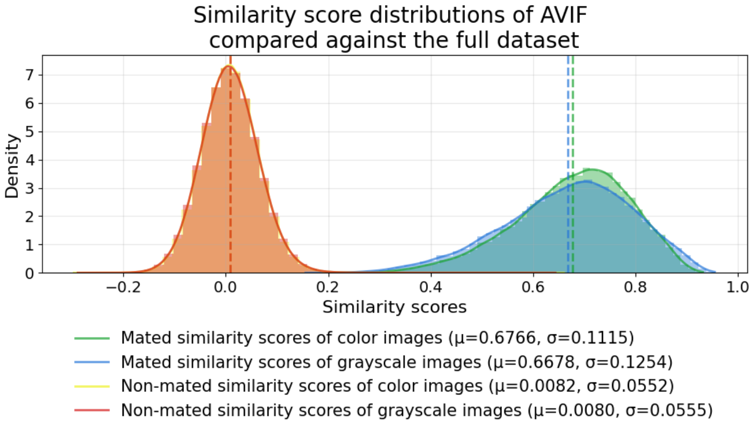


Figure D.60: Mated and non-mated similarity score distributions of AVIF-compressed color and grayscale images compared against the full dataset.

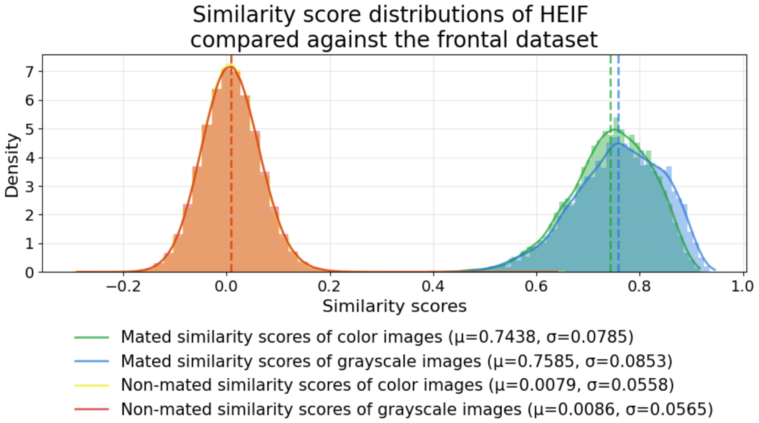


Figure D.61: Mated and non-mated similarity score distributions of HEIF-compressed color and grayscale images compared against the frontal dataset.

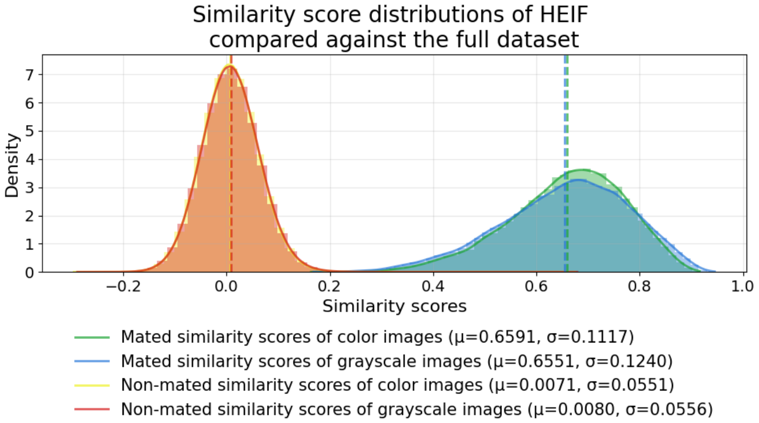


Figure D.62: Mated and non-mated similarity score distributions of HEIF-compressed color and grayscale images compared against the full dataset.

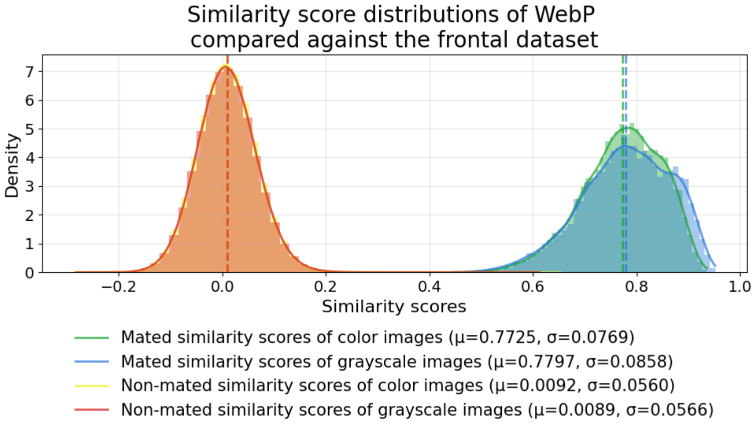


Figure D.63: Mated and non-mated similarity score distributions of WebP-compressed color and grayscale images compared against the frontal dataset.

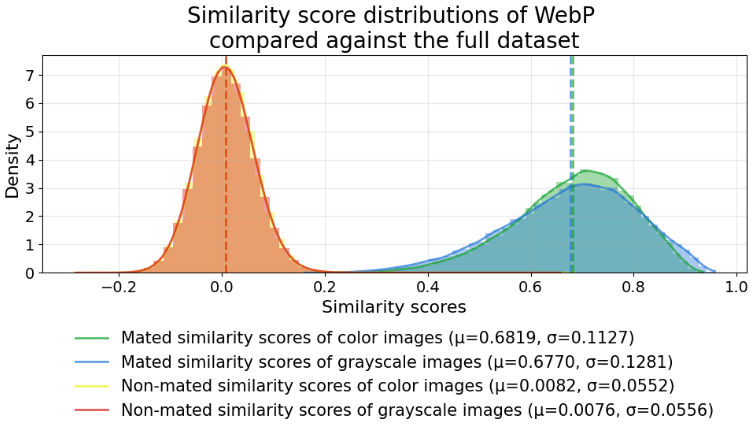


Figure D.64: Mated and non-mated similarity score distributions of WebP-compressed color and grayscale images compared against the full dataset.

D.4 mated similarity score distributions (each per algorithm)

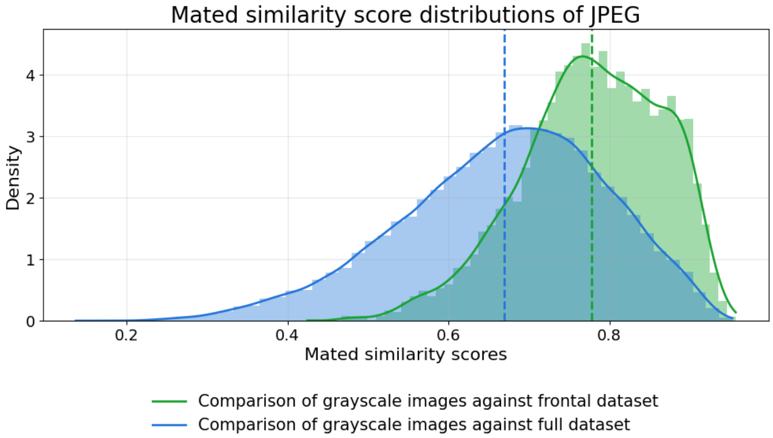


Figure D.65: Mated similarity score distributions of JPEG-compressed grayscale images compared against the frontal dataset and the full dataset.

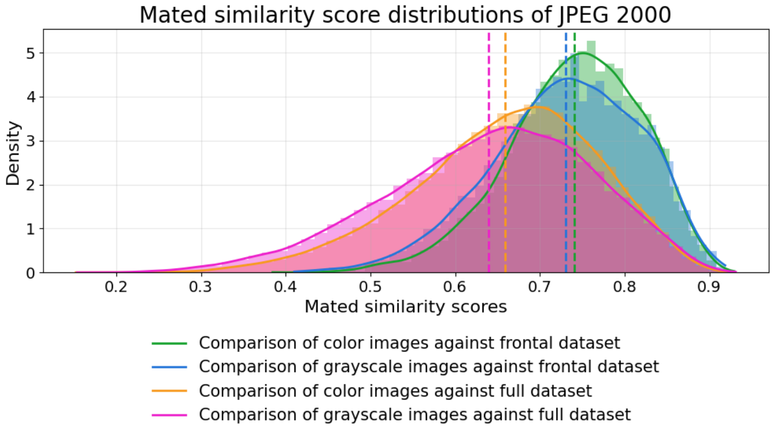


Figure D.66: Mated similarity score distributions of JPEG 2000-compressed color and grayscale images compared against the frontal dataset and the full dataset.

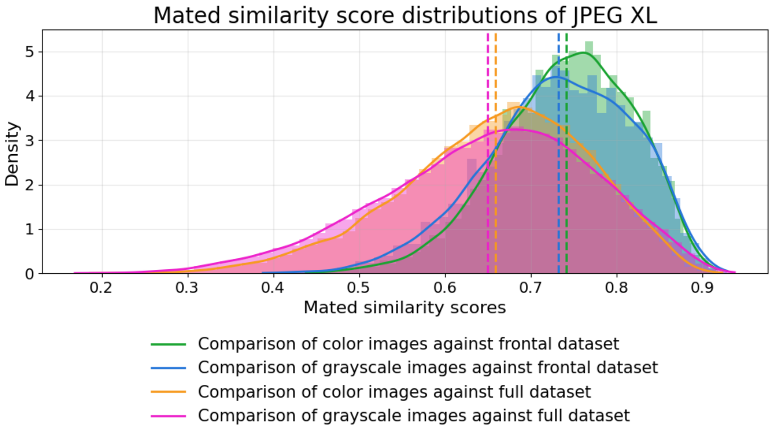


Figure D.67: Mated similarity score distributions of JPEG XL-compressed color and grayscale images compared against the frontal dataset and the full dataset.

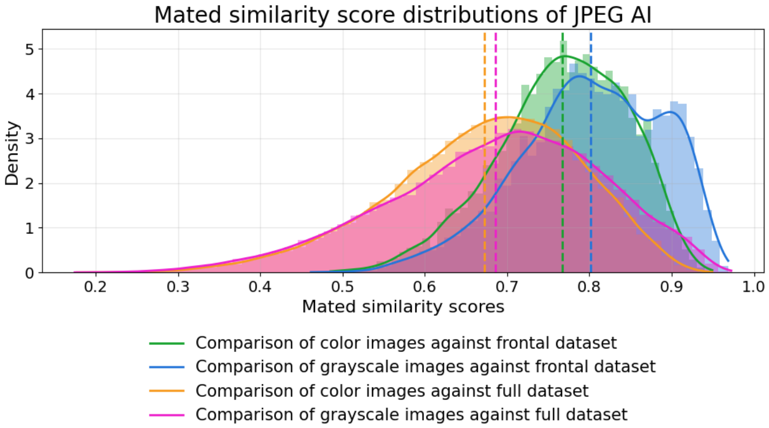


Figure D.68: Mated similarity score distributions of JPEG AI-compressed color and grayscale images compared against the frontal dataset and the full dataset.

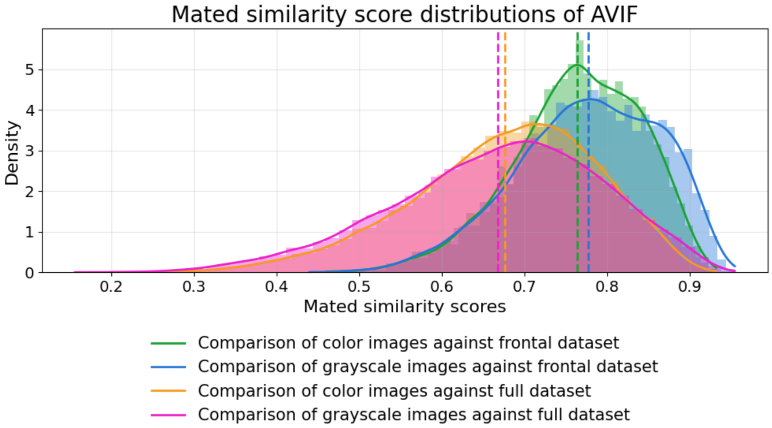


Figure D.69: Mated similarity score distributions of AVIF-compressed color and grayscale images compared against the frontal dataset and the full dataset.

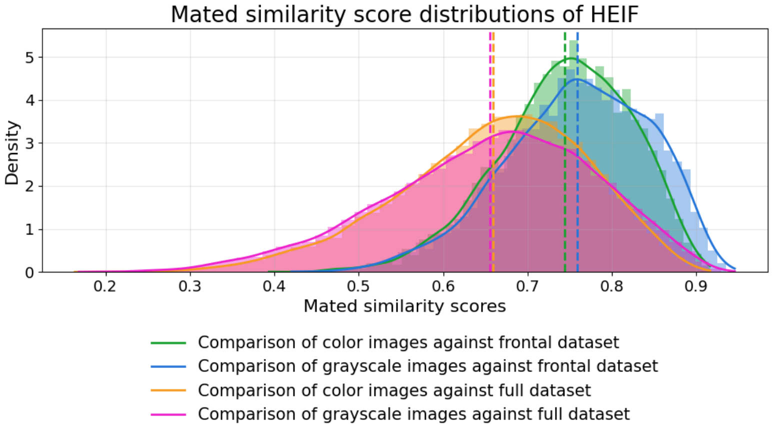


Figure D.70: Mated similarity score distributions of HEIF-compressed color and grayscale images compared against the frontal dataset and the full dataset.

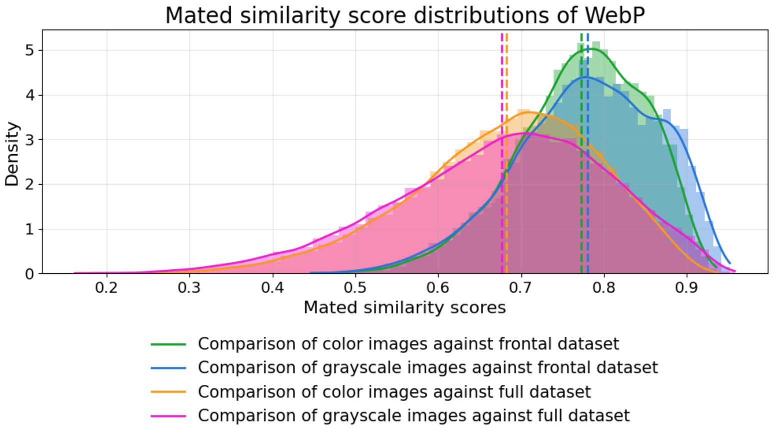


Figure D.71: Mated similarity score distributions of WebP-compressed color and grayscale images compared against the frontal dataset and the full dataset.

E
ERROR VERSUS DISCARD CHARACTERISTIC CURVES

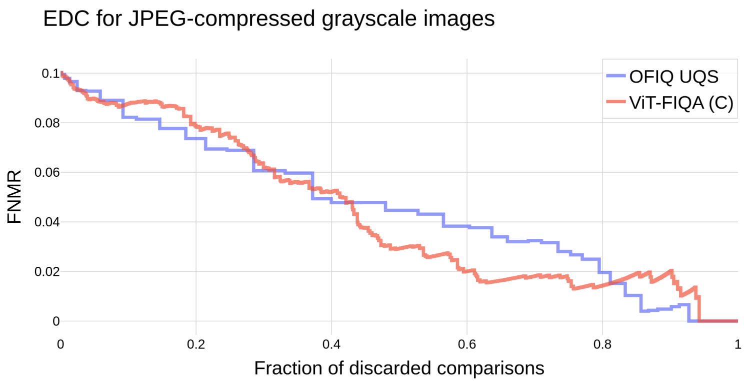


Figure E.1: EDC for JPEG-compressed grayscale images using optimized parameters for frontal comparison.

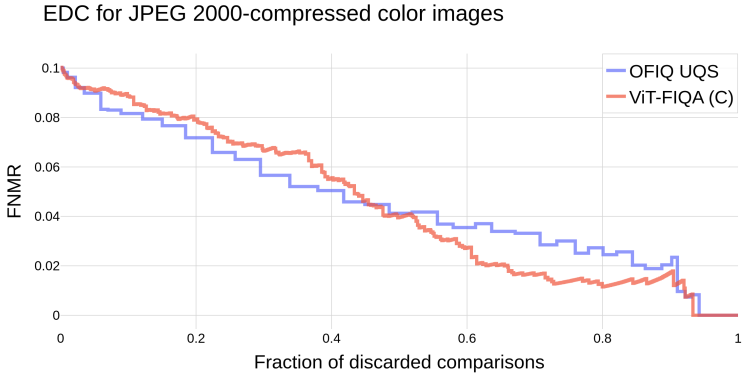


Figure E.2: EDC for JPEG 2000-compressed color images using optimized parameters for frontal comparison.

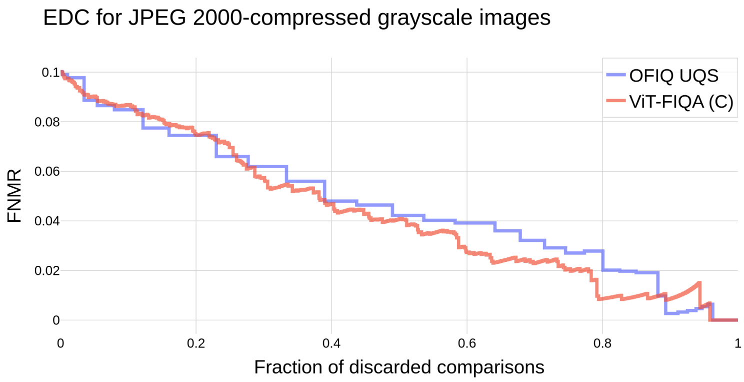


Figure E.3: EDC for JPEG 2000-compressed grayscale images using optimized parameters for frontal comparison.

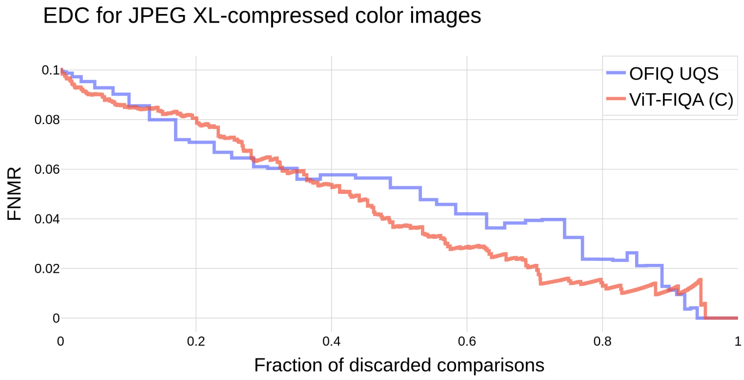


Figure E.4: EDC for JPEG XL-compressed color images using optimized parameters for frontal comparison.

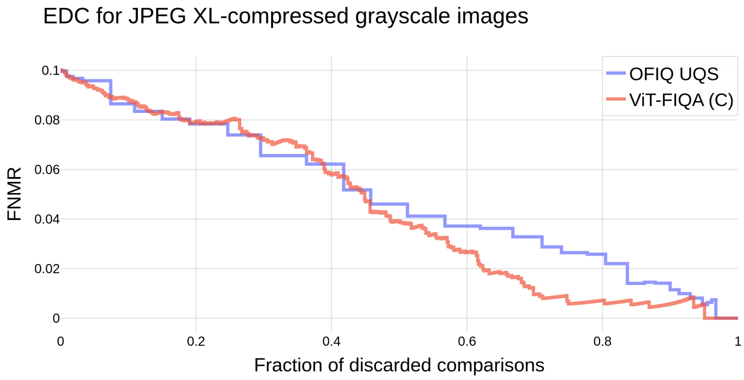


Figure E.5: EDC for JPEG XL-compressed grayscale images using optimized parameters for frontal comparison.

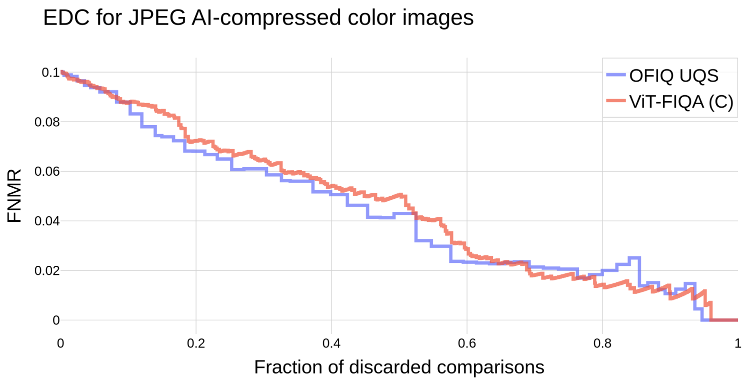


Figure E.6: EDC for JPEG AI-compressed color images using optimized parameters for frontal comparison.

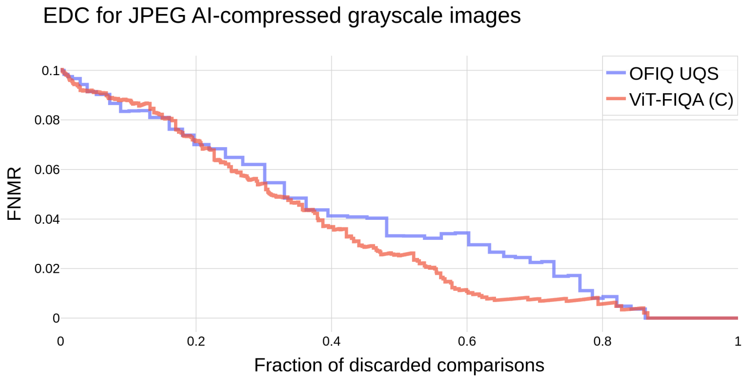


Figure E.7: EDC for JPEG AI-compressed grayscale images using optimized parameters for frontal comparison.

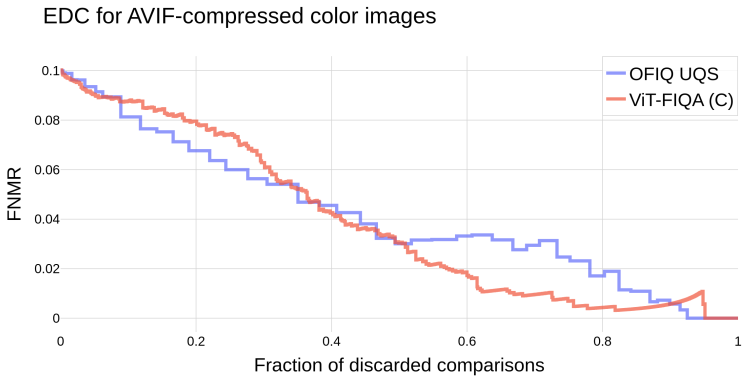


Figure E.8: EDC for AVIF-compressed color images using optimized parameters for frontal comparison.

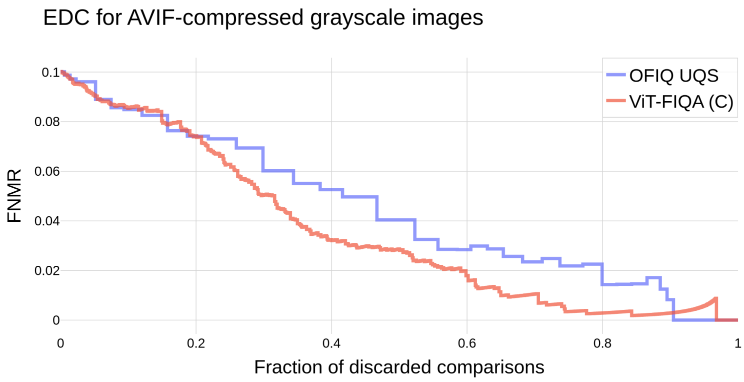


Figure E.9: EDC for AVIF-compressed grayscale images using optimized parameters for frontal comparison.

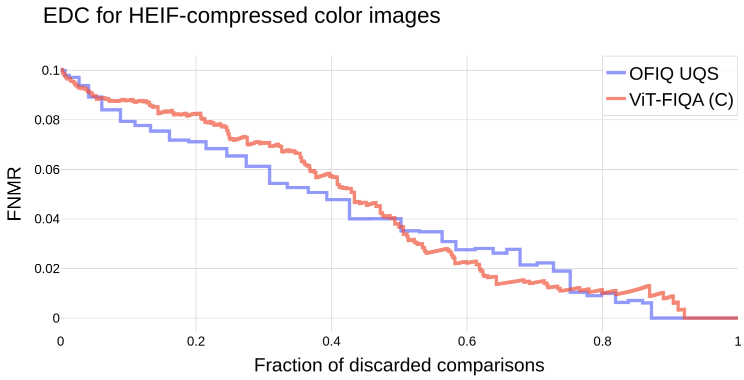


Figure E.10: EDC for HEIF-compressed color images using optimized parameters for frontal comparison.

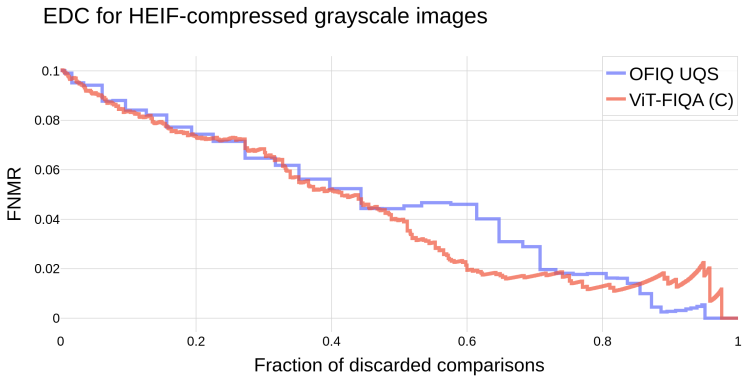


Figure E.11: EDC for HEIF-compressed grayscale images using optimized parameters for frontal comparison.

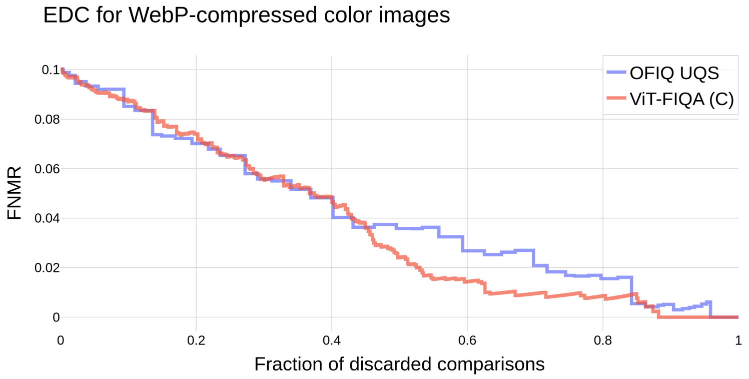


Figure E.12: EDC for WebP-compressed color images using optimized parameters for frontal comparison.

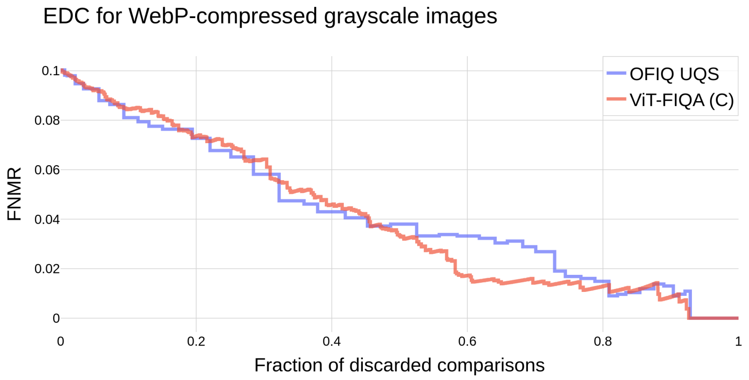


Figure E.13: EDC for WebP-compressed grayscale images using optimized parameters for frontal comparison.

[2994] ISO/IEC JTC 1/SC 29. *ISO/IEC 10918-1:1994 – Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*. International standard specifying core JPEG continuous-tone still image compression and coding; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Feb. 1994. URL: https://www.iso.org/standard/18902.html.

[2924a] ISO/IEC JTC 1/SC 29. *ISO/IEC 15444-1:2024 – Information technology — JPEG 2000 image coding system — Part 1: Core coding system*. International standard specifying the core JPEG 2000 image coding system; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Nov. 2024. URL: https://www.iso.org/standard/87632.html.

[2924b] ISO/IEC JTC 1/SC 29. *ISO/IEC 18181-1:2024 – Information technology — JPEG XL image coding system — Part 1: Core coding system*. International standard specifying the core JPEG XL image coding system; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), July 2024. URL: https://www.iso.org/standard/85066.html.

[2925a] ISO/IEC JTC 1/SC 29. *ISO/IEC 23008-12:2025 – Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format*. International standard specifying an interoperable image file format for single images, collections, and image sequences; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), July 2025. URL: https://www.iso.org/standard/89035.html.

[2925b] ISO/IEC JTC 1/SC 29. *ISO/IEC 23008-2:2025 – Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*. International standard specifying high efficiency video coding (HEVC) as part of the MPEG-H family; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Mar. 2025. URL: https://www.iso.org/standard/90502.html.

[2925c] ISO/IEC JTC 1/SC 29. *ISO/IEC 6048-1:2025 – Information technology — JPEG AI learning-based image coding system — Part 1: Core*

coding system. International standard specifying the core coding system of JPEG AI, a learning-based image coding technology; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Sept. 2025. URL: https://www.iso.org/standard/88911.html.

[3124a] ISO/IEC JTC 1/SC 31. *ISO/IEC 16022:2024 – Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification*. International standard specifying the Data Matrix symbology format and encoding rules; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), May 2024. URL: https://www.iso.org/standard/80926.html.

[3124b] ISO/IEC JTC 1/SC 31. *ISO/IEC 18004:2024 – Information technology — Automatic identification and data capture techniques — QR code bar code symbology specification*. International standard specifying the QR Code symbology characteristics and encoding rules; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Aug. 2024. URL: https://www.iso.org/standard/83389.html.

[3124c] ISO/IEC JTC 1/SC 31. *ISO/IEC 24778:2024 – Information technology — Automatic identification and data capture techniques — Aztec Code bar code symbology specification*. International standard defining the Aztec Code barcode symbology and encoding rules; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Apr. 2024. URL: https://www.iso.org/standard/82441.html.

[3711] ISO/IEC JTC 1/SC 37. *ISO/IEC 19794-5:2011 – Information technology — Biometric data interchange formats — Part 5: Face image data*. International standard specifying biometric face image data formats and photo constraints for biometric systems; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2011. URL: https://www.iso.org/standard/50867.html.

[3719] ISO/IEC JTC 1/SC 37. *ISO/IEC 39794-5:2019 – Information technology — Extensible biometric data interchange formats — Part 5: Face image data*. International standard specifying extensible biometric data interchange formats for face image data, including binary ASN.1 and XML schema formats; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2019. URL: https://www.iso.org/standard/72156.html.

[3722] ISO/IEC JTC 1/SC 37. *ISO/IEC 2382-37:2022 – Information technology — Vocabulary — Part 37: Biometrics*. International standard providing a systematic set of biometric vocabulary terms by ISO/IEC JTC 1/SC 37; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Mar. 2022. URL: https://www.iso.org/standard/73514.html.

[3725] ISO/IEC JTC 1/SC 37. *ISO/IEC 29794-5:2025 – Information technology — Biometric sample quality — Part 5: Face image data*. International standard specifying requirements and methods for quantifying the quality of face image data; accessed February 27, 2026. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Apr. 2025. URL: https://www.iso.org/standard/81005.html.

[AOM25] AOMediaCodec. *libavif: Library for Encoding and Decoding AVIF Image Files*. GitHub repository. Version v1.3.0 (or latest at time of use). 2025. URL: https://github.com/AOMediaCodec/libavif.

[Aki+19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. *Optuna: A Next-generation Hyperparameter Optimization Framework*. 2019. arXiv: 1907.10902 [cs.LG]. URL: https://arxiv.org/abs/1907.10902.

[Ala+23] Jyrki Alakuijala, Jon Sneyers, Luca Versari, and Jan Wassenberg. *JPEG XL Image Coding System: JPEG XL White Paper (Version 2.0)*. https://ds.jpeg.org/whitepapers/jpeg-xl-whitepaper.pdf. ISO/IEC JTC 1/SC 29/WG1 N100400, January 2023. 2023.

[All26a] Alliance for Open Media / AOMedia. *aom: AV1 Codec Library Reference Implementation*. https://aomedia.googlesource.com/aom. Official Git repository for the open-source AV1 codec (libaom) reference implementation; accessed February 27, 2026. 2026.

[All26b] Alliance for Open Media (AOMedia). *AV1 Image File Format (AVIF) Specification*. https://aomedia.org/specifications/avif/. Official specification and overview of the AVIF image format, maintained by AOMedia; accessed February 27, 2026. 2026.

[ABD25] Andrea Atzori, Fadi Boutros, and Naser Damer. *ViT-FIQA: Assessing Face Image Quality using Vision Transformers*. 2025. arXiv: 2508.13957 [cs.CV]. URL: https://arxiv.org/abs/2508.13957.

[BSI26] BSI-OFIQ. *OFIQ: Open Source Face Image Quality Library*. GitHub repository. Version v1.1.1 (or latest at time of use), reference implementation for facial image quality evaluation. 2026. URL: https://github.com/BSI-OFIQ/OFIQ-Project.

[Bou+22] Naima Bousnina, João Ascenso, Paulo L. Correia, and Fernando Pereira. "Impact of Conventional and AI-based Image Coding on AI-based Face Recognition Performance." In: *Proceedings of the 10th European Workshop on Visual Information Processing (EUVIP)*. Examines how conventional and learned image codecs affect the performance of an AI-based face recognition system (ArcFace) with compressed images; accessed February 27, 2026. Lisbon, Portugal, Sept. 2022. URL: https://www.it.pt/Publications/DownloadPaperConference/40369.

[DEN26] DENSO WAVE Inc. *About the patent – QR Code Patents*. https://www.qrcode.com/en/patent.html. Overview of QR-Code licensing and patent situation (QRcode.com), accessed February 27, 2026. 2026.

[Eur16] European Agency for the Management of Operational Cooperation at the External Borders of the Member States of the European Union, ed. *Guidelines for processing of third country nationals through automated border control*. en. Warsaw: FRONTEX, 2016. ISBN: 978-92-95205-51-2. DOI: 10.2819/86138.

[FFm26] FFmpeg Developers. *FFmpeg - libavcodec/libx265.c*. https://github.com/FFmpeg/FFmpeg/blob/master/libavcodec/libx265.c. Source file in the FFmpeg project implementing integration with the x265 encoder; accessed February 27, 2026. 2026.

[Goe15] Michel Goemans. *Shannon's Noiseless Coding Theorem*. https://math.mit.edu/~goemans/18310S15/noiseless-coding.pdf. Lecture notes from *Principles of Discrete Applied Mathematics* (MIT), discussing Shannon's noiseless coding theorem; accessed February 27, 2026. 2015.

[Goo26a] Google Developers. *WebP: A modern image format for the Web*. https://developers.google.com/speed/webp. Official documentation on the WebP image format, covering lossy and lossless compression, transparency, animation, and tools; accessed February 27, 2026. 2026.

[Goo26b] Google Developers. *WebP Compression: Lossy and Lossless Image Coding*. https://developers.google.com/speed/webp/docs/compression. Official documentation describing lossy and lossless compression methods in the WebP image format; accessed February 27, 2026. 2026.

[GNH25] Patrick Grother, Mei Ngan, and Austin Hom. *Face Recognition Technology Evaluation (FRTE): Preparation of Compact Face Images for 2D Barcodes*. en. Sept. 2025. DOI: https://doi.org/10.6028/NIST.SP.500-343. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-343.pdf.

[Guo+21] Jia Guo, Jiankang Deng, Alexandros Lattas, and Stefanos Zafeiriou. *Sample and Computation Redistribution for Efficient Face Detection*. 2021. arXiv: 2105.04714 [cs.CV]. URL: https://arxiv.org/abs/2105.04714.

[Han+21] Jingning Han et al. "A Technical Overview of AV1." In: *Proceedings of the IEEE* 109.9 (2021), pp. 1435–1462. DOI: 10.1109/JPROC.2021.3058584.

[Huf52] David A. Huffman. "A Method for the Construction of Minimum Redundancy Codes." In: *Proceedings of the IRE* 40 (1952). Cited in SCIRP reference database (ReferenceID 1604696); accessed February 27, 2026, pp. 1098–1101. DOI: 10.1109/JRPROC.1952.273898.

[Ica] *ICAO Doc 9303 — Machine Readable Travel Documents*. International standard defining specifications for machine-readable travel documents, including layout, biometrics, and electronic storage; accessed February 27, 2026. International Civil Aviation Organization (ICAO), 2021. URL: https://www.icao.int/publications/doc-series/doc-9303.

[ISO24] ISO/IEC JTC 1/SC29/WG1. *JPEG AI Overview Slides (Document N101029)*. ISO/IEC JTC 1/SC29/WG1 working group presentation. Presentation slides on the overview of JPEG AI; internal document from the ISO/IEC JPEG standardization activity; accessed February 27, 2026. 2024.

[Ins26] InsightFace. *InsightFace: Open Source Deep Face Analysis Library for 2D & 3D Face Recognition and Analysis*. https://www.insightface.ai/. Official website of InsightFace, a state-of-the-art toolbox and ecosystem for face detection, alignment, recognition, and analysis; accessed February 27, 2026. 2026.

[Int21] International Civil Aviation Organization. *Machine Readable Passports – Part 10 – Logical Data Structure (LDS) for Storage of Biometrics and Other Data in the Contactless Integrated Circuit (IC)*. https://www.icao.int/sites/default/files/publications/DocSeries/9303_p10_cons_en.pdf. Last accessed: 2021-11-23. International Civil Aviation Organization (ICAO), 2021.

[JHU22] Ehsaneddin Jalilian, Heinz Hofbauer, and Andreas Uhl. "Iris Image Compression Using Deep Convolutional Neural Networks." In: *Sensors* 22.7 (2022). ISSN: 1424-8220. DOI: 10.3390/s22072698. URL: https://www.mdpi.com/1424-8220/22/7/2698.

[Kim24] Minchul Kim. *CVLFace: High-Performance Face Recognition All-in-One Toolkit*. https://github.com/mk-minchul/CVLface. GitHub repository, MIT License, accessed February 27, 2026. 2024.

[KJL23] Minchul Kim, Anil K. Jain, and Xiaoming Liu. *AdaFace: Quality Adaptive Margin for Face Recognition*. 2023. arXiv: 2204.00964 [cs.CV]. URL: https://arxiv.org/abs/2204.00964.

[Kim+24] Minchul Kim, Yiyang Su, Feng Liu, Anil Jain, and Xiaoming Liu. *KeyPoint Relative Position Encoding for Face Recognition*. 2024. arXiv: 2403.14852 [cs.CV]. URL: https://arxiv.org/abs/2403.14852.

[Mas+19] Babak Maser, Jutta Hämmerle-Uhl, Tamara Lipowski, and Andreas Uhl. "Finger Vein Image Compression with Uniform Background." In: *Proceedings of the 3rd International Conference on Biometric Engineering and Applications (ICBEA)*. Examines effects of replacing finger-vein image backgrounds with uniform gray for improved compression and recognition; accessed February 27, 2026. Stockholm, Sweden: Association for Computing Machinery (ACM), 2019, pp. 23–27. ISBN: 978-1-4503-6305-1. DOI: 10.1145/3345336.3345347.

[Men+21] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. *MagFace: A Universal Representation for Face Recognition and Quality Assessment*. 2021. arXiv: 2103.06627 [cs.CV]. URL: https://arxiv.org/abs/2103.06627.

[Mer+24] J. Merkle et al. *Open Source Face Image Quality (OFIQ) - Implementation and Evaluation of Algorithms*. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/OFIQ/Projektabschlussbericht_OFIQ_1_0.pdf. Last accessed: 2025-02-22. Sept. 2024.

[Pak18] Alexey Pak. *Entropy Coding*. https://ies.iar.kit.edu/ies/download/lehre/idc/IDC-03-Entropy-Coding.pdf. Lecture material on entropy coding from *Image Data Compression* course, IES, Karlsruhe Institute of Technology; accessed February 27, 2026. 2018.

[Phi+00] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. "The FERET Evaluation Methodology for Face Recognition Algorithms." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.10 (2000), pp. 1090–1104.

[Phi+98] P. Jonathon Phillips, Harry Wechsler, J. Huang, and P. J. Rauss. "The FERET database and evaluation procedure for face recognition algorithms." In: *Image and Vision Computing* 16.5 (1998), pp. 295–306.

[San+16] Jose Sanchez del Rio, Daniela Moctezuma, Cristina Conde, Isaac Martin de Diego, and Enrique Cabello. "Automated border control e-gates and facial recognition systems." In: *Computers Security* 62 (2016), pp. 49–72. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2016.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0167404816300736.

[Sch24] Torsten Schlett. *Face Image Quality Toolkit (fiqat)*. https://code.fbi.h-da.de/hda10196/face-image-quality-toolkit. GitLab repository, MIT License; Source: Face Image Quality Toolkit

Project of Hochschule Darmstadt, accessed February 27, 2026. 2024.

[Sch+24] Torsten Schlett, Christian Rathgeb, Juan Tapia, and Christoph Busch. “Considerations on the Evaluation of Biometric Quality Assessment Algorithms.” In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 6.1 (2024), pp. 54–67. DOI: 10.1109/TBIOM.2023.3336513.

[Sch+23] Torsten Schlett, Sebastian Schachner, Christian Rathgeb, Juan Tapia, and Christoph Busch. *Effect of Lossy Compression Algorithms on Face Image Quality and Recognition*. 2023. arXiv: 2302.12593 [cs.CV]. URL: https://arxiv.org/abs/2302.12593.

[Str25] Tilo Strutz. *Bilddatenkompression: Grundlagen, fortgeschrittene Verfahren, Video, Kompressionssysteme, Standards*. Springer Vieweg Wiesbaden, 2025. ISBN: 978-3-658-49923-5. DOI: 10.1007/978-3-658-49923-5. URL: https://link.springer.com/book/10.1007/978-3-658-49923-5.

[U.S26] U.S. General Services Administration / Digital.gov. *Introduction to QR Codes*. https://digital.gov/resources/introduction-to-qr-codes. Introductory resource explaining the functionality and use cases of QR codes; accessed February 27, 2026. 2026.

[Via26] Via Licensing Alliance. *HEVC/VVC Patent Portfolio License*. https://via-la.com/licensing-programs/hevc-vvc/. Overview of the HEVC (H.265) and VVC (H.266) patent licensing program and associated royalty structures; accessed February 27, 2026. 2026.

[VD22] Swetha Vora and Yamuna Devi. *Comparative Analysis of Huffman and Arithmetic Coding Algorithms for Image Compression*. accessed February 27, 2026. 2022. URL: https://www.irjet.net/archives/V9/i9/IRJET-V9I903.pdf.

[WG124] WG1 JPEG Committee. *JPEG AI Reference Software: Reference Implementation for JPEG AI Image Compression*. GitLab repository. Created July 10, 2024, downloaded on October 18th, 2025. 2024. URL: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-reference-software.

[Web25] WebM Project. *libwebp: WebP Image Codec Library for Encoding and Decoding*. Chromium Git repository. Version v1.6.0 (or latest at time of use). 2025. URL: https://chromium.googlesource.com/webm/libwebp.

[lib25] libjpeg-turbo. *libjpeg-turbo: High-Performance JPEG Codec with SIMD Acceleration*. GitHub repository. Version 3.1.1 (or latest at time of use). 2025. URL: https://github.com/libjpeg-turbo/libjpeg-turbo.

[lib26] libjxl. *libjxl: JPEG XL Image Format Reference Implementation*. GitHub repository. Version v0.11.1 (or latest at time of use). 2026. URL: https://github.com/libjxl/libjxl.

[str26] strukturag. *libheif: HEIF and AVIF Image Format Decoder and Encoder*. GitHub repository. Version v1.21.0 (or latest at time of use). 2026. URL: https://github.com/strukturag/libheif.

[ucl25] uclouvain/openjpeg. *OpenJPEG: Open-source JPEG 2000 Codec Library and Tools*. GitHub repository. Version 2.5.4 (or latest at time of use). 2025. URL: https://github.com/uclouvain/openjpeg.

[wik26] wikipedia.org. *Sensitivity index*. https://en.wikipedia.org/wiki/Sensitivity_index. Wikipedia, The Free Encyclopedia. Retrieved February 28, 2026. 2026.