Optimizing Lossy Image Compression for Face Recognition within 1024 Bytes

1st Paul Andreas
Dept. of Computer Science
Hochschule Darmstadt
Darmstadt, Germany
paul.andreas@stud.h-da.de

2nd Torsten Schlett
Dept. of Computer Science - ATHENE
Hochschule Darmstadt
Darmstadt, Germany
torsten.schlett@h-da.de

3rd Christoph Busch
Dept. of Computer Science - ATHENE
Hochschule Darmstadt
Darmstadt, Germany
christoph.busch@h-da.de

***Abstract*—Biometric 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 JPEG2000 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 Data Matrix codes. 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. The results of this research show that AVIF is the most suitable compressor for images when using a resolution of 56x56 pixels and pre-smoothing outside the region of interest.**

***Index Terms*—face recognition, lossy compression, image processing, face image quality**

I. Introduction

Using faces for biometric recognition systems comes with several advantages. Humans are fairly good at face recognition, which makes such systems 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. Therefore face images are often included in identification documents, most of the time in a printed version as well as an electronic version stored on an RFID-chip [1]. Since temporary travel documents are destroyed after only a short period of use, equipping them with an RFID-chip is not resource-efficient. A solution here could be, to use a 2D Data Matrix code instead, which encodes the information of the face image. These codes are also easily machine readable while relying on optic capture, not needing any chips, making them a resource-saving alternative in comparison to RFID-chips. 2D Data Matrix codes come with a major downside in terms of their storage capacities, which are limited to around 1,500 bytes [2], while in comparison the RFID-chip for identification documents offers a capacity of 32 kB. Therefore the face image stored in a 2D barcode should only have a file size of 1024 bytes, instead of 12 kB as in other documents like a machine readable travel document (MRTD) defined by the International Civil Aviation Organisation (ICAO) [1].
There are several methods for reducing the file size of an image:

1) Resolution: By reducing the number of vertical and horizontal pixels, less information needs to be stored, which leads to a reduction in file size. This involves deleting information and can result in coarser textures.
2) Entropy encoding: This method tries to find more efficient ways to store the information in a lossless manner. The aim is to use less bits for sequences appearing more frequently and in exchange using more for sequences that appear less frequently.
3) Optimize on human vision: Since images are supposed to be visually processed by humans, another common mechanism is to optimize the colors and textures for the human perception. This involves exploiting various biological characteristics of the human eye, such as its enhanced perception of differences in brightness or shades of green.
4) Deleting other information: Another approach is the targeted deletion of information. This can involve metadata, for example, choosing a different image format with less overhead, as well as converting the image to grayscale, which removes the color information so that only the contrast information remains.

Most image compression algorithms use the second, the third, and the fourth mechanism to reduce file sizes. When used with caution those changes are nearly unnoticeable for human eyes. In order to support these algorithms, one can prepare images, for example by applying a mean kernel on regions without much interest, because most compression algorithms value regions of high contrast higher, due to human perception.

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

II. Related work

Grother et al. [3] propose in NIST SP 500-343 a method for compressing face images and encoding them in 2D barcodes. As in this study, the authors use both a reduction in resolution and a series of compression algorithms to reduce the images to a predetermined file size. One aspect that is not addressed in the study is the use of different settings for the compression algorithms and the manipulation of the images to tap into further potential for reducing the file size. NIST SP 500-343 shows that reducing the resolution has less of a negative impact on the resulting face recognition performance than the use of lossy compression. WebP and AVIF prove to be the most suitable formats.

Funk et al. [4] and recently Schlett et al. [5] analyzed the effect of lossy compression on face image quality and face recognition performance. To this end, they examine the effects of simply reducing the resolution or using various lossy compression algorithms from the JPEG family on the mated comparison scores and face image quality scores for various target file sizes above 2kB. As a result of their investigations, the authors of [5] recommend using JPEG XL for compression. They also showed that modern Face Image Quality Assessment (FIQA) algorithms are capable of detecting the loss of image quality caused by lossy compression.

"Iris Image Compression Using Deep Convolutional Neural Networks" by Jalilian et al. [6] investigated the suitability of machine learning for compressing biometric iris images. This form of image compression is a still young but promising approach to achieving higher compression with better image quality compared to conventional images. The study compares the effectiveness of traditional compression algorithms with those of Deep Semantic Segmentation-based Layered Image Compression (DSSLIC) on iris images. The study concludes that this new type of image compression delivers better results than conventional approaches, demonstrating that this new approach of lossy compression can also be applied to specific fields such as biometric data.

III. Experimental setup

In order to find an optimal configuration to reduce the image file sized to the desired amount of at most 1024 bytes, seven different compression algorithms were evaluated, namely JPEG, JPEG 2000, JPEG XL, JPEG AI, HEIF using HEVC, AVIF using the AV1 codec, and WebP. First different parameters for each algorithm are tested. After selecting the best compression parameters, different resolutions are tested using the best parameters. Finally, for the best result of compression parameters and resolution per algorithm, further image manipulations are tested to support the compression process. Afterwards, the effect of the compression on the face recognition performance is evaluated using the sensitivity index on resulting mated and non-mated score distributions from AdaFace. In addition the face recognition performance in terms of the False Non-Match Rate (FNMR) at a False Match Rate (FMR) of either 0.1% or 0.01% is reported.

A. Dataset

The ColorFERET dataset is used for the this work, as it contains uncompressed face images. [7] [8]. For the compression parameter optimization, only a small subset of ten frontal face images is used, due to limitations in resources. This is referred to in this paper as "minimal dataset". Subsequently the resulting compression configurations are applied to a larger subset of the dataset, which contains every frontal face image ("frontal dataset") resulting in 2638 images in total. Images with variations in pose and rotation will not be included in the compression process, since the compressed images in application scenarios are assumed to be face portrait images as required international standards such as ISO/IEC 19794-5 [9] and ISO/IEC 39794-5 [10].

B. Software

The command-line interface software offered in the repositories of the respective algorithms are used to compress the images. These include: libjpeg-turbo v3.1.1 [11], openjpeg v2.5.4 [12], libjxl v0.11.1 [13], jpeg-ai-reference-software [14], libwebp v1.6.0 [15], libheif v1.21.0 [16] with H.265 v3.5, and libavif v1.3.0 [17] with AOM v3.13.1.

In addition to the compression algorithms, the Open Face Image Quality (OFIQ) software library [18], [19] was used for the extraction of facial landmarks and unified quality scores.

For face recognition we used an openly available model from the CVLface library [20] with ViT-KPRPE architecture [21] trained with AdaFace [22] loss on WebFace12M, called "AdaFace" herein. In addition to OFIQ's unified quality score model, ViT-FIQA (C) [23] was also used in the face image quality assessment part of the evaluation.

C. Image preparation

To estimate the effect on the face recognition performance, the similarity score of the image before and after compression is computed. The idea behind this "self-similarity score" is that it expresses how far the compressed image has deviated from its original, whether due to information loss or compression artifacts. The goal is to keep this discrepancy as small as possible, so the experiments aim to keep the "self-similarity score" as high as possible.

In order for AdaFace to process the images, all images used must first be cropped to a size of 112x112 pixels, whereby five facial landmarks are used to align the image (two for the eyes, two for the mouth corners, one for the tip of the nose). This is already a first step towards achieving the target file size, because information in the image that is not relevant for face recognition is cropped out.

The next step is to generate a grayscale version of each color image, which inherently reduces the memory requirements prior to compression. All subsequent operations are now performed on both the color images and the grayscale images.

D. Compression parameter optimization

Now the compression parameter optimization can be performed. To do this, the first step is always to manually set

exactly one parameter for the respective compression algorithm and iteratively determine the compression rate that best meets the requirement of a target file size below or equal to 1024 bytes. This procedure is performed for the entire minimal dataset. Next, the average "self-similarity score" is determined to see whether the parameter has a positive, negative, or no effect on the result. After this has been done for all parameters, the Python library Optuna [24] is used to define a grid search on the parameters with positive effects, which determines an optimized combination of these parameters using the same procedure.
After finishing, for each compression algorithm a combination of improving parameters is left. In the next step the images used for compression are resized to the following resolutions: 56x56, 64x64, 75x75, 80x80, 96x96, 160x160, 180x180, 200x200, 224x224. For resolutions below 112 this is performed on the preprocessed data, for resolutions above, two versions are created, one upscaled and one using the original ColorFERET images and cropping and aligning them directly to the named resolutions. Resolutions of more than 112x112 are included, since JPEG AI needs a minimal resolution of 160x160. In theory upscaled images should be enough, since AdaFace later only uses a resolution of 112x112 and the further information will be discarded either way, but for the sake of completeness both versions are tested.
The entire process for this is shown in Fig. 1.

After determining the best resolutions for each algorithm the last optimization step is image manipulation. Therefore different manipulation strategies are evaluated (see Fig. 2):

- Removing the background: By fitting a rectangle around the region of eyes, nose, and the corners of the mouth with some additional padding (10% of the image resolution in every direction), the region of interest is defined. This includes the most important aspects used for face recognition. Everything outside the area is blacked out (a) or whitened out (b). This could benefit compression, since this area now consists of only one solid color. Of course this method deletes all information gained from hair, ears and background.
- Blurring: Since compression algorithms are designed to favor human perception, they tend to use more storage capacities for edges, which define contrast. By using a mean filter, these edges become less sharp, and may become easier to compress. Blur is always applied with a mean 3x3 kernel, but in three different ways. Full blur is applied to the entire image (c), rectangle blur is applied only to the area outside the region of interest defined above around the five facial landmarks (d), OFIQ-landmark-region-based blur uses the OFIQ software library for selecting a more accurate estimation of the relevant face area as landmarked region of interest and then smoothens the rest of the image (e).
- Low-pass filter: For this preparation, the images are converted into frequency domain and the top 10% of high frequencies are discarded. This has a similar effect to blurring, because high frequencies correspond to edges. This method is also used in some compression algorithms. This preprocessing is performed only for grayscale images and is applied on full images (f) and the outside of the rectangle defined earlier (g).

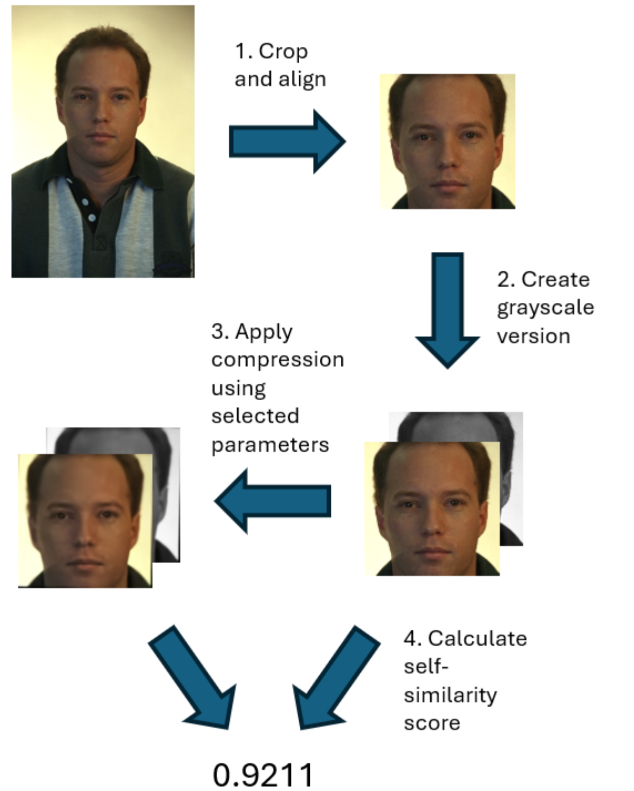

Fig. 1. Graphical representation of the parameter optimization process. The "self-similarity score" is always calculated using the compressed image (varying resolution) and the cropped and aligned version (resolution of 112x112 pixels) of the same color space.

The original image is included for comparison (h).

The search for optimized image manipulation is again performed by an Optuna grid-search for the seven different manipulations described and a nonmanipulated image resulting in eight possible settings in total for this step. The program code used for all operations can be found at https://github.com/dasec/1kB-FaceImage.

IV. Results

After determining the optimal settings for each of the seven compression algorithms for processing color and grayscale images based on the minimal dataset, these settings are applied to the dataset containing frontal face portrait images. All similarity scores are then calculated using the full ColorFERET dataset. This mimics an application scenario, in which the compressed frontal face images represent the biometric reference of an identification document, while the other images serve as the uncompressed biometric probes with varying poses. This procedure results in 2,638 self-similarity scores,

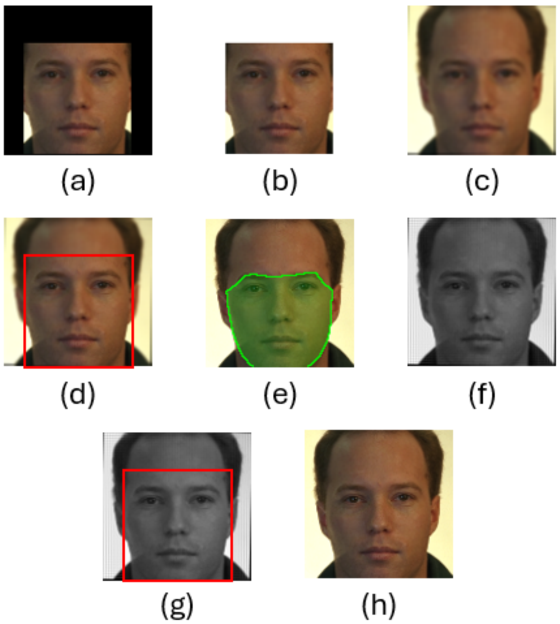

Fig. 2. Resulting images after image manipulation.

Fig. 3. These two images are used to calculate the "self-similarity score". On the left is the original face image, on the right is the AVIF-compressed version after optimized preprocessing.

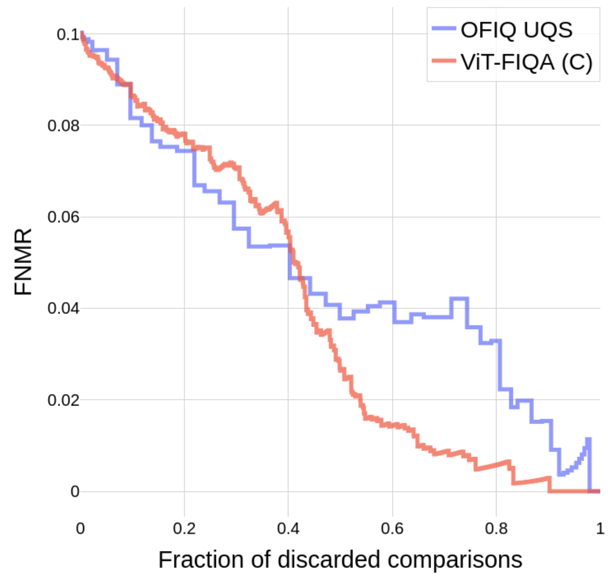

Fig. 4. Error versus Discard Characteristic curves of the AVIF-compressed color images. The comparison scores used for this evaluation were only calculated on the compressed images and not the original versions. The steeper the curve, the better the FIQA algorithm works on the data.

39,857 mated similarity scores, and 29,301,902 non-mated similarity scores.

Four different metrics are then used for evaluation, including the average "self-similarity score", the sensitivity index calculated from mated similarity score distribution and the non-mated similarity score distribution, FNMR when setting the threshold to the 99.9 percentile of non-mated similarity scores, which is in line with the recommendation of the European agency FRONTEX [25]. For FIQA the OFIQ Unified Quality Score (QFIQ UQS) and quality scores calculated by ViT-FIQA (C) were used in Error versus Discard Characteristic (EDC) curves.

Table I provides an overview of the results of the best settings for the most important metrics.

A. Best performing compression algorithm

As can be seen in Table I, AVIF delivers the best results in terms of face recognition performance on the color dataset of all algorithms and settings tested here. This result is consistent with the findings of Grother et al. [3]. In their study, they also conclude that AVIF is the most suitable compression algorithm for reducing face images to such small file sizes.

The AVIF-compressed images do not generate any false non-matches at an FMR of 0.1% and thus deliver even better results than the uncompressed images for this value. At an FMR of 0.01%, however, they do not perform better than the original images, but still beat all other compression algorithms examined. A look at the other metrics shows that AVIF does not achieve the best values here. Although AVIF's results for "self-similarity score" and sensitivity index are at a high level, they are surpassed by WebP. However, since the performance of a face recognition system is better represented by the FNMR results, AVIF is still the recommendation of this study. Fig. 3 provides a visual impression of the effects of AVIF compression on the input images. It shows the cropped and aligned image and the result image after compression. The settings used to achieve this result are listed in table II.

Further evidence that AVIF-compressed images are still suitable for face recognition is provided by the EDC curves derived from the quality scores of the two examined unified FIQA models. ViT-FIQA (C) in particular shows that quality assessment continues to work even on compressed images. OFIQ UQS, on the other hand, has problems correctly assessing quality at a discard rate of approximately 50% or higher, as can be seen in Fig. 4.

B. General findings

Throughout the experiments a few things stood out: the "self-similarity score" used as a metric for the quality of the compressed facial images only provides a limited degree of reliability. As can be seen in Table I, the grayscale images generate significantly better scores than the color

TABLE I
RESULTS OVERVIEW
sorted by Color FNMR at FMR 0.1% ascending then by Color FNMR at FMR 0.01% ascending

Compression algorithm	Color				Grayscale			
	Self-similarity score ↑	Sensitivity index ↑	FNMR in % at FMR 0.1% ↓	FNMR in % at FMR 0.01% ↓	Self-similarity score ↑	Sensitivity index ↑	FNMR in % at FMR 0.1% ↓	FNMR in % at FMR 0.01% ↓
no compression (baseline)	1.0000	7.6589	0.0025	0.0100	1.0000	6.7757	0.0050	0.0502
AVIF	0.8862	7.5987	0.0000	0.0151	0.9156	6.7815	0.0050	0.0828
JPEG XL	0.8582	7.5309	0.0000	0.0252	0.8761	6.7346	0.0075	0.1179
WebP	0.8902	7.6150	0.0025	0.0176	0.9312	6.7764	0.0075	0.0602
JPEG AI	0.8963	7.3949	0.0025	0.0201	0.9484	6.7852	0.0075	0.0753
JPEG 2000	0.8553	7.5513	0.0025	0.0251	0.8689	6.7032	0.0050	0.1179
HEIF	0.8596	7.4056	0.0025	0.0276	0.8941	6.7341	0.0151	0.1104
JPEG	n.a.	n.a.	n.a.	n.a.	0.9148	6.7302	0.0075	0.0878

TABLE II
FINAL COMPRESSION SETTINGS FOR AVIF

Parameter	Value
Color or grayscale	Color
YUV format	420
Speed	1
Resolution	56x56
Image manipulation	Rectangle blur

images. Looking at the sensitivity index and FNMR, it can be seen that, although a trend can be identified, the face recognition performance does not exclusively depend on the "self-similarity score". It should also be noted that, due to resource constraints, the parameters were optimized based on only 10 images, which is a relatively small set of data. For this reason the best three configurations were tested for each algorithm. In most cases the configurations found by optimizing for the highest "self-similarity scores" did not change.

Furthermore, smoothing the images before compression turned out to be an effective way to increase face recognition performance. Surprisingly the rectangle blur was often more effective than the OFIQ-landmark-region-based blur, although the latter fits to the geometry of the face much better, thus blurring more of the actual background area.

During parameter optimization, grayscale conversion showed a significant improvement in the "self-similarity scores." However, as the subsequent investigation of the effects on face recognition performance has shown, this step is not suitable for reducing file size. Instead grayscale images increased the error rate drastically.

Another unexpected finding is the interaction between resolution and compression. The initial assumption here was that these two methods of reducing file size are related to each other, so that an initial reduction of the resolution yields better results, up to a tipping point at which the compression rate becomes so low that the compression artifacts cause less loss of quality than the information lost by simply downscaling. In such a scenario, the course of the "self-similarity scores" should have taken the form of a downward-opening parabola. This is not the case; instead, the function zigzags. It follows that face recognition can handle certain resolutions better than others. No pattern has been noticed here.

V. CONCLUSION

The results of the study show that, with the exception of JPEG, the compression algorithms used here achieve the required target file size of 1024 B without exceeding FRONTEX's requirement of an FNMR below 5% at an FMR of 0.1%, at least for the examined dataset.

The recommended algorithm is AVIF, as it delivered the best face recognition performance. Regarding the comparatively unusual autoencoder approach of JPEG AI, the tested implementation was inferior to the best of the more conventional compression algorithms. It is also evident that downscaling the resolution is an appropriate measure to reduce file size, while converting the images to grayscale is not recommended. A further positive effect can be achieved by applying slight smoothing outside the relevant facial features.

ACKNOWLEDGMENT

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

REFERENCES

[1] 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, International Civil Aviation Organization (ICAO), 2021, last accessed: 2021-11-23.

[2] ISO/IEC, "ISO/IEC 16022:2024, information technology – automatic identification and data capture techniques – data matrix bar code symbology specification," International Organization for Standardization / International Electrotechnical Commission, 2024, international Standard.

[3] P. Grother, M. Ngan, and A. Hom, "Face Recognition Technology Evaluation (FRTE): Preparation of Compact Face Images for 2D Barcodes," Sep. 2025. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-343.pdf

[4] W. Funk, M. Arnold, C. Busch, and A. Munde, "Evaluation of image compression algorithms for fingerprint and face recognition," in *Proc. IEEE Information Assurance Workshop*. IEEE Press, June 2005.

[5] T. Schlett, S. Schachner, C. Rathgeb, J. Tapia, and C. Busch, "Effect of lossy compression algorithms on face image quality and recognition," 2023. [Online]. Available: https://arxiv.org/abs/2302.12593

[6] E. Jalilian, H. Hofbauer, and A. Uhl, "Iris image compression using deep convolutional neural networks," *Sensors*, vol. 22, no. 7, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/7/2698

[7] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[8] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[9] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 19794-5:2005. Information Technology - Biometric Data Interchange Formats - Part 5: Face Image Data*, International Organization for Standardization, June 2005.

[10] ——, *ISO/IEC 39794-5:2019 Information technology - Extensible biometric data interchange formats - Part 5: Face image data*, International Organization for Standardization, 2019.

[11] libjpeg-turbo, "libjpeg-turbo: High-performance jpeg codec with simd acceleration," GitHub repository, 2025, version 3.1.1 (or latest at time of use). [Online]. Available: https://github.com/libjpeg-turbo/libjpeg-turbo

[12] uclouvain/openjpeg, "OpenJPEG: Open-source jpeg 2000 codec library and tools," GitHub repository, 2025, version 2.5.4 (or latest at time of use). [Online]. Available: https://github.com/uclouvain/openjpeg

[13] libjxl, "libjxl: Jpeg XL image format reference implementation," GitHub repository, 2026, version v0.11.1 (or latest at time of use). [Online]. Available: https://github.com/libjxl/libjxl

[14] WG1 JPEG Committee, "JPEG AI Reference Software: Reference implementation for jpeg ai image compression," GitLab repository, 2024, created July 10, 2024, downloaded on October 18th, 2025. [Online]. Available: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-reference-software

[15] WebM Project, "libwebp: Webp image codec library for encoding and decoding," Chromium Git repository, 2025, version v1.6.0 (or latest at time of use). [Online]. Available: https://chromium.googlesource.com/webm/libwebp

[16] strukturag, "libheif: Heif and avif image format decoder and encoder," GitHub repository, 2026, version v1.21.0 (or latest at time of use). [Online]. Available: https://github.com/strukturag/libheif

[17] AOMediaCodec, "libavif: Library for encoding and decoding avif image files," GitHub repository, 2025, version v1.3.0 (or latest at time of use). [Online]. Available: https://github.com/AOMediaCodec/libavif

[18] BSI-OFIQ, "OFIQ: Open source face image quality library," GitHub repository, 2026, version v1.1.1 (or latest at time of use), reference implementation for facial image quality evaluation. [Online]. Available: https://github.com/BSI-OFIQ/OFIQ-Project

[19] J. Merkle, C. Rathgeb, B. Herdeanu, B. Tams, D. Lou, A. Dörsch, M. Schaubert, J. Dehen, L. Chen, X. Yin, D. Huang, A. Stratmann, M. Ginzler, M. Grimmer, and C. Busch, "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, September 2024, last accessed: 2025-02-22.

[20] mk-minchul, "CVLFace: High-performance face recognition all-in-one toolkit," GitHub repository, 2025, mIT License. [Online]. Available: https://github.com/mk-minchul/CVLface

[21] M. Kim, Y. Su, F. Liu, A. Jain, and X. Liu, "Keypoint relative position encoding for face recognition," 2024. [Online]. Available: https://arxiv.org/abs/2403.14852

[22] M. Kim, A. K. Jain, and X. Liu, "AdaFace: Quality adaptive margin for face recognition," 2023. [Online]. Available: https://arxiv.org/abs/2204.00964

[23] A. Atzori, F. Boutros, and N. Damer, "ViT-FIQA: Assessing face image quality using vision transformers," 2025. [Online]. Available: https://arxiv.org/abs/2508.13957

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," 2019. [Online]. Available: https://arxiv.org/abs/1907.10902

[25] 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*. Warsaw: FRONTEX, 2016.