

Article

# Impact of Image Compression on In Vitro Cell Migration Analysis

Ehsaneddin Jalilian <sup>\*</sup>, Michael Linortner  and Andreas Uhl 

Artificial Intelligence and Human Interfaces (AIHI) Department, University of Salzburg, 5020 Salzburg, Austria; uhl@cs.sbg.ac.at (A.U.)

\* Correspondence: ejalilian@cs.sbg.ac.at; Tel.: +43-662-80446336

**Abstract:** Collective cell movement is an indication of phenomena such as wound healing, embryonic morphogenesis, cancer invasion, and metastasis. Wound healing is a complicated cellular and biochemical procedure in which skin cells migrate from the wound boundaries into the wound area to reconstruct the injured skin layer(s). In vitro analysis of cell migration is an effective assay for measuring changes in cell migratory complement in response to experimental inspections. Open-source segmentation software (e.g., an ImageJ plug-in) is available to analyze images of in vitro scratch wound healing assays; however, often, these tools are error-prone when applied to, e.g., low-contrast, out-of-focus, and noisy images, and require manual tuning of various parameters, which is imprecise, tedious, and time-consuming. We propose two algorithmic methods (namely log gradient segmentation and entropy filter segmentation) for cell segmentation and the subsequent measurement of the collective cell migration in the corresponding microscopic imagery. We further investigate the effects of image compression on the algorithms' measurement accuracy, applying lossy compression algorithms (the current ISO standards JPEG2000, JPEG, JPEG-XL and AV1, BPG, and WEBP). We aim to identify the most suitable compression algorithm that can be used for this purpose, relating rate–distortion performance as measured in terms of peak signal-to-noise ratio (PSNR) and the multiscale structural similarity index (MS-SSIM) to the segmentation accuracy obtained by the segmentation algorithms. The experimental results show that the log gradient segmentation algorithm provides robust performance for segmenting the wound area, whereas the entropy filter segmentation algorithm is unstable for this purpose under certain circumstances. Additionally, the best-suited compression strategy is observed to be dependent on (i) the segmentation algorithm used and (ii) the actual data sequence being processed.



**Citation:** Jalilian, E.; Linortner, M.; Uhl, A. Impact of Image Compression on In Vitro Cell Migration Analysis. *Computers* **2023**, *12*, 98. <https://doi.org/10.3390/computers12050098>

Academic Editor: Lucia Maddalena

Received: 24 February 2023

Revised: 29 April 2023

Accepted: 30 April 2023

Published: 4 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** in vitro cell migration analysis; image compression; wound healing assay

## 1. Introduction

Image compression, in particular the lossy variant, is an important vision system enabler, especially in environments with limited resources. Its application is limited by compression artifacts, which potentially interfere with human perception or automated image analysis and vision technology. Therefore, the assessment of the applicability of a certain compression technique in a particular application context is important. Image compression algorithms are classically assessed with respect to human perception using the mean opinion score (MOS) or similar subjective image quality metrics (IQMs) to rate generic image quality. This approach has been chosen, e.g., when designing the JPEG default quantization (Q-) table, as this involves thousands of voluntary viewers. A second assessment option is to consider numerical rate-distortion criteria (e.g., employing *objective* IQMs such as PSNR or SSIM, as also performed for the JPEG Q-table optimization [1]). The major aim of such objective IQMs is to predict subjective image quality as reliably as possible. However, for applications in, e.g., pattern recognition (or any other automated image analysis or vision task), it is not obvious if compression algorithms exhibiting better

subjective or objective IQM are also better in a particular pattern recognition context. Compression algorithms tuned for applications in the pattern recognition context have thus been proposed, which are, e.g., based on the modification of the standard JPEG compression algorithm emphasizing middle and high frequencies and discarding low frequencies (the standard JPEG quantization matrix is rotated by 180 degrees) [2].

For these reasons, a compression assessment with respect to a specific application context may be desirable instead of relying on general-purpose IQMs. This can again be conducted in a subjective or an objective manner, e.g., in a medical context, in [3] two ophthalmologists graded jointly compressed fundus imagery in subjective assessment at a one month time interval for soft exudates, hard exudates, macular oedema, new vessels, intraretinal microvascular abnormalities (IRMAs), and retinal haemorrhages and/or microaneurysms. An example for objective assessment is the application of a computer-assisted diagnosis (CAD) system to compressed data [4] to grade colon mucosa under compressed high-magnification endoscopy. While the literature on general-purpose IQM is abundant, application-specific compression assessment can be tedious work. Thus, IQMs are often applied despite their often questionable evidence for certain applications. However, in some application areas, specific compression assessment and optimization are being considered, as reviewed in the following.

### 1.1. Biometrics

Apart from biometric sample compression standardization in ISO/IEC 19794-7:2021-“Biometric data interchange formats”, many research-oriented studies exist that dealt with the assessment of sample data compression and the impact on biometric recognition performance, e.g., for iris recognition [5–10], fingerprint recognition [11–14], vein recognition [15–18], and 3D face recognition [19]. Additionally for face image compression and its impact on face recognition performance, a significant corpus of research exists [20], considering, e.g., the impact of JPEG [21,22], JPEG 2000 [23,24], SPIHT [25], and H.264 [26]. In addition, face detection has been subject to similar investigations, see, e.g., [27] for JPEG and [28] for JPEG, JPEG2000, and JPEG XR. Deep-learning-based image compression has been successfully applied to iris sample data [29,30]. The need for such specific investigations has been demonstrated as rate-distortion performance can represent a poor predictor of biometric performance (recognition accuracy). In particular, for conventional iris segmentation techniques, the use of JPEG compression, which creates clear edge artefacts that assist iris texture boundary localization, was found to produce superior results compared with JPEG 2000 [31], although in a rate-distortion and perceptual sense, JPEG 2000 is clearly superior. The optimization of compression algorithms to meet the specific properties of the data to be compressed is a natural strategy. JPEG quantization matrix optimization was already considered in biometrics: a rate/distortion criterion was employed in the context of face recognition [21], and superior recognition performance was achieved compared with that of the standard matrix. We have designed optimized JPEG matrices for fingerprints [32] as well as for iris data compression [33–35], leading to recognition performance improvements as well. A further example is the optimization of JPEG 2000 Part 2 wavelet packet decomposition structures with respect to optimizing iris [36] and fingerprint [37] recognition accuracy, which provided better results than rate-distortion optimized wavelet packet structures. Even JPEG XR was optimized for the iris recognition context [38].

### 1.2. Medical Image Analysis

For medical image data, the application-specific compression assessment strategy has been employed as well. For example, refs. [39,40] compared different compression schemes for MRI based on objective image quality measures; a position paper by the European Society of Radiology [41] compared JPEG, JPEG 2000, and JPEG-LS applied to various typical medical data using PSNR; while [42] determined the perceptual quality of laparoscopic video after compression, based on medical experts scores. However, an assessment with respect to the impact on the actual diagnostic aim of the acquired imagery is more

beneficial and usually drastically increases the acceptance of such techniques among medical personnel [43]. For example, [44] investigated the effect of image compression and scaling on the automated scoring of immunohistochemical stainings and segmentation of tumor epithelium, while [45] studied the effects of MR image compression on tissue classification quality. The impact of compression on the detection of diabetic retinopathy by two ophthalmologists was assessed [3]. The effect of lossy compression techniques on texture classification and CBIR schemes as used in CAD support systems, i.e., computer-assisted tumor staging in colonoscopy, was analyzed in [4,46].

### 1.3. Other Areas

The impact of (JPEG) compression on general CBIR accuracy has been investigated [47–49], while specific (medical) image retrieval- focusing on high-magnification colonoscopy was addressed [46] considering more variety in the compression schemes. Similar investigations have been conducted in remote sensing [50–52] as well as in image forensics, such as for image age determination [53], copy-move detection, and PRNU-based device identification [54]. In microscopy imaging, being close to the particular scenario we address in this paper, ref. [55] quantified the effect of image compression on supervised learning applications in optical microscopy; ref. [56] also looked at the corresponding effects on deep-learning-based digital pathology image analysis. The role of image compression (JPEG and JPEG200) on the automated quantification of biomarkers in biological tissue samples has been assessed as well in a couple of other publications [57,58]. Some other researchers investigated the effect of image compression on the analysis of different human biological constituents such as blood cells. For example, in [59], the authors investigated the impact of image compression on the classification of red blood cell images during malaria infection using deep learning. Pomarico et al. [60] quantified the statistical distortions induced by compressing images of human neural stem cells and investigated the effect of the compression on outcomes of the cell segmentations. They reported considerable segmentation distortions when applying JPEG compression, specifically at higher compression ratios.

To the best of our knowledge, none of the previous studies have directly investigated the effect of lossy image compression on the *in vitro* cell analysis (segmentation) of wound healing assays, as we conducted in this study.

In this study, we developed two algorithms for the primary task of wound area detection and segmentation, which is the key step in *in vitro* cell migration (wound healing) assay analysis. We further investigated the impact of several (lossy and nearly lossless) image compression techniques on the cell images and the subsequent performance behavior of the segmentation algorithms. In particular, in addition to the standard IQMs such as PSNR or MS-SSIM, we aimed to understand if the relevant information of the cell structure is preserved with regard to computer-aided cell analysis processes.

## 2. Wound Area Segmentation

Wound healing assays are used to study cell migration processes in *in vitro* laboratory environments. Cells are grown in a container, also called a “well”, until they form a confluent monolayer. A scratch, which is a cell-free gap, is made into the monolayer, which mimics a wound. Therefore, these assays are also called scratch assays. After applying the scratch, the cells start to migrate into the gap until it is eventually closed. This allows modeling and studying wound healing processes under laboratory conditions. It is a well-established, cost-effective, and standard technique used to study collective cell migration in two-dimensional space. To measure the cell migration process, microscopic images are taken at regular time intervals, which may be further processed and used to automate the measuring process using computer vision techniques [61–63]. The first step to enable the analysis (measurement or quantification) of the *in vitro* cell migration assays is to articulate (segment) cell/non-cell (wound) pixels within the input cell images. A typical wound

healing process then can be calculated as the percentage of area reduction or wound closure as follows [63,64]:

$$\text{Wound Closure \%} = \left[ \frac{A_{t=0} - A_{t=\Delta t}}{A_{t=0}} \right] 100\%, \quad (1)$$

where  $A_{t=0}$  refers to the initial wound area, and  $A_{t=\Delta t}$  is the wound area measured after a time laps of  $\Delta t$ .

Studies already exist on in vitro cell migration assay analysis, focusing on measuring the cell gap using computer vision tools. In [65], software tool called *TScratch* was presented, which is a MATLAB based software with a graphical user interface (GUI). The user can load a sequence of cell images, adjust the parameters, and inspect the result. Technically, an edge-detection algorithm is used based on the discrete curvelet transform. The magnitudes of the curvelet coefficients are summed up and the resulting intensity provides a measure of the amount of details, based on which the cell area can be separated from the non-cell (wound) area by applying a threshold. Glass et al. [66] proposed applying an entropy filter on the images and then using topology-preserving level sets to find the edges between cell layer and wound area. If no wound gap is present in the image, the algorithm still finds a segmentation. To detect such false segmentations, they used an entropy-based heuristic to decide whether a wound area or a cell populated area has been segmented. In a follow-up study [67], they trained a support vector machine (SVM) to classify whether the algorithm detected a wound area or not. They implemented the algorithm as a plugin for *MiToBo*, an extension of *ImageJ*.

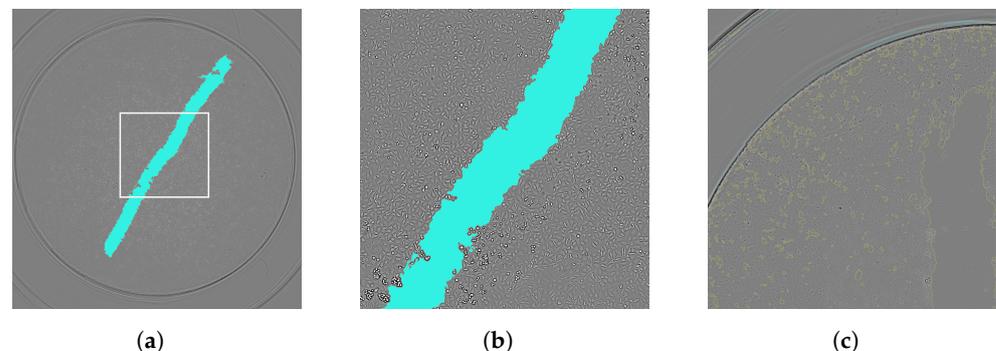
In [68], a texture-based method was used by computing the pixel's standard deviation within a sliding window. Based on the histogram of the obtained standard deviation values, a threshold was determined and subsequently used to segment the input image. Active contours (snakes) were used [69] to find the border between the wound area and cell layer. Cortesi et al. [70] presented a MATLAB based GUI for the wound area segmentation process. It also applies a local entropy filter to the input image and further determines a threshold based on Otsu's method. Suarez et al. [64] implemented a plugin for *ImageJ* using a texture filter based on local variance. After setting a sensitivity parameter, some local thresholding was applied in order to obtain the final segmentation result. In [71], a convolutional neuronal network (CNN) was trained to detect cells in the input image. To segment a wound area, a binary image was created where all detected cells were labeled, followed by morphological operations. The remaining nonlabeled area is identified as the wound area. Sinitca et al. [72] presented a software tool with a GUI, called *BCAnalyzer*, implemented in Python. For wound area detection, the Canny edge detection algorithm is first applied. In the following step, a local edge density is computed using a sliding window approach. In the final step, the resulting edge density image is binarized, applying a threshold that is determined by user input or using an automatic threshold determination technique such as Otsu's method.

We tested the publicly available implementations corresponding to the publications described above to check whether these tools are usable for the experiments on our dataset. We identified some severe problems in using these tools. Our dataset consists of images where the whole well is captured. The circular monolayer with the scratch area is situated in the image center part. Around that is a cell free area containing the border of the well. See Figure 1a for an example (the wound area is highlighted). The publicly available tools have been developed for analyzing the center of cropped well images, as depicted in Figure 1b. Thus, problems arise when directly using them on our data.

The Cell Invasiv-o-Meter from [70] is a MATLAB tool with a GUI, where, in the more recent MATLAB versions, the GUI does not work properly anymore. After some adaption to make the tool run, it produced incorrect outputs. Furthermore, the software downscales the images, which we do not desire as we do not want to affect the compression evaluation. After changing the code to handle our data in full scale, the software segmented no wound area. The software did not provide any parameters that could be tuned to adapt the output.

The *ImageJ* plugin *Wound Healing Size Tool (WHST)* from [64] for automatically analyzing wound areas was also designed for center cropped images. Depending on the parameters, it finds many cell-free areas in the image, mostly some small space between the cells in the monolayer. It assumes the area with the biggest size is the wound area and measures it. On our dataset, it identifies the well border as a wound area because it is the biggest cell-free area. It is possible to manually analyze all detected cell free areas and select the correct wound area and then use *ImageJ* to measure the enclosed size. For our compression experiments, this would necessitate an impossible amount of manual labor. Figure 1c shows the selected areas (in a detailed view of the top left corner of an image in order to provide a better visualization) produced by the *WHST* tool. Note that the cyan-marked area is identified as the biggest area, ergo falsely as the wound area.

The *BCAnalyzer* tool [72] also struggled to detect the correct wound areas due to the presence of the well border in the images. However, we were able to adapt the software so that we could use it for our experiments; see Section 5 for more details. An advantage of using the full-well images is that the whole scratch can be analyzed as opposed to just considering a part of it in the center of cropped images; compare Figure 1a and Figure 1b.



**Figure 1.** Example of a scratch assay image from our dataset with the well border present in image (a). The publicly available tools usually work on center-cropped images, as depicted in (b). The output of the *WHST* tool from [64] is shown in (c).

In the following, as one of the main objectives of this study, we describe two additional distinct algorithms, log gradient segmentation and entropy filter segmentation, to perform the challenging task of wound area segmentation on our dataset.

### 2.1. Log Gradient Segmentation

Edge detection is the most common technique used for characterizing object boundaries and is therefore widely used for object segmentation tasks. First-order derivative operators (e.g., Sobel, Prewitt, Roberts, etc.) work best when the gray-level transition is abrupt. For smoother transitions (e.g., low-contrast images such as cell images), it is more useful to compute the second-order derivative, where the zero crossing points of the second derivatives are considered. A common convolutional operator used for the estimation of the second derivative is the Laplacian operator, which is often applied to images that have been already smoothed with, e.g., a Gaussian smoothing filter (Laplacian of Gaussian (LoG)). The LoG is mathematically defined as:

$$\Delta(G_{\sigma} \times I) = \left[ \frac{\partial^2 G_{\sigma}(x, y)}{\partial x^2} + \frac{\partial^2 G_{\sigma}(x, y)}{\partial y^2} \right] \times I(x, y), \quad (2)$$

where  $G_{\sigma}(x, y)$  is a Gaussian filtering function with standard deviation  $\sigma$ . Still, the LoG can be very sensitive, and many points of zero crossing in the output map are often false (positive) edge points. This is mainly caused by intensity inhomogeneity or noise. Using (inverse) gradient information as the edge indicator within an energy function is one

effective way to further smooth the homogeneous regions and enhance the edges of an object [73]:

$$E^{LoG}(L) = \iint_{\Omega} g(|\nabla I|) \times (L - 0)^2 + (1 - g(|\nabla I|)) \times (L - (G_{\sigma} \times I))^2 dx dy, \quad (3)$$

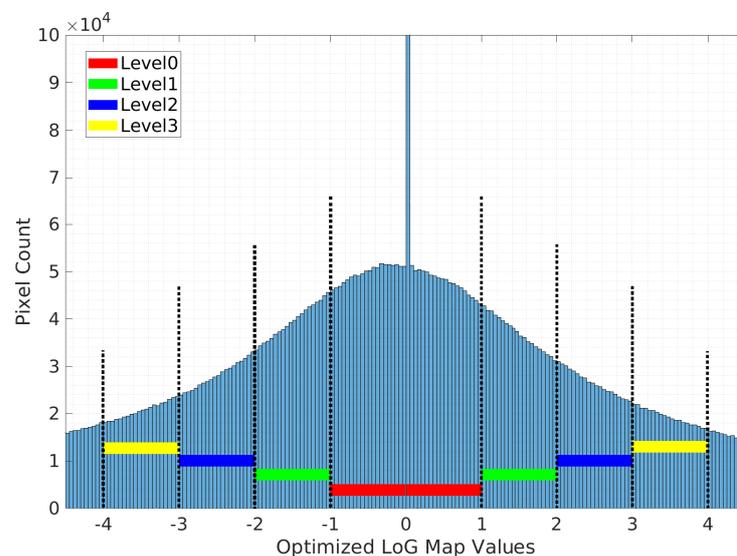
where  $L$  is the value of the optimized LoG, and  $g(|\nabla I|)$  is the edge indicator. The values of  $g(|\nabla I|)$  are small and rather often equal to 0 at the positions near the object boundaries. The values of  $g(|\nabla I|)$  are large and approximately equal to 1 in homogeneous areas.  $(L - 0)^2$  is the data fitting term that computes the proximity between the optimized LoG and the zero plane. If the energy decreases, the term  $g(|\nabla I|) \times (L - 0)^2$  pushes  $L$  close to 0 in the homogeneous regions. Thus, it is helpful to smooth the homogeneous regions. At the same time,  $(L - \Delta(G_{\sigma} \times I))^2$  is the data-fitting term that computes the proximity between the optimized LoG and the original LoG. Minimizing the energy function, we obtain the following Euler–Lagrange equation:

$$g(|\nabla I|) \times L - (1 - g(|\nabla I|)) \times (L - \Delta(G_{\sigma} \times I)) = 0. \quad (4)$$

The steepest gradient descent (SGD) can be used to solve the above equation, where we obtain the following iterative equation:

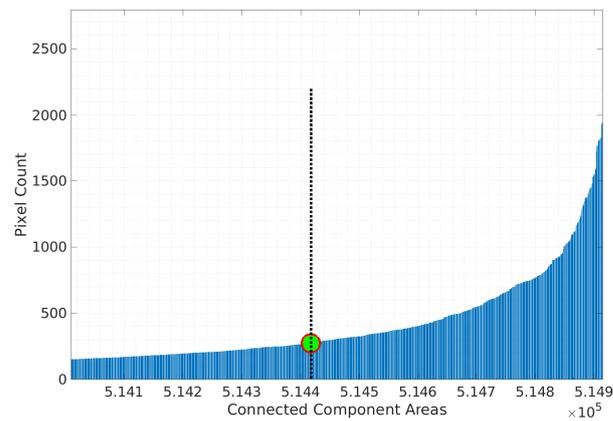
$$\frac{\partial L}{\partial t} = \times (g(|\nabla I|) \times L - (1 - g(|\nabla I|)) \times (L - \Delta(G_{\sigma} \times I))), \quad (5)$$

where the value of the  $t = 0.009$ . The output of the SGD is smoother in homogeneous regions, which mainly refers to the non-cell (wound) areas, and is steeper near the object boundaries, which mainly indicates cell areas. Analysis of the LoG map histogram enables us to distinguish three pixel value regions in the map: (I) non-cell-area pixels, whose values mainly range between zero and one; (II) cell-area pixels, whose values are greater than four; and (III) uncertain area pixels whose values are between one and four. To enable optimal classification of uncertain area pixels, we use a hysteresis thresholding (HT) mechanism. Figure 2 illustrates the HT levels on the output map histogram. The principle of the HT mechanism is based on hierarchical connectivity checking, where the first-level pixel values, ranging from one to two are checked if they are connected to the second-level pixel values ranging between two to three. The connected pixels values are kept, while those that are not connected are discarded. Similarly, the second-level pixel values, ranging between two and three are checked if they are connected to the third-level pixel values ranging between three and four, and so on.

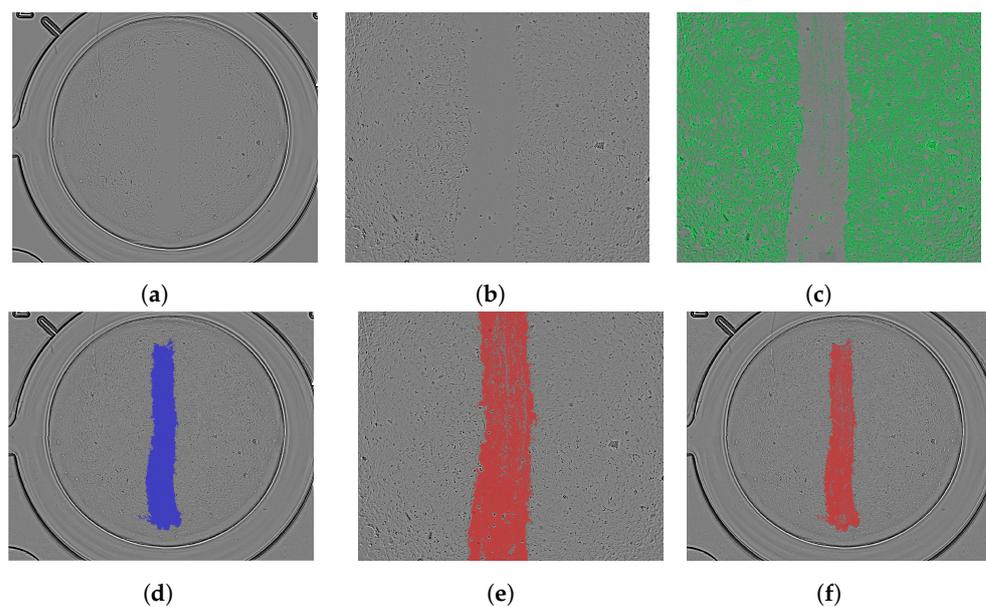


**Figure 2.** Optimized LoG output map and hysteresis thresholding (HT) levels.

The HT process can be further continued if a higher level of confidence is demanded in the algorithm. While the HT mechanism helps to refine the uncertain edges to a large extent, some isolated edge components form during the refinement process in the map. To further smooth the non-cell (wound)-area pixels, we apply an auto-thresholding process on the output map. For this, we first calculate the area of each connected component found in the output map in terms of pixel number. Then, we sort them in ascending order, so that we can enable a sort of classification among them based on their sizes. Starting from the smallest component, the pixel difference ( $D$ ) of each component to its predecessor is calculated. The threshold is then set, where the first abrupt point ( $D \geq 2$ ) is found in the sequence. Figure 3 illustrates this process. The threshold is then used to remove the isolated components, which most probably belong to the outlier type and noise. Using the initial image output segmentation, we define the wound area by specifying the biggest connected component on the inverse cell segmentation and filling the subsequent area, as demonstrated in Figure 4d. The wound (ROI) mask then is used in the subsequent wound images in the sequence to mask out the cell-area pixels. Thus, in the case where the integrity (connectivity) of the wound and the cell pixels in the initial segmentation are not well maintained, the wound area definition, and thus the subsequent calculations (wound area pixel count in the next images in the sequence), will be incorrect.



**Figure 3.** HT output map and the connected component autothresholding.



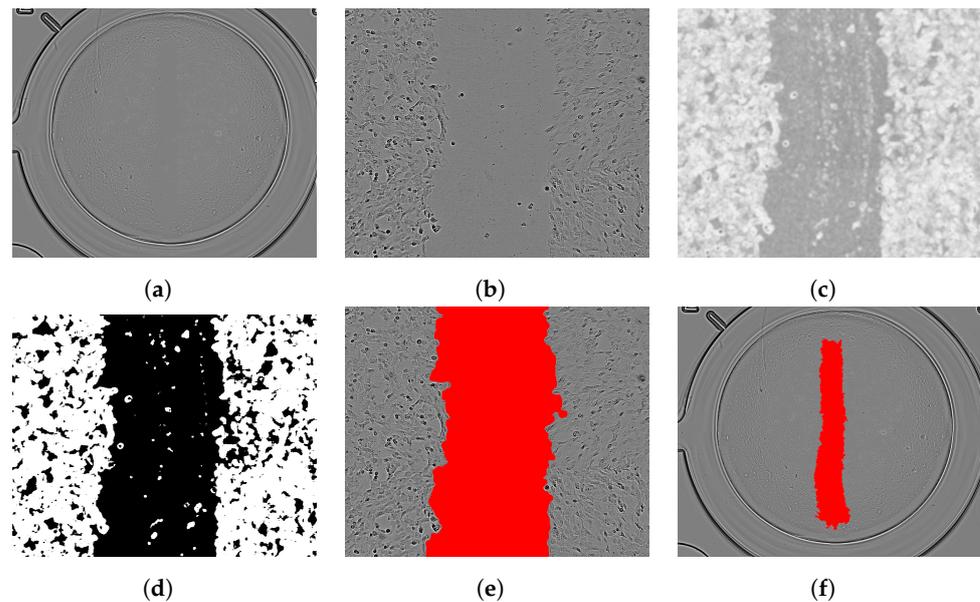
**Figure 4.** Example input image (a), detailed view of the center (b), the corresponding cell-area segmentation (c), defined wound area (d), and detailed (e) and final wound area segmentation (f), using the log gradient segmentation algorithm.

## 2.2. Entropy Filter Segmentation

The second segmentation method uses a holistic approach by trying to separate the textures of the confluent cell layer and the non-cell (wound) area. To enhance the texture, an entropy filter is applied that computes the local entropy in a region (local neighborhood  $N$ ) of a gray-scale image using the following equation:

$$e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i), \quad (6)$$

where  $z_i$  is the pixel intensity value,  $p(z)$  is the histogram value of the pixel values in a region  $N$ , and  $L$  is the number of possible intensity values [74]. As there is more irregularity in the area where cells are located due to the cell structure, a higher response of the entropy filter is expected compared with that of the cell-free area. Figure 5c shows the filter response of the cell area depicted in Figure 5b, and a clear enhancement in the cell covered area is visible. Additionally, noise and possible contamination in the wound area are enhanced. The filter response is used to determine a global threshold by applying Otsu's method in order to separate both texture areas (cell and wound area). After the binarization process, small areas in the wound gap are present, which belong to noise or contamination, and the border of the cell layer is not closed everywhere to form a clear borderline, but small islands and bays appear, which can be seen in Figure 5d. To address these effects, several morphological operations are applied. First, small connected areas (blobs) are removed to eliminate the noise and contamination artifacts in the wound gap. This operation also removes some small islands at the cell border that should not be removed as they are part of the border line. Therefore, after removing the small blobs, a dilation operation is applied that lets the cell layer grow toward the middle of the wound area and closes the border line. The result is combined with the previous result via a logical "and" operation, and the small islands in the border region are restored. Afterward, a closing operation is performed to try to obtain a closed border line by avoiding expansion into the wound area. In the final step, small holes in the cell area are filled, and the remaining biggest hole is identified as the wound area. Note that in the first image, the size of the wound area is the largest compared with that in the later stages in the sequence and is quite clearly separable. Therefore, it is easier, in the first image, to identify the biggest hole in the cell layer as the wound area. In the later stages, cells migrate into the wound area, small islands of cells may form there, and the wound area defragments into several small holes. Therefore, these morphological operators need to be used more cautiously, and the sizes of the structuring elements need to be carefully designed in order to achieve a reasonable trade-off between removing small wound areas and falsely identifying small (isolated) cell areas as wound areas. Especially, the border between the cell area and non-cell area appears not so clear. Thus, the segmented wound area of the first image of a sequence is used as a ROI for the subsequent images, and the cell area outside the ROI is not considered anymore, as there could also be less confluent areas because of the constantly migrating cells, which otherwise may falsely be identified as small wound areas. Figure 5e,f show an overlay of the final segmentation result. The idea of the entropy filter segmentation algorithm is, to a certain extent, comparable to that proposed by [70], where an entropy filter was used, and Otsu's method was applied for thresholding. In [64], they used a local variance filter instead of the entropy filter. In this study however, the algorithm contained more elaborate postprocessing and was able to handle the full-scale images including the well border.



**Figure 5.** Example input image (a), detailed view of the center (b), the entropy filter response (c), after binarization (d), detailed view with the segmented wound area (e), and final wound area segmentation (f) using the entropy filter segmentation algorithm.

### 3. Image Compression

Two types of redundancy exist in an image: Spatial redundancy refers to the correlation between neighboring pixels. This is the redundancy formed by the patterning or self-similarity within an image. Spectral redundancy, on the other hand, refers to the correlation between different colors of planes or spectral bands. Image compression techniques aim at reducing the number of bits needed to represent an image by eliminating the spatial and spectral redundancies in an image.

Several image compression techniques have been proposed over the years, the most important ones being proposed by the Joint Photographic Experts Group (JPEG), which is the name of the joint ISO/CCITT committee that has created a series of image compression standards starting in 1992. Baseline JPEG compression involves (fixed) block-based compression. Data reduction is achieved by subsampling of the color information by the quantization of the discrete cosine transformation (DCT) coefficients, and by Huffman Coding (reorder and coding). This algorithm is lightweight and fast, however causes blocking artifacts (particularly visible near sharp edges with high contrast), which can be disturbing at high compression ratios. Later on, in the year 2000, the group introduced JPEG 2000 (J2K), which uses the wavelet transform (WT) instead of the DCT to reduce the amount of information contained in an image. The WT converts the information into the local scale and amplitude over time and is thus more efficient for the compact representation of images than DCT. J2K provides higher scalability than baseline JPEG, yet it requires more processing power/time than JPEG. Proposed in 2022, JPEG-XL is based on the VarDCT (variable-blocksize DCT) encoding mode. It utilizes the same DCT algorithm as baseline JPEG, but instead of using fixed blocks, it uses variable block sizes and can even use other transforms (e.g., AFV or Hornuss). The VarDCT mode is based on the (lossy) PIK (<https://github.com/google/pik> accessed on 12 December 2022) encoder and Cloudinary's Free Universal Image Format (FUIF) (<https://github.com/cloudinary/fuif> accessed on 12 December 2022). JPEG-XL is claimed to offer significantly better image quality and compression rate than baseline JPEG and offers efficient computation and short specification.

Created in 2014 by Fabrice Bellard (<https://bellard.org/bpg> accessed on 19 January 2023), BPG image compression was proposed as a replacement for JPEG to provide a more compression-efficient alternative in terms of image quality and file size. It uses the intraframe (I-frame) encoding of the High-Efficiency Video Coding (HEVC) (<https://>

[//www.itu.int/rec/T-REC-H.265](https://www.itu.int/rec/T-REC-H.265) accessed on 19 January 2023) video compression standard. BPG is an I-frame version of HEVC with a reduced header to make it more efficient. BPG files are significantly smaller than those of baseline JPEG and have the ability to include metadata as well. WEBP (lossy) compression was first announced in September 2010 and subsequently released in April 2018 by Google Inc. The algorithm is based on the intraframe block prediction compression technique that is utilized in VP8 (<https://www.rfc-editor.org/rfc/pdf/rfc6386.txt.pdf> accessed on 19 January 2023) video compression to compress key frames in videos. It uses block adaptive quantization, Boolean arithmetic encoding, and predictive coding. The predictive coding uses the values in adjacent blocks of pixels to predict the values in a block and then encodes the difference. Images generated by WEBP compression are 25–34% smaller than those of baseline JPEG images; however, WEBP files can lose detail and texture ([https://developers.google.com/speed/webp/docs/webp\\_study](https://developers.google.com/speed/webp/docs/webp_study) accessed on 19 January 2023). Released in 2018 by the Alliance for Open Media (<https://aomediacodec.github.io/av1-avif/> accessed on 19 January 2023), AV1 compression enables about 30% compression gains over its predecessor VP9 (<https://www.webmproject.org/vp9/> accessed on 19 January 2023). The AV1 encoder operates on pixel blocks. Each pixel block is processed with a predictive-transform coding mechanism, where the prediction comes from intraframe reference pixels. A unitary transform then is applied to the residuals to further remove the spatial correlations, and their coefficients are quantized. Both the prediction syntax and the transform coefficient indexes are entropy-coded. The main drawback of the AV1 compression is slower encoding and even decoding performance due to the high complexity of the algorithm.

#### 4. Experimental Framework

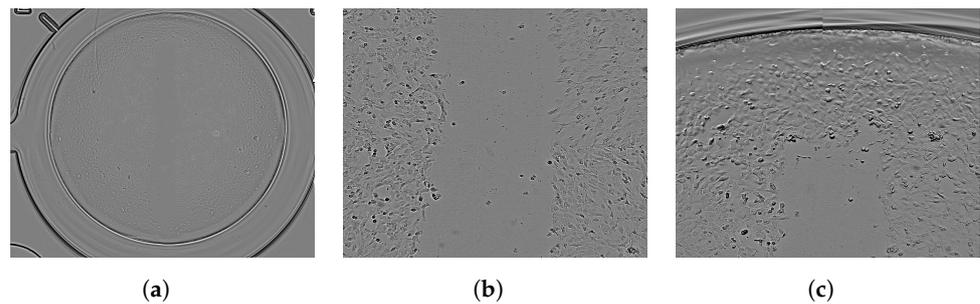
In this section the dataset and metrics for our experimental setup are introduced.

##### 4.1. Main Dataset

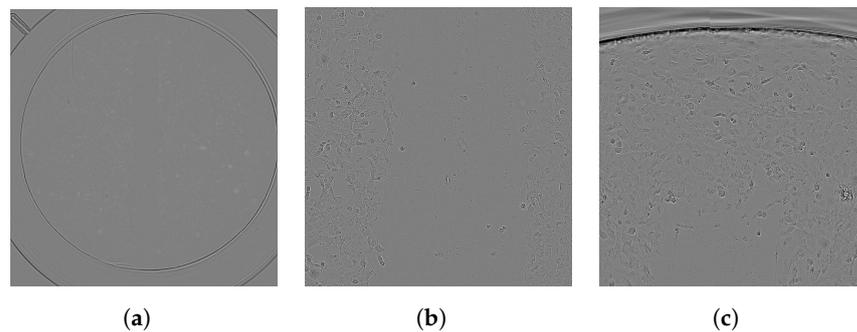
For this study, we were provided with images of wound healing assays acquired with a Spark Cyto cell plate reader from the company Tecan using bright-field imaging. The probe contained monolayers of type 3T3 cells located in a 96-well microplate from Greiner (Cellstar TC, Catalog Nr. 650160). The plate reader recorded high-quality images of each well at 4× and 10× magnification. These images were stored in TIFF format and subsequently used to analyze the cell populations as described. The dataset is organized as follows: In total, there are 10 image sequences in the dataset, and each sequence contains 9 (wound) time series images in two magnifications of 9824 × 10,260 pixels (10×) and 4912 × 4015 pixels (4×). The elapsed time between each image in each sequence is 3 hours, and images are composed of smaller image patches. Each image sequence is identified with an alphanumeric code, which represents the coordinates of the location of the well on the microplate. For example, sequence A2 refers to the cell layer images of the well in row A, column 2. The size of the 4× magnification images is around 50 megabytes, and the 10× magnification images have a size of around 260 megabytes. This motivated us to compress these images to reduce the needed storage capacity and to speed up data transfer processes. Figures 6 and 7 demonstrate sample images from the dataset.

##### 4.2. Data Sequence X

In addition to the dataset, we had a single sequence of 29 cell migration images with magnification 10×. This sequence was already JPEG-compressed; the original uncompressed data were not available. However, for this sequence, a ground truth is provided. The ground truth was provided by the manufacturer of the well reader that captured the images of our dataset. It was partly manually labeled by a nonspecialist; the remaining part was fed into a neuronal network (U-Net) segmentation algorithm. These data were manually corrected by a microscopy specialist for the so-far final ground truth (which still exhibits some errors). The images in this dataset are center-cropped images and do not contain the challenging (in particular for wound area detection and segmentation) border area.



**Figure 6.** Example image from the dataset: first image of the the first sequence with  $4\times$  magnification (a), detailed view of the center (b), and detailed view of the border (c) where a degradation of the quality of the cell structure is visible.



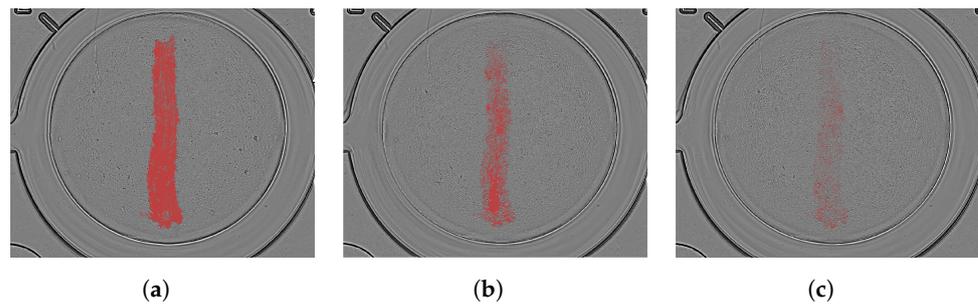
**Figure 7.** Example image from the dataset: first image of the the first sequence with  $10\times$  magnification (a), detailed view of the center (b), and detailed view of the border (c).

#### 4.3. Metric and Measures

To measure the compression performance, we used the peak signal-to-noise ratio (PSNR) and the multiscale structural similarity index measure (MS-SSIM) [75]. Additionally, we assessed segmentation behavior relating the results of different compression rates to the results computed on the uncompressed data.

### 5. Segmentation Experiments and Analysis

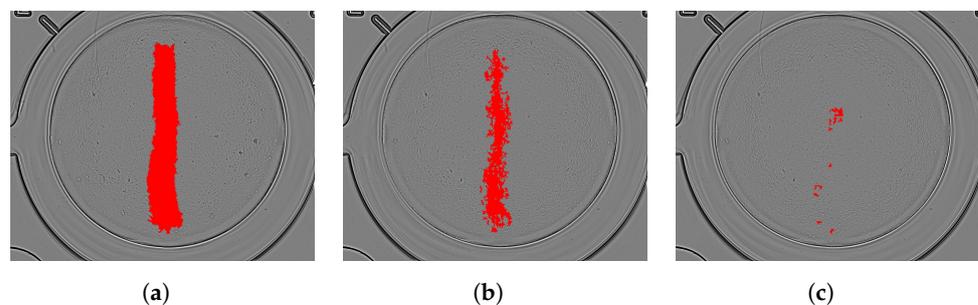
As the primary step in evaluating the segmentation algorithms' performance and to maintain a benchmark and a reference for the compression evaluation experiments, we first ran the segmentation algorithms on the images of the main dataset without applying compression. To measure the wound area, the total number of segmented wound-area pixels were counted in our experiment. As mention before, in the literature, the wound area has been used to measure the wound healing process; see Equation (1). Figure 4 shows an input image sample (Figure 4a), the corresponding cell area segmentation (Figure 4c), a defined wound area (Figure 4d), and the corresponding wound (non-cell)-area segmentation (Figure 4f) using the log gradient segmentation (HT level 2). As shown in Figure 4c, the algorithm was able to differentiate and segment the cell-area pixels from the non-cell-area (wound) pixels with high accuracy. The classifying functionality of the HT mechanism was mainly visible in the segmentation of the indistinct, sparse, and isolated structures (distributed especially on the non-cell/wound area) and at the cell area boundaries. It should be noted that while such indistinct structures were effectively filtered out in the wound area, the pixel groups that belonged to the significant cell structures inside the wound area were well preserved (see Figure 4c). The accuracy of the cell-area segmentation at this step plays a key role in defining the target wound area (ROI) in the initial cell image (which is used as the wound-area reference) and subsequently the system accuracy. Figure 8 demonstrates the algorithm performance on the first (Figure 8a), fifth (Figure 8b), and the last (Figure 8c) image in the sequence.



**Figure 8.** Final wound-area segmentation in the first (a), fifth (b), and ninth (c) image (slides) of the sample sequence A2 using the log gradient segmentation algorithm.

Figure 5 visualizes the segmentation process when applying the entropy filter segmentation algorithm: Figure 5a shows the gray-scale input image with the scratch in a cell layer, Figure 5c visualizes the entropy filter response on which the threshold is determined, and Figure 5d shows the binarized result after applying the computed threshold. As shown in Figure 5d, noise and spots of remaining cells or cell fragments are present in the wound area. With morphological operations, these are removed, and the border of the wound area is connected to finally obtain the segmentation result, as depicted in Figure 5e and Figure 5f, respectively. Figure 9 shows the segmented wound area extracted from a sequence of wound healing images using the entropy filter segmentation algorithm.

To provide a better idea of the available in vitro cell migration tools' performance and to enable some comparison with our proposed algorithms, we applied *BCAnalyzer* software [72] to our dataset. As mentioned, the software is not applicable out of the box to our dataset. After adapting the software, the output of the software was not directly usable as it also segmented the border area of the image as a cell-free area, which is technically correct but was not part of the wound area. The reason for this is that the software was developed to be run on data that are center-cropped, as shown in Figure 10b. The output of the *BCAnalyzer* image processing part is shown in Figure 10c. As it can be seen, the border area where the well was located was also identified as a wound area as was the more degenerated cell pattern at the border of the cell layer, which needed to be masked out. This, in particular, highlights the key role of precise segmentation and wound area detection because, if the algorithm fails in this step, the overall algorithm performance is affected. To address this problem of the *BCAnalyzer* tool, we created an ROI by taking the segmentation result of the first image in each sequence produced by one of our algorithms (entropy filter segmentation algorithm). This area was enlarged by a margin of 150 pixels for the 4× magnification and 300 pixels for the 10× magnification images. This ROI is depicted in Figure 10d. The result of masking the *BCAnalyzer* output with the ROI is presented in Figure 10e and as an overlay in the original image in Figure 10f. Figure 11 depicts the segmentation results of the *BCAnalyzer* tool on a sequence of images.

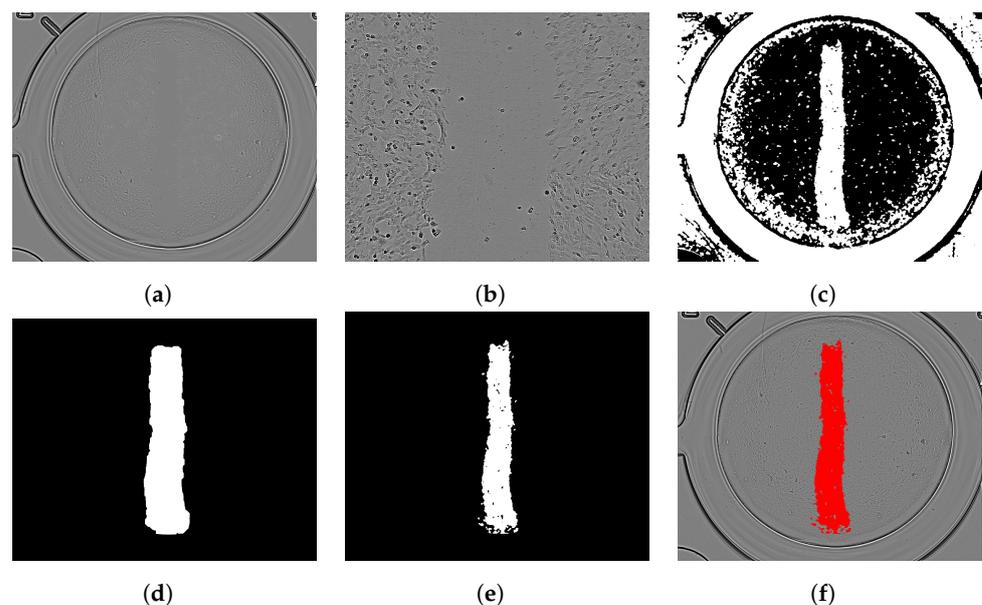


**Figure 9.** Final wound area segmentation in the first (a), fifth (b), and ninth (c) image in the sample sequence A2 using the entropy filter segmentation algorithm.

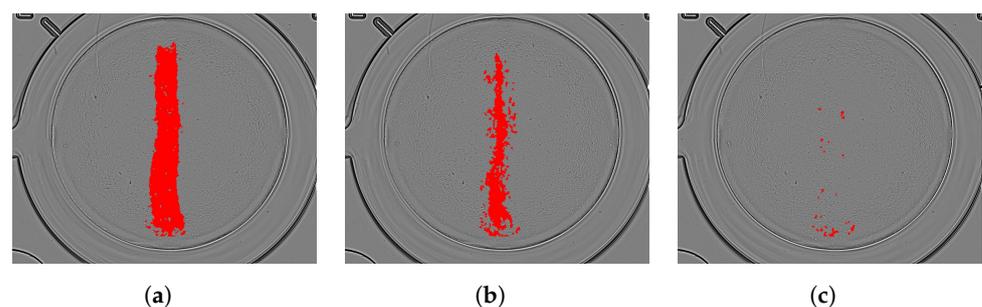
As a ground truth was available for the data sequence X solely, this dataset was used to provide an understanding of the algorithms' segmentation accuracy on this particular

dataset. The graph in Figure 12 shows the segmentation result by measuring the segmented wound area in pixels (plotted on the  $y$  axis). On the  $x$  axis, the time stamps are plotted, each representing one image in the cell migration sequence  $X$ . The performance was also compared with a ground truth that was provided by the manufacturer of the well reader that captured the images of our dataset.

The performance graph in Figure 12 shows that all three algorithms followed the trend of the ground truth, although the log gradient segmentation and entropy filter segmentation algorithms tended to overestimate the wound area in comparison with the ground truth, especially in the middle of the sequence. This could be due to the higher precision of these algorithms in detecting more (true) wound area pixels than the automatically generated ground truth and the manually corrected version. The *BCAnalyzer* algorithm followed the ground truth best. Only in the first time steps, it seemed to underestimate the size of the wound area. Figure 13b provides a detailed view on the sixth image of sequence  $X$  with an overlay of the corrected ground truth segmentation in green and the resulting segmentation of the entropy filter segmentation algorithm in red. It clearly depicts why the entropy filter segmentation algorithm overestimates the wound area size in comparison with the ground truth: the algorithm finds the border closer toward the cell layer, and more smaller holes and bays are included compared with the ground truth. Some deeper discussion with cell migration experts will be helpful to clarify how problematic this discrepancy is. When looking at the detailed view, it seems that the ground truth segmentation is also not exact at all positions.



**Figure 10.** Example input image (a), detailed view of the center (b), the entropy filter response (c), and the output of *BCAnalyzer* software (d). The ROI used to mask the output of *BCAnalyzer* software is shown in (d); the resulting wound area segmentation is shown in (e,f) as an overlay in the input image.



**Figure 11.** Final wound area segmentation in the first (a), fifth (b), and ninth (c) image in the sample sequence A2 using the (adapted) *BCAnalyzer* tool.

Therefore, the quality of the ground truth should be discussed. Due to the high resolution of the images, it is a challenging task to manually produce ground truth data or to manually correct an algorithm output. Therefore, it could be the case that there is room for improvement on the ground truth data itself. Figure 13a shows an overlay of the automatically produced ground truth in green and the manually corrected version in red.

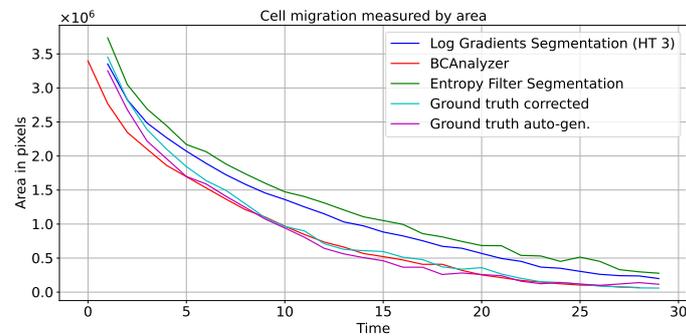


Figure 12. Segmentation algorithms performance applied on the sequence X.

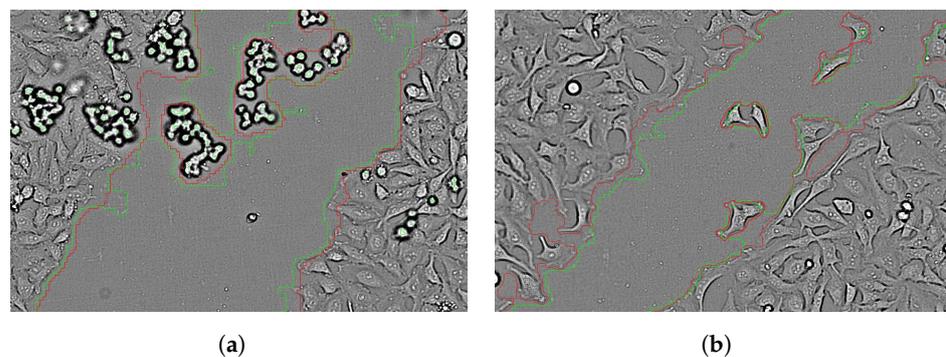


Figure 13. Visualization of the automatically generated ground truth in green in comparison with the manually corrected values in red as overlay on a detailed view of the wound area in the first image of sequence X (a). (b) An overlay on the sixth image in sequence X to compare the ground truth (green) with the result of the entropy filter segmentation algorithm (red).

## 6. Compression Experiments and Analysis

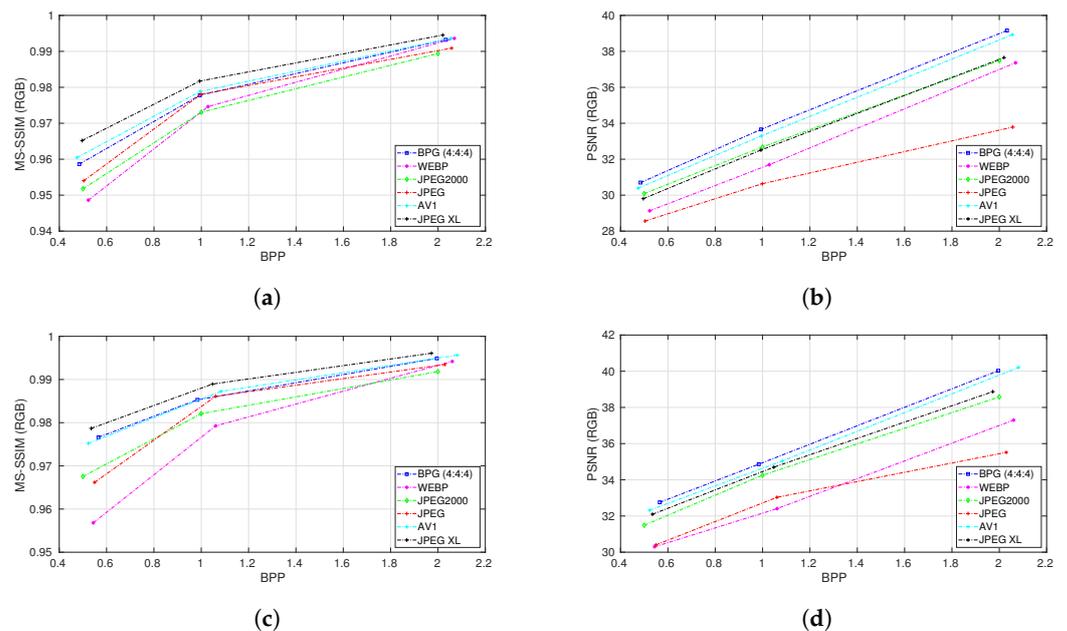
To evaluate the effect of the compression on the *in vitro* cell images' quality and, subsequently, the segmentation algorithms' performance, we applied the compression algorithms explained in Section 3 (JPEG, JPEG2000, WEBP, AV1, BPG, and JPEG-XL) to our main dataset. To address the fixed bandwidth/storage limit requirement (mandatory to enable a fair and transparent comparison), we set three different bandwidth limits of A (0.5 BPP), B (1.0 BPP), and C (2.0 BPP), corresponding to high, medium, and low compression rates, respectively, in terms of bits per pixel (BPPs). The original BPP for the uncompressed images for  $4\times$  and  $10\times$  magnification, on average, was about 6.8.

It should be noted that not all algorithms allowed us to set the exact output file size. Thus, we optimized the compression parameter for each algorithm so that we could achieve the closest ( $\max \pm 0.03$  BPP) possible BPPs to our three target rates. Table 1 shows the selected compression parameters (par) and the resulting BPPs per algorithm, averaged over all 10 image sequences.

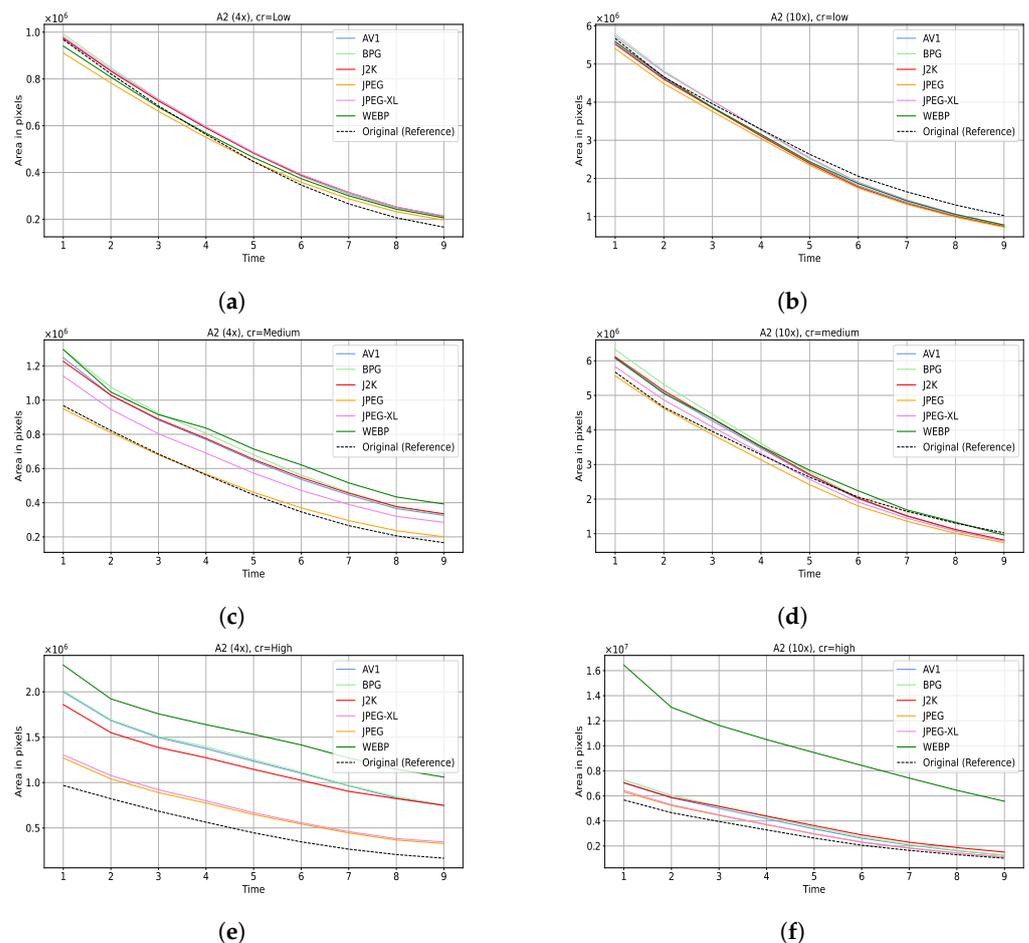
**Table 1.** Selected compression parameters (par) and their corresponding compression performance in bits per pixel (BPPs) for each algorithm.

Parameter & Output BPPs:	High		Medium		Low	
	par	bpp	par	bpp	par	bpp
BPG	33	0.48	29	0.99	25	2.03
J2K	16	0.49	8	0.99	4	1.99
JPEG	23	0.50	61	0.99	85	2.05
WEBP	26	0.52	53	1.02	83	2.06
JPEG-XL	61	0.49	81	0.99	89	2.01
AV1	34	0.47	26	0.99	15	2.05
BPP Level:	A (0.50)		B (1.00)		C (2.00)	

Figure 14 shows the compression algorithms' performance results on the 4× (Figure 14a,b) and the 10× (Figure 14c,d) magnification images. Interestingly, we observed differences in performance using different image quality metrics. In particular, while the best-performing algorithm in terms of PSNR was BPG, when considering the MS-SSIM metric, JPEG-XL produced the best compression performance. In addition, while the JPEG baseline performed the worst in terms of PSNR, the worst-performing algorithms in terms of MS-SSIM (especially using the high (A) compression rate) were J2K and WEBP. Overall, AV1 showed stable performance on both image quality evaluation metrics (PSNR and MS-SSIM). If reliability and compression performance are the primary concerns, AV1 seems to be the best choice. This inference is even more convincing as we note that AV1 showed quite competitive (only slightly below) performance compared with the top-performing algorithms for both PSNR and MS-SSIM. Interestingly, when investigating the compression performance on the larger (10×) images, we observed similar trends, and the algorithms' performance was quite comparable to their performance on the smaller (4×) images. The only notable difference was the performance of WEBP, which seemed to be even worse for the larger (10×) images.

**Figure 14.** Compression of algorithms' performance on the 4× (a,b) and 10× (c,d) magnification images using 0.5 BPPs, 1.0 BPPs, and 2.0 BPPs bandwidths, corresponding to high (A), medium (B), and low (C) compression rates, respectively.

To enable the evaluation of the compression effects on the segmentation algorithms' performance, we proceeded with running the segmentation algorithms on the compressed image sequences (using the three different compression rates). Figure 15 shows the performance of log gradient segmentation on the A2 sequence images. The graphs show the segmented wound area (area pixels) against time (the sequence images), before (dashed line) and after applying the various compression techniques (solid lines). The analysis of the algorithm performance on the uncompressed image sequences demonstrated the quite stable performance of the algorithm on the sequence data. As it can be seen in the results (dashed lines in the graphs and the corresponding output segmentation images presented in Figure 8), accurate segmentations were able to illustrate the wound healing process quite well. It should be noted that the decreasing wound area (detected pixel counts) was coincident with the actual shrinking wound area depicted in the images sequences, and pixel counts at each step (image slide) showed less wound area than their sequence predecessors. As can be seen in Figure 15b, applying the low-level (C (2.0)) compression rate did not notably affect the algorithm performance on the sequence images. This was rather true for results of the medium (B (1.0))-rate-compressed (10 $\times$ ) images. Yet, as can be seen in the results, the algorithm tended to slightly overestimate when applied to the medium (B (1.0))-rate-compressed (4 $\times$ ) images (except for JPEG compression). This was mainly due to the more severe effect of the compression on the cell structures (texture) in these images, which were lower resolution than the 10 $\times$  images.



**Figure 15.** Segmentation performance on the C (a,b), B (c,d), and A (e,f) compressed images on sequence A2 using the log gradient segmentation algorithm. In the left column, results are depicted for 4 $\times$  magnification and in the right column for 10 $\times$  magnification images.

Applying a high (A (0.5)) compression rate to the images, however, started to more clearly affect the algorithms' performance. For the 10 $\times$  images, we observed a considerable

overestimation on the WEBP-compressed images. Such an overestimation was not far from our expectation, as we already knew (as discussed in Section 3) that the images compressed by WEBP (specifically with lower rates) performed poorly in terms of PSNR and MS-SSIM. On the other hand, for the  $4\times$  images, the good accuracy on the JPEG-compressed sequences was unexpected given the corresponding IQMs' results (whereas good JPEG-XL results were expected). Again, overall, the overestimation was more notable on the high (A (0.5))-rate-compressed ( $4\times$ ) images, which, as already mentioned, was due to the more severe effect of the compression on the cell structures in these images. Nonetheless, we note that despite this observed overestimation, the segmentation algorithm was stable in depicting the progressive wound healing process (decreasing wound area trend), in particular if a suitable compression scheme is chosen. The analysis of the further sequences (A3, etc.) in the dataset (both the  $4\times$  images and  $10\times$  images) showed similar performance, as presented for the sample sequence A2 here.

The behavior of the entropy filter segmentation approach when applied to compressed data is depicted in Figures 16 and 17. Figure 16 shows the performance when measuring the wound area over time for sequence A2 for both magnifications. For the majority of the compression methods used, at the low (C) compression rate (Figure 16a,b), the segmentation results only slightly differed from the reference result, whereas the results produced by J2K and WEBP notably diverged in the later states of the sequence on the  $10\times$  magnification images. Here, the algorithm overestimated the size of the wound area relative to the reference measure. The effect was more severe in the WEBP results, which sometimes even indicated a reversed wound healing process. On the  $4\times$  magnification images, the differences were hardly significant only AV1 produced underestimations in this case. For the medium (B) compression level (Figure 16c,d), the segmentation underestimated the wound area size, with WEBP and JPEG-XL producing results closest to the original. In the sequences with the highest compression rate (Figure 16e,f), wound area underestimation was even worse, again with WEBP and JPEG-XL being the best. Figure 18 shows a center crop of image 6 ( $10\times$  magnification), where the divergence is the highest according to the result plot. The segmentation result for the original image is shown in Figure 18d; those on the J2K-compressed images with a low (C) compression rate are shown in Figure 18a; results for the medium (B) rate are shown in Figure 18b; and for the high (A) compression rate, the results are shown in Figure 18c. The segmented wound area is highlighted. The illustrations show that in the original image as well as in the image with a low (C) compression rate, the algorithm overestimated the wound area, and some cells were included in the segmented wound area. In the image with a medium (B) compression rate, some, but clearly fewer, cell pixels are still included than in the prior example. It seems that almost no cell pixels are included in the segmented wound area in the image with a high (A) compression rate only. However, this was just a subjective visual analysis as there were no ground truth data available for these data.

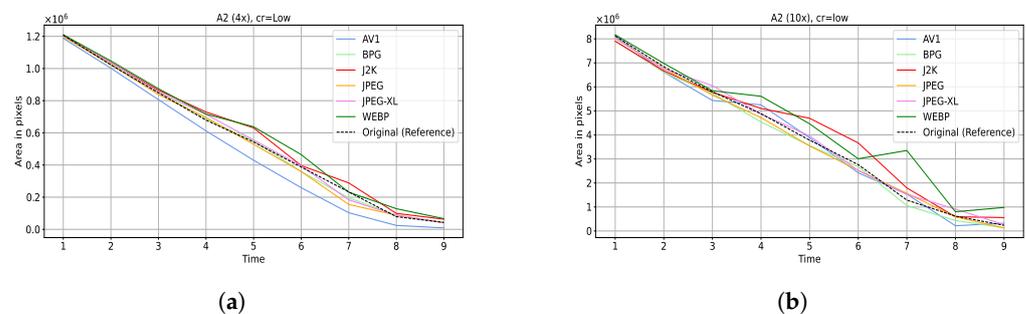
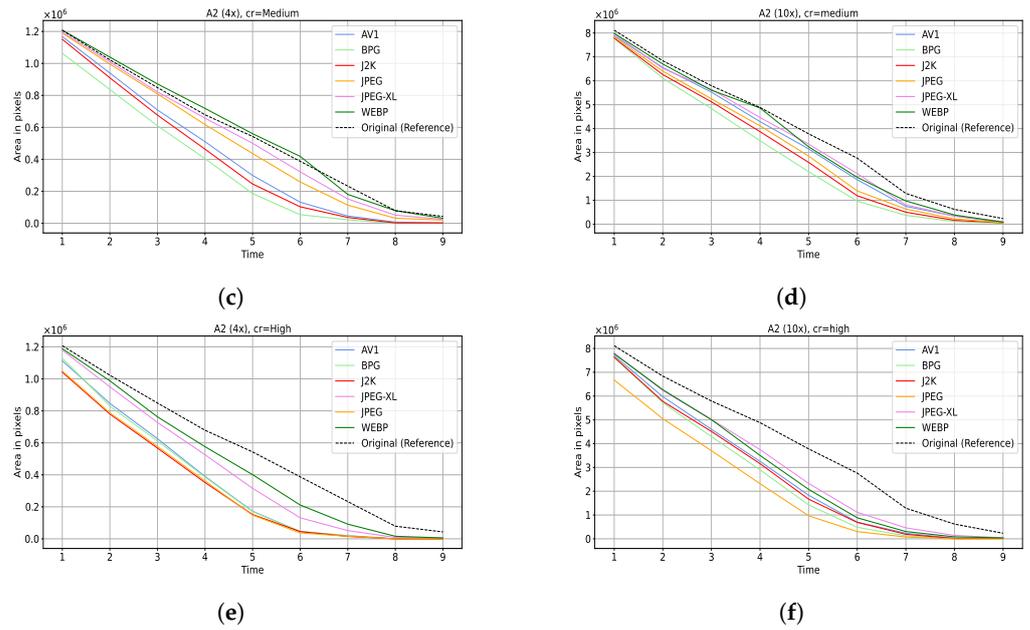
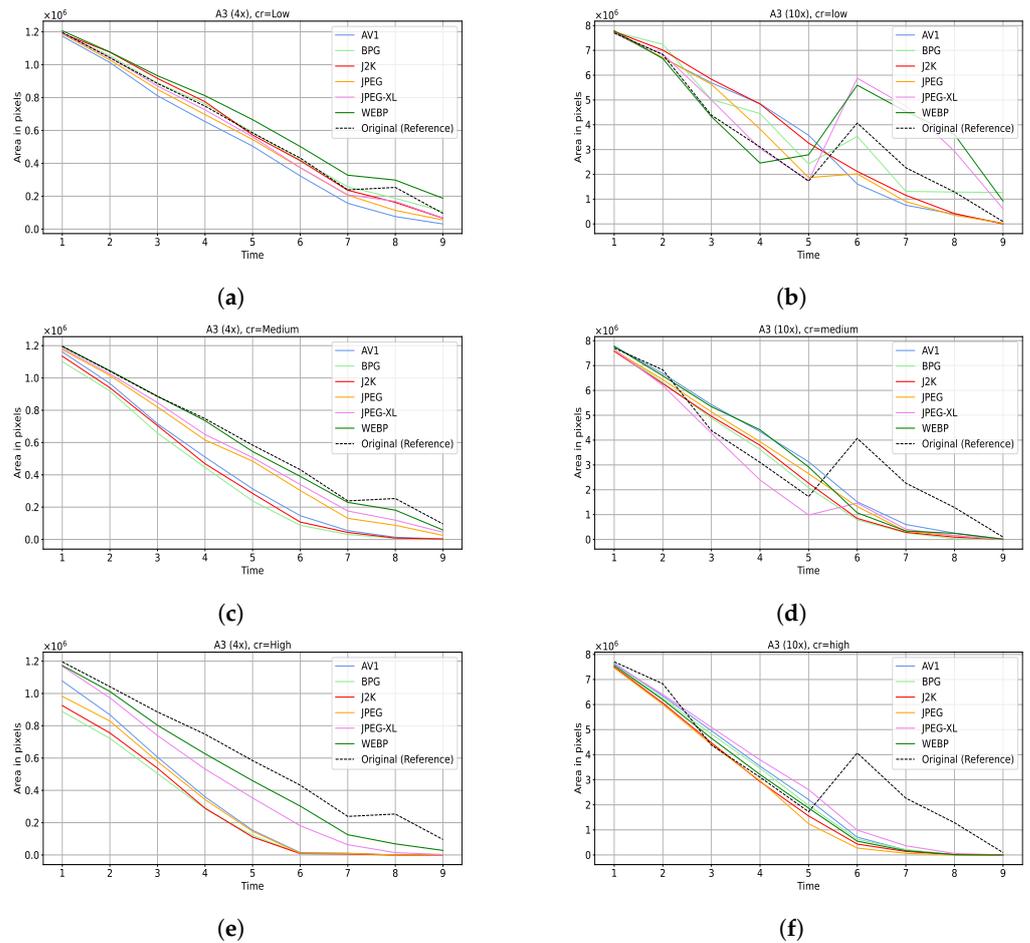


Figure 16. Cont.



**Figure 16.** Segmentation performance on the C (a,b), B (c,d), and A (e,f) compressed images on sequence A2 using the entropy filter segmentation algorithm. In the left and right columns, the results are depicted for the 4× and 10× magnification images, respectively.



**Figure 17.** Segmentation performance on the C (a,b), B (c,d), and A (e,f) compressed images on sequence A3 using the entropy filter segmentation algorithm. In the left and right columns, the results are depicted for the 4× and 10× magnification images, respectively.

For an additional cell sequence (A3), the behavior is somehow different. First, we observed an irregularity in the reference performance in the  $10\times$  magnification sequence: between images 5 and 6, an increase in the size of the measured wound area appeared. Naturally, the wound area would decrease in the probe, but the algorithm suddenly detected more non-cell (wound) area. Here, the algorithm seemed to experience a problem in determining the required threshold, possibly due to a global change in the filter response. Interestingly, when running the algorithm on the compressed data, this leap did not appear for the medium (B) and high (A) compression rates (except for JPEG-XL at the medium (B) compression rate, see Figure 17d,f). For the images with a low (C) compression rate (Figure 17b), the leap vanished for the AV1-, J2K-, and JPEG-compressed data, whereas in the other cases, it was preserved. In general, on the low (C) compressed data, there was a wider fluctuation in the measured wound area size. The increase (difference) in the identified wound area between image 5 and 6 was around 3.7% relative to the total area of the cell monolayer. The absolute increase in the size of the wound area was about  $2.4\times$  between these two images. Figure 19 shows the segmented wound area in the center of uncompressed images 5 and 6, where the jump in the size of the segmented wound area appeared according to the result graph in Figure 17. In image 6, more cells were included in the wound area than in the previous image 5. In the compressed image, however, the algorithm seemed to perform the segmentation more correctly. The results in the remaining cell image sequences (not depicted here) rather followed the results obtained for sequence A2. For the  $4\times$  magnification images of the A3 sequence, there was only a small leap present, and it appeared at a later time step, i.e., between images 7 and 8. The leap completely disappeared for the medium (B) and high (A) compression rates, and the algorithm followed the trend already seen for sequence A2.

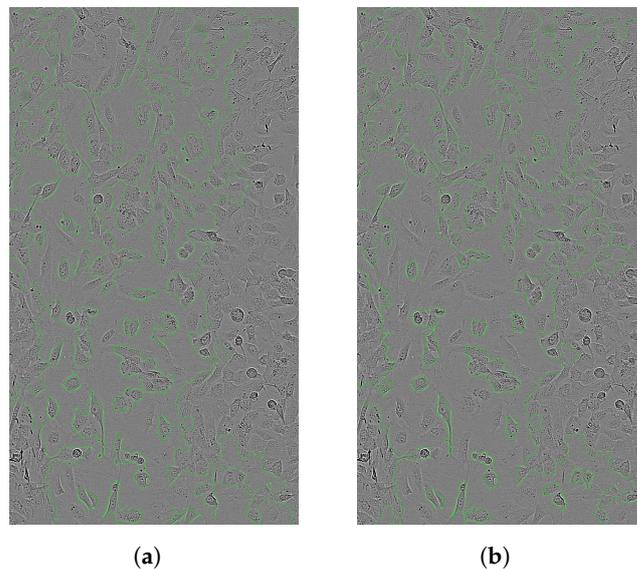
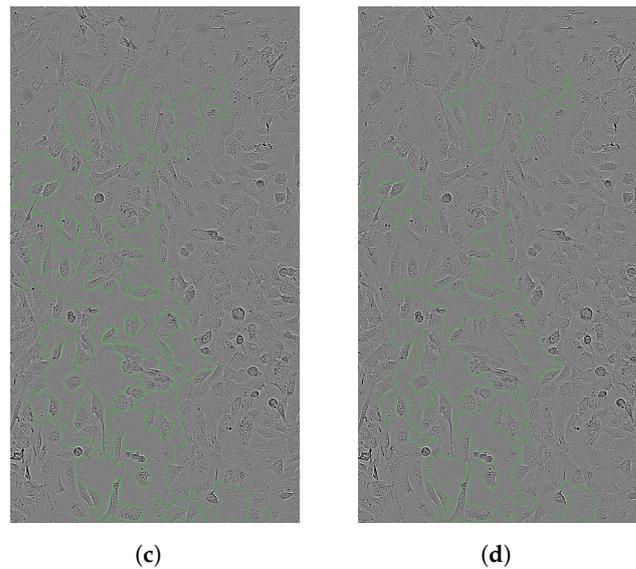
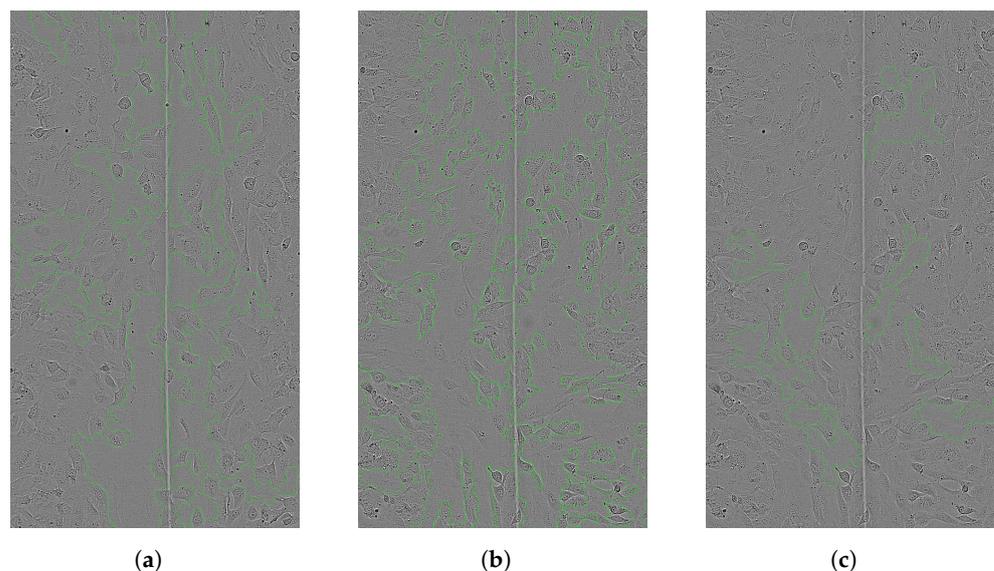


Figure 18. Cont.

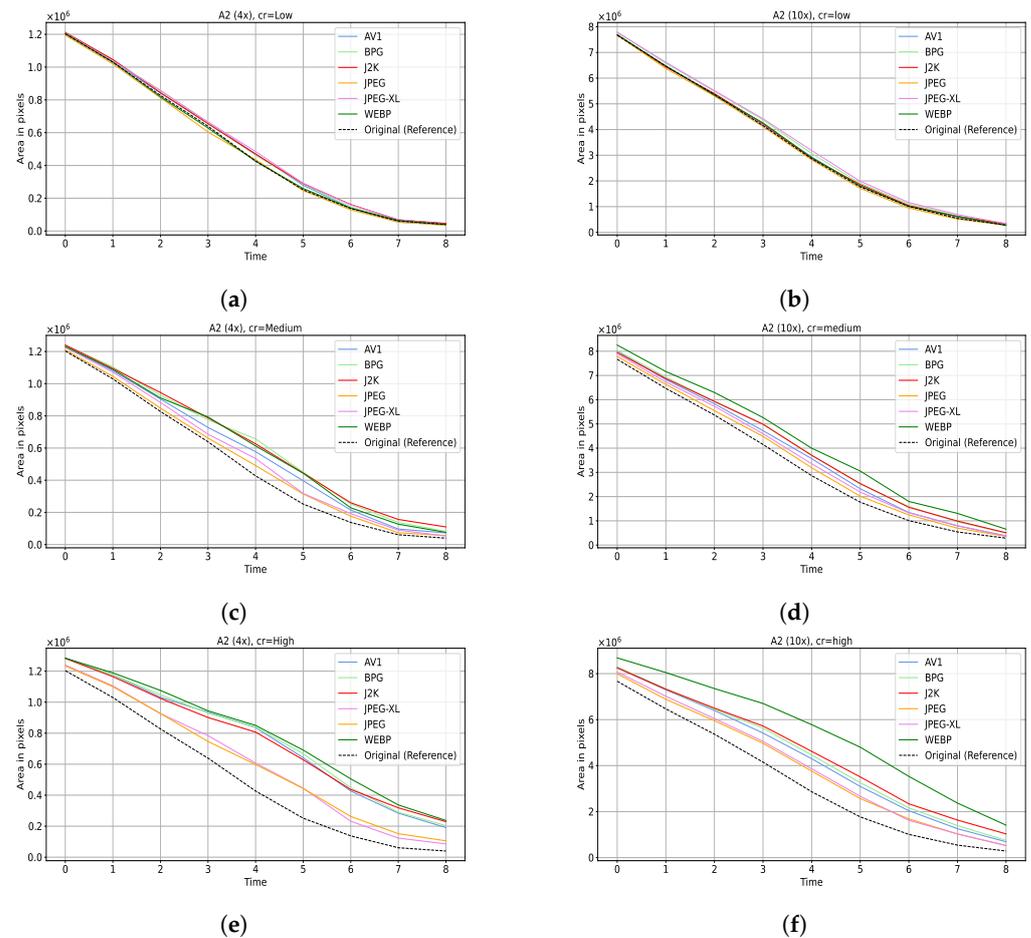


**Figure 18.** Center cropped area of image 6 of the A2 sequence. The highlighted segmented wound area (a) shows the segmentation on the original image, whereas the other three are J2K-compressed images with C (b), B (c), and A (d) compression rates. All images are from the 10× magnification sequence.

The behavior of the *BCAnalyzer* algorithm on the compressed data is shown in the graphs of Figure 20 for the A2 sequence of our dataset. It can be seen that when a low (C) compression rate was applied on the images, the algorithm's output was only slightly affected. There was almost no divergence from the reference result on the uncompressed images, as shown in Figure 20a,b. With increasing compression rate, the divergence increased for all compression types, and the algorithm always overestimated the wound area size in comparison with the reference segmentation on the uncompressed data. The JPEG baseline and JPEG-XL provides the accuracy closest to the uncompressed case; again, WEBP gave the worst results. The behavior was consistent for all sequences independent of the magnification factor.



**Figure 19.** Visualization of the segmentation result of sequence A3 where the jump in the segmented wound area size appeared in the reference result according to the results in Figure 17. (a) The center of image 5 and (b) the center of image 6 of the original images. (c) The J2K-compressed image 6 at high (A) compression rate. All images were from the 10× magnification sequence.



**Figure 20.** Segmentation performance on the C (a,b), B (c,d), and A (e,f) compressed images on sequence A2 using the (adapted) *BCAnalyzer* tool. In the left and right columns, the results are depicted for the  $4\times$  and  $10\times$  magnification images, respectively.

Table 2 shows the algorithms' processing time on SMP Debian 5.10.162-1, with an Intel R-Core(TM) i7-6700 CPU @ 3.40GHz and 32 GB RAM. Note, the entropy filter segmentation and log gradient segmentation were implemented in MATLAB 2019a and the *BCAnalyzer* algorithm in Python 3.

**Table 2.** Average processing time of each algorithm (in seconds) for a single image.

Magnification	Entropy Filter Seg.	Log Gradient Seg.	<i>BCAnalyzer</i>
$4\times$	$5.1 \pm 0.9$	$12.2 \pm 0.5$	$0.8 \pm 0.0$
$10\times$	$26.5 \pm 2.3$	$240.6 \pm 27.3$	$4.0 \pm 0.1$

All result plots for the 10 sequences for both magnifications and algorithms can be downloaded from our web page (<https://wavelab.at/publications.shtml#Jalilian23a> or directly from [https://wavelab.at/papers\\_supplement/Jalilian23a\\_supplement.pdf](https://wavelab.at/papers_supplement/Jalilian23a_supplement.pdf)).

## 7. Conclusions

In this paper, we presented two algorithms for the challenging task of biological cell/wound-area segmentation, as the key step in the analysis of the in vitro wound healing assay. log gradient segmentation showed stable performance on all the images/sequences and both images sizes ( $9824 \times 10,260$  pixels and  $4912 \times 4015$  pixels). Applying this algorithm, we observed that the segmentation (detected pixel counts) at each step/time produced a consistently smaller wound area than their temporal predecessors, which is

coincident with an actual shrinking wound area. The entropy filter segmentation algorithm appeared unstable in detecting the actual wound area under all circumstances, which was mainly due to difficulties in automatically detecting the correct threshold.

Motivated by the large size of the cell images, we applied a set of popular lossy compression algorithms to the images in our dataset, and we studied the compression effect on the segmentation algorithm's accuracy considering three distinct compression bitrates (high (A), medium (B), and low (C)). We observed difference in the compression performance by applying two image quality metrics (PSNR and MS-SSIM). While JPEG-XL and BPG performed the best in terms of MS-SSIM and PSNR, overall, AV1 compression seems to be the better choice. The effect of low (C) and medium (B) compression on the log gradient segmentation was negligible; however, applying certain compression algorithms, which have been proven to cause significant distortions to the reconstructed images (e.g., WEBP, as used in our experiments), caused the algorithm to overestimate the actual wound area. With entropy filter segmentation, high (A) compression seemed to benefit the algorithm in some cases (not yet confirmed with real ground truth data); in contrast to log gradient segmentation, underestimation of the wound area was observed under medium (B) and severe compression. The (adapted) *BCAnalyzer* tool behave similar to log gradient segmentation under compression, however, with a higher spread in terms of the different compression algorithms' effects.

Overall, it is difficult to provide a clear conclusion regarding the best-suited compression strategy, as the best choice is dependent on (i) the segmentation algorithm used and (ii) the actual data sequence being processed. The observed under- or overestimation of the wound area under compression depended on the segmentation algorithm employed. One observation is crucial: the compression rankings as obtained using objective IQMs (with AV1, BPG, and JPEG-XL in the lead) did not necessarily carry over to the compression rankings obtained when applying segmentation to the compressed data (where we regularly observed JPEG among the top-performing algorithms although being poor in terms of IQMs).

**Author Contributions:** Conceptualization, A.U.; methodology, E.J. and M.L.; software, E.J. and M.L.; validation, E.J. and M.L.; formal analysis, E.J. and M.L.; investigation, E.J. and M.L.; resources, E.J. and M.L.; data curation, E.J. and M.L.; writing—original draft preparation, E.J., M.L. and A.U.; writing—review and editing, E.J., M.L. and A.U.; visualization, E.J. and M.L.; supervision, A.U.; project administration, A.U.; funding acquisition, A.U. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Salzburg State Government within the Science and Innovation Strategy Salzburg 2025 (WISS 2025) under the project AIIV-Salzburg (Artificial Intelligence in Industrial Vision), project no 20102-F2100737-FPR.

**Data Availability Statement:** Supplemental material containing all result plots for the main dataset can be downloaded from our web page <https://wavelab.at/publications.shtml#Jalilian23a> or directly from [https://wavelab.at/papers\\_supplement/Jalilian23a\\_supplement.pdf](https://wavelab.at/papers_supplement/Jalilian23a_supplement.pdf).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fong, W.; Chan, S.; Ho, K. Designing JPEG quantization matrix using rate-distortion approach and human visual system model. In Proceedings of the IEEE International Conference on Communications (ICC'97), Montreal, QC, Canada, 12 June 1997; Volume 3, pp. 1659–1663.
2. Chen, M.; Zhang, S.; Karim, M. Modification of standard image compression methods for correlation-based pattern recognition. *Opt. Eng.* **2004**, *43*, 1723–1730.
3. Conrath, J.; Erginay, A.; Giorgi, R.; Leclaire-Collet, A.; Vicaut, E.; Klein, J.C.; Gaudric, A.; Massin, P. Evaluation of the effect of JPEG and JPEG2000 image compression on the detection of diabetic retinopathy. *Eye* **2007**, *21*, 487–493. [[CrossRef](#)]
4. Elmer, P.; Häfner, M.; Tamaki, T.; Tanaka, S.; Thaler, R.; Uhl, A.; Yoshida, S. Impact of Lossy Image Compression on CAD Support Systems for Colonoscopy. In Proceedings of the Computer-Assisted and Robotic Endoscopy (CARE'15), Munich, Germany, 5 October 2015; Springer Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9515, pp. 1–11.

5. Rakshit, S.; Monro, D. An Evaluation of Image Sampling and Compression for Human Iris Recognition. *IEEE Trans. Inf. Forensics Secur.* **2007**, *2*, 605–612. [[CrossRef](#)]
6. Matschitsch, S.; Tschinder, M.; Uhl, A. Comparison of Compression Algorithms' Impact on Iris Recognition Accuracy. In Proceedings of the Proceedings of the 2nd International Conference on Biometrics 2007 (ICB'07), Seoul, Republic of Korea, 27–29 August 2007; Lee, S.W., Li, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4642, pp. 232–241.
7. Daugman, J.; Downing, C. Effect of Severe Image Compression on Iris Recognition Performance. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 52–61. [[CrossRef](#)]
8. Ives, R.W.; Broussard, R.P.; Kennell, L.R.; Soldan, D.L. Effects of image compression on iris recognition system performance. *J. Electron. Imaging* **2008**, *17*, 011015. [[CrossRef](#)]
9. Ives, R.W.; Bishop, D.; Du, Y.; Belcher, C. Iris Recognition: The Consequences of Image Compression. *EURASIP J. Adv. Signal Process.* **2010**, *2010*, 680845. [[CrossRef](#)]
10. Grother, P. Quantitative Standardization of Iris Image Formats. In Proceedings of the Biometrics and Electronic Signature Special Interest Group Conference (BIOSIG 2009), Darmstadt, Germany, 17–18 September 2009; pp. 143–154.
11. Figueroa-Villanueva, M.; Ratha, N.; Bolle, R. A comparative performance analysis of JPEG2000 vs. WSQ for fingerprint compression. In *Proceedings of the AVBPA*; Kittler, J., Nixon, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2688, pp. 385–392.
12. Kidd, R. Comparison of wavelet scalar quantization and JPEG for fingerprint image compression. *J. Electron. Imaging* **1995**, *4*, 31–39. [[CrossRef](#)]
13. Mascher-Kampfer, A.; Stögner, H.; Uhl, A. Comparison of compression algorithms' impact on fingerprint and face recognition accuracy. In Proceedings of the Visual Communications and Image Processing 2007 (VCIP'07), San Jose, CA, USA, 28 January–1 February 2007; Chen, C., Schonfeld, D., Luo, J., Eds.; Number 6508 in Proceedings of SPIE; SPIE: San Jose, CA, USA, 2007; p. 650810.
14. Funk, W.; Arnold, M.; Busch, C.; Munde, A. Evaluation of image compression algorithms for fingerprint and face recognition systems. In Proceedings of the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop, West Point, NY, USA, 15–17 June 2006; Cole, J., Wolthusen, S., Eds.; IEEE Computer Society: Washington, DC, USA, 2006; pp. 72–78.
15. Ablinger, V.; Zenz, C.; Hämmerle-Uhl, J.; Uhl, A. Compression Standards in Fingerprint Recognition. In Proceedings of the 9th IAPR/IEEE International Conference on Biometrics (ICB'16), Halmstad, Sweden, 13–16 June 2016; pp. 1–7.
16. Kauba, C.; Uhl, A. Robustness Evaluation of Hand Vein Recognition Systems. In Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'15), Darmstadt, Germany, 9–11 September 2015; pp. 1–8. [[CrossRef](#)]
17. Lipowski, T.; Maser, B.; Hämmerle-Uhl, J.; Uhl, A. Finger-vein Sample Compression in Presence of Pre-Compressed Gallery Data. In Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'18), Darmstadt, Germany, 26–28 September 2018; pp. 1–8.
18. Schörgenhofer, K.; Dafir, T.S.; Uhl, A. Multi-sample Compression of Finger Vein Images using H.265 Video Coding. In Proceedings of the 12th IAPR/IEEE International Conference on Biometrics (ICB'19), Crete, Greece, 4–7 June 2019; pp. 1–8.
19. Granai, L.; Tena, J.; Hamouz, M.; Kittler, J. Influence of compression on 3D face recognition. *Pattern Recognit. Lett.* **2009**, *30*, 745–750. [[CrossRef](#)]
20. Delac, K.; Grgic, S.; Grgic, M. Image Compression in Face Recognition—A Literature Survey. In *Recent Advances in Face Recognition*; Delac, K., Grgic, M., Bartlett, M., Eds.; I-Tech: Vienna, Austria, 2008; pp. 236–250.
21. Jeong, G.M.; Kim, C.; Ahn, H.S.; Ahn, B.J. JPEG Quantization Table Design for Face Images and Its Application to Face Recognition. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2006**, *E69-A*, 2990–2993. [[CrossRef](#)]
22. Mascher-Kampfer, A.; Stögner, H.; Uhl, A. Multiple Re-Watermarking Scenarios. In Proceedings of the 13th International Conference on Systems, Signals, and Image Processing (IWSSIP 2006), Budapest, Hungary, 21–23 September 2006; pp. 53–56.
23. Delac, K.; Grgic, M.; Grgic, S. Face recognition in JPEG and JPEG2000 compressed domain. *Image Vis. Comput.* **2009**, *27*, 1108–1120. [[CrossRef](#)]
24. Quinn, G.; Grother, P.J. *Technical Report NIST Interagency Report 7830*; Information Technology Laboratory, The National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.
25. Kamasack, M.; Sankur, B. Face recognition under lossy compression. In Proceedings of the International Conference on Pattern Recognition and Information Processing (PRIP'99), Minsk, Belarus, 18–20 May 1999; pp. 27–32.
26. Klare, B.; Burge, M. Assessment of H.264 Video Compression on Automated Face Recognition Performance in Surveillance and Mobile Video Scenarios. In *Proceedings of the Biometric Technology for Human Identification VII*; Kumar, V., Prabhakar, S., Ross, A., Eds.; Proceedings of SPIE; SPIE: Bellingham, WA, USA, 2010; Volume 7667, p. 76670X.
27. Korshunov, P.; Ooi, W. Video quality for face detection, recognition, and tracking. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2011**, *7*, 14. [[CrossRef](#)]
28. Elmer, P.; Lupp, A.; Sprenger, S.; Thaler, R.; Uhl, A. Exploring Compression Impact on Face Detection Using Haar-like Features. In Proceedings of the 19th Scandinavian Conference on Image Analysis (SCIA'15), Copenhagen, Denmark, 15–17 June 2015; Volume 9127, pp. 53–64.
29. Jalilian, E.; Hofbauer, H.; Uhl, A. Deep Iris Compression. In *Proceedings of the Pattern Recognition—ICPR International Workshops and Challenges, Proceedings, Part V*; Springer: Milan, Italy, 2021; Volume 12565, pp. 1–15.

30. Jalilian, E.; Hofbauer, H.; Uhl, A. Iris Image Compression Using Deep Convolutional Neural Networks. *Sensors* **2022**, *22*, 2698. [[CrossRef](#)]
31. Rathgeb, C.; Uhl, A.; Wild, P. Effects of Severe Image Compression on Iris Segmentation Performance (Best Poster Award). In Proceedings of the IAPR/IEEE International Joint Conference on Biometrics (IJCB'14), Clearwater, FL, USA, 29 September–2 October 2014.
32. Herzog, T.; Uhl, A. JPEG Optimisation for Fingerprint Recognition: Generalisation Potential of an Evolutionary Approach. In Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'15), Darmstadt, Germany, 9–11 September 2015; p. 8. *accepted*.
33. Konrad, M.; Stögner, H.; Uhl, A. Custom Design of JPEG quantization tables for compressing iris polar images to improve recognition accuracy. In Proceedings of the 3rd International Conference on Biometrics 2009 (ICB'09), Alghero, Italy, 2–5 June 2009; Tistarelli, M., Nixon, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5558, pp. 1091–1101.
34. Konrad, M.; Stögner, H.; Uhl, A. Evolutionary Optimization of JPEG Quantization Tables for Compressing Iris Polar Images in Iris Recognition Systems. In Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, ISPA '09, Salzburg, Austria, 16–18 September 2009.
35. Kostmayer, G.; Stögner, H.; Uhl, A. Custom JPEG Quantization for Improved Iris Recognition Accuracy. In Proceedings of the Emerging Challenges for Security, Privacy and Trust—Proceedings of the 24th IFIP International Information Security Conference 2009 (IFIP SEC'09), Pafos, Cyprus, 18–20 May 2009; Gritzalis, D., Lopez, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 297, pp. 76–86.
36. Hämmerle-Uhl, J.; Karnutsch, M.; Uhl, A. Evolutionary Optimisation of JPEG2000 Part 2 Wavelet Packet Structures for Polar Iris Image Compression. In Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP'13), Havana, Cuba, 20–23 November 2013; Volume 8258, pp. 391–398.
37. Mühlbacher, B.; Stütz, T.; Uhl, A. JPEG2000 Part 2 wavelet packet subband structures in fingerprint recognition. In Proceedings of the Visual Communications and Image Processing 2010 (VCIP'10), Huangshan, China, 11–14 July 2010; Frossard, P., Li, H., Wu, F., Girod, B., Li, S., Wei, G., Eds.; Number 7744 in Proceedings of SPIE; SPIE: Bellingham, WA, USA, 2010; p. 77442C.
38. Horvath, K.; Stögner, H.; Uhl, A. Optimisation of JPEG XR quantisation settings in iris recognition systems. In Proceedings of the 4th International Conference on Advances in Multimedia (MMEDIA 2012), Chamonix/Mont Blanc, France, 29 April–4 May 2012; pp. 88–93.
39. Belloulata, K.; Baskurt, A.; Benoit-Cattin, H.; Prost, R. Fractal coding of medical images. In *Proceedings of the Medical Imaging 1996: Image Display*; Kim, Y., Ed.; SPIE Proceedings; SPIE: Newport Beach, CA, USA, 1996; Volume 2707, pp. 598–609.
40. Panych, L. Theoretical comparison of Fourier and Wavelet Encoding in Magnetic Resonance Imaging. *IEEE Trans. Med Imaging* **1997**, *15*, 141–153. [[CrossRef](#)]
41. European Society of Radiology (ESR). Usability of irreversible image compression in radiological imaging. A position paper by the European Society of Radiology (ESR). *Insights Imaging* **2011**, *2*, 103–115. [[CrossRef](#)]
42. Schoeffmann, K.; Fabro, M.D.; Böszörményi, L.; Smulders, J.; Jakimowicz, J. Investigation of the Impact of Compression on the Perceptual Quality of Laparoscopic Videos. In Proceedings of the 27th International Symposium on Computer-Based Medical Systems (CBMS'14), New York, NY, USA, 27–29 May 2014; pp. 153–158.
43. Cosman, P.C.; Gray, R.M.; Olshen, R.A. Evaluating quality of compressed medical images: SNR, Subjective Rating, and Diagnostic Accuracy. *Proc. IEEE* **1994**, *82*, 919–932. [[CrossRef](#)]
44. Konsti, J.; Lundin, M.; Linder, N.; Haglund, C.; Blomqvist, C.; Nevanlinna, H.; Aaltonen, K.; Nordling, S.; Lundin, J. Effect of image compression and scaling on automated scoring of immunohistochemical stainings and segmentation of tumor epithelium. *Diagn. Pathol.* **2012**, *7*, 29. [[CrossRef](#)]
45. Santalla, H.; Meschino, G.; Ballarin, V. Effects on MR images compression in tissue classification quality. *J. Phys. Conf. Ser.* **2007**, *90*, 012061. [[CrossRef](#)]
46. Elmer, P.; Häfner, M.; Tamaki, T.; Tanaka, S.; Thaler, R.; Uhl, A.; Yoshida, S. Compression-scenarios for LIRE-based CBIR on colonoscopy data. In Proceedings of the Bildverarbeitung für die Medizin 2016 (BVM'16), Berlin, Germany, 13–15 March 2016; pp. 152–157.
47. Chao, J.; Al-Nuaimi, A.; Schroth, G.; Steinbach, E. Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with JPEG-encoded query images. In Proceedings of the 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, Italy, 30 September–2 October 2013; pp. 029–034. [[CrossRef](#)]
48. Schäfer, G. Does compression affect image retrieval performance? *Int. J. Imaging Syst. Technol.* **2008**, *18*, 101–112. [[CrossRef](#)]
49. Sultana, Z.; Nahar, L.; Tasnim, F.; Hossain, M.S.; Andersson, K. Lossy Compression Effect on Color and Texture Based Image Retrieval Performance. In *Proceedings of the Intelligent Computing & Optimization*; Vasant, P., Weber, G.W., Marmolejo-Saucedo, J.A., Munapo, E., Thomas, J.J., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 1159–1167.
50. Zabala, A.; Pons, X. Effects of lossy compression on remote sensing image classification of forest areas. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 43–51. [[CrossRef](#)]
51. Garcia-Vichez, F.; Munoz-Mari, M.Z.J.; Blanes, I.; Gonzales-Ruiz, V.; Camps-Valls, G. On the Impact of Lossy Compression on Hyperspectral Image Classification and Unmixing. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 253–257. [[CrossRef](#)]
52. Lau, W.L.; Li, Z.L.; Lam, W.K. Effects of JPEG compression on image classification. *Int. J. Remote Sens.* **2003**, *24*, 1535–1544. [[CrossRef](#)]

53. Joechl, R.; Uhl, A. Effects of Image Compression on Image Age Approximation. In Proceedings of the 20th International Workshop on Digital-Forensics and Watermarking (IWDW2021), Beijing, China, 20–22 November 2021; pp. 102–116. [\[CrossRef\]](#)
54. Remy, O.; Strumegger, S.; Hämmerle-Uhl, J.; Uhl, A. Comparative Compression Robustness Evaluation of Digital Image Forensics. In Proceedings of the Computational Science and Its Applications—ICCSA 2022, Malaga, Spain, 4–7 July 2022; Volume 13376, pp. 236–246.
55. Pomarico, E.; Schmidt, C.; Chays, F.; Nguyen, D.; Planchette, A.; Tissot, A.; Roux, A.; Pagès, S.; Batti, L.; Clausen, C.; et al. Quantifying the effect of image compression on supervised learning applications in optical microscopy. *arXiv* **2020**, arXiv:2009.12570. [\[CrossRef\]](#)
56. Chen, Y.; Janowczyk, A.; Madabhushi, A. Quantitative Assessment of the Effects of Compression on Deep Learning in Digital Pathology Image Analysis. *JCO Clin. Cancer Inform.* **2020**, *4*, 221–233. [\[CrossRef\]](#)
57. López, C.; Lejeune, M.; Bosch, P.E.R.; Salvadó, M.T.; Pons, L.; Baucells, J.; Cugat, X.; Álvaro, T.; Jaén, J. Effects of image compression on automatic count of immunohistochemically stained nuclei in digital images. *J. Am. Med. Inform. Assoc.* **2008**, *15*, 794–798. [\[CrossRef\]](#)
58. Lejeune, M.; López, C.; Bosch, R.; Korzyńska, A.; Salvadó, M.T.; García-Rojo, M.; Neuman, U.; Witkowski, Ł.; Baucells, J.; Jaén, J. JPEG2000 for automated quantification of immunohistochemically stained cell nuclei: A comparative study with standard JPEG format. *Virchows Arch.* **2011**, *458*, 237–245. [\[CrossRef\]](#)
59. Dong, Y.; Pan, W.D.; Wu, D. Impact of Misclassification Rates on Compression Efficiency of Red Blood Cell Images of Malaria Infection Using Deep Learning. *Entropy* **2019**, *21*, 1062. [\[CrossRef\]](#)
60. Pomarico, E.; Schmidt, C.; Chays, F.; Nguyen, D.; Planchette, A.; Tissot, A.; Roux, A.; Pagès, S.; Batti, L.; Clausen, C.; et al. Statistical distortion of supervised learning predictions in optical microscopy induced by image compression. *Sci. Rep.* **2022**, *12*, 3464. [\[CrossRef\]](#)
61. Liang, C.C.; Park, A.Y.; Guan, J.L. In vitro scratch assay: A convenient and inexpensive method for analysis of cell migration in vitro. *Nat. Protoc.* **2007**, *2*, 329–333. [\[CrossRef\]](#)
62. Jonkman, J.E.N.; Cathcart, J.A.; Xu, F.; Bartolini, M.E.; Amon, J.E.; Stevens, K.M.; Colarusso, P. An introduction to the wound healing assay using live-cell microscopy. *Cell Adh. Migr.* **2014**, *8*, 440–451. [\[CrossRef\]](#)
63. Grada, A.; Otero-Vinas, M.; Prieto-Castrillo, F.; Obagi, Z.; Falanga, V. Research Techniques Made Simple: Analysis of Collective Cell Migration Using the Wound Healing Assay. *J. Investig. Dermatol.* **2017**, *137*, e11–e16. [\[CrossRef\]](#)
64. Suarez-Arnedo, A.; Figueroa, F.T.; Clavijo, C.; Arbeláez, P.; Cruz, J.C.; Muñoz-Camargo, C. An image J plugin for the high throughput image analysis of in vitro scratch wound healing assays. *PLoS ONE* **2020**, *15*, e0232565. [\[CrossRef\]](#)
65. Gebäck, T.; Schulz, M.M.P.; Koumoutsakos, P.; Detmar, M. TScratch: A novel and simple software tool for automated analysis of monolayer wound healing assays. *Biotechniques* **2009**, *46*, 265–274. [\[CrossRef\]](#) [\[PubMed\]](#)
66. Glaß, M.; Möller, B.; Zirkel, A.; Wächter, K.; Hüttelmaier, S.; Posch, S. Scratch Assay Analysis with Topology-Preserving Level Sets and Texture Measures. In *Proceedings of the Pattern Recognition and Image Analysis*; Vitrià, J., Sanches, J.M., Hernández, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 100–108.
67. Glaß, M.; Möller, B.; Zirkel, A.; Wächter, K.; Hüttelmaier, S.; Posch, S. Cell migration analysis: Segmenting scratch assay images with level sets and support vector machines. *Pattern Recognit.* **2012**, *45*, 3154–3165. [\[CrossRef\]](#)
68. Topman, G.; Sharabani-Yosef, O.; Gefen, A. A standardized objective method for continuously measuring the kinematics of cultures covering a mechanically damaged site. *Med. Eng. Phys.* **2012**, *34*, 225–232. [\[CrossRef\]](#) [\[PubMed\]](#)
69. Möller, B.; Posch, S. Comparing active contours for the segmentation of biomedical images. In Proceedings of the 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), Barcelona, Spain, 2–5 May 2012; pp. 736–739. [\[CrossRef\]](#)
70. Cortesi, M.; Pasini, A.; Tesei, A.; Giordano, E. AIM: A Computational Tool for the Automatic Quantification of Scratch Wound Healing Assays. *Appl. Sci.* **2017**, *7*, 1237. [\[CrossRef\]](#)
71. Javer, A.; Rittscher, J.; Sailem, H.Z. DeepScratch: Single-cell based topological metrics of scratch wound assays. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 2501–2509. [\[CrossRef\]](#) [\[PubMed\]](#)
72. Sinitca, A.M.; Kayumov, A.R.; Zelenikhin, P.V.; Porfiriev, A.G.; Kaplun, D.I.; Bogachev, M.I. Segmentation of patchy areas in biomedical images based on local edge density estimation. *Biomed. Signal Process. Control* **2023**, *79*, 104189. [\[CrossRef\]](#)
73. Ding, K.; Xiao, L.; Weng, G. Active contours driven by region-scalable fitting and optimized Laplacian of Gaussian energy for image segmentation. *Signal Process.* **2017**, *134*, 224–233. [\[CrossRef\]](#)
74. Gonzalez, R.C.; Woods, R.E.; Eddins, S.L. *Digital Image Processing Using Matlab*; Prentice Hall: Hoboken, NJ, USA, 2003.
75. Wang, Z.; P-Simoncelli, E.; C-Bovik, A. Multiscale structural similarity for image quality assessment. In Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 9–12 November 2003; Volume 2, pp. 1398–1402.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.