Automated crack segmentation via saturation channel thresholding, area classification and fusion of modified level set segmentation with Canny edge detection

Uche A. Nnolim*
Department of Electronic Engineering, Faculty of Engineering, University of Nigeria, Enugu, Nigeria

ARTICLE INFO
Keywords:
Computer science
Modified active contour energy function
Adaptive region-based segmentation
Adaptive threshold-based edge detection binary fusion segmentation
Fuzzy c-means clustering area classification-based morphological processing
Saturation component-based thresholding
Wavelet multiscale local-global enhancement

ABSTRACT
Automatic detection of complex cracks on rough concrete surfaces via image processing is a challenging task. The most current effective methods involve deep learning schemes. These are usually computationally and structurally complex. Recently, relatively simplified algorithms were developed for effective segmentation of crack features. However, these approaches still could not consistently and accurately extract such features from extremely noisy images of rough concrete surfaces with complex crack patterns. This study describes crack feature segmentation algorithms based on wavelet coefficient adjustment, nonlinear filter pre-processing, saturation channel extraction, adaptive threshold-based edge detection and fuzzy clustering-based area classification. Additional modifications include a new energy function for active contour segmentation algorithm. Adaptive localized mask generation is also proposed for automatic region-based segmentation. Furthermore, a binary fusion stage is incorporated for improved edge feature extraction. The quantitative and visual evaluation of the proposed schemes show improvement in results compared to several recent state-of-the-art algorithms.

1. Introduction
Automated segmentation of crack features from images of concrete surfaces is still an important problem in the field. Various schemes utilize images obtained from manned or unmanned vehicles [1]. Majority of research in this area focuses on relatively simple crack patterns on smooth surfaces. Additionally, several datasets are composed of high resolution images where the cracks have more refined contrast compared to the background texture. This leads to failure of most algorithms when confronted with images acquired under non-ideal conditions. Such image scenes exhibit considerable uneven illumination, extremely coarse or rough surfaces, stains, line markings, smears and complex crack patterns. These issues are compounded by poor image resolution, extremely poor contrast between foreground and background and compression artifacts in some cases. These degradations are difficult to resolve with conventional thresholding and segmentation algorithms. Furthermore, deep learning architectures for combating these problems are constantly increasing in complexity. Thus in this study, we aim to show that such problematic images can be handled by relatively less complex image processing, mathematical and machine learning-based hybrids. Two algorithms, which incorporate these concepts are presented for addressing both the general and specific problems outlined. The paper is outlined as follows; section 2 presents the background, motivation, relevant concepts and key contributions. Section 3 initiates preliminary analysis of core components, methodology and justification for selection. Section 4 presents the proposed schemes and their essential features. Section 5 provides the experimental results and comparisons. The final section contains the conclusion.

2. Background and motivation
Crack detection approaches include basic (pixel-based) thresholding [10, 11, 12], (region-based) edge detection [4], morphological processing, and (texture-based) filtering (Gabor) [13]. Others are transform-based (wavelets), evolutionary (genetic) algorithms [14] and machine learning schemes [15]. Recently, deep learning-based approaches have been extensively reported [2, 3, 4, 5, 6, 7, 16, 17]. The simpler methods are fast but less effective, while the complex and deep-learning approaches are usually computationally intensive and require massively parallel processors. They also require large datasets and extensive runtime to train the networks effectively. Moreover, it is difficult to obtain datasets with mainly complex crack patterns on rough...

* Corresponding author.
E-mail address: uche.nnolim@unn.edu.ng.

https://doi.org/10.1016/j.heliyon.2020.e05748
Received 17 July 2020; Received in revised form 10 October 2020; Accepted 14 December 2020
2405-8440/© 2020 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
We explore the following methods and concepts in order to exploit their advantages. These will hopefully grant us insights in improving performance of crack feature segmentation for standard and problematic image scenes.

2.1. Relevant concepts

2.1.1. Active contour segmentation

The active contour segmentation (ACS) algorithm is a popular PDE-based segmentation technique employing level set curves [15]. Several variants of ACS can be essentially classified as either edge- and region-based active contour models [27]. The edge-based models utilize image gradient data for operation [27]. However, this implies that problems arise for images that exhibit noisy and weak/smoothened edges [27]. The region-based models utilize statistical and curvature data within and outside the contour for operation [27]. These ensure robustness to images exhibiting noisy and blurred edges [27]. The most popular formulations include the Mumford-Shah [28] and Chan-Vese models [28, 29, 30]. We utilize the Chan Vese model, which utilizes the average intensity computed across the image region [27]. The ACS algorithm is not well suited to segmentation of crack features due to their unique properties unlike medical images. This challenge is addressed in subsequent discussions.

2.1.2. Colour coordinate systems

Colour space conversions are vital in numerous computer vision and image processing applications [32][33]. They have been invaluable in computational photography, underwater and haze image processing, High Dynamic Range imaging, segmentation, etc. Classes include additive colour systems such as red-green-blue (RGB) and subtractive types such as cyan-magenta-yellow (CMY) [32, 33]. Others include video colour systems such as YIQ and non-linear colour space transformations such as hue-saturation-intensity (HSI) [32, 33]. It is hoped that the colour images of cracks would provide insight into certain key features, which may be lost in the colour to greyscale conversion process.

2.2. Motivation

In previous work, we encountered problems when segmenting images of complex crack patterns on rough concrete surfaces [8]. Attempts from basic thresholding to combined smoothing and edge detection failed for such images. We then developed the crack image enhancement and detection (CIEAD-1 and CIEAD-2) algorithms to rectify this issue [9]. There were improvements using the latter scheme, however, some images with complex crack patterns were still difficult to segment automatically without deep learning approaches. These difficult images also exhibited uneven illumination and poor contrast in addition to the rough surfaces. Thus, we hope to advance the work respecting these particular images in this study. There are two proposed methods. The first scheme combines the results of level set segmentation with edge detection to yield an improved result. The second algorithm exploits the saturation component in the hue-saturation-intensity (HSI) colour space coupled with edge detection and area classification for morphological operations.

2.3. Key contributions

Key contributions and features of the proposed algorithms include:

- New energy function for active contour segmentation combined with adaptive mask computation and generation to reduce computation time of active contour segmentation algorithm (faster convergence).
- Multi-scale, local-global contrast and detail enhancement with adaptive shock filter for simultaneous smoothing and sharpening to minimize rough textures, while preserving crack features and binary fusion of edges and segmented results to improve detected crack features.
- Adaptive binary threshold computation via Otsu method for Canny edge detection to avoid arbitrary tuning and adaptive area classification assignment (obtained via binary image structure labelling and fuzzy c-means clustering) for determining size of structuring element utilized in morphological operations.
- Saturation channel processing and pseudo-thresholding for rough texture and uneven illumination elimination.

3. Preliminary analysis and methodology

We proceed with prior analysis of various smoothing, colour space conversion, contrast enhancement, edge detection and thresholding algorithms to assess performance, which forms part of the selection criteria.

3.1. Analysis of relevant smoothing algorithms

We compare the various algorithms such as FABD, TVR and their variants, specifically the original TVR (ROFTV) and the adaptive fractional order TVR (FOTVR) [34]. The sample results are shown in Figure 1. Results indicate that variational methods yield best smoothing of rough background textures as evidenced in previous work [9].

3.2. Colour space comparisons

We compare colour space conversions in HSI, YCbCr, YUV, YIQ, XYZ and HSV domain [33]. The results are shown in Figure 2 and indicate that HSI colour space yields the best outcomes. This is followed by the HSV domain in terms of retaining most of the crack features, while minimizing background texture information. The YUV and YIQ conversions also remove the background texture features but also retain fewer crack feature regions in the images. The YCbCr and XYZ colour spaces still retain the undesired background surface textures. Based on visual observation, we select the HSI domain for further analysis.

3.2.1. HSI colour space analysis

We need to fully understand what the hue and saturation channel depicts and how this is useful to crack feature interpretation. Typically, the initial stage of crack feature segmentation algorithms involves converting the RGB colour image to greyscale. Converting the RGB image to HSI/HSV colour space yields surprising outcomes. Firstly, most RGB colour images of cracks are of low resolution with minimal surface texture roughness, hues and colours. The hue channel is blocky when observed in HSI/HSV colour space. The intensity channel yields the typical result used in image-based crack detection pre-processing schemes. We will also analyze the histograms of the HSI images of cracks. Figure 3 shows the effects of processing in RGB and HSI space. The histograms of the original and enhanced HSI images indicate that the contrast enhancement can be used to separate out the crack feature from the background. The RGB histogram indicates that enhancement in RGB domain is ineffective. The HSI histogram indicates that the saturation
channel is the desired information band of the image. Also, the histogram shows an almost fully segmented image in HSI space. Further analysis of this channel may result in better detection of crack features. Thus, the choice of HSI colour space is validated and we proceed to compare edge detection results in the HSI domain.

3.2.2. Edge detector comparisons

In this section we evaluate the efficacy of several linear and nonlinear edge filters on the saturation channel regarding clarity of extracted crack features. The nonlinear edge filter algorithms include variance, interquartile range, mean difference, relative mean difference, median absolute deviation, range and standard deviation filters [9, 33]. For the linear edge filters, we test the Sobel, Prewitt, Roberts, Canny, Laplacian of Gaussian (LoG) and zero-crossing algorithms [33]. A sample of the results are shown in Figure 4. The median difference filter appears to be the most effective non-linear edge filter for the saturation channel in detecting crack features in Figure 4(a). In Figure 4(b), the Canny edge detection yields the best results. Thus, we select the Canny edge filter in this case. Also, it appears that the contrast stretched image is better than histogram equalized image for crack detection, since it minimizes illumination distortion and over-enhancement.

3.2.3. Contrast enhancement comparisons

We also compare contrast enhancement algorithms and their effectiveness in separating the crack foreground from the background image. We test several algorithms including standard and PDE-based formulations of minimax (PDE_MINMAX), global, adaptive and contrast limited adaptive histogram equalization (GHE, AHE, CLAHE, PDE_GHE, PDE_AHE, PDE_CLAHE), specification (HS, PDE_HS) [36, 37, 38, 39]. Other conventional and PDE versions are piece-wise linear transform (PWL, PDE_PWL), contrast stretching (CS, PDE_CS) and enhancement (PDE_CE) [36, 37, 38, 39]. We also compare Anisotropic Diffusion enhancement (ADE1, ADE2, AD_GOC3), spatial and frequency domain homomorphic filter (SHF, FDHF, PDE enhancement + HF) [36, 37, 38, 39]. Also included were typical and PDE-based versions of gain offset correction (GOC1, GOC2, GOC3, PDE_GOC2, PDE_GOC3), tonal correction (TC), splitting signal (SSAR), generalized un-sharp masking (GUM), single and multiscale Retinex (SSR, MSR, MSRCR) [36, 37, 38, 39]. New developed algorithms added were the global-local PDEs (PDE-GOC-CLAHE, fast PDE-GOC-CLAHE), adaptive clip limit PDE-GOC-CLAHE (ACL-PDE-GOC-CLAHE), GOC-regularized entropy optimized adaptive clip limit PDE (EG-PDE-GOC-CLAHE) [36, 37, 38, 39].

The results are shown for the RGB and corresponding HSI images in Figure 5. The clarity of the crack features indicates the best algorithms for crack detection in HSI domain using saturation channel. However, most of the algorithms do not handle uneven illumination well as seen in Figure 5(a). This shortcoming is eliminated by utilizing the saturation channel as seen in Figure 5(b). Also, the algorithms that emphasize local contrast more than global contrast improve the clarity of the crack features.

3.2.4. Analysis of relevant thresholding algorithms for crack detection

In this section we analyze results of various thresholding algorithms applied to crack detection. The thresholding algorithms tested include Otsu [40], Fuzzy C-means [41], Liu et al [42], Raja et al [43]. Additional algorithms include adaptive threshold [44], Bradley threshold [45], Triangle threshold [46], Differential evolution [47] and method by Sarkar et al [48]. Others are the Entropy threshold [49], Fuzzy entropy method [50], Gonzalez method [33], mean threshold, multi-threshold [51], Niblack [52], Nick [53], Sauvola [54], Raja et al [43], standard deviation and the Wolf thresholding algorithm.

The results are shown in Figure 6 for comparison of various thresholding algorithms. Results indicate that the contrast enhancement and saturation channel processing enhances thresholding results for most algorithms. This also eliminates the shadows that interfere with the thresholding process. The crack features are clearly outlined in most cases, ensuring minimal morphological steps to clean the images.
However, the level of success is still dependent on the intrinsic image features. For example, images with relative minimal differences between foreground and background contrast with rough texture require additional processing as in previous work [9]. Furthermore, small, low-resolution images of concrete surfaces with rough cracks, which exhibit the aforementioned properties are the hardest to extract crack features from [9]. However, the intermediate saturation channel can be utilized as input to a relatively complex neural network. This would reduce the number of layers needed for improved results.

In summary, we have analyzed the effects of:

- Colour spaces such as YUV, YCbCr and YIQ [33] on the saturation channel of RGB colour images of pavement surfaces with cracks.
- Linear and non-linear edge detectors on the saturation channel of colour images with cracks.
- Contrast enhancement on the saturation channel with respect to crack feature amplification.
- Threshold value on the area of detected crack features in images using various edge detection algorithms.
- Various thresholding algorithms on images of cracks.

### 4. Proposed approaches

We now present in detail, the two algorithms for crack detection; the first algorithm involves the fusion of edge detection and active contour segmentation results via adaptive mask. The second method utilizes the HSI colour space and saturation channel for segmentation and thresholding. This is then combined with Canny edge detection and morphological refinement using iterative fuzzy clustering and area classification.

#### 4.1. Binary fusion of edge detection and adaptive mask-based active contour segmentation

The key stages of the first proposed algorithm (PA1) include the pre-processing, analysis and segmentation and post-processing stages. The pre-processing stage consists of illumination correction, contrast enhancement and smoothing. The edge map and binary mask generation followed by modified active contour segmentation comprise the analysis and segmentation stage. The post-processing stage is composed of the fusion and morphological operations.
The flowchart of PA1 is shown in Figure 7, which is the generalized algorithm for handling all types of colour or greyscale crack images. This is the reason the algorithm appears to be heavily involved. The RGB2gray operation is performed if the input is a RGB colour image, otherwise this stage is bypassed. Wavelet contrast enhancement is utilized for images with faint cracks. Also it is useful for images where background texture intensity level is less than or equal to the intensity level of the crack features. The Wavelet scheme enhances the crack detail features, while minimizing the values of the approximation coefficients at multiple scales. However, this increases noise slightly and the Super-Resolution Shock filter addresses this by suppressing the noise, while preserving and enhancing details. This is followed by the computation of the Otsu threshold, but avoiding binarization. The Otsu threshold is simply a scalar value in the range 0–1 and the graythresh function in MATLAB performs this operation. This threshold value is then used as an input parameter for the Canny edge detection.

Figure 4. Crack detection results using (a) nonlinear filters [35] and (b) linear filters acting on the saturation channel threshold.
The obtained binary edge map is then labeled and region area properties are computed from which we derive the minimum area. The image is then processed with closing, filling and dilation operations using disk structuring element set to the size of the minimum area. This results in the refined edge map being the first fusion input.

Then the initial greyscale image is segmented with the modified ACS algorithm using the computed edge map. Additional iterative computation of area ratio using cluster centers is performed to check for any new unwanted features incorrectly introduced by the ACS. If there are spurious features, additional morphological operations using computed area ratio are performed until a clean image is obtained. The algorithm also checks for the image size. Large images, which are classed in the megapixel range and higher do not require all the steps shown. Such images are usually easier to segment with simple to intermediate steps in the algorithm. Thus, the pre-processing stage and most of the analysis and post-processing stages will be bypassed. The segmentation stage remains the critical path for most crack images with ideal features.

4.1.1. Pre-processing stage

We first convert the image to grayscale if it is an RGB colour image to reduce data and processing time. Then we perform illumination...
Figure 6. Comparison of various thresholding algorithms for (a) Intensity channel (b) Saturation channel (c) Key to figures.
correction using a previously developed fuzzy homomorphic enhancement algorithm [8, 55, 56]. This allows the normalization of illumination without any edge or noise enhancement. This is then followed by contrast enhancement using a multi-level wavelet decomposition scheme [9, 57]. This multi-scale approach improves the local and global contrast of the image. With the normalized illumination, this process makes the crack features much more prominent without increasing the background texture contrast. We subsequently perform nonlinear smoothing to de-emphasize the roughness of the background texture while preserving the crack features. This is achieved using the adaptive shock filter algorithm proposed by Xiao et al [26].

4.1.2. Analysis and segmentation stage: edge map generation

We obtain the edge map from the smoothed image using the Canny edge detection algorithm [58]. Using a less suitable edge detector would enhance the noise and rough texture features that were not completely suppressed by the smoothing stage. Furthermore, we utilize the adaptive threshold computation of the Otsu method to obtain the threshold value to be utilized in the Canny edge detection algorithm.

4.1.3. Analysis and segmentation stage: binary mask generation

In the original Chan-Vese algorithm, a mask is utilized in performing the segmentation, from which the curves evolve to take the shape of the object under investigation. Using default masks will not work for all images and can cause the segmentation to fail, even after numerous iterations. Yu et al developed a scalable region-based level set method using adaptive bilateral filter for noisy image segmentation [59]. However, this adds more computational complexity due to bilateral filtering. Chen et al developed a matting scheme based on full feature coverage [60] and this also adds more complexity.

Figure 8 shows the customized masks utilized to segment various cracks and their orientations. These masks were initially manually derived. However, we wish to develop an automated scheme to generate suitable masks for each input image. The steps for developing an appropriate mask for each input image. Thus, we desired a much more adaptive and practical method of developing masks suited to the particular input image. The steps for developing a mask adaptively are;

- Step 1: Perform binary image labelling using eight connected components.
- Step 2: Compute the area of all components and obtain the minimum area to describe the size of the structuring element.
- Step 3: Perform image closing using the defined structuring element (a disk).
- Step 4: Fill the holes in the processed image.
- Step 5: Perform image dilation using the same disk structuring element.

4.1.4. Analysis and segmentation stage: modified active contour segmentation (MACS)

The basic formulation for the Chan-Vese ACS algorithm [28, 29] is given as;

\[
\arg \min_{\varphi \in [0,1]} \left\{ E(\varphi, c_1, c_2) = \alpha \int_{\Omega} (I(x,y) - c_1)^2 \varphi \text{d}x\text{d}y \\
+ \beta \int_{\Omega} (I(x,y) - c_2)^2 (1 - \varphi) \text{d}x\text{d}y + \gamma \int_{\Omega} \nabla \varphi \cdot \text{d}x\text{d}y \right\}
\]  

In (1), \(c_1\) and \(c_2\) are the mean values of inside and outside of the contour, \(\varphi\), while \(\alpha, \beta\) and \(\gamma\) are parameters [28]. The image to be
segmented is $I(x, y)$ with $\Omega$ as the image domain and the values of $c_1$ and $c_2$ are computed [28] as;

$$c_1 = \frac{\int_{\Omega} I(x, y) \, dx \, dy}{\int_{\Omega} \phi \, dx \, dy} \quad \text{and} \quad c_2 = \frac{\int_{\Omega} I(x, y)(1 - \phi) \, dx \, dy}{\int_{\Omega} (1 - \phi) \, dx \, dy} \quad \text{while} \quad \phi = \{0, 1\} \quad \text{to} \quad \{0, 1\}$$

The term given as

$$E(\phi, c_1, c_2) = \alpha \int_{\Omega} I(x, y) - c_1 \phi \, dx \, dy + \beta \int_{\Omega} (I(x, y) - c_2)(1 - \phi) \, dx \, dy + \gamma \int_{\Omega} \nabla \phi \, dx \, dy$$

is the energy function. The improved energy function is given as;

$$E(\phi, c_1, c_2) = \alpha \int_{\Omega} I(x, y) - c_1 \phi \, dx \, dy + \beta \int_{\Omega} (I(x, y) - c_2)(1 - \phi) \, dx \, dy + \gamma \int_{\Omega} \nabla \phi \, dx \, dy$$

Reduction of resolution also leads to faster convergence of the active contour algorithm. However, minute details may be lost. Using the original image resolution is much more accurate but results in a larger number of iterations. Furthermore, results may not always converge, leading to disjointed and incomplete segmentation. We note the effect of the enhancement stages on the segmentation result with the default energy function as shown in Figure 9 for 500 iterations. The mask shown in Figure 9(e) is manually created to overlap with the specified crack feature. However, this is a trivial case and we require various orientations to process the various images with much more complex crack patterns.

Figure 10 shows the results of both non-adaptive and adaptive masks coupled with the new energy function. We can see that the adaptive mask enables faster convergence of the ACS algorithm along with the new energy function. It is tempting to utilize the generated mask as the desired result. However, this is not always ideal and will require further processing to obtain the best result for other images with more complex crack patterns.

### 4.1.5. Post processing stage: binary fusion of edge and segmented images

In most cases, neither the mask nor the segmented result is optimal. Also, the Candy edge map may be desirable only for simple crack features on relatively smooth surfaces and fails for images with rough textures. Any further adjustment of threshold values results in loss of edge features or increase in spurious edges. Thus, we propose the fusion of the edge map and the segmented image.

The Candy edge map usually led to minimal edge features based on the adaptive threshold computation. This is especially critical for images with rough concrete surfaces to avoid detection of false features. Furthermore, this enables the ACS to focus on the region of interest, even if incomplete. The final segmented value can then be obtained much more quickly. The combination of the two leads to preserving the outline of the crack along with recovering previously missing sections of the crack segment.

We utilize the maximum (or union operation in set theory) function to merge the two images. In Figure 11, there are some breakages in the edge image, which are linked in the segmented image and vice versa. However, the noise from the segmented image is also carried over to the final fused result. This can easily be removed with few additional morphological operations. The results are adequate and show improvements not possible with either the default ACS or the Candy edge detection algorithm. Further adjustment of the threshold may improve the results in the upper right and link the breakages. However, more noise would be included.

### 4.1.6. Post processing stage: morphological cleaning operations

In most cases, spurious features remain after segmentation and need to be eliminated to yield a clean image. Thus, we fill and open regions less than a specified area. We also remove internal pixels, leaving the crack feature outline, which enables comparison with the ground truth segmented results as shown in Figure 12.

### 4.2. Saturation channel and area classification-based crack detection

The saturation channel interestingly, yields an image mostly similar to a binary map. It depicts a pseudo-thresholded image and preserves desired crack features. It removes redundant data and quantifies the contrast degree between background and foreground. This makes it easy to obtain a partially segmented image without utilizing a threshold value. Furthermore, it facilitates edge detection and eliminates the need for smoothing algorithms, which also affect crack features. We designate the second proposed algorithm as PA2 and the flowchart of the algorithm is shown in Figure 13. The equations describing the key steps are shown in Eqs. (4), (5), (6), (7), (8), (9), (10), (11), (12), and (13). In order to obtain cleaner features, we perform contrast enhancement on the original RGB image to obtain the result, $U_{RGRB}(x, y)$ as shown in eqn.(4);

$$U_{RGRB}(x, y) = \text{contrast enhancement}\{U_{RGRB}(x, y)\}$$

The enhanced RGB colour image is subsequently converted to the HSI colour space;

$$U_{HSI}(x, y) = \text{RGB2HSI}\{U_{RGRB}(x, y)\}$$

We split the enhanced HSI image, $U_{HSI}(x, y)$, into the hue (H), saturation (S) and intensity (I);

$$\{H, S, I\} = \text{split}\{U_{HSI}(x, y)\}$$

We compute the Otsu threshold for saturation ($T_s$) and intensity ($T_i$) channels as shown in Eq. (7);

$$T_s = \text{graythresh}(S); \quad T_i = \text{graythresh}(I)$$

![Figure 9](image-url) (a) Original image; (b) ACS using default energy function (c) ACS using default energy function + $\text{imadjust} + \text{FHF}$ and (d) ACS using proposed energy function + $\text{imadjust} + \text{FHF}$ (500 iterations) using (e) fixed binary mask; (f) binary result of (b); (g) binary result of (c); (h) binary result of (d).
This is followed by the computation of the joint saturation-intensity threshold ($T_{SI}$) as:

$$T_{SI} = T_S \cdot T_I$$

(8)

This obtained threshold is utilized in the subsequent Canny edge detection algorithm;

$$S_{canny} = Canny\_edge\_detection(S, T_{SI})$$

(9)

The area array is obtained after object labelling operations;

$$Area\_array = Object\_labelling(S_{canny})$$

(10)

This is followed by the computation of the histogram of areas (HOA);

$$P_{Area} = histogram(Area\_array)$$

(11)

Fuzzy c-means clustering [41] is an unsupervised machine learning technique that has been effectively utilized in various segmentation tasks. Cracks exhibit irregular geometry, which make it difficult to classify shapes and areas using standard methods. Thus, it is hoped that fuzzy-based methods would be more robust in removing spurious regions in processed images with ambiguous shapes and areas. The HOA data is paired with the area data to obtain fuzzy cmeans cluster centers, $C_1$ and $C_2$;

$$\{C_2, C_1\} = fuzzy\_cmeans(Area\_array, P_{Area})$$

(12)

These clusters are utilized in computing the area ratio value defined as shown;

$$Area\_ratio = \begin{cases} C_2 & C_2 > C_1 \\ C_1 & C_2 \leq C_1 \end{cases}$$

(13)

We then perform morphological operations utilizing the computed area ratio to obtain a clean image. We define PA2 as the HSI-based crack detection (HSI_CD) method. This algorithm is much faster than the previous proposed algorithm (PA1), though not as versatile.

The proposed scheme works best for images with relatively simple crack patterns. It fails to completely detect all edge features for large images of cracks in concrete surfaces. This is due to the computation of the threshold value used in the Canny edge detector. Nevertheless, the proposed scheme has shown efficacy in most cases, while being of relatively minimal computational complexity. Additional processing is only required for the most difficult of images. This reduces processing time and improves efficiency of the crack detection algorithm. Also, the HSI-based processing eliminates the need for smoothing filters and illumination correction algorithms, which further contribute to the processing time. Additionally, smoothing algorithms are usually not required for large images since the background surface texture in such images are relatively smooth compared to the foreground crack features. Thus, the
problems are observed mostly for the smaller sized images with fine or rough surface texture.

The scalar magnification of the saturation channel yields undesired results. However, statistical contrast enhancement of the saturation channel improves clarity. Amplification of the saturation channel features is better observed in the contrast enhancement of the original colour image. This is observed for the comparison with the ground truth segmented result in Figure 14.

4.3. Modifications to proposed algorithm for difficult images with complex crack patterns

Despite the improvements in the proposed algorithm, certain images such as those with similar intensity between foreground and background led to poor results. Thus, additional techniques were required to ensure optimal crack feature extraction from such images. These will feature as pre-processing operations and prior to the edge detection stages.

4.3.1. Global histogram equalization and localized wavelet based enhancement

Based on the analysis of the most problematic cases, it was discovered that local-global or multi-scale contrast enhancement improved the detection results for such images. Thus, the global histogram equalization normalized the distribution and led to fewer intensities, which is desirable. Furthermore, the wavelet-based enhancement algorithm amplified the detail coefficients. It also suppressed the approximation coefficients, increasing contrast and sharpness. This further increased the level of detail for images with intricate crack patterns, leading to improved crack detection. We designate this version as the modified HSI-based crack detection (HSI_CD2).

4.3.2. Canny edge detection guided by decision-based adaptive threshold

The edge detection process is the key stage after conversion to the HSI space. Thus, the choice of threshold value is critical to performance. The previous scheme described in Eqs. (7) and (8) was devised to prevent over- or under-selection of edge features. However, with the additional multi-scale enhancement, this previous scheme leads to over-selection of edges resulting in an extremely noisy image. It can also result in under-selection of features when the Otsu threshold of the intensity channel is zero. Thus, we realized a scheme for such cases. This is given as;

$$T(T_s, T_i) = \begin{cases} T_s, & T_s > 0 \\ T_i, & T_s < T_i or T_s = 0 \\ T_i, & T_i < T_s or T_i = 0 \\ 0.75, & \text{else} \end{cases}$$

(14)

The appropriate threshold is once more utilized in the Canny edge detection algorithm;

$$S_{\text{Canny}} = \text{Canny\_edge\_detection}(S, T)$$

(15)

This however, also increases the noise as well during the edge detection process. We designate this variant as the modified HSI-based crack detection with small (200 pixels) or large (1000 pixels) area analysis and evaluation (HSI_CD2_200/HSI_CD2_1000).

![Diagram](image-url)
4.3.3. Median filter pre-processing

Initial sharpening of the image prior to the conversion leads to increased noise, which requires a median filter to clean the image. However, this also smoothens out sections of the crack feature. In order to guide the edge detection process, some pre-processing is necessary for images with both foreground features and rough background textures exhibiting similar intensities.

Incorporating PDE-based smoothing filters increased complexity but with minimal improvement. Thus, experimentation led to the use of a $3 \times 3$ median filter to pre-process such images. This led to the suppression of unwanted features of the saturation component, making it easier to extract crack features from such images. Figure 15 shows the visual outcomes of such modifications for the initial algorithm and improvements. However, the modifications lead to introduction of spurious artifacts, which require further morphological processing. Experiments resulted in fixed parameter values to resolve such issues.

The proposed schemes work well for other crack image datasets used in the analysis of crack detection algorithms. The images which result in failure cases are a small subset, which have been mostly resolved using the proposed modifications. We designate this variant the modified HSI-based crack detection with small/large area evaluation and median filter pre-processing (HSI_CD2_Medfilt).

5. Experiments and results

We perform experiments and comparisons with results obtained with various crack segmentation algorithms from the literature. We utilize the crack forest dataset (CFD) [20] and Concrete Crack Segmentation Dataset (CCSD) [61]. The CFD consists of a hundred and eighteen (118) RGB colour images with dimensions of $480 \times 320$ pixels, acquired from road pavements in Beijing, China via an I-phone 5. The manually segmented crack contours were used to produce the ground truth images as shown in Figures 6(c) and 9(f). The CCSD dataset consists of four hundred and fifty-eight (458) high resolution original RGB colour and segmented ground truth images with dimensions of $3024 \times 4032$ pixels acquired from the Middle East Technical University [61].

Though there are other additional datasets, the CFD dataset is the most widely used and one of the most difficult to automatically segment based on experiments. Thus, for the basis of comparison, we focus on CFD dataset. The runtime for each image in the CFD is about 1.70 s. For the CCSD dataset, runtime is about 67 s. The execution times are obtained on an Intel® Core i7-6500U x64-based 2.59 gigahertz (GHz) processor with 12 gigabyte (GB) random access memory (RAM) with a 64-bit operating system and NVIDIA® GeForce™ 940M graphics processing unit (GPU) (compute capability of 5.0). However, such large images from the CCSD dataset do not require extensive processing. The essential steps needed for such high resolution images reduces the runtime to just under 3 s.

The overall computational complexity of the PA1 and PA2 algorithms is roughly $16N^2\log N$ due to the convolution-based filters for a $N \times N$ saturation image. The complexity of the wavelet routines and median filters would be higher. However, PA1 is heavily iterative while PA2 is not.

5.1. Subjective analysis

We compare PA1 and PA2 with various algorithms from the literature. These include Canny [58], local thresholding by Niblack [52], Free Form Anisotropy (FFA), CrackTree [20], CrackForest [20] and Structured Prediction with Convolutional Neural Network (SPCNN) [3]. Additional algorithms are Multi-Scale Image Fusion Crack Detection (MFCD) [22], Percolation methods [21], improved Crack Image Enhancement And Detection-2 (CIEAD-2) algorithm [9] and Densely connected deep neural network by Mei et al [6]. The sample results are shown in Figures 16 and 17, which contain visual results from Qu et al [21] compared with CIEAD, PA1 and PA2.

Though PA1 and PA2 perform well in Figure 16, some images still require additional post-processing to remove the unwanted artifacts. However, PA2 is the least complex algorithm. In Figure 17, PA2 yields good results and matches the more complex improved CIEAD2 algorithm from previous work [9].

Figure 18 includes image results from Mei et al [6], compared with saturation channel extraction, PA1 and PA2. In Figure 18, we compare the saturation channel result, the proposed approach (PA2) and its modification with a recent convolutional neural network-based method by Mei et al [6]. The PA2 performs adequately and on par with the method by Mei et al.

Figure 15. Comparison of effects on the saturation channel (top row) and edge map (bottom row) for the (a) initial algorithm (PA) and (b) with modifications for outlier images.
5.2. Objective analysis

We utilize several metrics usually applied in quantitative evaluation of various algorithms. These include Precision, Recall and F-Measure [62]. These are expressed as:

Precision,

\[ Pr = \frac{TP}{TP + FP} \]  

Recall,

\[ Re = \frac{TP}{TP + FN} \]  

F-Measure,

\[ F1 = \frac{2 \times Pr \times Re}{Pr + Re} \]

The terms TP, FP and FN in Eqs. (16), (17), and (18), are the true positive (measure of features present in both ground truth and processed images), false positive (measure of features present only in the processed images) and false negative (measure of features present only in the ground truth images) values respectively. We utilized additional measures such as Accuracy, PSNR, Pseudo F-measure [62], etc. However, all work found in the literature present only the Precision, Recall and F-Measure metrics. Thus, there would be no way to fully compare additional metrics with other works since their software implementations are mostly unavailable. This was the reason for the utilization of the published results presented by the various authors.

We compare numerical results in Table 1 using the CFD dataset. The table is from Fan et al [3] with additional results from Li et al [22] amended with the results of the improved CIEAD-2 [9] and Mei et al [6].

Figure 16. Comparisons with various algorithms for images with rough surfaces and uneven illumination: (a) Original images (b) ground truth and results from (c) FFA (d) CrackTree (e) CrackForest (f) Qu et al (g) improved CIEAD-2 (h) ACS + Canny_fusion (PA1) (i) Saturation-based thresholding (PA2).

Figure 17. Comparisons with various algorithms for large images with smooth surfaces: (a) Original images (b) Hoang et al (c) CIEAD-2 (d) improved CIEAD-2 (e) PA2.
We present only objective results of PA2 since it runs much faster than PA1 and much more accurately. Objective evaluations indicate that PA2 and its variants yield the best results of all the algorithms. The HSI_CD2_200/HSI_CD2_1000/HSI_CD2_Medfilt are best suited to difficult and complex crack patterns, though they still perform well for simple cracks. However, they slightly involve more runtime than the HSI_CD variant. Thus, the effect of HSI colour space can be seen in the improved results since it is unaffected by uneven illumination, rough surfaces or texture. This makes it easier to handle images with rough background textures.

6. Conclusion

We have proposed a modified active contour algorithm combined with Canny edge detection to improve results of adaptive crack segmentation. Alternatively, a relatively less complex but more targeted algorithm employing HSI colour space improves speed and crack detection performance. With additional modifications, the scheme also solves the issue of complex crack patterns, noise and rough textures interfering with accurate segmentation. Subjective and objective comparisons with existing and state-of-the-art methods show overall and consistently good performance. This is achieved without extensive deep learning or machine learning methods. Further study would involve full automation of the process to account for both generally good and outlier images, which still pose problems for automated crack feature extraction. Ultimately, such advantages and insights gained from the study of the HSI colour space will be explored in other image thresholding applications.

Table 1. Results from [3, 6, 9, 22] compared with PA2 using objective measures.

| Measures | Precision | Recall | F1 Measure |
|----------|-----------|--------|------------|
| Canny [58] | 0.4377/0.1223 | 0.7307/0.2215 | 0.4570/0.1576 |
| Local thresholding [52] | 0.7727 | 0.8274 | 0.7418 |
| CrackForest [20] | 0.7466/0.8228 | 0.9514/0.8944 | 0.8318/0.8571 |
| SPCNN [8] | 0.9119 | 0.9481 | 0.9244 |
| FPA [18] | 0.7856 | 0.6843 | 0.7315 |
| MFCD [22] | 0.8990 | 0.8947 | 0.8804 |
| CrackTree [20] | 0.7322 | 0.7645 | 0.7080 |
| ResNet152-FCN [63] | 0.8783 | 0.8819 | 0.8801 |
| VGG19-FCN [64] | 0.9280 | 0.8549 | 0.8853 |
| CrackNet-V [65] | 0.9258 | 0.8603 | 0.8918 |
| UNet [66] | 0.8680 | 0.7797 | 0.8183 |
| FPHBN [67] | 0.9588 | 0.8779 | 0.9153 |
| Mei et al (with regular loss function) [6] | 0.9202 | 0.9113 | 0.9158 |
| Mei et al (with new loss function) [6] | 0.9100 | 0.9322 | 0.9199 |
| Improved CIEAD-2 [9] | 0.9909 | 0.9675 | 0.9789 |
| PA2 (HSI_CD) | 0.9943 | 0.9381 | 0.9647 |
| PA2 (HSI_CD2,200) | 0.9909 | 0.9799 | 0.9853 |
| PA2 (HSI_CD2,1000) | 0.9908 | 0.9888 | 0.9898 |
| PA2 (HSI_CD2_Medfilt) | 0.9908 | 0.9823 | 0.9865 |

Figure 18. Comparisons with various algorithms for images with simple to complex crack features on rough surfaces with markings: (a) Original images (b) Ground truth (c) Mei et al [6] (d) Mei et al (with post-processing) (e) Saturation channel (f) PA2 (g) modified PA2.

Declarations

Author contribution statement

U. Nnolim: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed materials, analysis tools or data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
Data availability statement

Data will be made available on request.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] Xingu Zhong, Xiong Peng, Shengkun Yan, Yinhan Shen, Yinyin Zhai, Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles, Autom. Construt. 89 (2019) 49–57.

[2] Zheng Teng, Jie Gao, Haitao Zhang, Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks, Constr. Build. Mater. 146 (2017) 779–787.

[3] Zhan Fan, Yuming Wu, Jiewei Lu, Wenhui Li, Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network, Feb 1 2018 arXiv preprint arXiv:1802.02308.

[4] Nhat Duc Hoang, Quoc Lam Nguyen, Van Duc Tran, Automatic recognition of asphalt pavement cracks using metamorphetic optimized edge detection algorithms and convolution neural network, Autom. Construt. 94 (2018) 203–213.

[5] Cao Vu Dung, Le Duc Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Autom. Construt. 99 (2019) 52–58.

[6] Qipei Mei, Mustafa Gul, Md Riasat Azim, Densely connected deep neural network considering connectivity of pixels for automatic crack detection, Autom. Construt. 110 (103018) (2020) 1–12.

[7] Yupeng Ren, et al., Image-based concrete crack detection in tunnels using deep fully convolutional networks, Constr. Build. Mater. 234 (117367) (2020) 1–12.

[8] U.A. Nnolim, Partial differential equation-based enhancement and crack detection, Math. Prbhl Eng. (2019) 1–16. March 26 2019.

[9] U.A. Nnolim, Fully adaptive segmentation of cracks on concrete surfaces, Comput. Electr. Eng. 83 (May 2020) 1–15 Online.

[10] Ahmed Mahgoub Ahmed Talab, Zhangcan Huang, Fan Xi, Liu HaMing, Detection crack in images using Otu method and multiple filtering in image processing techniques, Optik-Int. J. Light Electron. Optic. 127 (3) (February 2016) 1030–1033.

[11] Sattar Doraftabian, Magdoure Catru, Xuefeng Zhao, Automatic Surface Crack Detection in Concrete Structures Using Otu Thresholding and Morphological Operations, Utah State University, Apr 2016. Report.

[12] Nhat Duc Hoang, Detection of surface crack in building structures using image processing technique with an improved Otu method for image thresholding, Adv. Civ. Eng. 2018 (2018) 1–10.

[13] Roberto Medina, Joseph Llamas, Jaime Gomez-Garcia-Bermejo, Eduardo Zalama, Miguel Jose Segarra, Crack detection in concrete tunnels using a gabor filter invariant to rotation, Sensors 17 (1670) (20 July 2017) 1–16.

[14] Arun Mohan, Sumathi Poobal, Crack detection using image processing: a critical review and analysis, Alexandria Eng. J. (14 January 2017) 1–12.

[15] Gang Li, Xiaoqiong Zhao, Kai Du, Feng Ru, Yubo Zhang, Recognition and evaluation of crack videos, EURASIP J. Image and Video Process. 2020 (1) (2020) 1–17.

[16] H. Yu, F. He, Y. Pan, A novel segmentation model for medical images with internal inhomogeneity based on adaptive perturbation, Multimed. Tool. Appl. 78 (9) (2019) 11779–11798.

[17] R.C. Gonzalez, R.E. Woods, Digital Image Processing, second ed., Prentice Hall, 2002.

[18] U.A. Nnolim, Analysis of Proposed PDE-Based Underwater Image Enhancement Algorithm, Computational Intelligence and Neuroscience, 2016 [Online]. Available from: arXiv.org/pdf/.

[19] U.A. Nnolim, Entropy-guided Retinex Anisotropic Diffusion Algorithm Based on Partial Differential Equations (PDE) for Illumination Correction, University of Nigeria, Enugu, 2017 [Online]. Available from: http://arxiv.org/pdf/.

[20] U.A. Nnolim, A PDE-Based Log-Agnostic Illumination Correction Algorithm, University of Nigeria, Nsukka, 2017. Technical Report 2017 [Online]. Available from: http://arxiv.org/pdf/1705.04272.

[21] Nobuyuki Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. SAC-9 (1) (1979) 62–66.

[22] Mathworks, Fuzzy C-Means Clustering: Fuzzy Logic Toolbox for Use with MATLAB, Document 2013 [Online]. Available from: www.mathworks.com.

[23] F. Liu, Y. Luo, X. Song, D. Hu, Active surface model-based adaptive thresholding algorithm by repulsive external force, J. Electron. Imag. 12 (3) 299–306.

[24] S. Sarkar, S. Paul, R. Burman, S. Das, S.S. Chaudhuri, A fuzzy entropy based multiscale image segmentation using fuzzy c-means algorithm, Pattern Recogn. 35 (2) (2002) 342–349.

[25] G.N. Zack, W.E. Rogers, S.A. Lut, Automatic measurement of sifier chromatic exchange frequency, J. Histochem. Cytochem. 25 (7) (1977) 741–753.

[26] Sujoy Paul, Shouknat Datta, Swagatam Das, Rough-fuzzy collaborative multi-level image thresholding: a differential evolution approach, in: R. Matosik (Ed.), Advances in Intelligent Systems and Computing (ASCI), 378, Springer, Cham International Publishing, Switzerland, 7 June 2015, pp. 329–341.

[27] M.-L. Peng, Y.-P. Fan, Adaptive binarization method for document image analysis, IEEE Int. Conf. Multimed. Expon. 1 (2004) 339–342.

[28] Derek Bradley, Gerhard Roth, Adaptive thresholding using the integral image, J. Graph. Tool. 12 (2) (January 2007) 13–21.

[29] S. Sarkar, S. Burman, S. Saha, S.S. Chaudhuri, A fuzzy entropy based multi-level image thresholding using differential evolution, in: International Conference on Image Processing, Evolutionary, and Memetic Computing (SEMCOCO 2014), 8947, July 16 2015, pp. 386–395.

[30] Yang Xiao, Zhiguo Cao, Tianxu Zhang, Entropic thresholding based on gray-level inhomogeneity based on adaptive perturbation, J. Vis. Commun. Image Represent. 40 (18 June 2016) 168–177.

[31] Yangyang Song, Guohua Peng, A fast two-pass active contour model for intensity inhomogeneous image segmentation, PloS One 14 (4) (2019).

[32] Jinheng Xiao, et al., Adaptive shock filter for image super-resolution and enhancement, J. Vis. Commun. Image Represent. 40 (18 June 2016) 168–177.

[33] Yangyang Song, Guohua Peng, A fast two-pass active contour model for intensity inhomogeneous image segmentation, PloS One 14 (4) (2019).

[34] H. Yu, F. He, Y. Pan, A novel segmentation model for medical images with internal inhomogeneity based on adaptive perturbation, Multimed. Tool. Appl. 78 (9) (2019) 11779–11798.

[35] H. Yu, F. He, Y. Pan, A novel segmentation model for medical images with internal inhomogeneity based on adaptive perturbation, Multimed. Tool. Appl. 78 (9) (2019) 11779–11798.
[58] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 6 (November 1986) 679–698.
[59] H. Yu, F. He, Y. Pan, A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation, Multimed. Tool. Appl. 79 (2020) 5743–5765.
[60] X. Chen, F. He, H. Yu, A matting method based on full feature coverage, Multimed. Tool. Appl. 78 (9) (2019) 11173–11201.
[61] Çaglar Firat Özgenel, April 3) Concrete Crack Segmentation Dataset, Mendeley Data, 2019 v1. [Online].
[62] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437.
[63] S. Bang, S. Park, H. Kim, H. Kim, Encoder-decoder network for pixel-level road crack detection in black-box images, J. Comput. Aided Civ. Infrastruct. Eng. (2019).
[64] X. Yang, et al., Automatic pixel-level crack detection and measurement using fully convolutional network, J. Comput. Aided Civ. Infrastruct. Eng. 33 (12) (2018) 1090–1109.
[65] Y. Fei, et al., Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet V, IEEE Trans. Intell. Transp. Syst. (2019).
[66] Z. Liu, Y. Gao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Autom. Constr. 104 (2019) 129–139.
[67] F. Yang, et al., Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Trans. Intell. Transp. Syst. (2019).