Region- and Edge-based Active Contour for Automatic Crack Segmentation Refinement

Yung-An Hsieh

Georgia Institute of Technology

Final Project: ECE 6560 Partial Differential Equations in Image Processing and Computer Vision

Instructor: Dr. Anthony Yezzi

Spring 2020

Contents

1. Motivation

Cracking appears in different kinds of structures, such as pavements, buildings, and bridges. Since cracking can accelerate the deterioration process, the occurrence and severity of cracking serve as important indicators that maintenance is required. Therefore, efficient and accurate automatic crack detection methods have always been an important research topic. In recent years, deep learning (DL) models, or more specifically the Convolutional Neural Networks (CNNs), have been largely utilized on automatic image-based crack detection. With the advanced development of CNNs, research efforts have already gained successful results on detecting cracks in pixel level, or crack segmentation (Fig. 1).

However, while the CNN-based segmentation models can detect most of the crack patterns accurately, a remaining problem is that the segmented results are not able to capture the fine details of the cracks. This is because most of the CNN-based segmentation models have an encoder-decoder structure, in which the spatial size of feature maps will be continuously decreasing at the encoder and then gradually restored to the original input size at the decoder. The upsampling process at the decoder caused the segmentation results to be coarse (Fig. 1). While the skip connections [1] and residual techniques [2] can provide improvements on the segmentation details, the fine details of the cracks are still hard to capture because of the extremely thin width (sometimes even with only one or two pixels wide). Therefore, the objective of this study is to utilize partial differential equations (PDEs) techniques as a post-processing step to refine the crack segmentation results generated by the CNN-based models.

2. Mathematical Interpretation

The results generated from CNN-based segmentation models are binary images, in which the white pixels indicate crack pixels and the black pixels indicate non-crack pixels. In other words, the white pixels on the result images form the initial contour of the cracks. To refine the segmentation results, we need to evolve the initial contours to better fit the cracks

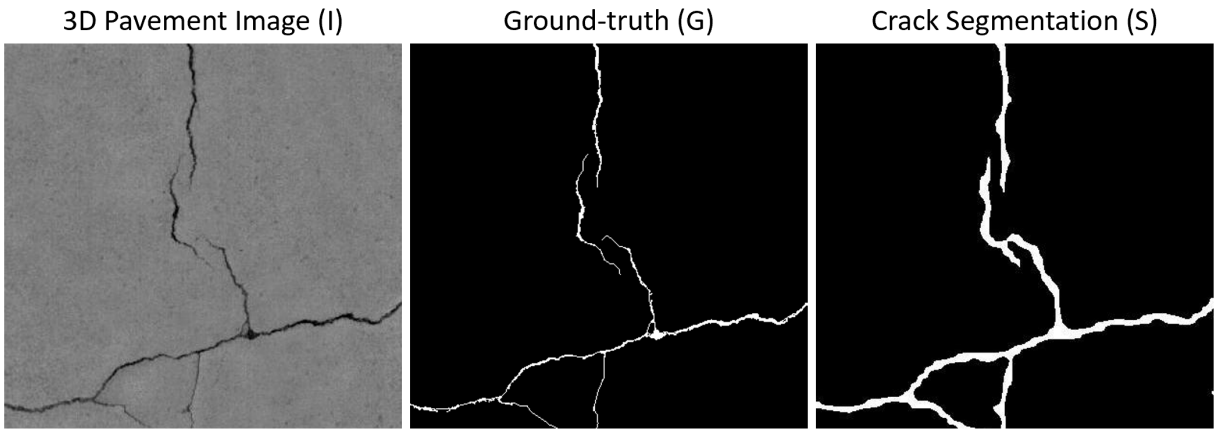

Fig. 1: An example of a 3D pavement surface image with crack (left), a manually labeled crack ground-truth (middle), and a crack segmentation generated by DL-based model (right).

on the original images. Therefore, the problem of refining segmented cracks is formulated as contours evolution and the active contour model comes in handy.

In this study, the active contour model was implemented using the variational gradient descent PDEs. Therefore, an energy functional needs to be formulated to guide the evolution of the contours. This is done by incorporating the information of the original images, which contain the cracks we want to detect, into the energy functional. With the fact that the cracks appear darker on the original images, two kinds of information are utilized in the task of refining crack segmentation, which are the regions and edges of the dark pixels. The next section presents the details of incorporating the information into the energy functional and the formulation of the region- and edge-based active contour models.

3. Mathematical Formulation

3.1. Region-based active contours

3.1.1. Derivation

Assume C is the initial contour in a 2D space, R is the interior of C, R^c is the exterior of C, and f is a function in the region R. An energy functional inside C for region-based active contour can be defined as:

$$E(C) = \iint_R f dx dy$$

The objective here is to minimize the energy when we evolve C. To do this, we need to get the gradient of E and perform gradient descent on C. Suppose C varies in time, we can obtain the gradient of E by:

$$\frac{d}{dt}E(C(t)) = \langle C_t, \nabla_C E \rangle = \int_c C_t \cdot (\nabla_C E) ds \tag{1}$$

where s is the arclength parameter and the gradient $\nabla_C E$ is independent to time t. Suppose F is a vector field such that $\nabla \cdot F = f$, by using the divergence theorem we can obtain:

$$E(C) = \iint_R f dx dy = \iint_R (\nabla \cdot F) dx dy = \int_C F \cdot N ds$$

$$\begin{aligned}\frac{dE}{dt} &= \frac{d}{dt}\int_C F \cdot N ds = \frac{d}{dt}\int_0^1 (F \cdot N)||C_p|| dp = \int_0^1 \frac{d}{dt}(F \cdot N)||C_p|| dp \\ &= \int_0^1 (\frac{d}{dt}F) \cdot N||C_p|| + F \cdot (\frac{d}{dt}N)||C_p|| + (F \cdot N)(\frac{d}{dt}||C_p||) dp\end{aligned}$$

We know that:

$$\frac{d}{dt}F = \frac{\partial F}{\partial x}\frac{dx}{dt} + \frac{\partial F}{\partial y}\frac{dy}{dt} = \left[\frac{\partial F}{\partial X}\right] C_t$$

$$\frac{d}{dt}N = J(\frac{d}{dt}\frac{C_p}{||C_p||}) = J(\frac{C_{pt}}{||C_p||} - \frac{C_p}{||C_p||^2}||C_p||_t) = \frac{1}{||C_p||}J(N(C_{pt} \cdot N)) = \frac{-T(C_{pt} \cdot N)}{||C_p||}$$

$$||C_p||_t = (\sqrt{C_p \cdot C_p})_t = \frac{C_{pt} \cdot C_p}{||C_p||} \tag{2}$$

Thus:

$$\begin{aligned}\frac{dE}{dt} &= \int_0^1 (\left[\frac{\partial F}{\partial X}\right] C_t) \cdot N||C_p|| - F \cdot (\frac{T(C_{pt} \cdot N)}{||C_p||})||C_p|| + F \cdot N(\frac{C_{pt} \cdot C_p}{||C_p||})dp \\ &= \int_C (\left[\frac{\partial F}{\partial X}\right] C_t) \cdot N||C_p|| - (F \cdot T)(C_{ts} \cdot N)||C_p|| + (F \cdot N)(C_{ts} \cdot T)||C_p||\frac{ds}{||C_p||} \\ &= \int_C ([\dot{F}]C_t) \cdot N - C_{ts} \cdot (F \cdot T)N + C_{ts} \cdot (F \cdot N)Tds\end{aligned}$$

Using integrate by parts on the second and third terms:

$$\begin{aligned}\frac{dE}{dt} &= \int_C C_t \cdot (\dot{F}^T N) + C_t \cdot ((F \cdot T)N)_s - C_t \cdot ((F \cdot N)T)_s ds \\ &= \int_C C_t \cdot [\dot{F}^T N + ((F \cdot T)N)_s - ((F \cdot N)T)_s]ds\end{aligned}$$

Thus, we get:

$$\begin{aligned}\nabla_C E &= \dot{F}^T N + ((F \cdot T)N)_s - ((F \cdot N)T)_s \\ &= \dot{F}^T N + (F_s \cdot T)N + (F \cdot T_s)N + (F \cdot T)N_s - (F_s \cdot N)T - (F \cdot N_s)T - (F \cdot N)T_s \\ &= \dot{F}^T N + (T^T \dot{F} T)N - \kappa(F \cdot N)N + \kappa(F \cdot T)T - (N^T \dot{F} T)T - \kappa(F \cdot T)T + \kappa(F \cdot N)N \\ &= \dot{F}^T N + (T^T \dot{F} T)N - (N^T \dot{F} T)T \\ &= ((T^T \dot{F}^T N)T + (N^T \dot{F}^T N)N) + (T^T \dot{F} T)N - (N^T \dot{F} T)T \\ &= (N^T \dot{F} T)T + (N^T \dot{F} N)N + (T^T \dot{F} T)N - (N^T \dot{F} T)T \\ &= (N^T \dot{F} N)N + (T^T \dot{F} T)N \\ &= trace(\dot{F})N = (\nabla \cdot F)N = fN \end{aligned} \tag{3}$$

Note that κ in the above equations is the curvature. Now we consider the energy functional outside C:

$$E(C) = \iint_{R^c} f dxdy = \iint_{\Omega} f dxdy - \iint_R f dxdy = -\iint_R f dxdy = -fN \tag{4}$$

where Ω is the whole 2D space.

3.1.2. The Chan-Vese segmentation

To incorporate the information of the original images (the regions of black pixels) into the region-baesd active contour, the Chan-Vese algorithm [3] is utilized. The energy functional is

now defined as:

$$E(C, u, v) = \iint_R (I - u)^2 dxdy + \iint_{R^c} (I - v)^2 dxdy$$

where I is the original image, u is the mean value of I in R, and v is the mean value of I in R^c. The intuition here is to find a binary mask, with pixel value u in R and pixel value v in R^c, that best matches the original image I. Using Eq. (3) and (4), we can obtain the gradient of E:

$$\begin{aligned}\nabla_C E &= ((I - u)^2 - (I - v)^2)N \\ &= (I^2 - 2uI + u^2 - I^2 + 2vI - v^2)N \\ &= (u - v)(u + v - 2I)N\end{aligned}$$

We can then obtain the gradient descent for the contour C:

$$C_t = -\nabla_C E = (u - v)(2I - u - v)N \tag{5}$$

3.2. Edge-based active contours

3.2.1. Derivation

An energy functional for the edge-based active contour, or more specifically the geodesic active contour (GAC), can be defined as:

$$E(C) = \int_C \phi ds$$

where the function ϕ should be independent to the curve. Also, we should choose ϕ to be small at the desirable location, which is the boundary of the cracks in this study. Again, the objective here is the minimize the energy as we evolve C, so we need to obtain the gradient of the energy E. We can use the same method as in Eq. (1) to obtain the gradient:

$$\begin{aligned}\frac{dE}{dt} &= \frac{d}{dt} \int_C \phi(C(s, t)) ds \\ &= \frac{d}{dt} \int_0^1 \phi(C(p, t)) ||C_p|| dp \\ &= \int_0^1 (\nabla\phi \cdot C_t ||C_p|| + \phi ||C_p||_t) dp\end{aligned}$$

Using Eq. (2) and transfer p back to s:

$$\frac{dE}{dt} = \int_C (\nabla\phi \cdot C_t||C_p|| + \phi(C_{ts} \cdot C_s||C_p||))\frac{ds}{||C_p||} = \int_C (\nabla\phi \cdot C_t + \phi(C_{ts} \cdot C_s))ds$$

Apply integrate by parts on the second term, we get:

$$\begin{aligned}
\frac{dE}{dt} &= \int_C C_t \cdot \nabla\phi - C_t \cdot (\phi C_s)_s ds \\
&= \int_C (C_t \cdot \nabla\phi - C_t \cdot T(\nabla\phi \cdot T) + C_t \cdot \phi\kappa N)ds \\
&= \int_C (C_t \cdot (\nabla\phi \cdot N)N + C_t \cdot \phi\kappa N)ds \\
&= \int_C C_t \cdot (\phi\kappa + \nabla\phi \cdot N)Nds
\end{aligned}$$

Thus, we get:

$$\nabla_C E = \phi\kappa N + (\nabla\phi \cdot N)N \tag{6}$$

We can then obtain the gradient descent of the contour C:

$$C_t = -\nabla_C E = -\phi\kappa N - (\nabla\phi \cdot N)N \tag{7}$$

In this study, we need to have small values of ϕ at the boundary of the cracks, which usually have large gradient values. Therefore, ϕ is set as:

$$\phi = \frac{1}{1 + ||\nabla I||^2} \tag{8}$$

3.3. Regularization terms

3.3.1. Region-based regularization

The region-based regularization can be defined as:

$$E(C) = a \iint_R 1dxdy$$

which is associated with the area of the region inside the contour. The constant a controls its weighting. The effect of this term is dilation if a is negative and erosion if a is positive when

we try to minimize the energy. Using Eq. (3), we can obtain the gradient of E and the gradient descent:

$$\nabla_C E = aN$$

$$C_t = -\nabla_C E = -aN \tag{9}$$

3.3.2. Edge-based regularization

Similarly, we can define the energy functional of the edge-based regularization term as:

$$E(C) = b \int_C 1 ds$$

which is associated with the edge of the contour. The constant b is the weighting. This term controls the smoothness of the contour. With larger weighting, the contour will be smoother when the energy is minimized. Using Eq. (6), we can obtain the gradient of E and the gradient descent by setting $\phi = 1$:

$$\nabla_C E = b\kappa N$$

$$C_t = -\nabla_C E = -b\kappa N \tag{10}$$

3.4. Level sets method

The contour, or curve, C can be represented implicitly using level sets ψ:

$$C = \{(x, y) : \psi(x, y) = constant\}$$

Suppose $\psi(x, y, t)$ is the time evolving implicit representation of the contour $C(p, t)$. This indicates that:

$$\psi(C(p, t), t) = constant \tag{11}$$

In this study, the signed distance function is used as ψ and $\psi(C(p, t), t) = 0$. From Eq. (11), we can get:

$$\frac{d}{dt}(\psi(C(p, t), t)) = \nabla\psi \cdot C_t + \psi_t = 0$$

$$\psi_t = -\nabla\psi \cdot C_t$$

Assume $C_t = \alpha T + \beta N$ and note that $N = \frac{\nabla\psi}{||\nabla\psi||}$, we can get:

$$\psi_t = -\nabla\psi \cdot (\alpha T + \beta N) = -\nabla\psi \cdot (\alpha J\frac{\nabla\psi}{||\nabla\psi||} + \beta\frac{\nabla\psi}{||\nabla\psi||}) = -\beta||\nabla\psi||$$

With this equation, we can now write the level sets implementation of the gradient descent equations we obtain previously (Eq. (5), (7), (9), (10)) as the following equations respectively:

$$\psi_{t-(5)} = ||\nabla\psi||(u-v)(u+v-2I) \tag{12}$$

$$\psi_{t-(7)} = ||\nabla\psi||(\phi\kappa + (\nabla\phi \cdot N)) = ||\nabla\psi||(\phi\kappa + (\nabla\phi \cdot \frac{\nabla\psi}{||\psi||})) \tag{13}$$

$$\psi_{t-(9)} = ||\nabla\psi||a \tag{14}$$

$$\psi_{t-(10)} = ||\nabla\psi||b\kappa \tag{15}$$

Finally, by combining the above equations and assign a weighting λ between region- and edge-based level sets, we can obtain:

$$\psi_t = ||\nabla\psi||\Big[\lambda(u-v)(u+v-2I) + (1-\lambda)(\phi\kappa + (\nabla\phi \cdot \frac{\nabla\psi}{||\psi||})) + a + b\kappa\Big] \tag{16}$$

We can then update the level sets in each time step T using the forward time scheme:

$$\psi^{(T+1)} = \psi^{(T)} + \Delta t(\psi_t^{(T)}) \tag{17}$$

4. Implementation

4.1. Discretization and CFL conditions

For the purpose of computational simulation, we have to discretize the PDE (Eq. (16)) and make sure that the simulation time step is small enough for the evolution of PDE to be stable. The stability criterion is called the Courant–Friedrichs–Lewy (CFL) condition. The discretization and CFL condition of each term in Eq. (16), which are Eq. (12) to (15), are shown as follows.

4.1.1. Region-based active contour

Assume the pre-computed term before updating the level set function, $(u-v)(u+v-2I)$, as α. Eq. (12) can then be written as:

$$||\nabla\psi||(u-v)(u+v-2I) = \alpha||\nabla\psi|| = \alpha\sqrt{\psi_x^2+\psi_y^2}$$

This morphological filter can be implemented using the entropy upwind scheme:

$$\psi_t = \alpha\sqrt{max^2(D_x^+\psi,0)+min^2(D_x^-\psi,0)+max^2(D_y^+\psi,0)+min^2(D_y^-\psi,0)} \text{ if } \alpha > 0 \quad (18)$$

$$\psi_t = \alpha\sqrt{min^2(D_x^+\psi,0)+max^2(D_x^-\psi,0)+min^2(D_y^+\psi,0)+max^2(D_y^-\psi,0)}, \text{ if } \alpha < 0 \quad (19)$$

where $D_x^+\psi$ and $D_x^-\psi$ are the forward and backward differences respectively. The CFL condition of this implementation is:

$$|\alpha|\Delta t < \frac{1}{\sqrt{2}}\Delta x \quad (20)$$

4.1.2. Edge-based active contour

Eq. (13) can be further separated into two terms. The first term can be written as:

$$\phi||\nabla\psi||\kappa = \phi||\nabla\psi||\nabla\cdot(\frac{\nabla\psi}{||\nabla\psi||}) = \phi\frac{\psi_x^2\psi_{yy}-2\psi_x\psi_y\psi_{xy}+\psi_y^2\psi_{xx}}{\psi_x^2+\psi_y^2}$$

Note that the fraction part is a geometric heat equation, which can be implemented using central differences scheme with the CFL condition of:

$$\phi\Delta t \leq \frac{1}{2}\Delta x^2 \quad (21)$$

Where ϕ are some values that can be pre-computed before updating the level set function.

Now consider the second term of Eq. (13). Assume $\nabla\phi$ is a external vector field $\overrightarrow{v} = (v_1, v_2)$, Eq. (13) can then be written as:

$$||\nabla\psi||(\nabla\phi\cdot\frac{\nabla\psi}{||\psi||}) = \nabla\phi\cdot\nabla\psi = \overrightarrow{v}\cdot\nabla\psi = v_1\psi_x + v_2\psi_y$$

which can be implemented using the upwind difference scheme with the CFL condition of:

$$|v|\Delta t < \frac{1}{\sqrt{2}}\Delta x \quad (22)$$

4.1.3. Region-based regularization

Eq. (14) is similar to Eq. (12) with the pre-computed term, $(u - v)(u + v - 2I)$, replaced by a constant a. Therefore, follow by section 4.1.1, we can use entropy upwind scheme for implementation and obtain the CFL condition according to Eq. (20):

$$|a|\Delta t < \frac{1}{\sqrt{2}}\Delta x \tag{23}$$

4.1.4. Edge-based regularization

Eq. (15) is the same as the first term of Eq. (13) with the pre-computed ϕ replaced by a constant b. Therefore, we can use central differences scheme for implementation and get the CFL condition from Eq. (21):

$$|b|\Delta t \leq \frac{1}{2}\Delta x^2 \tag{24}$$

4.1.5. Determine the time step

During the contour evolution, the Δx is one pixel. Also, the magnitudes of the constants a and b are both set to be less than one in the experiments, and from Eq. (8) we know that ϕ will also have values less than one. Therefore, by combining these constraints and the CFL conditions calculated above, the time step Δt is set to 0.3

4.2. Level sets re-initialization

During the evolution of the level-set function ψ, the function can develop gradients that are too steep and/or too flat. This impacts the numerical stability in subsequent iterations and causes unpredictable outcomes. Therefore, we need to perform a distance reinitialization step at the end of every few iterations, such that the zero level-curve location remains unchanged. In this study, the method proposed by Sussman et al. [4] was used to perform the level sets

re-initialization, which evolve the PDE to a steady-state with the following equation:

$$\psi_t + sign(\psi_0)(||\nabla\psi|| - 1) = 0$$

where ψ_0 is the level sets before the re-initialization.

4.3. Stopping criterion

In this study, the evolution of level set function was halted by 1) a threshold set on the number of iterations ($iters$), and 2) checking if the change of new segmentation is smaller than a certain pixels ($thresh$) for five continuous iterations. The $iters$ parameter was set to 500 and the $thresh$ parameter is set to 20 during the experiments.

4.4. Dataset

In this study, 19 3D pavement surface images (I) with cracking were used to evaluate the crack segmentation refinement method. The pavement images are gray-scale images, where darker pixels indicate deeper pavement surface. The height and width of the images range from 300 to 415 pixels. Also, these images are selected such that the dataset contains different types (e.g. longitudinal, transverse, or both) and widths of cracks.

Each of the pavement images is accompanied by two binary images. The first binary image is the crack segmentation result (S) generate by a DL-based segmentation model, the fully convolutional network (FCN) [1] with ResNet50 [5] as the backbone. This segmentation result is the initial contour that will go through the contour evaluation process. The second binary image is the ground-truth of the crack segmentation (G), which is manually labeled. The white pixels in S and G indicate crack pixels and the black pixels indicate non-crack pixels. An example of these three images, I, G, and S, is shown in Fig. 1.

4.5. Metrics

To evaluate the results of refined crack segmentation, the Enhanced Hausdorff Distance (EHD) metric proposed by Tsai and Chatterjee [6] was utilized.

The Hausdorff distance between two sets of points A and B can be calculated by:

$$HD(A,B) = max(h_{HD}(A,B), h_{HD}(B,A))$$

Where the penalty is determined by the pixel with the maximum Euclidean distance from the other set of pixels:

$$h_{HD}(A,B) = max_{a\in A} min_{b\in B} ||a-b||$$

Based on the Hausdorff distance, Kaul et al. [7] proposed the buffered Hausdorff distance metric for crack segmentation algorithms. This is done by taking the mean instead of the maximum of the shortest distances from each pixel to the other set of pixels when calculating the penalty. Also, an upper limit u is used to ensure no further penalty is added after one can clearly determine a predicted crack is not an actual crack. The penalty of buffered Hausdorff distance metric is calculated by:

$$h_{BHD}(A,B) = \frac{1}{|A|}\sum_{a\in A} sat_u(min_{b\in B}||a-b||)$$

To consider the subjectivity issue of ground-truth labeling, Tsai and Chatterjee [6] incorporated a penalty-free buffer around the ground truth and predicted crack when calculating the penalty:

$$h_{EHD}(A,B) = \frac{1}{|A|}\sum_{a\in A} f(||a-b||), \text{ where } f(x) = \begin{cases} u-l & \text{for} \quad u < x \\ x-l & \text{for} \quad l \le x \le u \\ 0 & \text{for} \quad x \le l \end{cases} \tag{25}$$

In Eq. (20), l is the lower limit of the Euclidean distance after which a penalty is applied. The lower limit provides a penalty-free region around the crack to address the subjectivity issue of ground-truth labeling. On the other hand, u is the upper limit of the Euclidean distance after which the penalty will not further increase. In this study, the lower limit is set to 0 pixel since we are interested how the width of segmented cracks changes during the contour evaluation On

the other hand, the upper limit is set to 10 pixels. The set of data points, A, is the segmented crack pixels and the set of data points, B, is the ground-truth crack pixels. With Eq. (20), the EHD score is given by the following equation, which higher score indicates better performance:

$$score(A, B) = 100 - \frac{max(h_{EHD}(A, B), h_{EHD}(B, A))}{u - l} \times 100 \tag{26}$$

Also, the false positive (FP) and false negative (FN) penalties are given by the following two equations such that the higher the FP/FN penalty, the more FP/FN pixels are presented in the segmentation results.

$$\text{FP penalty} = \frac{h_{EHD}(A, B)}{u - l} \tag{27}$$

$$\text{FN penalty} = \frac{h_{EHD}(B, A)}{u - l} \tag{28}$$

5. Experimental Results

5.1. Weighting between region- and edge-based active contour

From Eq. (16), there are three parameters we can decide during the implementation, which are λ, a, and b. In this part, the experiment on the effect of weighting (λ) between the region- and edge-based active contour terms was conducted with a set to -0.01 and b set to 0.01. Table 1 shows the evaluation results with different values of λ. Note that the column S in Table 1 is the evaluation result using the initial segmentation without contour evolution.

λ	S	0	0.1	0.3	0.5	0.7	0.9	1
EHD Score ↑	82.2064	85.0191	85.5948	86.6937	87.292	87.6024	**87.6865**	87.3883
FP penalty ↓	0.1779	0.138	0.1341	0.1251	0.1205	0.1207	0.123	0.1261
FN penalty ↓	0.0204	0.0453	0.043	0.0411	0.0412	0.0372	0.0375	0.031

Table 1: The evaluation results with different values of λ.

As shown in Table 1, applying the active contour method can provide an improvement of five points on the EHD score when comparing to the initial segmentation, which indicates that the method can largely refine the crack segmentation results. Also, by looking at the different EHD scores of different λ values, we can observe that the region-based active contour is more important than edge-based active contour in this task. A possible explanation is that although the initial segmentation is close to the crack, there is still lots of noise on the image around the crack. Since we are calculating the gradient of the image in Eq. (8) for edge-based active contour, the noise can make the contour evolve to the undesired boundary. However, solely relies on region-based active contour cannot provide the best performance, which means that the edge-based information is still useful.

From Fig. 3 and Fig. 4, we can see that the refinement method is not affected by the types, or patterns, of the cracks. However, the width of cracks largely affects performance. When the crack width is sufficient to make the crack appears clearly on the image (the first two columns of both figures), the method can perform well on the refinement. For thin cracks that are not obvious on the images (the latter two columns of both figures), both region- and edge-based information are not sufficient to properly guide the contour evolution, which leads to broken segmentation and noisy boundaries.

5.2. Improve performance with gradient enhancement

As mentioned in section 5.1, the current problem of the refinement method is its low performance on thin cracks. A simple but possible solution is to enhance the region- or edge-based information that can guide the contour evolution. In this part, the edge-based information, or the gradient, was enhanced to seek possible improvement. However, enhancing the gradient of the whole image will also enhance lots of noise. Therefore, the initial segmentation from the DL model was utilized. Since we know that most of the initial segmentation results are correct and the true cracks are inside these segmentations, we can enhance only the gradient inside the initial segmentation. By multiplying the gradient inside the initial segmentation with a factor of 10, the results in Table 2 and Fig.2 are obtained.

λ	S	0	0.1	0.3	0.5	0.7	0.9	1
EHD Score ↑	82.2064	84.6481	85.5026	87.0915	87.8199	**88.0672**	87.8728	87.3883
FP penalty ↓	0.1779	0.1439	0.1352	0.1191	0.111	0.1133	0.1204	0.1261
FN penalty ↓	0.0204	0.0351	0.0359	0.0395	0.0473	0.0439	0.0386	0.031

Table 2: The evaluation results with enhanced gradient and different values of λ.

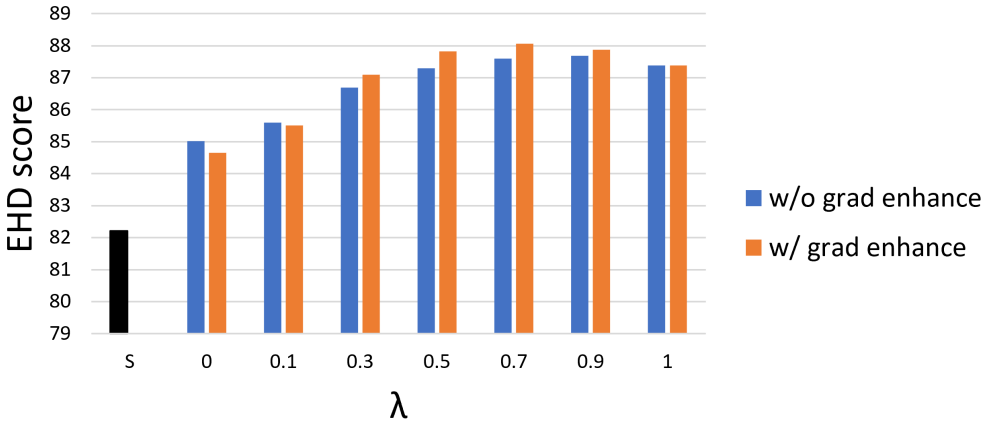

Fig. 2: The EHD scores comparison of refined segmentation results with and without gradient enhancement.

From both Table 2 and Fig. 2, we can observe that the enhanced gradient can help to improve the performance. Also, the observation in section 5.1, which is combining both the region- and edge-based active contour but with more weighting on region-based information provides the best performance, can be seen in Table 2. Fig. 3 and Fig. 4 show more details on the improvement with the enhanced gradient. For wide cracks (first two columns of the figures), the refined segmentation results are almost identical with and without gradient enhancement. However, by looking at the thin cracks (last two columns of the figures), we can observe improvement by using gradient enhancement, especially for the noisy boundaries problem discussed in section 5.1 (more evident in Fig. 4).

Notice that in both Table 1 and Table 2, we reduced the FP penalty of initial segmentation by applying the active contour refinement method. However, the FN penalty actually increases after the refinement method was applied. This is because while the width of crack segmenta-

tion is shrunk during the refinement process, the segmentation of cracks that are too thin can also be removed. This can be observed in the second and fourth columns in Fig. 4. Although in this task, the FP penalty is usually much larger than the FN penalty because the number of FP pixels dominates FN pixels and the EHD score is thus mainly depend on the FP penalty according to Eq. (26), the FN penalty is still an important indicator when we are trying to use the crack segmentation results for further pavement condition assessment. The FN problem can affect the measurement of important crack properties such as crack length. Therefore, the trade-off between the FP and FN penalties when we try to tune the parameters of Eq. (16) is an important aspect that should be further explored in future works.

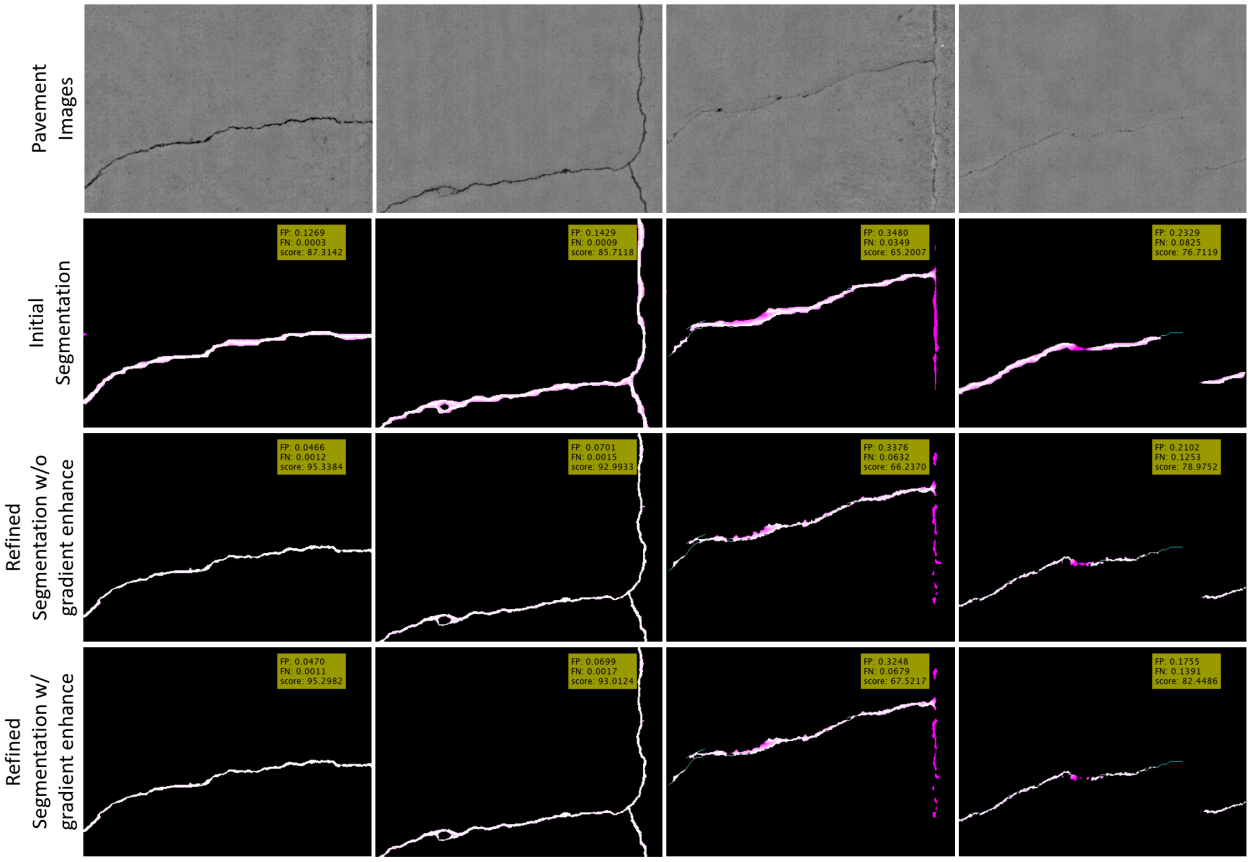

Fig. 3: Results of refined crack segmentation. Note that for segmentation images, magenta indicates FP pixels, blue indicates FN pixels, and white indicates true positive (TP) pixels.

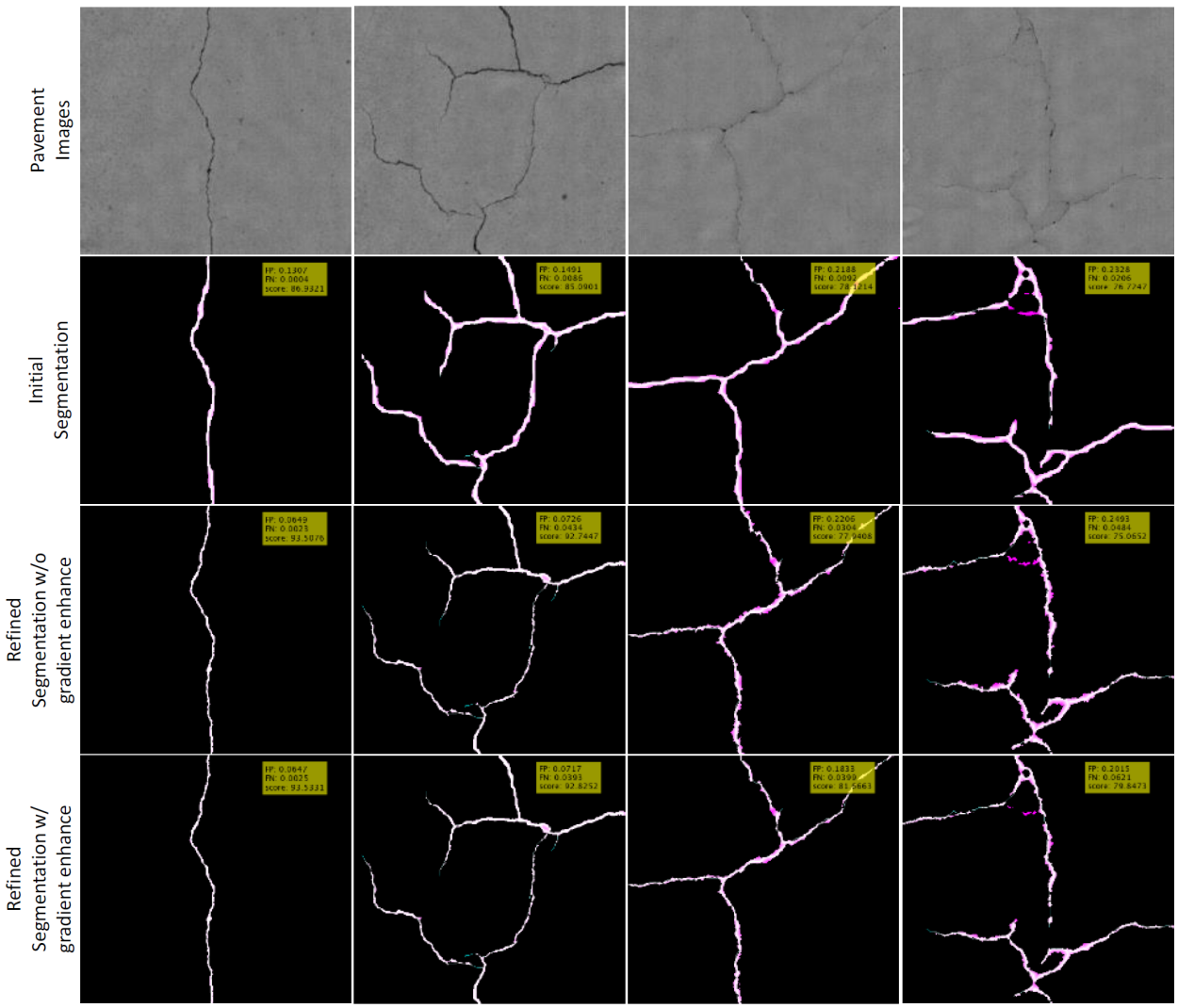

Fig. 4: Results of refined crack segmentation. Note that for segmentation images, magenta indicates FP pixels, blue indicates FN pixels, and white indicates true positive (TP) pixels.

5.3. The effects of regularization terms

In this section, the parameters a and b were adjusted with λ set to 0.7 and the gradient enhancement in section 5.2 used. The evaluation results of different values of a and b are shown in Table 3 and Table 4.

a	0.05	0.01	0	-0.01	-0.05
EHD Score ↑	**88.8085**	88.3186	88.2168	88.0672	87.1369
FP penalty ↓	0.0997	0.1086	0.1105	0.1133	0.125
FN penalty ↓	0.0714	0.0499	0.0473	0.0439	0.0349

Table 3: The evaluation results with different values of a.

b	0.05	0.01	0	-0.01	-0.05
EHD Score ↑	87.9046	**88.0672**	87.9785	87.9568	87.865
FP penalty ↓	0.1141	0.1133	0.1141	0.1146	0.1172
FN penalty ↓	0.0447	0.0439	0.044	0.0429	0.0387

Table 4: The evaluation results with different values of b.

The parameter a is the weighting of the region-based regularization term. In Table 3, we can see that a larger value of a leads to better performance. As mentioned in section 5.2, the EHD score mainly depends on the FP penalty in this task. Therefore, the more erosion we apply as we use a larger value of a, the FP pixels and thus the FP penalty is decreased, which leads to a better EHD score. However, the trade-off between the FP and FN penalties discussed in section 5.2 can also be observed here. As we increase the value of a, the FN penalty is also increasing. Therefore, we have to consider this trade-off when we try to select the value of a.

The parameter b controls the smoothness of the contour. As shown in Table 4, applying a slight smoothing effect leads to the best EHD score. By using a larger value of b, the details of cracks are removed by the smoothing effect. On the other hand, using negative values of b can lead to complex and noisy curves.

6. Conclusions

In this study, an active contour method was utilized to refine the pavement crack segmentation generated by the DL-based segmentation model. Both region- and edge-based active contour methods and regularization terms were considered and experiments on how these terms affect the refinement results were conducted. The experiment results showed that the active contour method largely improved the segmentation results with over five points higher on the EHD score. Also, combining both the region- and edge-based active contour but with more weighting on region-based information provides the best performance. Although the method performed well on wide cracks, it resulted in broken segmentation and noisy boundary on thin cracks. By providing more edge-based information to guide the contour evolution with gradient enhancement, better performance can be obtained by reducing the noisy boundary problem on thin cracks.

Although the active contour method has demonstrated its effectiveness in crack segmentation refinement, there are still several aspects that can be further investigated. First, the trade-off between the FP and FN penalties as we tune the parameters should be considered with the crack assessment we want to conduct using the refined crack segmentation. Second, as shown in Fig. 3 and Fig. 4, the FP and FN crack predictions that already exist in the initial segmentation cannot be removed using the implemented active contour method. In other words, the currently implemented method can only refine the correct predictions in the initial segmentation. Further improvements on the refinement method can be explored to potentially remove the FP and FN crack predictions. Finally, there are several parameters that need to be tuned to obtain good refinement results. However, the optimal values of these parameters can be different when applying the method on different types and conditions of pavements, and even different types of devices to capture the images. Therefore, an automatic and adaptive parameter selection method can be investigated to prevent frequent tuning of the parameters.

References

[1] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3] Chan, Tony F., and Luminita A. Vese. "Active contours without edges." IEEE Transactions on image processing 10.2 (2001): 266-277.

[4] Sussman, Mark, Peter Smereka, and Stanley Osher. "A level set approach for computing solutions to incompressible two-phase flow." (1994): 146-159.

[5] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[6] Tsai, Yi-Chang, and Anirban Chatterjee. "Comprehensive, quantitative crack detection algorithm performance evaluation system." Journal of Computing in Civil Engineering 31.5 (2017): 04017047.

[7] Kaul, Vivek, Yichang Tsai, and Russell M. Mersereau. "Quantitative performance evaluation algorithms for pavement distress segmentation." Transportation research record 2153.1 (2010): 106-113.

[8] ECE 6560 lecture notes by professor Yezzi