

CNN based multi-output regression model to estimate infrastructural surface crack dimensions adopting a generalized patch size and FWHM-based width quantification

Sudipta Debroy (✉ sudipta_rs@civil.nits.ac.in)

National Institute of Technology Sylchar <https://orcid.org/0000-0001-7370-4402>

Arjun Sil

National Institute of Technology Sylchar

Research Article

Keywords: segmentation, convolution neural network, CNN, multi-output regression, crack, length, width, estimation, crack patch, FWHM, uncertainty

Posted Date: January 15th, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-3854981/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

1 Highlights

2 CNN based multi-output regression model to estimate infrastructural surface crack
3 dimensions adopting a generalized patch size and FWHM-based width quantification

4 Sudipta Debroy, Arjun Sil

- 5** • Research highlight 1: A multi-output Regression model based on Convolution Neural Network
6 to directly estimate crack dimensions from noisy input images.
- 7** • Research highlight 2: Domain adaptation of a robust morphological algorithm called FILFINDER
8 for efficient crack segmentation.
- 9** • Research highlight 3: FWHM based crack-width quantification.

10 **CNN based multi-output regression model to estimate**
11 **infrastructural surface crack dimensions adopting a**
12 **generalized patch size and FWHM-based width**
13 **quantification**

14 Sudipta Debroy^{a,*}, Arjun Sil^a

^a*Department of Civil Engineering, National Institute of Technology Silchar, Assam, 788010, India*

15 **Abstract**

To cater the need for real-time crack monitoring of infrastructural facilities, a CNN-Regression model is proposed to directly estimate the crack properties from patches. RGB crack images and their corresponding masks obtained from a public dataset are cropped into patches of 256 square-pixels that are classified with a pre-trained Deep Convolution Neural Network, the True Positives are segmented and crack properties extracted using two different methods. A statistical test has been performed for comparison and a database prepared with the more suitable method which is then fed into the neural network model to predict crack-length, crack-width and width-uncertainty directly. The proposed model has been tested on crack patches collected from different locations. Huber Loss has been used to ensure robustness of the proposed model selected from a set of 288 different variations of it. In spite of using a limited sized data set, quite satisfactory results have been achieved.

16 *Keywords:* segmentation, convolution neural network, CNN, multi-output regression, crack,
17 length, width, estimation, crack patch, FWHM, uncertainty

18 **1. Introduction**

19 In all civil engineering structures, cracks are the earliest indications of deterioration of the strength
20 and serviceability. Hence, continuous monitoring of a structure is necessary to facilitate timely
21 remedial measures. In the last few years, several attempts have been very made in this direction to
22 develop an efficient segmentation and quantification methodology to equip the field engineers with
23 a real-time monitoring system. *In this study, the authors have attempted to develop a generalized*
24 *methodology such that the cracks can be segmented and quantified irrespective of its complexity or*
25 *tortuosity and magnitude of the background noise or illumination effects.*

*Corresponding author

Email addresses: sudipta_rs@civil.nits.ac.in (Sudipta Debroy), arjun@civil.nits.ac.in (Arjun Sil)

26 1.1. Related work

27 A distressed structure first shows its signs on its surface in the form of cracks that need to be
28 continuously monitored and addressed to prevent any further damage or collapse leading to loss of
29 lives. With this objective, several studies and proposals have been put forward by various researchers
30 who adopted different techniques for detection, segmentation and quantification of infrastructural
31 cracks from image or video sources. Some proposed mathematical morphology based methods to
32 extract the region of interest (ROI) from the image background [18, 23, 30, 38, 48, 52]. In 2014
33 Adhikari et al developed an integrated model for numerical representation of cracks comprising of
34 a crack quantification model that evaluates crack lengths based on the crack skeleton parameter
35 considering the tortuosity of the crack, a model to detect the temporal changes in damage patterns
36 based on Fourier transform of digital images eliminating the need for image registration as required
37 in traditional models, and, a neural network for prediction of crack depths and 3D visualizations of
38 crack patterns considering crack density as primary attribute [1]. Su & Yang proposed a novel edge
39 detection technique [48] while Dorafshan et al performed a comparative study on six common edge
40 detection schemes based on their data-set [12]. Nnolim used them in combination with other methods
41 [38] while Wang et al explored the limitations of edge detection methods and suggested remedial
42 measures [52]. Dung et al proposed an image segmentation model to obtain more precise information
43 such as crack path and density [13]. Images with complex backgrounds were used in some studies
44 to produce efficient and versatile models. [7, 19, 27, 57]. To produce sound models, tests to ensure
45 robustness against corrupt images has been performed [54] or complexity was introduced in the source
46 data by deliberately adding some noise to it [53]. Some authors proposed their segmentation models
47 based on the Unet, a CNN architecture originally proposed for biomedical image segmentation and
48 achieved improvement in comparison to state-of-the-art CNN models [29, 41]. Models proposed by
49 some [27, 32, 50, 59] were made ready to be deployed on a real-time monitoring system while Ni
50 et al [36] exclusively studied the measurement and analysis capability of concrete crack features
51 of an Android smartphone app. Few authors proposed combined methodologies involving wavelet-
52 based approaches to enhance contrast of input images [38] or to eliminate the shortcomings of
53 edge-detection methods [52]. Pixel-level or semantic segmentation models were suggested by some
54 researchers [7] who leveraged the capability of Deep learning networks, while Kim et al. demonstrated
55 an improvement of speed and accuracy using hierarchical convolution neural network and a multi-
56 loss update method [21]. Liu and Yeoh preferred a two-stage methodology in which a pixel-level
57 segmentation was performed on image patches from the original images followed by retrieval of crack
58 properties using fast-marching distance transform approach [30]. Up-to-date research on semantic
59 segmentation was also reviewed [41]. A downstream model for pixel-level crack detection using a
60 semantic segmentation base model was proposed that yielded a satisfactory generalization score [47].
61 Tanveer et al. studied the performance of five Deep Learning semantic segmentation models from

62 heavyweight and lightweight models categorized upon the number of parameters [50]. Recently, an
63 efficient Deep-learning based semantic segmentation and quantification model named Cracklab has
64 been proposed for efficient pixel-level crack segmentation and quantification [57] whereas a flexible
65 pixel-level crack detection network has been developed by amalgamation of Transformer and CNN
66 models [58]. The performance of Deep-learning based models has been compared with traditional
67 methods by several authors [7, 11, 12, 31, 50, 53, 54, 55, 57, 60, 61, 62]. Among them, some
68 also used transfer-learning to obtain better results with smaller data-sets [13, 27]. Authors tried
69 to automate their crack detection, classification, segmentation and quantification based on various
70 algorithms. For example, Lins et al designed a particle filter based crack detection system followed
71 by a measurement algorithm which estimated the width and length of cracks from pixel resolution
72 multiplied by the number of pixels detected [28]. Yolo architecture was used by Zhang et al. [59]
73 or used in combination with DeepLab to develop more efficient detection and segmentation systems
74 by Sarmiento et al [43]. Others evaluated the capability of various Deep learning architectures.
75 [25], [27], [60]. Nnolim and Liu et al. proposed combined methodologies in which they used Active
76 Contour Models along with other techniques to produce precise crack maps [38, 30]. New data-sets
77 were developed and provided for current and future studies from concrete structures [21, 50], brick
78 masonry walls [31], concrete dams [58], concrete railroad bridge [10] and pavements and bridges [60].
79 Relating to the study of different structural components or materials used, worthwhile contribution
80 was made by [14, 32, 46, 48] in the study concerning the defects or cracks in sewer pipes or systems
81 while some studies [11, 29, 43, 61] were focused on cracks causing deterioration in pavements or
82 concrete roads. Important contributions concerning underwater cracks occurring in dams were
83 made recently by Li et al [27] who achieved impressive model performance for complex under-water
84 scenarios and also developed a GUI based real-time pattern-detection software; whereas Zhang et al
85 proposed a pixel-level crack detection network by amalgamation of Transformer and CNN models
86 and ascertained its flexibility by testing on three public data-sets [58]. Recently, Li and Zhao [26]
87 made a first-attempt to model concrete spalling and crack damages based on Generative adversarial
88 network to generate high-resolution synthetic images to overcome the tedious problem of manual
89 image acquisition for training Deep learning networks, while, Qiu and Lau [42] proposed a rapid
90 crack detection system of tiled side-walks by integrating a YOLO based platform on an unmanned
91 aerial vehicle after considering the influence of object shadows or lighting conditions.

92 1.2. *Research gap*

93 From the extensive literature review, it has been observed that most of the researchers have adopted
94 either morphological processes or Deep Learning based processes to segment the Region-of-Interest
95 from image data. Thereafter, they have applied an additional methodology to retrieve the crack
96 dimensions from the segmented masks. The existing studies were limited mostly to a single type of

97 structural material or surface and direct estimation of crack properties from input images have not
98 been attempted so far as the authors' knowledge goes.

99 **2. Present study**

100 *2.1. Research objectives and contributions*

101 In this study, the authors have successfully eliminated the need for such a tedious methodology
102 by training a Convolution Neural Network having a terminal Regression layer, with image patches
103 labelled with values of crack properties such as length, width and width-uncertainty to build a
104 prediction model. For building the neural network training database, an efficient algorithm already
105 in use for astronomical image segmentation has been successfully adapted and implemented for the
106 first time in the current study domain (referred as Method-2 in this study) [22]. Additionally, a
107 modification of the segmentation and quantification method proposed by Liu and Yeoh [30] has also
108 been suggested (Method-1). A comparative study of the Top-1 neural network model predictions
109 with the outputs by Method-2 and Method-1 has been performed on test images collected by the
110 authors from various infrastructural surfaces and materials and the results summarised in Table 9.

111 To the best of the authors' knowledge this is the first-of-its-kind implementation of a CNN-based
112 Regression model to directly predict the crack parameters from image patches.

113 *2.2. Data Preparation*

114 *2.2.1. Public dataset:*

115 The public segmentation dataset provided by [40] have been used in this study to build-up the
116 database required to fulfil *Objective 3*. The available images were cropped into smaller 256 x 256
117 pixel sub-images due to the reason explained in *Section 2.3.1*. The corresponding masks have been
118 used to identify the *True classes* (cracked and un-cracked classes) of the cropped image patches
119 and data-imbalance corrections have been applied to the prediction probabilities as recommended
120 by [8]. Since a CNN model is free to accept any image size on which it is trained, the C15 model
121 proposed by [8] has been retrained on 256 x 256 patches cropped out of larger images from the
122 [40] data-set (which has not been used to fulfil either *Objective 1* or *Objective 2*) to prepare it for
123 the patch classification in this study. The parent image has been cropped to smaller sub-images of
124 256 x 256 pixels as per the grid shown in **Fig.3**. A first scan was performed by the retrained C15
125 model to detect cracked patches. A second set of sub-images were obtained by shifting the grids
126 with a random offset as shown in the figure. Care has been taken such that those cracked pixels
127 which previously lay at the edges of the patches now lie sufficiently inside the patches. Another
128 scan was performed on the second set of patches. Thus, there will be two sets of patches, and two
129 scans by the retrained C15 model as illustrated in the figure. The *False Positives* are eliminated

130 from the *predicted Positives* and only the True Positives are used for subsequent segmentation and
131 quantification.

132 *2.2.2. Test data collected by the authors:*

133 The following **Fig.1(a)** presents a the partial ground floor plan of the academic building in the
134 institute campus with test locations of DS1 & DS2 highlighted and labelled in red and **Fig. 1(b),**
135 **(c) and (d)** shows the pictorial views of the DS1 & DS2 locations and a partial elevation view
136 of the building respectively. **Table 2** lists out the specifications of the IphoneSE camera used for
137 data collection by the authors. The parameters that governed collection of test data were shooting
138 distance, measured crack length and measured crack width for the images. The shooting distance
139 and the conversion factor from image dimension in pixels to the corresponding object dimension
140 in millimeter for the NITS test data are presented in **Table 3**. This conversion factor have been
141 computed from the law of similar triangles schematically shown in **Fig. 2(f).**

142

Table 1: Dataset characteristics

Dataset source	ID	Image size	No. of images	No. of used images	Cropped sub-image size (pixels)	No. of sub-image patches	Remarks
[40]	PD1	2448 x (001, .., 018)	448 3264 / 4032 x 3024	3	256 x 256	313	Public dataset
	DS1	2560 x 768	1	1	256 x 256	30	Lintel of Acad bldg., NITS
	DS2	512 x 512	1	1	256 x 256	4	Column of Acad bldg., NITS
	DS3	1536 x 1536	1	1	256 x 256	36	Campus road, NITS
NITS	DS4	2048 x 2048	1	1	256 x 256	64	Hostel Room Wall, NITS
campus	DS12	3024 x 3024	1	1	2816 x 2816	1	Pavement of MSH, NITS
	DS13	3024 x 4032	1	1	2101 x 2191	1	Indoor wall of MSH, NITS
	DS14	3024 x 3024	1	1	851 x 857	1	Stair cover column of MSH, NITS

Table 2: Specifications of the device used for test data collection

Item	Description
Device	iPhone SE (1st generation) rear camera
Focal length	4.15 mm
Color space	sRGB
Exposure mode	Auto exposure
Pixel X dimension	3024
Pixel Y dimension	3024
Camera pixel size	1.22 μ m

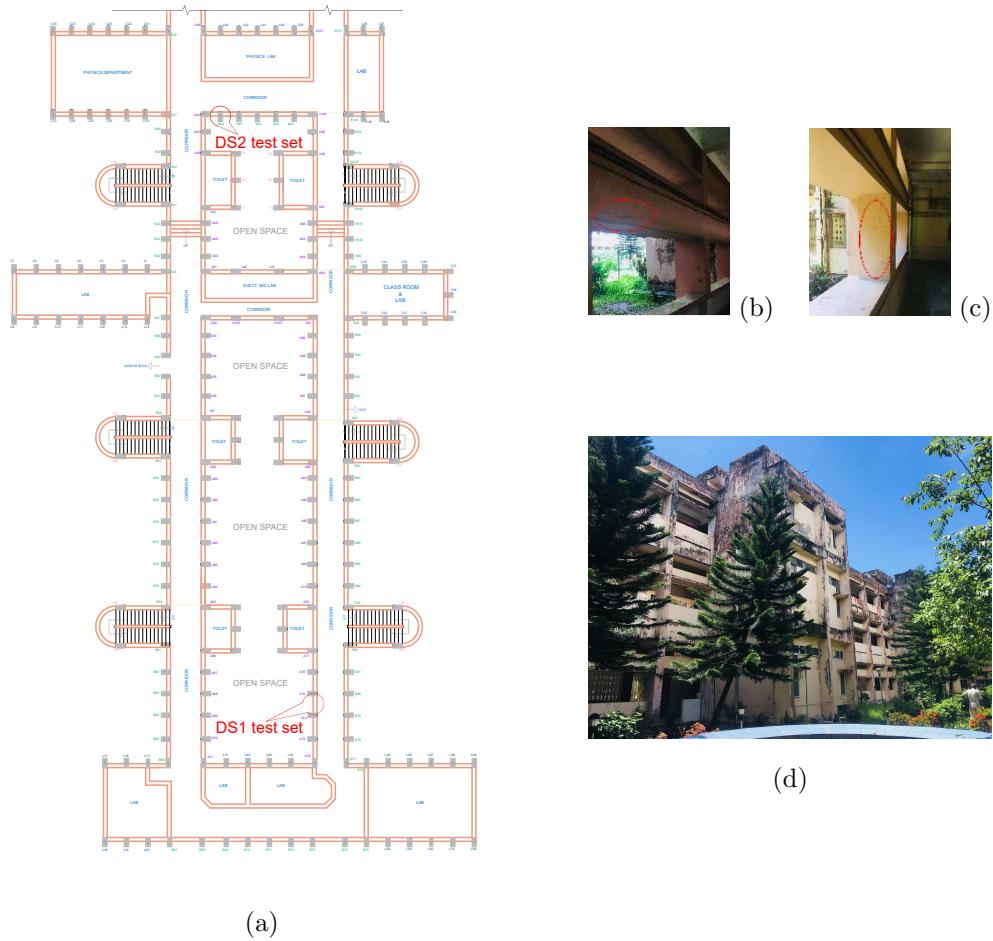


Fig. 1. (a) Partial Ground Floor Plan of the structure located within the NITS campus considered for collection of test samples (b) location of DS1 (lintel), (c) location of DS2 (column) and (d) Partial Elevation of the structure.

Table 3: Shooting distance and conversion factors from image dimension (pixels) to physical dimension (mm) for various of test specimens collected from the NIT Silchar campus

Image Id	Shooting distance (mm)	Image dim (pixel)	Object dim (mm)	Location
DS1 [LINT-0446]	1067	1	0.3136	Academic bldg.
DS2 [COL-0735]	890	1	0.2616	Academic bldg.
DS3 [PAV-0789]	773	1	0.2273	campus road
DS4 [WALL-0807]	384	1	0.1129	Hostel Room
DS12 [PAVE-5487]	965	1	0.2837	MSH
DS13 [WALL-5345]	300	1	0.0882	MSH
DS14 [WALL-5347A]	1310	1	0.3851	MSH

143 2.3. *Methodologies*

144 2.3.1. *Method1: Modification of the segmentation and quantification method proposed by Liu and*
145 *Yeoh*

146 Studying the work done by Liu and Yeoh [30], it has been observed that, if images containing cracks
147 of considerably larger dimensions are cropped into smaller patches of 32x32 pixels, then, some of the
148 crack patches are completely surrounded by the cracked pixels as illustrated in **Fig. 2 (a)**. This
149 might have been due to larger shooting distance in case of wider cracks, thereby requiring retraining
150 the neural network with a different patch size. The problem of requirement of a larger data set of
151 parent images arising out of the increase in patch-size can be solved by data augmentation [33] while
152 training a neural network. Data augmentation technique involves creating altered copies of an image
153 by rotating, flipping, mirroring, shearing, zooming etc. to the existing database thus increasing its
154 size.

155 Hence, for a generalised approach, the following modification have been proposed: The *True Positives*
156 obtained from the above procedure are cropped to sub-images of 256 x 256 pixels. The RGB sub-
157 images pertaining to the *True Positive class* were converted to 8-bit grayscale. Thereafter, Otsu's
158 thresholding was applied to generate the mask. An initial contour was drawn graphically around the
159 Region-of-interest (ROI). The Active contour segmentation has been performed with the level-set
160 function.

161

162 **Active Contour Segmentation:**

163 The Active contour model is a framework introduced by [20] for segmenting an image feature from a
164 2D noisy image. Based on the mathematical implementation, there are two approaches: (i) Snakes
165 and (ii) Level-sets. Snakes as they are commonly called are used to segment structure boundaries
166 precisely. They are based on the energy minimization principle of the initial contour that is fitted
167 to that feature of the image which relates to the ROI. A simple elastic snake is defined by a set of
168 n number of points v_i for $i = 0, \dots, n - 1$. As the number of points is constant, it should be ensured
169 that the initial snake has enough points to capture the details of the final contour. Energy consists
170 of internal energy (contour shape), external energy (Image features) and constraint energy which
171 accommodates the user interaction to guide the snakes that includes the initial contour placement.

172 The energy functional equation is given by :

$$E_{\text{snake}} = E_{\text{snake}}(v(s))ds = (E_{\text{internal}}(v(s)) + E_{\text{image}}(v(s)) + E_{\text{con}}(v(s)))ds \quad (1)$$

173 where, $v(s)$ represents a set of points $x(s), y(s)$;

174 E_{internal} = internal energy of the spline due to bending;

175 E_{image} = image forces that push the snake towards salient image features like lines, edges &

176 subjective contours;

177 E_{con} = external constraint forces responsible for putting the snake towards a desired local minimum.

178 These may be the guiding forces generated from a user interface.

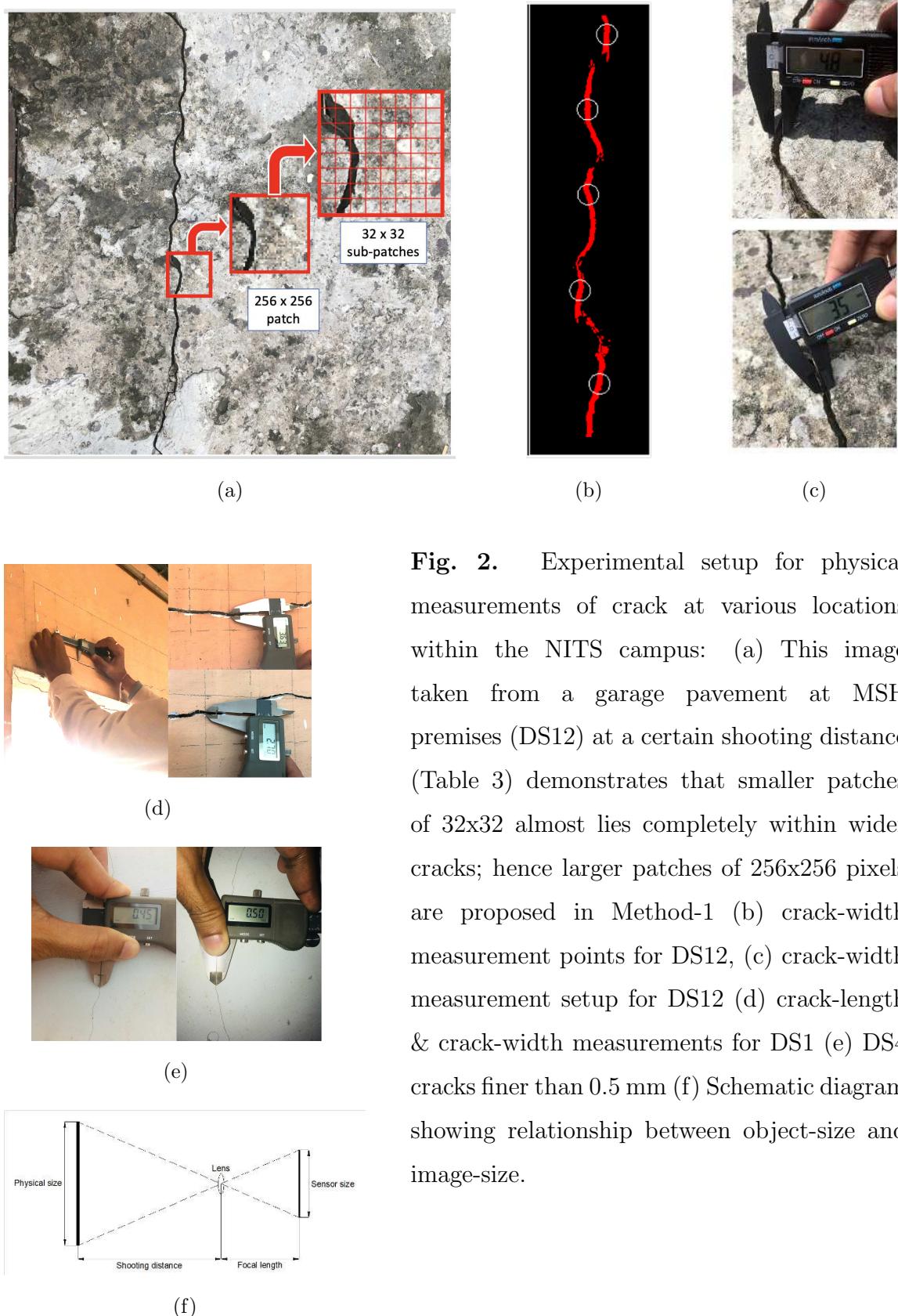


Fig. 2. Experimental setup for physical measurements of crack at various locations within the NITS campus: (a) This image taken from a garage pavement at MSH premises (DS12) at a certain shooting distance (Table 3) demonstrates that smaller patches of 32x32 almost lies completely within wider cracks; hence larger patches of 256x256 pixels are proposed in Method-1 (b) crack-width measurement points for DS12, (c) crack-width measurement setup for DS12 (d) crack-length & crack-width measurements for DS1 (e) DS4 cracks finer than 0.5 mm (f) Schematic diagram showing relationship between object-size and image-size.

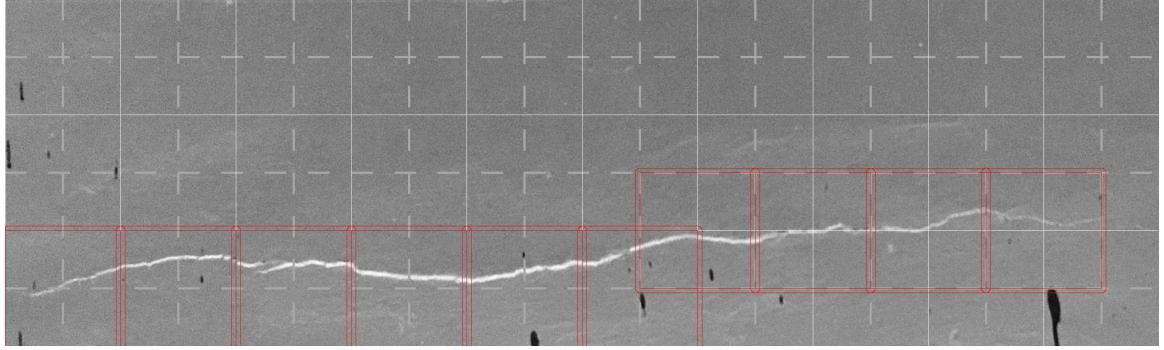


Fig. 3. Test-image DS1 sub-divided into 256 x 256 patches (continuous lines) for first scan [denoted by patch_ID_a] & 256 x 256 patches shifted by a random offset (dashed lines) for second scan [denoted by patch_ID_b] to contain those cracked pixels which lie at the patch boundary in the first scan

179 Level-set method :

180 The Level-set method approaches the contour implicitly as a plane intersecting a surface. It optimises
181 level-set function to change contour. Suppose we have a 3D surface defined by $\phi(x, y, z)$. When a
182 plane intersects the surface, we get a 2D curve defined by $\phi(x, y)$ for a certain value of z such that
183 when $\phi(x, y) = 0$, the point bearing coordinates (x,y) lies on the surface. If $\phi(x, y) > 0$, the point
184 (x,y) lies outside the surface, and when $\phi(x, y) < 0$, it lies within the surface.

185 As the height at which the plane intersects the curve changes, the shape of the curve varies. This
186 curve is called the level curve or, curve of a level-set. This level-curve can be implicitly defined by
187 intersecting the surface with a given plane. Then, the c-level set of this surface is given by:

$$\phi(x, y) = c \quad (2)$$

188 Evolution of the level-set function: In the adjacent **Fig. 4(a)** as the plane changes position with
189 the change in value of z , different level-curves can be obtained at different points of time. In this
190 case, we want to track the level-curve at $c=0$, which is the zero level set of ϕ . This curve and surface
191 evolution can be parameterized with a temporal variable t such that:

$$\phi(x, y(t), t) = 0 \quad (3)$$

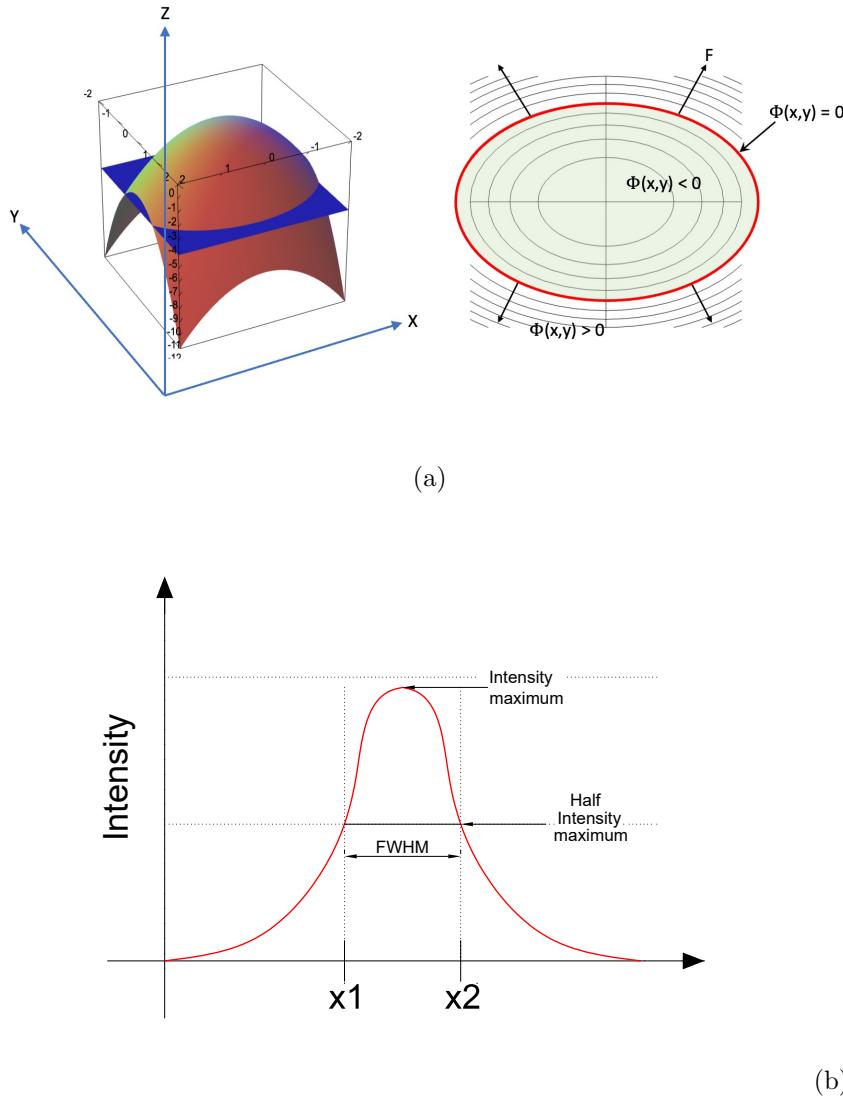


Fig. 4. (a) Level-set evolution as a horizontal plane intersects a 3D surface [39]; (b) Schematic diagram of FWHM of a curved function [15, 49]

192 To track the propagation of the level curve which is in a normal direction governed by a force F , the
193 equation of motion of ϕ is derived with respect to time t . Finally, the level set PDE is given by,

$$\phi^t = \phi + \Delta t F \|\nabla \phi\| \quad (4)$$

194 Here, F is a vector field which tells us the magnitude and direction of movement of the surface ϕ .
195 For an image to be segmented, the value of F is needed to be determined from the image. For the
196 curve to propagate quickly in the background of the image and to move slowly or stop at the edge
197 or boundary of the Region-of-Interest (ROI), F needs to be high everywhere in the image except at
198 the boundary. Thus, the value of F can be derived by taking gradients of the image,

$$g(I) = \frac{1}{1 + ||\nabla I||^2} \quad (5)$$

199 In this study, the segmentation has been performed on Fiji open-source platform with the default
200 values of the following parameters except the Advection which was adjusted by trial and error [45].

201 *Advection, A*: The speed at which the contour progresses towards the target boundary. Increasing
202 its value may speed up the process but miss the target.

203 *Propagation, P*: This value governs the expansion of the contour.

204 *Curvature, Cu*: It determines the weight of curvature in contour progression.

205 *Grayscale tolerance, Gr*: This value acts as a threshold for comparison of the grayscale values of the
206 two successive contours.

207 *Convergence criterion, Cc*: This factor is used for the spatial adjustment of the progressive contour
208 with target boundary.

209 If the contour progresses in the normal direction with speed v , then the speed function at current
210 iteration i is given by:

$$v(i) = v(A, P, Cu, Gr, Cc) \quad (6)$$

211 2.3.2. *Method 2: Mathematical morphology based crack-filament segmentation and FWHM-based
212 crack-width retrieval*

213 :

214 **FWHM**: The full-width at half-maximum is a parameter used to describe the width of the crest of
215 a curvilinear function at that point when its value reaches half of its maximum value as illustrated in
216 **Fig. 4(b)** where the abscissa represents the spatial position and the ordinate represents brightness
217 intensities of crack pixels.

218 This method is based on a set of complex algorithm named FILFINDER proposed by Koch and
219 Rosolowsky [22] for identification, segmentation and quantification of filamentary structures from
220 noisy astronomical images. The authors have adapted and implemented this robust algorithm in the
221 current study domain to effectively segment and quantify infrastructural cracks from surface images
222 with noisy backgrounds. Practically cracks occurring in structural surfaces are often complex enough
223 with multiple branch-outs and long enough to continue beyond the standard frame dimension of a
224 single image from a smartphone camera. Even if such a crack can be accommodated within a
225 single image frame by increasing the shooting distance, the extraction procedure by this method
226 becomes computationally expensive due to the complexity of the algorithm. Therefore, the source
227 image is sub-divided into smaller patches of 256 x 256 pixels keeping parity with the concept of

228 a generalised patch size adopted in Method-1. To reduce the uncertainty in the FWHM-based
229 width computation [4] of segmented skeleton filaments, fine-tuning has been performed involving
230 a trade-off between noise-level and the minimum skeleton length for generation of the mask from
231 the input image as recommended in the source literature. A thinning procedure called Medial Axis
232 Transform is performed on the masks to isolate the skeletons of the cracks. By definition, the
233 skeleton consists of a set of all region points which have the same minimum distance from two
234 separate boundary points and the minimum distance of the chosen pixel is greater than that of its
235 neighbouring ones. Subsequently, the smaller branch-outs remaining after the previous operation
236 are pruned without disturbing the spinal connectivity to obtain the smooth skeleton structure where
237 each pixel is connected to not more than two neighbouring pixels. Multiple skeleton fragments are
238 generated from a single crack patch where there are discontinuities along the path as shown in **Fig.**
239 **8 (j2),..(j6)**. The algorithm determines the filament widths by creating radial profiles around the
240 skeleton by considering a distance transform, to which a model is fitted either with a Gaussian
241 background, or using a non-parametric method to estimate the width. Where the Gaussian profile
242 cannot be fitted to the radial profile, the operation can be experimented with or without padding.
243 Here, padding means the number of pixels to be cushioned around each image. The peak and
244 background intensities in the profile are determined using the 5th and 99th percentiles respectively,
245 and FWHM value is obtained by interpolation. The uncertainty is given by the distance between
246 45th and 55th percentiles. The Rolling Hough Transform (RHT) measures the linear intensity as
247 a function of orientation in images which can be used to identify linear structure in images and to
248 map the image intensity from a 2D to a 3D space. Apart from detecting the linear structure, this
249 algorithm examines the probability of a boundary pixel of being a part of the filamentary structure
250 of a crack. The fact the RHT detects a linear structure irrespective of the overall brightness of the
251 region of interest enhances the suitability of its application to the problem of this study [6]. **Fig. 8**
252 (a_i) **to** (j_i) demonstrates the step-wise outputs by this method for some of the test patches.
253

254 *Relative advantages and disadvantages.*

255 *Method-1. :*

256 *Advantages:* The Method-1 requires manual selection of initial contour around the ROI. By implementing
257 the freehand interactive GUI-based demarcation of the initial contour around the ROI, the inclusion
258 of holes (or, noise) near the boundary can be significantly eliminated.

259 *Disadvantages:* More manual intervention required as compared to Method-2.

260

261 *Method-2. :*

262 *Advantages:* If the crack patterns continue grazing the edge of the cropped patches, this method is

263 still able to segment the crack filaments with a reasonable accuracy, unlike Method-1, as is evident
264 from the example figure above. Therefore, a second scan to contain the cracked pixels away from
265 the edges is seldom required.

266 *Disadvantages:* The Method-2 requires fine-tuning of the percentile value during the mask generation
267 and setting the cut-off skeleton filament length to suit the variation in background-noise density
268 present in the image. During the fine-tuning of percentile threshold in the masking procedure, some
269 of the background noise was still unavoidable, the area or length of which were in close proximity
270 to that of some smaller crack fragments. These smaller fragments contribute to the error in the
271 segmentation process.

272 *Statistical Test. :*

273 Let us assume a *Null Hypothesis* which states that there exists no significant difference between the
274 results of Method-1 and Method-2,

$$H_0 : \mu_{Method1} = \mu_{Method2} \quad (7)$$

275 where, $\mu_{Method1}$ and $\mu_{Method2}$ are the mean values of relative errors in both the cases of crack-length
276 and crack-width computation by the two methods respectively. Before conducting the t-test, the
277 outliers present in the data have been detected and removed by Inter-Quartile Range technique
278 [51]. The presence and absence of outliers of the individual sample groups has been illustrated with
279 corresponding box-plots in **Fig. 5 B.**

280 A paired t-test has been performed which yields a p-value of 0.000068 for crack-length-error and
281 0.030711385 for crack-width-error corresponding to 5 % significance level that enables to safely reject
282 the Null Hypothesis [9]. From a close observation of the comparative results presented in **Table 6,**
283 **7 & 8** produced by the Methods 1 and 2 based on the NITS test data DS1, DS2, DS3 & DS4, it is
284 evident that the mean error by Method-2 is slightly greater than that produced by Method-1. But,
285 Method-2 has been preferred over Method-1 to build the neural network input database due to the
286 reasons stated below:

- 287** i) lesser human intervention is required in executing Method-2 algorithm, hence, building the the
288 input database is less tedious.
- 289** ii) for crack length errors, the measured distance is actually the shortest linear distance of the crack
290 path. Hence, the relative error does not guarantee the lesser accuracy of the Method-2 output.
- 291** iii) the principle for width calculation by Method-2 based on FWHM principle is more robust.
- 292** iv) an estimate of the uncertainty involved in calculated width can be obtained.
- 293** v) It can more efficiently handle noisy data.

294

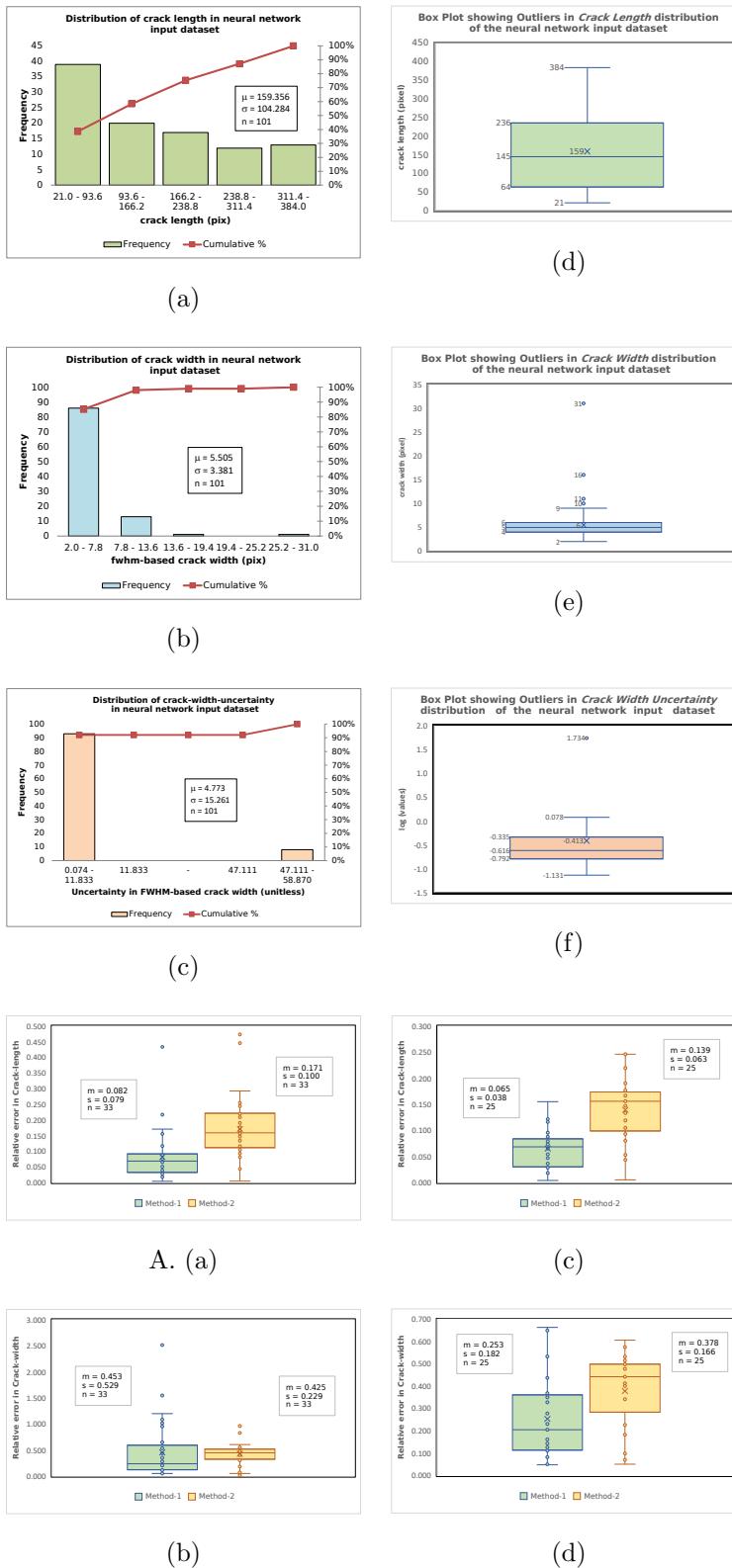


Fig. 5. A. Histograms and Box Plots of the crack-parameters extracted by Method-2 from the dataset PD1 for training and validation of neural network models: (a) Distribution of crack length (b) Distribution of FWHM-based crack width (c) Distribution of uncertainty in FWHM-based crack width (d) Outliers in crack-length distribution (e) Outliers in crack-width distribution (f) Outliers in crack-width-uncertainty distribution. **B.** (a) & (b) Box plots showing presence of outliers in the crack-length-relative-error and crack-width-relative error outputs produced by Method 1 & 2 respectively on NITS test-data, and (c) & (d) corresponding figures showing outliers removed for prerequisite of performing paired t-test.

295 2.3.3. Convolution Neural Network based Regression model:

296 The authors have tried to mimic the output of Method-2 algorithm by designing a Convolution
297 Neural Network based multi-output Regression model to predict crack length, FWHM-based crack
298 width, and, Uncertainty in crack-width predictions similar to the output produced by the algorithm
299 but without the necessity of performing the intermediate morphological operations. The neural
300 network model was fed with various crack images with three attributes connected to each image.
301 The original dimensions of each image as well as the values of the attributes were unique due to the
302 stochastic nature of occurrence of the cracks and subsequent crack properties retrieval procedure
303 used. The purpose of the convolution operation(s) was to associate a higher level of representation
304 of the crack pixels with the attributes and approximate a linear output in the final regression layer.
305 Previously, CNN based Regression models have been developed and proposed by some authors
306 in other domains like computer science and statistical investigations [34, 37, 44] or in biomedical
307 imaging applications [2, 56]. But, to the best of the authors' knowledge, this is the first of its kind
308 implementation for crack segmentation and quantification.

309

310 Input data:

311 Fig. 5A presents the distribution of the crack parameters: length, width and width-uncertainty
312 in the input data (classified from the PD1 data-set) for neural network model training, validation
313 & testing. These properties have been extracted by Method-2 decided upon the Statistical Test
314 explained below. The box plots highlight absence of any outliers in the length distribution. The
315 few outliers present in the width distribution are not excluded from our following experiments as
316 critical data points may be lost from the relatively small data-set. The few outliers present in the
317 width-uncertainty distribution can be attributed to the few non-parametric width representation by
318 Method-2.

319 Training, validation and test data-set preparation:

320 A total of 101 image patches cropped to 256 x 256 pixels from larger images of the public data-set
321 PD1 as specified in **Table 1**. The patches are then classified by the C15 model proposed by [8]
322 and the *True Positives* are considered for length and width extraction by Method-2. In the process,
323 one or multiple number of image stamps of various dimensions less than 256 pixels were generated
324 containing crack filaments along with their property information, namely crack-length, FWHM-
325 based crack-width and width-uncertainty involved. 10% of the images were split from these 101
326 images and kept as held-out test data-set. 15% of the the remaining 90% rounded to 13 images were
327 further split aside as held-out validation data-set. Finally, the models were trained on remaining
328 77 image files. The input layer of the neural network was supplied with the training images, all
329 resized to a common input size of 256 x 256 pixels along with the true values of three objective

330 variables as image labels. The input data comprised of resized (256 x 256 pixels) crack filament
331 stamps obtained from the selected cracked patches. The extracted crack length and FWHM-based
332 crack-width along with its corresponding uncertainty obtained through *Method-2* were also fed (as
333 a Dataframe) to the input layer. Due to the time-consuming process of building such a database,
334 the size of the training data-set has been limited. The input data consisted of resized (256 x 256
335 square pixels) crack patches tagged with the three attributes namely, crack-length, fwhm-width and
336 width-uncertainty.

337

338 Model development:

339 Two types of models were created at first based on the number of layers: 8 layers and 16 layers
340 respectively. These models were sub-classed into two types on whether data augmentation [33]
341 was performed or not resulting in 4 categories. Each category were trained on 6 different types of
342 Optimizers, namely SGD, RMSprop, Adam, Adagrad, Adadelta and Nadam. The learning rates
343 were kept constant for one instance and adaptive for another. Dropouts of 20% and 50% were
344 added in between the Dense layers to prevent overfitting at separate occasions. Thus, a total of
345 288 variation of models were obtained as summarised in **Table 4 & 5**.

346

347 Model architecture:

348 8-layer model:

349 A simple convolution neural network model has been constructed [see **Fig. 6(a)**] starting with an
350 *Input Layer* followed by *one block* of *Convolution and AveragePooling Layer*. The convolution layer
351 consisted of 16 filters with a kernel size of 3 x 3 with a ReLU activation function which produced 16
352 stacks of two-dimensional feature map of size 127 x 127 pixels. The output tensor is then flattened
353 and fed into two successive repetitions of *Dense Layer* [3] of 1024 neurons. A Dropout is provided
354 in between the Dense Layers to prevent over-fitting of the model during training. The output layer
355 of the functional model is Regression layer with 3 nodes to output the predicted values of Crack
356 length, FWHM-based crack width and Predicted width Uncertainty as obtained from Model-2. The
357 total number of trainable parameters were 265,311,683.

358 16-layer model:

359 A variation of the above model was achieved in this case [see **Fig. 6(b)**] with four more repetitions
360 of the Convolution and Average Pooling layer with 32, 64, 128 & 256 filters successively until the
361 size of feature maps were narrowed down to 6 x 6 pixels with 256 stacks. The remaining operations
362 were kept identical resulting in 10,883,491 trainable parameters.
363 All the 288 models were trained on 77 training images with Huber Loss function (for a default
364 Delta value = 1). The Top-3 models from each of the four category A, B, C and D were selected

365 based on the minimum validation loss achieved by the models for the Held-out Validation data set
366 consisting of 13 images. The Top-3 models from each category were then re-trained with three
367 variations of Huber Loss (for Delta = 0.1, 1.0 & 10.0) producing 9 different models from each of
368 the four categories. The Top-3 models from each category were further filtered out based on the
369 minimum validation loss resulting in best 12 models as illustrated in **Fig. 10**. These models were
370 then reconstructed with their saved 'best weights' and evaluated on the held-out validation data-set.
371 To overcome size limitations, each of the 12 models were repeatedly evaluated over the same data-set
372 (due to the stochastic nature of the learning algorithm) [5]. Finally, the **Top-3 models** were selected
373 from the evaluation results. The prediction performance of the Top-3 Models were then tested on
374 both the 'Held-out Test Data set' & 'NITS Test dataset' for Huber Loss delta values of 0.1, 1.0 & 10.0.
375

376 *Activation functions.* :

377 For the convolution layers, ReLU activation function is used while a linear activation function is
378 used for the Regression layer.

379 *Optimizers.* :

380 Six different optimizers namely, SGD (Stochastic Gradient Descent), RMSprop, Adam, Adagrad,
381 Adadelta and Nadam form the backbone of six different type of Models in each category A, B, C
382 and D for a particular type of learning rate and a specific Dropout value.

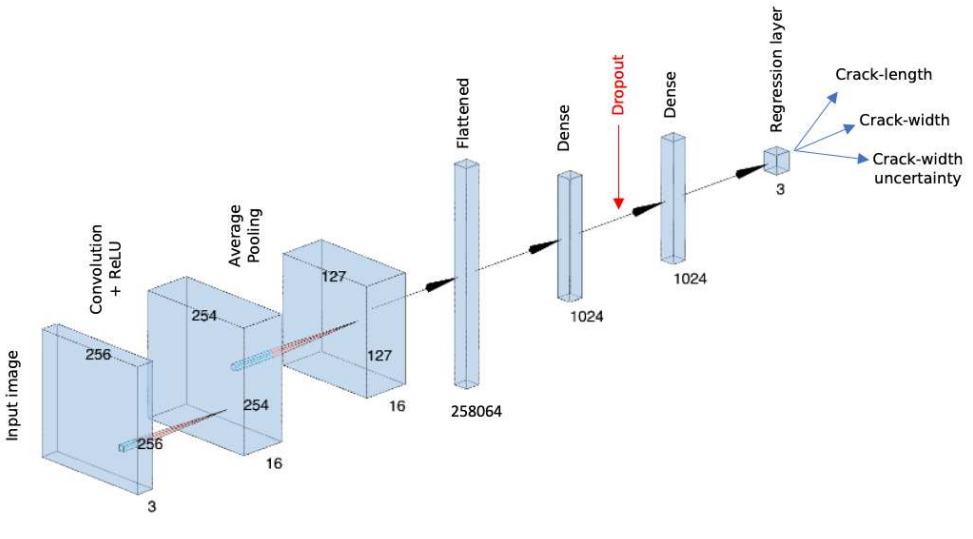
383

384 *Loss function.* :

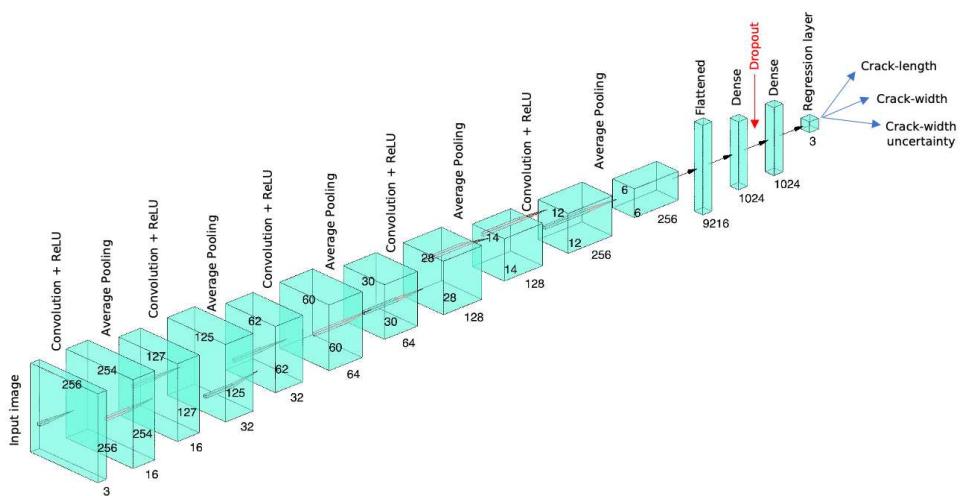
385 Since, a small data set have been used for training the neural network, the few outliers present in
386 the data, visible in **Fig. 5A (e) & (f)**, were not removed to prevent loss of critical data points. To
387 compensate this shortcoming, a suitable loss function known as *Huber Loss* which is less sensitive
388 to outliers has been used [16, 17] and performance comparison has been shown with other *loss*
389 *functions*. The δ parameter that controls the transition of the Huber Loss from an absolute value
390 function to its quadratic form has been chosen manually as discussed later in conjunction with **Fig.**
391 **13**.

392 Mathematically, Huber Loss is given by:

$$L_\delta(\text{error}) = \begin{cases} \frac{1}{2} \cdot \text{error}^2 & \text{for } |\text{error}| \leq \delta, \\ \delta(|\text{error}| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \quad (8)$$



(a)



(b)

Fig. 6. CNN based Regression model architectures (a) model with eight layers (including Dropout) (b) model with sixteen layers (including Dropout)

Table 4: Parameters and their values which were kept constant in training the neural network models

Parameters	constant values
Train: Validation: Test ratio	1: 0.17: 0.14
Activation function	ReLU; Linear
Train batch size	4
Validation batch size	1
Test batch size	1
Loss function	Huber
Early stopping patience	50
Early stopping min-delta	1e-7
ROP factor	0.1
ROP patience	25

393 *Fine-tuning the hyper-parameters.* :

394 Momentum is used to accelerate the speed of training and, hence, it has not been used here. Instead,
395 a Learning Rate Scheduler is employed to reduce the learning rate by 10% if the validation loss does
396 not improve after a certain number of epochs (patience) [provide citation]. The adaptive learning
397 rate is governed by the equation:

$$lr_{k+1} = \eta * lr_k \quad (9)$$

398 where,

399 lr = learning rate
400 k = current iteration
401 k+1 = next iteration
402 η = Reduce-on-plateau (ROP) factor

403 The following **Table 4** presents the constant parameters involved and **Table 5** shows the nomenclature
404 of the CNN based Regression models. A total of 288 models have been generated by combining the
405 following variables: No. of layers, Data augmentation, Learning rate, Dropout and type of optimizer
406 considered. The model names vary as per i=1,2,3 & j=a,b,c. Initial learning rate varies as 0.1, 0.01
407 & 0.001 as both i & j varies. The early stopping call back ends the training if the validation loss
408 does not improve after 50 epochs. The min-delta signifies the amount by which the validation loss
409 has to improve.

Table 5: Nomenclature of the neural network models

No. of layers	Data augmentation	Learning rate	Dropout																							
			0.2						0.5						SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
8	No	constant	Ai	Bi	Ci	Di	Ei	Fi	2Ai	2Bi	2Ci	2Di	2Ei	2Fi	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
		adaptive	Aj	Bj	Cj	Dj	Ej	Fj	2Aj	2Bj	2Cj	2Dj	2Ej	2Fj	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
	Yes	constant	Gi	Hi	Ii	Ji	Ki	Li	2Gi	2Hi	2Ii	2Ji	2Ki	2Li	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
		adaptive	Gj	Hj	Ij	Jj	Kj	Lj	2Gj	2Hj	2Ij	2Jj	2Kj	2Lj	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
16	No	constant	AAi	BBi	CCi	DDi	EEi	FFi	2AAi	2BBi	2CCi	2DDi	2EEi	2FFi	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
		adaptive	AAj	BBj	CCj	DDj	EEj	FFj	2AAj	2BBj	2CCj	2DDj	2EEj	2FFj	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
	Yes	constant	GGi	HHi	IIi	JJi	KKi	LLi	2GGi	2HHi	2IIi	2JJi	2KKi	2LLi	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	
		adaptive	GGj	HHj	IIj	JJj	KKj	LLj	2GGj	2HHj	2IIj	2JJj	2KKj	2LLj	SGD		RMSprop		Adam		Adagrad		Adadelta		Nadam	

410 **3. Results and Discussions**

411 Both the methods proposed here can be applied to segment parent un-cropped images. The
412 tedious human intervention is increased significantly in case of Method-1 in such a case. On the
413 other hand, the Method-2, although more automated but is computationally more expensive due
414 to the complexity of the algorithm involved. Hence, the decomposition of a larger image with
415 complex cracks into smaller patches containing simpler cracks with fewer branch-outs reduce the
416 time complexity of the algorithm. **Fig.7 B and 8** illustrates the step-wise outputs of segmentation
417 and quantification procedure of sample image patches from the test-set DS1 by Method 1 and 2
418 respectively (see in conjunction with **Table 6, 7 and 8**).

419 For segmentation by Method-1, the operations have been performed on the Fiji open-source platform
420 [45] in which an RGB image patch is first converted into a single channel gray scale image on which
421 Otsu thresholding is performed followed by morphological operations are performed to generate a
422 binary (8-bit) mask. Then, an initial contour is graphically drawn just outside the perimeter of
423 the region-of-interest to be segmented and level-set segmentation is performed with default values of
424 parameters as described in Section (Methodologies). The area of the ROI in square pixels is obtained
425 and a skeleton of the crack ROI is generated by applying a thinning algorithm [24]. The length of
426 the skeleton is extracted and the average width of the crack is computed by dividing the area with
427 it. A maximum number of 900 iterations were required for the level-set to converge which took less
428 than 120 secs on a Mac M1. The mean relative error in crack length computed for DS1, DS2, DS3 &
429 DS4 against length measured with a slide caliper [see **Fig. 2(c),(d) and (e)**] are 0.033, 0.303, 0.217
430 and 0.078 respectively whereas 0.792, 0.717, 0.064 and 0.249 are the corresponding errors obtained
431 in crack-width retrieval as compiled in **Table 6 and 8** respectively.

432 **Fig. 8** presents the results obtained at different steps of the segmentation and quantification
433 procedure by **Method 2** such as: generation of a binary mask of the crack patch, the identification
434 and segmentation of the filamentary structure of the crack contained in the patch from a noisy
435 background, and the generation of separate skeletons of multiple crack fragments resulting out of
436 the discontinuities in between. The length of each skeleton is depicted in the corresponding skeletons
437 both in image pixels and corresponding equivalent physical dimensions (millimeter). The lengths
438 and widths of the cracks for all the patches used have been quantified and provided in **Tables 6 &**
439 **8** respectively.

440

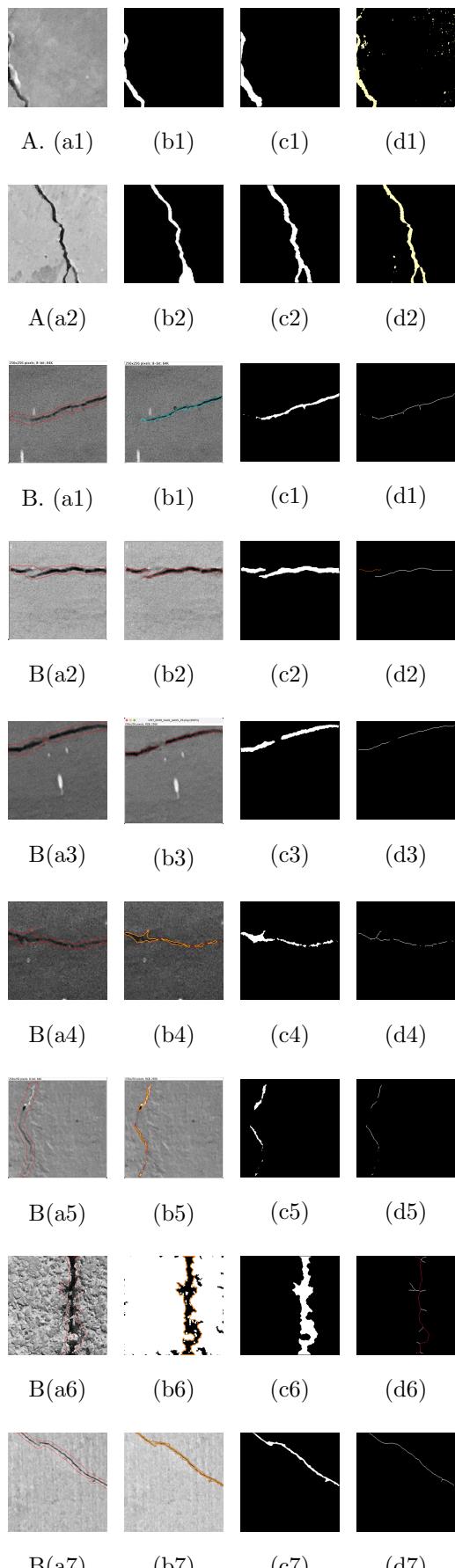


Fig. 7. **A.** Comparison among *masks provided by the original data source of PD1* and those generated by the Method-1 and Method-2 for *typical training patches (for the neural network)* (a1) PD1-018-patch-010: crack grazing the edge; (a2) PD1-018-patch-42 : crack away from the edge; (b1 & b2) binary mask by [40]; (c1 & c2) binary mask by Method-1; (d1 & d2) binary mask by Method-2; **B.** Segmentation results by Method-1 on some test patches: DS1-patch20_a, DS1-patch22_a, DS1-patch25_a, DS1-patch18_b, DS2-patch11, DS3-patch02 & DS4-patch13: (a_i) Initial single contour; (b_i) Converged level-set along the ROI boundary; (c_i) Generated binary mask; and, (d_i) longest path of the crack skeleton.

a

^aIt is to be noted that the source mask has been cropped from a larger mask of the parent image, whereas, the masks presented in figures (c) & (d) are generated from the corresponding RGB patches only.

Owing to the considerably rough surface texture of the bituminous pavement compared to the concrete or masonry surfaces, the patch DS3-02 needed extra pre-processing steps for ROI segmentation and skeleton generation.

Table 6: Crack lengths quantified by Method-1 against measured lengths on validation datasets DS1, DS2, DS3 & DS4 collected from structures and pavements at randomly chosen locations within the NITS campus

Image	Patch	Shooting	Calc.	Calc.	Meas.	Rel.	Remarks
Id	Id	distance (mm)	skeleton length (pixels)	physical length (mm)	physical length (mm)	error	[col 5]
	patch_20_a		265.066	83.125	81.45	0.021	
	patch_21_a		269.882	84.635	80.72	0.049	
	patch_22_a		275.053	86.257	80.97	0.065	
	patch_23_a		268.326	84.147	82.67	0.018	
DS1	patch_24_a		262.468	82.310	81.95	0.004	shortest
	patch_25_a	1067	248.196	77.834	80.24	0.030	
(lintel)	patch_26_a		253.912	79.627	81.35	0.021	crack-length
	patch_15_b		245.400	76.957	80.06	0.039	
	patch_16_b		245.082	76.858	79.64	0.035	
	patch_17_b		243.822	76.463	80.66	0.052	
	patch_18_b		251.652	78.918	81.16	0.028	
Mean (μ)			257.169	80.648	80.99	0.033	
DS2	patch_01	890	206.209	53.944	65.11	0.171	-do-

(col.)	patch_11	135.610	35.476	62.63	0.434	
	Mean (μ)	170.910	44.710	63.87	0.303	
<hr/>						
DS3 (bitu. pave.)	patch_02	773	307.794	69.962	57.51	0.217
<hr/>						
	Mean (μ)	307.794	69.962	57.51	0.217	
<hr/>						
	patch_01	280.563	31.676	29.52	0.073	
	patch_02	349.357	39.442	38.35	0.028	
	patch_03	43.042	4.859	4.47	0.087	
	patch_13	281.848	31.821	29.25	0.088	
	patch_14	262.049	29.585	27.69	0.068	
	patch_24	49.284	5.564	5.37	0.036	
	patch_25	283.894	32.052	29.81	0.075	
DS4	patch_26	268.811	30.349	28.79	0.054	shortest
(masonry wall)	patch_27	384	285.309	32.211	29.92	0.077
	patch_40	162.255	18.319	17.50	0.047	length
	patch_41	232.711	26.273	23.54	0.116	
	patch_51	51.799	5.848	5.68	0.030	

patch_52	240.853	27.192	23.54	0.155
patch_53	289.208	32.652	29.80	0.096
patch_54	298.664	33.719	31.25	0.079
patch_55	52.385	5.914	5.63	0.050
patch_65	232.409	26.239	24.35	0.078
patch_66	302.664	34.171	30.60	0.117
patch_67	290.480	32.795	29.24	0.122
Mean (μ)	224.083	25.299	23.38	0.078

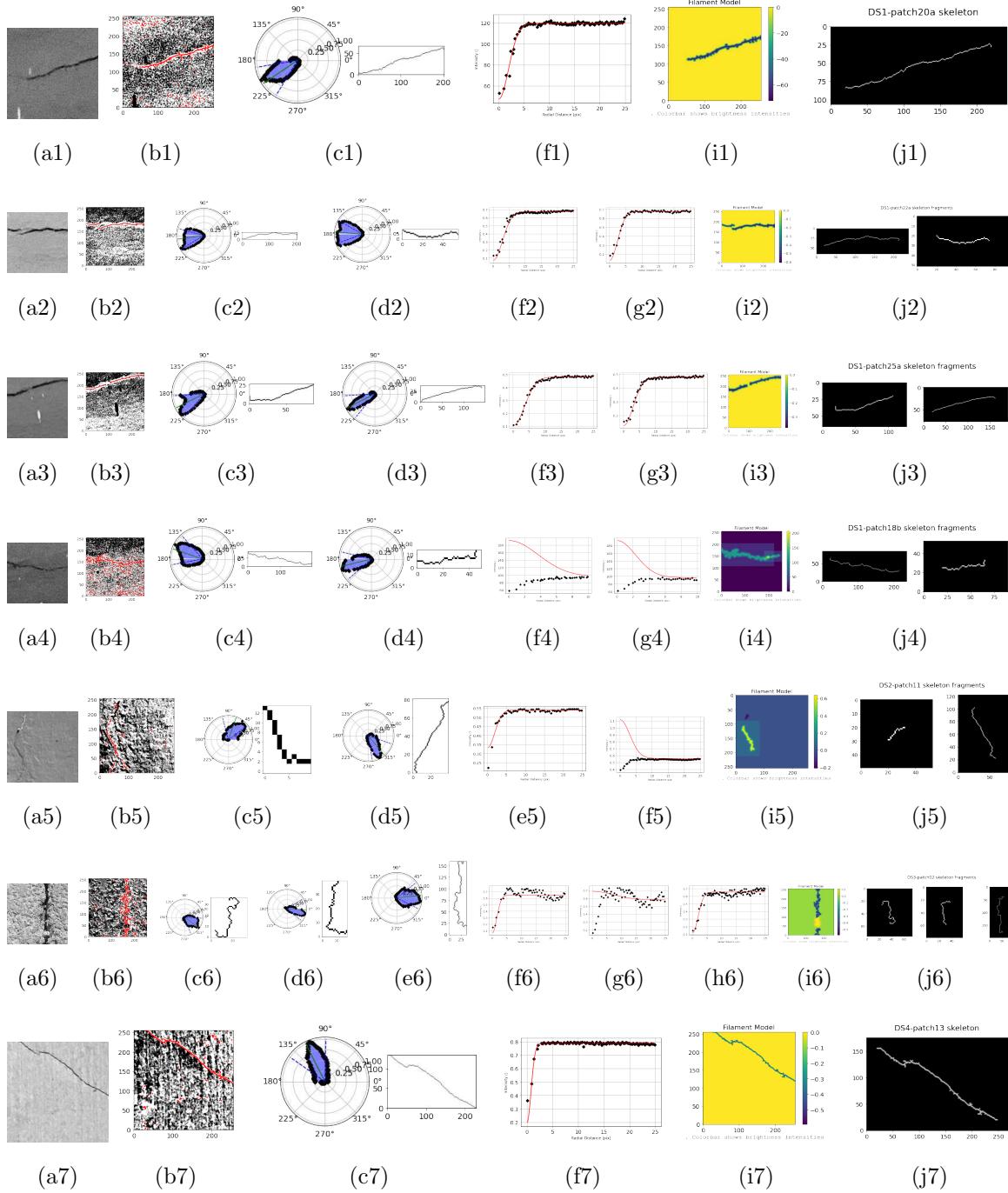


Fig. 8. Output at different steps during the segmentation and quantification procedure by Method-2 on some test patches DS1-patch20_a, DS1-patch22_a, DS1-patch25_a, DS1-patch18_b, DS2-patch11, DS3-patch02 & DS4-patch13: (a_i) Input image; (b_i) Generated mask; (c_i),(d_i) & (e_i) RHT; (f_i),(g_i) & (h_i) Radial profile (i_i) Filament model (j_i) crack fragments segmented as separate filament stamps along with their corresponding length, FWHM-width and width-uncertainty.

Table 7: Crack lengths quantified by Method-2 against measured lengths on validation datasets DS1, DS2, DS3 and DS4

Image	Patch	Shooting	Calc.	Calc.	Meas.	Rel.	Remarks
Id	Id	distance	skeleton	physical	physical	error	[col 5]
		(mm)	length	length	length		
			(pixels)	(mm)	(mm)		
	patch_20_a		245.948	77.129	81.45	0.053	
	patch_21_a		258.628	81.106	80.72	0.005	
	patch_22_a		303.768	95.262	80.97	0.177	
DS1	patch_23_a		308.545	96.760	82.67	0.170	shortest
	patch_24_a	1067	299.820	94.023	81.95	0.147	crack-length
(lintel)	patch_25_a		283.126	88.788	80.24	0.107	
	patch_26_a		290.313	91.042	81.35	0.119	
	patch_15_b		284.993	89.374	80.06	0.116	
	patch_16_b		328.25	102.939	79.64	0.293	
	patch_17_b		310.844	97.481	80.66	0.209	
	patch_18_b		195.594	61.338	81.16	0.244	
Mean		-	282.712	88.658	80.99	0.149	
(μ)							
DS2	patch_01	890	218.898	57.264	65.11	0.121	-do-
(col.)	patch_11		125.819	32.914	62.63	0.474	

	Mean (μ)	-	172.359	45.089	63.87	0.297	
DS3							
(bitu. pave.)	patch_02	773	365.947	83.180	57.51	0.446	-do-
	Mean (μ)	773	365.947	83.180	57.51	0.446	
	patch_01		302.840	34.191	29.52	0.158	
	patch_02		354.512	40.024	38.35	0.044	
	patch_03		49.674	5.608	4.47	0.255	
	patch_13		300.773	33.957	29.25	0.161	
	patch_14		268.029	30.260	27.69	0.093	
	patch_24		59.275	6.692	5.37	0.246	
	patch_25		305.953	34.542	29.81	0.159	
DS4	patch_26		297.492	33.587	28.79	0.167	
(masonry wall)	patch_27	384	315.423	35.611	29.92	0.190	-do-
	patch_40		171.250	19.334	17.50	0.105	
	patch_41		259.820	29.334	23.54	0.246	
	patch_51		61.574	6.952	5.68	0.224	
	patch_52		218.169	24.631	23.54	0.046	
	patch_53		305.251	34.463	29.80	0.156	

patch_54	313.666	35.413	31.25	0.133
patch_55	60.825	6.867	5.63	0.220
patch_65	249.317	28.148	24.35	0.156
patch_66	304.382	34.365	30.60	0.123
patch_67	279.723	31.581	29.24	0.080
Mean (μ)	235.681	26.608	23.38	0.156

1

¹The measured length of the crack gives the linear distance between the initial and final points of the crack path ignoring the discontinuity in between. In contrast, the calculated length is the length of the actual skeleton of the crack excluding the discontinuities. The length of the segmented crack by Method-1 depends more on human intervention for drawing the initial contour, whereas the process by Method-2 is more automatic and precisely excludes the discontinuities. Observation: Thus, the larger errors in length calculation by Method-2 can be justified.

Table 8: Comparison between mean crack-width retrieved by two methods against measured mean crack-width on validation datasets DS1, DS2, DS3 & DS4

Image	Patch Id	Meas.	Meth-1,	Rel.	Meth-2,	Rel.	Meth-2
	Id	mean	calc.	error	calc.	error	width-
		width (mm)	width		width		uncertainty
	patch_20_a	≈ 1.0	1.663	0.663	1.442	0.442	0.150
	patch_21_a	1.58	2.605	0.649	1.735	0.098	0.095
	patch_22_a	2.23	3.205	0.437	1.308	0.413	0.181
	patch_23_a	2.81	3.263	0.161	2.614	0.070	0.112
	patch_24_a	3.05	3.675	0.205	2.495	0.182	0.137
DS1	patch_25_a	2.61	2.907	0.114	2.480	0.050	0.164
(lintel)	patch_26_a	2.12	3.069	0.448	1.963	0.074	0.257
	patch_15_b	1.81	3.537	0.954	2.356	0.302	0.097
	patch_16_b	1.47	3.746	1.548	2.084	0.418	0.231
	patch_17_b	≈ 1.0	3.515	2.515	1.960	0.960	0.183
	patch_18_b	≈ 1.0	2.014	1.014	1.980	0.980	0.447
Mean (μ)		1.88	3.018	0.792	2.038	0.363	-
DS2	patch_01	≈ 1.0	2.199	1.199	1.015	0.015	2.427

(column)	patch_11	≈ 1.0	1.235	0.235	1.551	0.551	28.791
	Mean (μ)	1.00	1.717	0.717	1.283	0.283	-
DS3							
(bitu. pave.)	patch_02	4.86	4.547	0.064	0.835	0.828	1.177
	Mean (μ)	4.86	4.547	0.064	0.835	0.828	-
	patch_01	0.50	0.684	0.368	0.387	0.226	0.802
	patch_02	0.50	0.395	0.211	0.234	0.531	0.536
	patch_03	0.50	1.042	1.083	0.243	0.515	0.253
	patch_13	0.50	0.541	0.082	0.213	0.575	0.145
	patch_14	0.50	0.323	0.354	0.243	0.514	0.087
	patch_24	0.50	0.572	0.144	0.197	0.605	0.395
	patch_25	0.50	0.444	0.112	0.250	0.499	0.119
DS4	patch_26	0.50	0.437	0.126	0.261	0.477	0.112
(masonry wall)	patch_27	0.50	0.325	0.350	0.255	0.489	0.169
	patch_40	0.50	0.361	0.278	0.251	0.498	0.149
	patch_41	0.50	0.577	0.154	0.275	0.449	0.148
	patch_51	0.50	0.571	0.143	0.278	0.444	0.662
	patch_52	0.50	0.456	0.087	0.329	0.342	0.527

patch_53	0.50	0.385	0.231	0.292	0.416	0.355
patch_54	0.50	0.474	0.051	0.275	0.450	0.204
patch_55	0.50	0.336	0.328	0.240	0.520	0.501
patch_65	0.50	0.475	0.050	0.300	0.400	0.591
patch_66	0.50	0.476	0.048	0.327	0.347	0.565
patch_67	0.50	0.766	0.533	0.329	0.341	0.767
Mean (μ)	0.50	0.507	0.249	0.273	0.455	-

2

²Due, to surface roughness in the case of DS1 & DS2, cracks finer than 1.0 mm could not be measured accurately under the naked eye. Hence, the approximate crack-width of 1.0 mm is used for Relative Error calculation. For DS4, the average crack width has been found to be finer than 0.5 mm but it has been impossible for the naked eye to fetch accurate dimensions although the least count of the slide caliper used was 0.01 mm. Hence, an approximated value of 0.5 mm has been adopted for relative error calculation.

On the other hand, due to the immense surface roughness of the bituminous pavement in comparison to the concrete structural components, the calculated crack-width by the Method-1 is in close agreement to that of the measured width whereas the FWHM-based width output by Method-2 differs largely.

* Since, a patch may contain multiple fragments of a crack during segmentation, the mean values of uncertainty for the fragments are provided.

441 *3.1. Performance of the CNN-based Regression model*

442 The CNN based Regression models presented in **Table 5** were trained on the training data consisting
443 of 77 images as per the data split stated earlier. All the models have been trained for a maximum
444 number of 1000 epochs with an *early stopping function* to monitor the *minimum validation loss* on
445 13 (thirteen) held-out validation images and employing the robust **Huber Loss** function with a
446 default value of the delta parameter equal to 1.

447

448 The minimum validation loss achieved by each series of models namely A,...,F, 2A,...,2F, G,...,L,
449 2G,...2L, AA,...,FF, 2AA,...,2FF, GG,...,LL, and 2GG,...,2LL are illustrated in the bar charts in
450 **Fig. 10 (a-h)**. Thereafter, the Top-3 models achieving minimum validation losses are designated
451 and are used to make predictions on both the held-out test data (11 images) and 33 image patches
452 collected by the authors from four random locations within the NITS campus. Two of the Top-3
453 models belonged to the data-augmented class. Finally, the predictions for **crack length** and **crack**
454 **width** made by the best neural network model on both the test data sets are compared with the
455 outputs of Model 1 and Model 2.

456

457 The following figures **9 (a), (b) and (c)** present the loss curves obtained during training the
458 Top-3 neural network models respectively. In each of the figures, the logarithmic (base 10) values
459 of Huber Loss for a specific delta value (indicated in parenthesis) against increasing epochs have
460 been considered. The held-out validation data-set has been used for validation. The comparison of
461 training loss and validation loss are shown using different line-types as indicated in the legends. Due
462 to random initialization of model weights, each model is run for three successive times and the mean
463 value has been considered. The average minimum validation loss achieved by the models 2HHc-Z,
464 2HHc-X and C3-X are 15.910, 18.583 and 18.916 respectively.

465

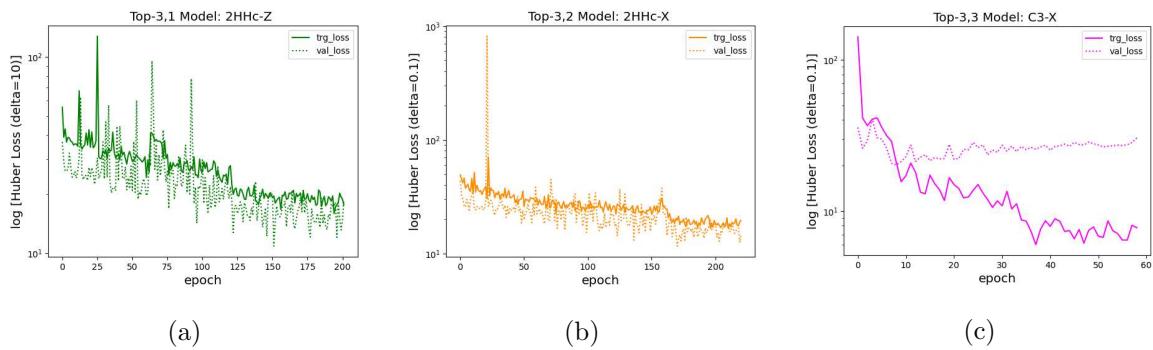


Fig. 9. Loss curves of Top-3 neural network models

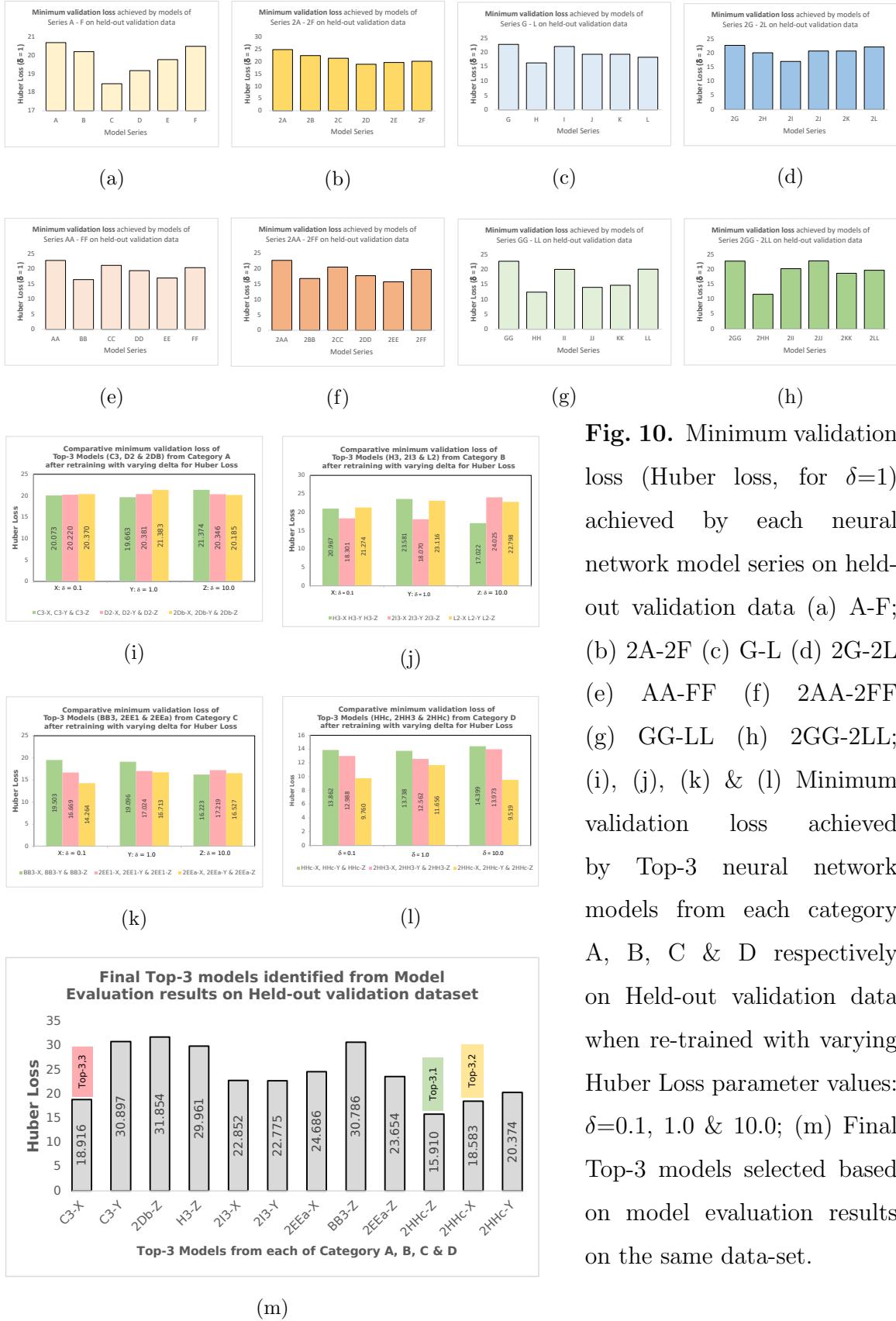


Fig. 10. Minimum validation loss (Huber loss, for $\delta=1$) achieved by each neural network model series on held-out validation data (a) A-F; (b) 2A-2F (c) G-L (d) 2G-2L (e) AA-FF (f) 2AA-2FF (g) GG-LL (h) 2GG-2LL; (i), (j), (k) & (l) Minimum validation loss achieved by Top-3 neural network models from each category A, B, C & D respectively on Held-out validation data when re-trained with varying Huber Loss parameter values: $\delta=0.1$, 1.0 & 10.0; (m) Final Top-3 models selected based on model evaluation results on the same data-set.

466 ***3.2. Model Predictions:***

467 The Top-3 models were tested on both the test data sets and the predictions of crack-length, crack-
468 width and width-uncertainty are presented in the form of scatter plots in **Fig. 11** with *true values*
469 along the x-axis and corresponding *predicted values* along the y-axis. The 95% confidence intervals
470 for each of the linear fits are displayed and the Pearson Correlation coefficients for each of the
471 predictions are indicated. It is observed that the *C3-X* model achieved the highest correlation
472 of 92% and 91% for crack length and crack-width respectively while Model *2HHc-X* predictions
473 achieved maximum positive correlation of 0.45 for width-uncertainty on the Held-out test set. On
474 the other hand, looking at the predictions for NITS Test Data, highest value of $r = 0.507$ has been
475 achieved by the Model *2HHc-X* for crack-length whereas no positive correlations were achieved for
476 the crack-width predictions. For width-uncertainty predictions, the maximum positive correlation
477 achieved was 0.073 by Model *2HHc-Z*. The width uncertainty scatter plots are shown in log-scale to
478 accommodate the outlying true values obtained from Method-2 due to the non-parametric method
479 used by Method-2 for width estimation resulting in large width-uncertainties (see **Table 8**).
480

481 A comparative study of the mean predicted and mean true values of the objective crack parameters
482 for both the test data sets and for the Top-3 models has been graphically presented in the form of
483 bar charts in **Fig. 12**. For, the Held-out test data set, the maximum and minimum relative errors
484 obtained for length predictions are 0.284 and 0.017 respectively, those for width predictions are 0.159
485 and 0.086 respectively, and, the maximum and minimum relative errors for width-uncertainty are
486 2.667 and 0.473 respectively.
487

488 The squared error loss accentuates the contribution of the outliers in the training data during the
489 regression fitting, which is mitigated considerably by the Huber Loss function. Although the model
490 trained using the value of the delta parameter equal to 10 has been found to be the top performing
491 model *2HHc-Z*, the predicted values of the three target variables *crack-length*, *crack-width* & *crack-*
492 *width uncertainty* for both the test data sets have been graphically plotted in **Fig. 13** for different
493 values of *delta* to demonstrate the variation of Huber Loss compared to the Squared Error and
494 Absolute Error. For held-out test set, the loss curves corresponding to $\delta = 1$ and $\delta = 10$ practically
495 coincides except for the crack-length where the Absolute Error coincides with the loss curve for
496 $\delta = 1$. On the other hand, for NITS test set, loss curve for $\delta = 1$ coincides with the Absolute
497 error curve except for the crack-width predictions in which the loss curves for $\delta = 1$ and 10 partially
498 coincides in the lower range of the Absolute Error.
499

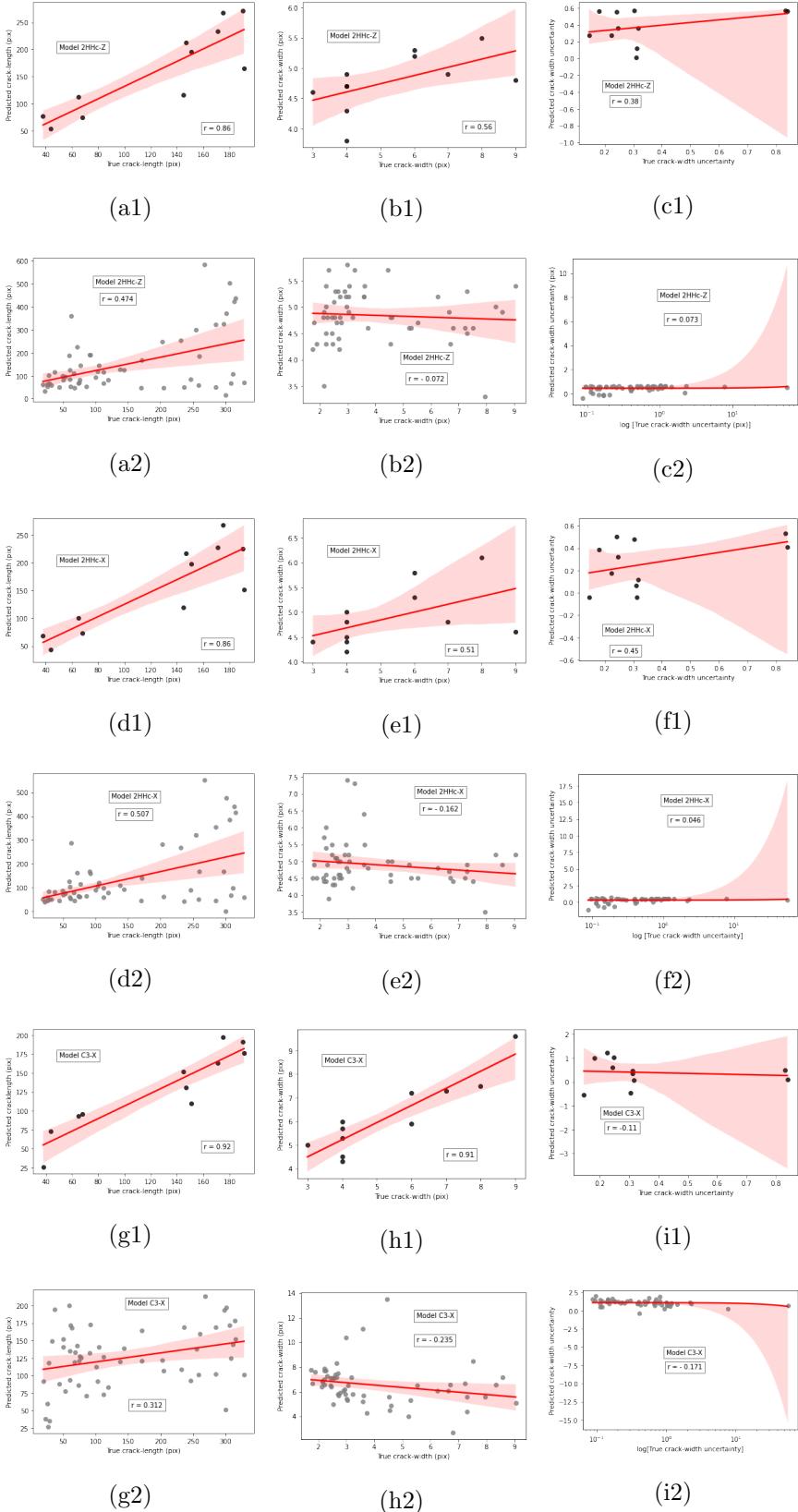


Fig. 11. Scatter plots illustrating predictions on (a1,...,i1) *Held-out test data* and (a2,...,i2) *NITS test data* against 95% Confidence Intervals by **Top-3 neural network models: 2HHC-Z** [(a),(b) & (c)]; **2HHC-X** [(d),(e) & (f)]; & **C3-X** [(g),(h) & (i)] respectively: (a),(d) & (g) Predicted length vs. True Length (b),(e) & (h) Predicted Width vs. True Width (c),(f) & (i) Predicted Width-uncertainty vs True Width-uncertainty

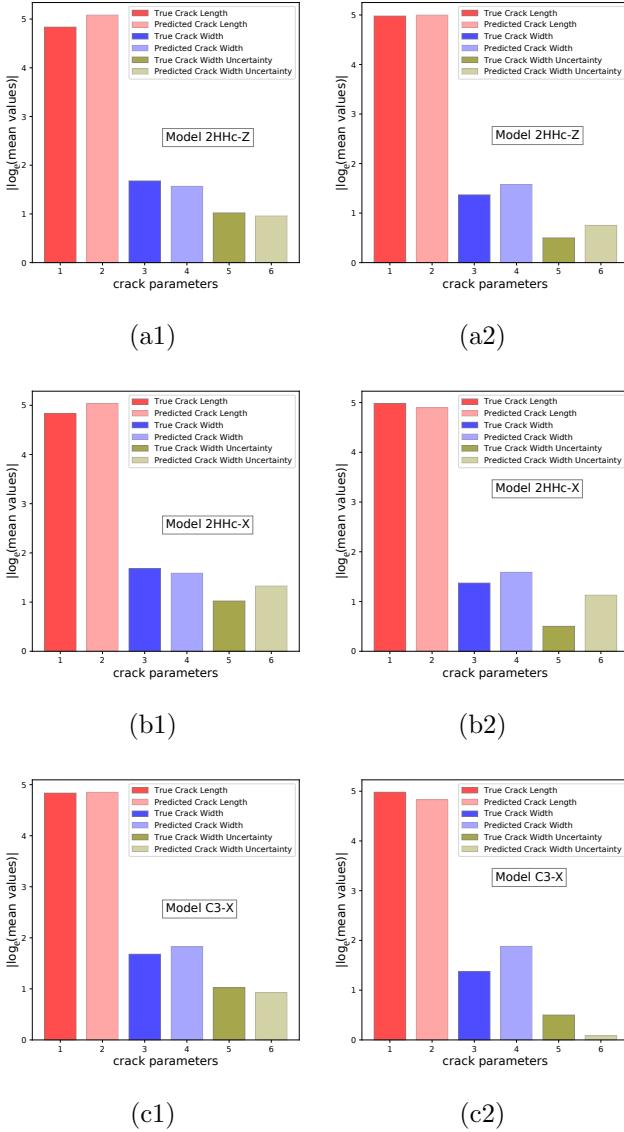


Fig. 12. Mean Predicted vs Mean True values by Top-3 neural network models 2HHc-Z, 2HHc-X & C3-X (a1),(b1) & (c1) on *Held-out Test data*, and (a2), (b2) & (c2) on *NITS Test Data*. For Held-out Test data, Mean predicted lengths are (a1) 161.64 pixels (b1) 153.82 pixels & (c1) 128 pixels against Mean True Length of 125.91 pixels; mean predicted widths are (a1) 4.79 pixels (b1) 4.9 pixels & (c1) 6.21 pixels against Mean True Width of 5.36 pixels; mean predicted width-uncertainties are (a1) 0.38 pixels (b1) 0.26 pixels & (c1) 0.4 pixels against mean true width-uncertainty of 0.36 pixels. For NITS Test data, Mean predicted lengths are (a2) 172.86 pixels (b2) 120.77 pixels & (c2) 115.19 pixels against Mean True Length of 145.42 pixels; mean predicted widths are (a2) 8.38 pixels (b2) 16.14 pixels & (c2) 18.41 pixels against Mean True Width of 3.94 pixels; mean predicted width-uncertainties are (a2) 2.43 pixels (b2) 2.47 pixels & (c2) 6.05 pixels against mean true width-uncertainty of 1.65 pixels.

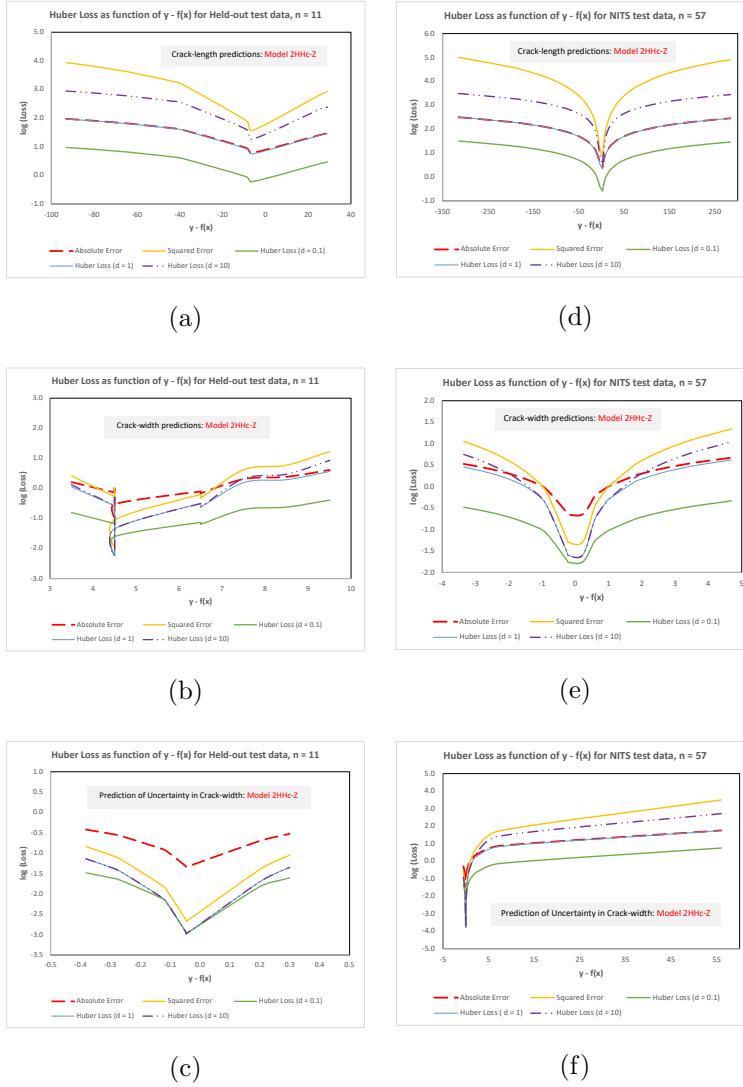


Fig. 13. Variation of Huber Loss as compared to Absolute Error Loss and Squared Error Loss as function of $y - f(x)$ for proposed neural network model (**Model 2HHc-Z**) predictions on *Held-out Test data*: (a) prediction of crack length, (b) prediction of crack width & (c) prediction of uncertainty in crack-width values and *NITS Test data*: (d) prediction of crack length, (e) prediction of crack width & (f) prediction of uncertainty in crack-width values

Table 9: Comparison of Predicted mean values (mm) of crack parameters with those computed by two existing methods based on test samples collected by authors

Sample ID	Measured parameters		Modified Liu and Yeoh [30] (Method-1)				FILFINDER and FWHM based quantification [22] (Method-2)				Proposed CNN-Regression model(Top-3,1) predictions			
	length	width	length	Rel. error	width	Rel. error	length	Rel. error	width	Rel. error	length	Rel. error	width	Rel. error
	DS1	80.99	1.88	80.648	0.004	3.018	0.605	88.658	0.095	2.038	0.084	26.417	0.702	1.477
DS2	63.87	1.00	44.710	0.300	1.717	0.717	45.089	0.294	1.283	0.283	67.946	0.507	1.248	0.027
DS3	57.51	4.86	69.962	0.217	4.547	0.064	83.180	0.446	0.835	0.828	190.769	1.292	1.272	0.523
DS4	23.38	0.50	25.299	0.082	0.507	0.014	26.608	0.138	0.273	0.454	36.787	0.383	0.548	1.007

^a

^aThe relative error for the neural network predictions have been computed with respect to the corresponding values produced by Method-2

500 4. Conclusion

501 In this study, the crack segmentation and quantification method proposed by [30] has been reviewed
502 and a modification proposed regarding the input patch size and selection of initial contour for level-
503 set segmentation. To reduce the manual involvement in Method-1 and make the segmentation
504 and quantification process more automated as well as robust, an alternative method originally
505 developed for astronomical image segmentation has been implemented for this domain and proposed
506 as Method-2. The two methods were compared statistically on a test data set collected by authors
507 from different locations inside the NITS campus which established a significant difference between
508 them. Although, both the methods exhibited appreciable accuracy for random and noisy crack data
509 obtained from diverse surfaces and materials such as concrete members, bituminous and concrete
510 pavements as well as indoor masonry walls with crack-widths ranging from less than 0.5 mm to
511 4.5 mm, Method-2 is hereby recommended due to its more automated process. The authors went
512 one step further to develop a Convolution Neural Network based Regression model to predict the
513 identical crack parameters as those produced by Method-2, namely, crack-length, crack-width and
514 width-uncertainty directly from the RGB patches of cracks. The model was trained on a public
515 data set and tested on the NITS dataset. In spite of a significantly small training data, the scatter
516 plots in **Fig. 11**, the comparative bar charts in **Fig. 12**, and, the Huber loss plots in **Fig. 13**
517 demonstrate optimistic results. The algorithm adopted for the Method-1 and Method-2, and the
518 structure of the CNN-Regression model from input to output have been graphically summarised in
519 **Fig. 14 (a), (b) and (c)** in the Appendix.

520 4.1. Limitations and Future scope

521 Limitations of this work in practical applications: The images of cracks containing multiple prominent
522 branches need to be pre-processed to generate image-stamps containing branch-less cracks. These
523 stamps can only be fed in the model to generate respective crack-lengths and crack-widths within a
524 limited confidence interval. Due to the time-consuming factor of building the training database, the
525 authors look forward to explore the prediction capabilities of a DCNN based Regression model in a
526 separate article in future when larger training database becomes available. For choosing the optimal
527 value for the δ parameter of the Huber Loss function, a range of 0.1 to 10 with a multiplication
528 factor of 10 has been arbitrarily chosen to obtain three experimental values unlike adopting a grid
529 search method as recommended by [35] which can be explored in future works.

530

531 Notation:

532 The following symbols are used in this paper:

533 A - Advection

534 P - Propagation

535 C_u - Curvature

536 G_r - Grayscale tolerance

537 C_c - Convergence criterion

538 DCNN - Deep Convolutional Neural Network

539 FWHM - Full-width at Half-maximum

540 3D - three dimensional

541 GUI - Graphical User Interface

542 RGB - Red-Green-Blue

543 MSH - Married Scholars' Hostel

544 ROI - Region of Interest

545 RHT - Rolling Hough Transform

546 PD - Public dataset

547 DS - Dataset

548 H_0 - Null hypothesis

549 μ - Mean

550 NITS - National Institute of Technology Silchar.

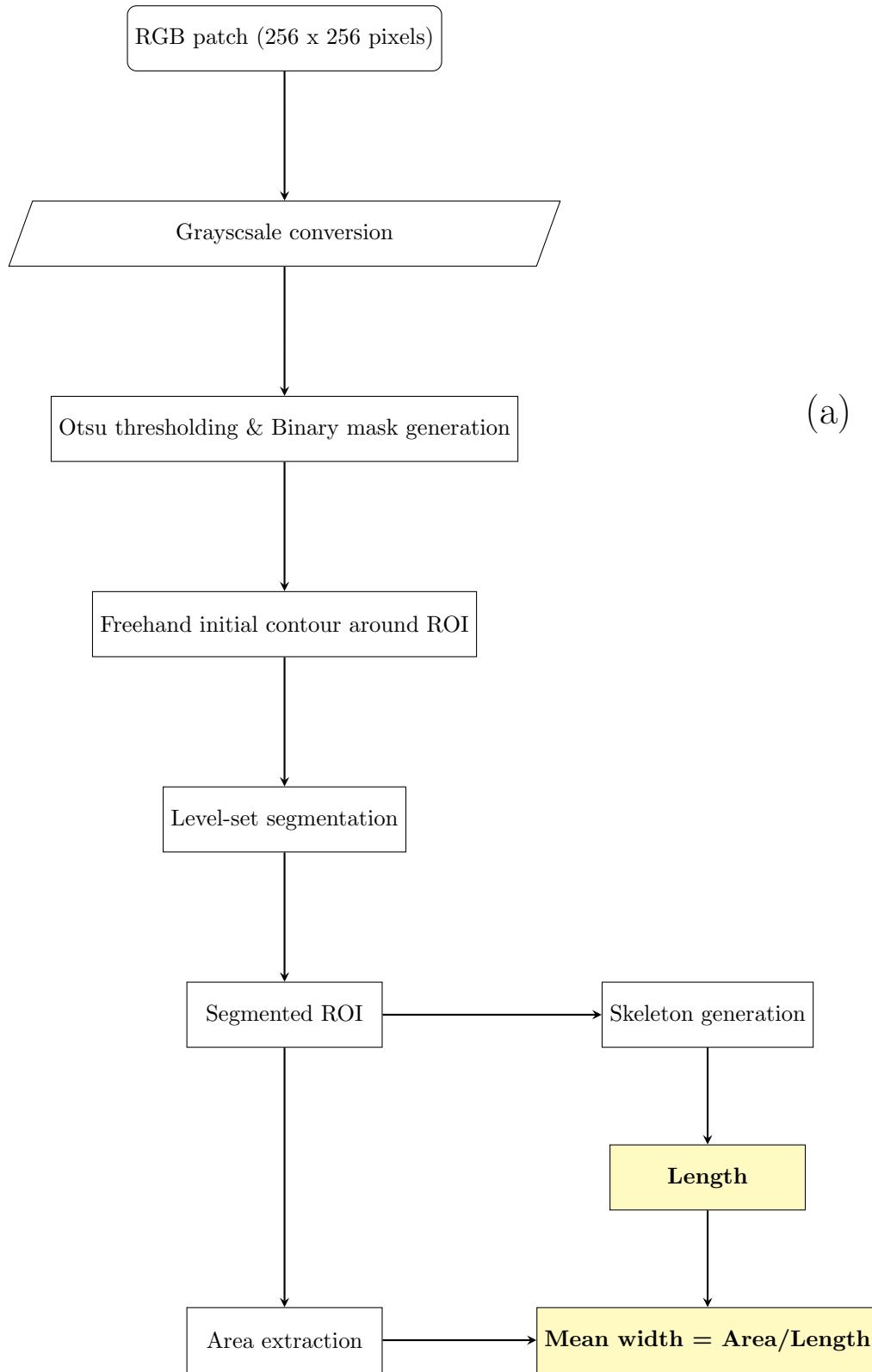
551 RMSprop - Root Mean Squared Propagation

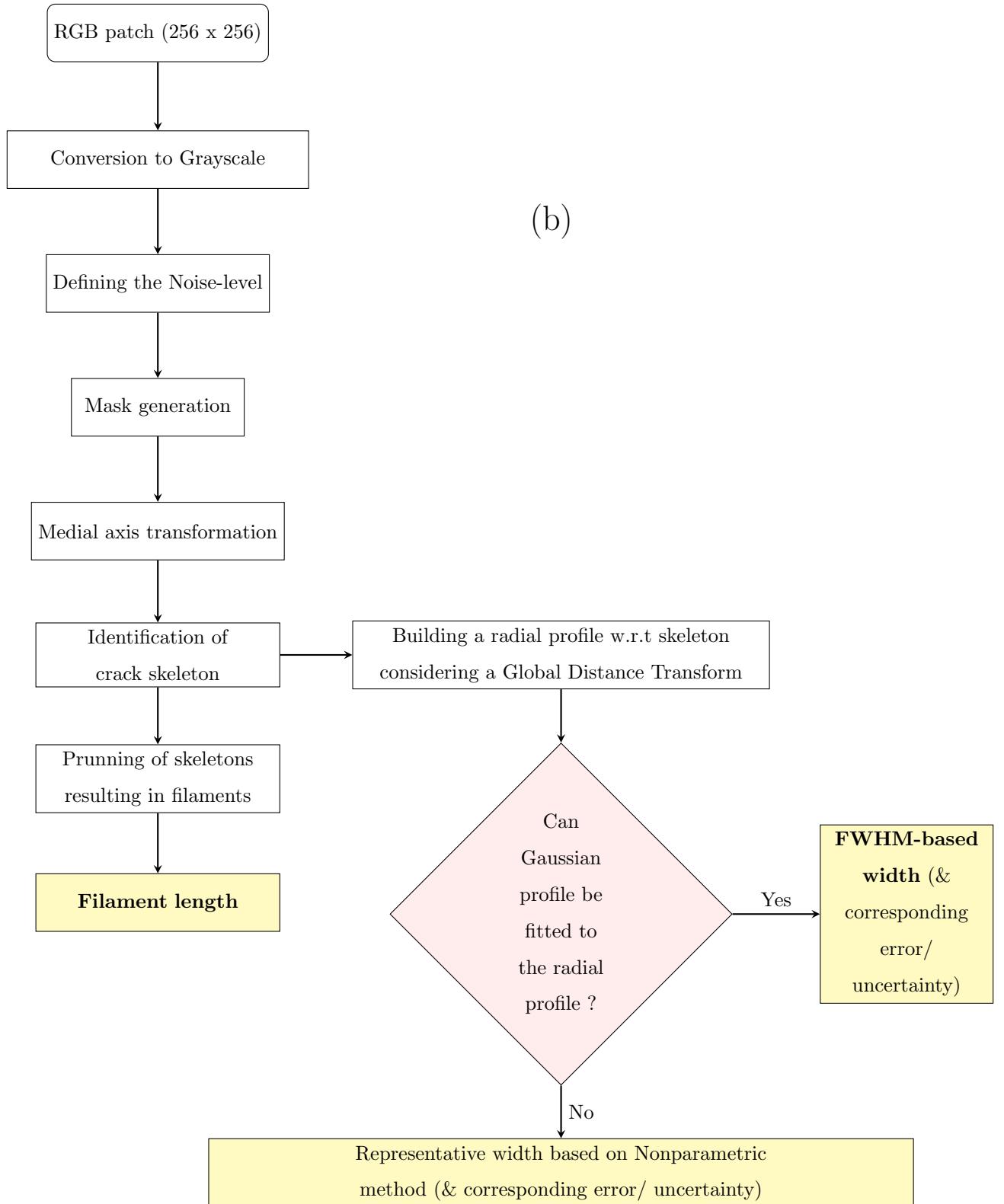
552 ROP - Reduce-on-plateau

553 w.r.t - with respect to

554

555 Appendix





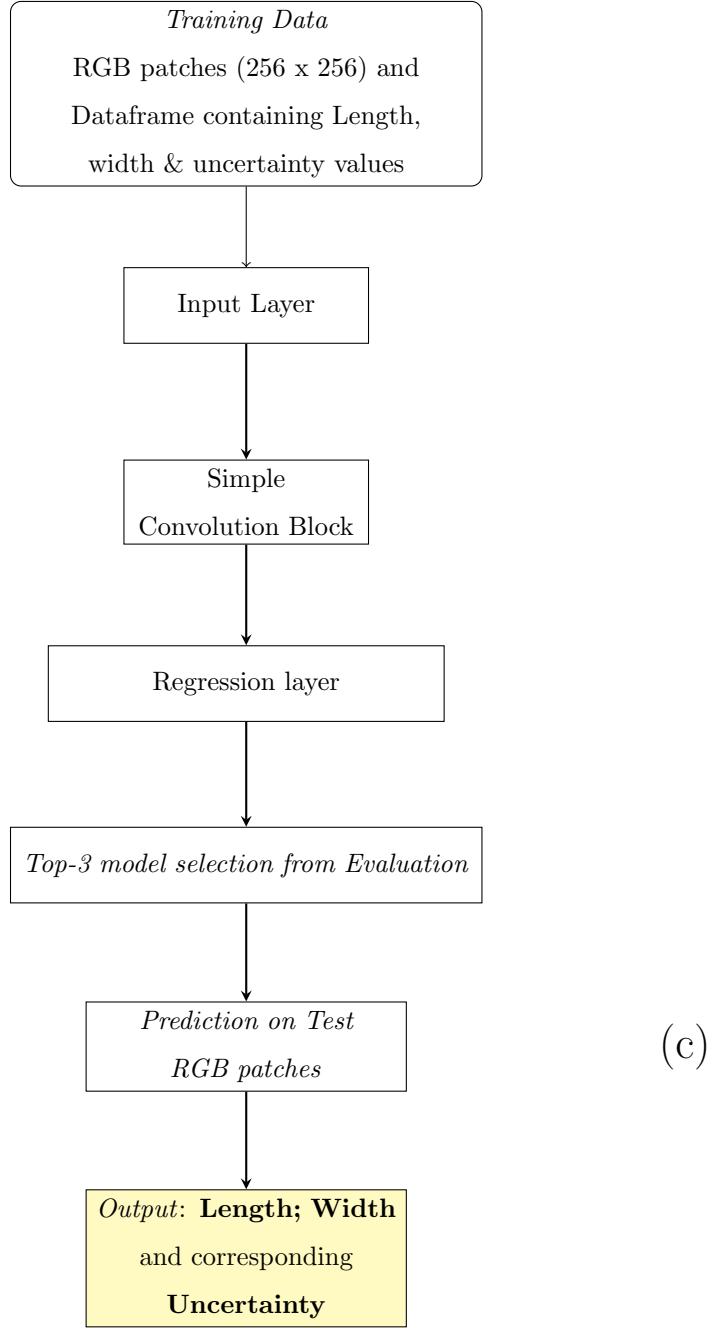


Fig. 14. Frameworks showing (a) crack-length and mean crack-width retrieval by Method-1, (b) extraction of crack-filament lengths, corresponding FWHM-based crack-width and uncertainty associated with crack-width values by Method-2, based on [22], and (c) Neural Network Model development and prediction of target variables from image patches

556

557 **References**

- 558 [1] Adhikari, R., Moselhi, O., Bagchi, A., 2014. Image-based retrieval of concrete crack properties
559 for bridge inspection. *Automation in construction* 39, 180–194.
- 560 [2] Ameri, A., Akhaee, M.A., Scheme, E., Englehart, K., 2019. Regression convolutional neural
561 network for improved simultaneous emg control. *Journal of neural engineering* 16, 036015.
- 562 [3] Bitvai, Z., Cohn, T., 2015. Non-linear text regression with a deep convolutional neural network,
563 in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
564 and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short
565 Papers), pp. 180–185.
- 566 [4] Bouma, H., Bescós, J.O., Vilanova, A., Gerritsen, F.A., 2007. Unbiased vessel-diameter
567 quantification based on the fwhm criterion, in: Medical Imaging 2007: Image Processing, SPIE.
568 pp. 903–911.
- 569 [5] Chollet, F., 2021. Deep learning with Python. Simon and Schuster.
- 570 [6] Clark, S., Peek, J., Putman, M., 2014. Magnetically aligned h i fibers and the rolling hough
571 transform. *The Astrophysical Journal* 789, 82.
- 572 [7] Dais, D., Bal, I.E., Smyrou, E., Sarhosis, V., 2021. Automatic crack classification and
573 segmentation on masonry surfaces using convolutional neural networks and transfer learning.
574 *Automation in Construction* 125, 103606.
- 575 [8] Debroy, S., Sil, A., 2022. An apposite transfer-learned dcnn model for prediction of structural
576 surface cracks under optimal threshold for class-imbalanced data. *Journal of Building Pathology
577 and Rehabilitation* 7, 1–18.
- 578 [9] Dekkers, O.M., 2019. Why not to (over) emphasize statistical significance.
- 579 [10] Deng, L., Sun, T., Yang, L., Cao, R., 2023. Binocular video-based 3d reconstruction and length
580 quantification of cracks in concrete structures. *Automation in Construction* 148, 104743.
- 581 [11] Dong, J., Wang, N., Fang, H., Wu, R., Zheng, C., Ma, D., Hu, H., 2022. Automatic damage
582 segmentation in pavement videos by fusing similar feature extraction siamese network (sfe-
583 snet) and pavement damage segmentation capsule network (pds-capsnet). *Automation in
584 Construction* 143, 104537.

- 585 [12] Dorafshan, S., Thomas, R.J., Maguire, M., 2018. Comparison of deep convolutional neural
586 networks and edge detectors for image-based crack detection in concrete. Construction and
587 Building Materials 186, 1031–1045.
- 588 [13] Dung, C.V., et al., 2019. Autonomous concrete crack detection using deep fully convolutional
589 neural network. Automation in Construction 99, 52–58.
- 590 [14] Fang, X., Li, Q., Zhu, J., Chen, Z., Zhang, D., Wu, K., Ding, K., Li, Q., 2022. Sewer defect
591 instance segmentation, localization, and 3d reconstruction for sewer floating capsule robots.
592 Automation in Construction 142, 104494.
- 593 [15] Foundation, W., 2023. Full width at half maximum. URL: https://en.wikipedia.org/wiki/Full_width_at_half_maximum.
- 595 [16] Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer
596 Series in Statistics, Springer New York Inc., New York, NY, USA.
- 597 [17] Huber, P.J., 1992. Robust estimation of a location parameter. Breakthroughs in statistics:
598 Methodology and distribution , 492–518.
- 599 [18] Iyer, S., Sinha, S.K., 2005. A robust approach for automatic detection and segmentation of
600 cracks in underground pipeline images. Image and Vision Computing 23, 921–933.
- 601 [19] Kang, D., Benipal, S.S., Gopal, D.L., Cha, Y.J., 2020. Hybrid pixel-level concrete crack
602 segmentation and quantification across complex backgrounds using deep learning. Automation
603 in Construction 118, 103291.
- 604 [20] Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International
605 journal of computer vision 1, 321–331.
- 606 [21] Kim, J., Shim, S., Cha, Y., Cho, G.C., 2021. Lightweight pixel-wise segmentation for efficient
607 concrete crack detection using hierarchical convolutional neural network. Smart Materials and
608 Structures 30, 045023.
- 609 [22] Koch, E.W., Rosolowsky, E.W., 2015. Filament identification through mathematical
610 morphology. Monthly Notices of the Royal Astronomical Society 452, 3435–3450.
- 611 [23] Lee, B.Y., Kim, J.K., Kim, Y.Y., Yi, S.T., 2007. A technique based on image processing for
612 measuring cracks in the surface of concrete structures .
- 613 [24] Lee, T.C., Kashyap, R.L., Chu, C.N., 1994. Building skeleton models via 3-d medial surface
614 axis thinning algorithms. CVGIP: graphical models and image processing 56, 462–478.

- 615 [25] Li, J., Wang, Q., Ma, J., Guo, J., 2022a. Multi-defect segmentation from façade images using
616 balanced copy–paste method. Computer-Aided Civil and Infrastructure Engineering 37, 1434–
617 1449.
- 618 [26] Li, S., Zhao, X., 2023. High-resolution concrete damage image synthesis using conditional
619 generative adversarial network. Automation in Construction 147, 104739.
- 620 [27] Li, Y., Bao, T., Huang, X., Chen, H., Xu, B., Shu, X., Zhou, Y., Cao, Q., Tu, J., Wang,
621 R., et al., 2022b. Underwater crack pixel-wise identification and quantification for dams via
622 lightweight semantic segmentation and transfer learning. Automation in Construction 144,
623 104600.
- 624 [28] Lins, R.G., Givigi, S.N., 2016. Automatic crack detection and measurement based on image
625 analysis. IEEE Transactions on Instrumentation and Measurement 65, 583–590.
- 626 [29] Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V.C.S., Ding, L., 2020. Automated pavement
627 crack detection and segmentation based on two-step convolutional neural network. Computer-
628 Aided Civil and Infrastructure Engineering 35, 1291–1305.
- 629 [30] Liu, Y., Yeoh, J.K., 2021. Robust pixel-wise concrete crack segmentation and properties retrieval
630 using image patches. Automation in Construction 123, 103535.
- 631 [31] Loverdos, D., Sarhosis, V., 2022. Automatic image-based brick segmentation and crack detection
632 of masonry walls using machine learning. Automation in Construction 140, 104389.
- 633 [32] Ma, D., Fang, H., Wang, N., Zheng, H., Dong, J., Hu, H., 2022. Automatic defogging,
634 deblurring, and real-time segmentation system for sewer pipeline defects. Automation in
635 Construction 144, 104595.
- 636 [33] Maharana, K., Mondal, S., Nemade, B., 2022. A review: Data pre-processing and data
637 augmentation techniques. Global Transitions Proceedings 3, 91–99.
- 638 [34] Malek, S., Melgani, F., Bazi, Y., 2018. One-dimensional convolutional neural networks for
639 spectroscopic signal regression. Journal of Chemometrics 32, e2977.
- 640 [35] Meyer, G.P., 2021. An alternative probabilistic interpretation of the huber loss, in: Proceedings
641 of the ieee/cvf conference on computer vision and pattern recognition, pp. 5261–5269.
- 642 [36] Ni, T., Zhou, R., Gu, C., Yang, Y., 2020. Measurement of concrete crack feature with android
643 smartphone app based on digital image processing techniques. Measurement 150, 107093.
- 644 [37] Nibali, A., He, Z., Morgan, S., Prendergast, L., 2018. Numerical coordinate regression with
645 convolutional neural networks. arXiv preprint arXiv:1801.07372 .

- 646 [38] Nnolim, U.A., 2020. Automated crack segmentation via saturation channel thresholding, area
647 classification and fusion of modified level set segmentation with canny edge detection. *Heliyon*
648 6, e05748.
- 649 [39] Nykamp, D.Q., . Math insight. URL: https://mathinsight.org/applet/level_curves_eliptic_paraboloid_graph.
- 651 [40] Özgenel, Ç.F., 2019. Concrete crack segmentation dataset. Mendeley Data 1.
- 652 [41] Panella, F., Lipani, A., Boehm, J., 2022. Semantic segmentation of cracks: Data challenges and
653 architecture. *Automation in Construction* 135, 104110.
- 654 [42] Qiu, Q., Lau, D., 2023. Real-time detection of cracks in tiled sidewalks using yolo-based method
655 applied to unmanned aerial vehicle (uav) images. *Automation in Construction* 147, 104745.
- 656 [43] Sarmiento, J.A., 2021. Pavement distress detection and segmentation using yolov4 and
657 deeplabv3 on pavements in the philippines. arXiv preprint arXiv:2103.06467 .
- 658 [44] Sateesh Babu, G., Zhao, P., Li, X.L., 2016. Deep convolutional neural network based regression
659 approach for estimation of remaining useful life, in: International conference on database
660 systems for advanced applications, Springer. pp. 214–228.
- 661 [45] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch,
662 S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.Y., White, D.J., Hartenstein, V., Eliceiri, K.,
663 Tomancak, P., Cardona, A., 2012. Fiji: an open-source platform for biological-image analysis.
664 *Nature Methods* 9, 676–682. URL: <https://doi.org/10.1038/nmeth.2019>, doi:10.1038/nm
665 eth.2019.
- 666 [46] Sinha, S.K., Fieguth, P.W., 2006. Segmentation of buried concrete pipe images. *Automation*
667 in Construction 15, 47–57.
- 668 [47] Siriborvornratanakul, T., 2022. Downstream semantic segmentation model for low-level surface
669 crack detection. *Advances in Multimedia* 2022.
- 670 [48] Su, T.C., Yang, M.D., 2014. Application of morphological segmentation to leaking defect
671 detection in sewer pipelines. *Sensors* 14, 8686–8704.
- 672 [49] svi.nl, S.V.I., . Halfintensitywidth. URL: <https://svi.nl/HalfIntensityWidth>.
- 673 [50] Tanveer, M., Kim, B., Hong, J., Sim, S.H., Cho, S., 2022. Comparative study of lightweight
674 deep semantic segmentation models for concrete damage detection. *Applied Sciences* 12, 12786.
- 675 [51] Vinutha, H., Poornima, B., Sagar, B., 2018. Detection of outliers using interquartile range
676 technique from intrusion dataset, in: *Information and Decision Sciences: Proceedings of the*
677 *6th International Conference on FICTA*, Springer. pp. 511–518.

- 678 [52] Wang, S., Wu, X., Zhang, Y., Liu, X., Zhao, L., 2020. A neural network ensemble method
679 for effective crack segmentation using fully convolutional networks and multi-scale structured
680 forests. Machine Vision and Applications 31, 1–18.
- 681 [53] Wang, W., Su, C., 2021. Semi-supervised semantic segmentation network for surface crack
682 detection. Automation in Construction 128, 103786.
- 683 [54] Wang, W., Su, C., 2022. Automatic concrete crack segmentation model based on transformer.
684 Automation in Construction 139, 104275.
- 685 [55] Xiang, C., Wang, W., Deng, L., Shi, P., Kong, X., 2022. Crack detection algorithm for concrete
686 structures based on super-resolution reconstruction and segmentation network. Automation in
687 Construction 140, 104346.
- 688 [56] Xie, Y., Xing, F., Kong, X., Su, H., Yang, L., 2015. Beyond classification: structured regression
689 for robust cell detection using convolutional neural network, in: International conference on
690 medical image computing and computer-assisted intervention, Springer. pp. 358–365.
- 691 [57] Yu, Z., Shen, Y., Sun, Z., Chen, J., Gang, W., 2022. Cracklab: A high-precision and
692 efficient concrete crack segmentation and quantification network. Developments in the Built
693 Environment 12, 100088.
- 694 [58] Zhang, E., Shao, L., Wang, Y., 2023. Unifying transformer and convolution for dam crack
695 detection. Automation in Construction 147, 104712.
- 696 [59] Zhang, J., Qian, S., Tan, C., 2022. Automated bridge surface crack detection and segmentation
697 using computer vision-based deep learning model. Engineering Applications of Artificial
698 Intelligence 115, 105225.
- 699 [60] Zhou, S., Canchila, C., Song, W., 2023. Deep learning-based crack segmentation for civil
700 infrastructure: data types, architectures, and benchmarked performance. Automation in
701 Construction 146, 104678.
- 702 [61] Zhou, S., Song, W., 2020. Concrete roadway crack segmentation using encoder-decoder networks
703 with range images. Automation in Construction 120, 103403.
- 704 [62] Zhou, S., Song, W., 2021. Crack segmentation through deep convolutional neural networks and
705 heterogeneous image fusion. Automation in Construction 125, 103605.