







UNIVERSITÀ POLITECNICA DELLE MARCHE  
DEPARTMENT OF ENGINEERING  
PHD IN MECHANICAL ENGINEERING

---

# **Crack Analyser: a novel image-based NDT approach for measuring crack severity**

Candidate:  
**Ing. Nicola Giulietti**

Advisor:  
**Prof. Gian Marco Revel**

Coadvisor:  
**Prof. Paolo Chiariotti**

Academic Year 2021-2022





UNIVERSITÀ POLITECNICA DELLE MARCHE  
DEPARTMENT OF ENGINEERING  
PHD IN MECHANICAL ENGINEERING

---

# **Crack Analyser: a novel image-based NDT approach for measuring crack severity**

Candidate:  
**Ing. Nicola Giulietti**

Advisor:  
**Prof. Gian Marco Revel**

Coadvisor:  
**Prof. Paolo Chiariotti**

Academic Year 2021-2022

---

UNIVERSITÀ POLITECNICA DELLE MARCHE  
PHD IN MECHANICAL ENGINEERING  
DEPARTMENT OF ENGINEERING  
Via Brezze Bianche – 60131 Ancona (AN), Italy

*To my family,  
my fellow students and my friends*





# Abstract

Europe's ageing transport infrastructure needs effective and proactive maintenance in order to continue its safe operation during the entire life cycle; European countries have to allocate huge resources for maintaining their serviceability level [1]. This give rise to the necessity of an urgent need to adopt faster and more reliable monitoring inspection approaches to help tackling these issues. The deterioration of structures is most often foreseen by the formation of cracks on concrete surface [2]. The presence of cracks can be a symptom of various problems like expansion and shrinks due to temperature differences, settlement of the structure, improper cover provided during concreting, corrosion of reinforcement steel, heavy load applied, insufficient vibration at the time of laying the concrete or loss of water from concrete surface shrinkage [3], therefore the identification, measurement and monitoring of cracks on the concrete surface becomes of primary importance. The main currently adopted inspection methods rely on visual marking and rulers [4], long and cumbersome activity, prone to errors and poorly objective on quantitative analysis because it strongly depends on operator experience. According to UNI EN 1992-1-1:2005 standard , the maximum admitted concrete crack width is  $0.3mm$  [5]. For this reason, to accurately and reliably measure the target dimension, it is necessary to employ measurement instruments with suitable metrological characteristics (e.g. precision and accuracy at least one order lower than the value to be measured). Otherwise, the crack severity could be misclassified.

This thesis work proposes a novel automatic image-based approach able to locate and measure cracks on concrete surfaces respecting the metrological constraint imposed by UNI EN 1992-1-1:2005 standard. Using only one image, the developed method is able to automatically and rapidly locate and measure the average width and length of a crack in an existing concrete structure. The measurement system developed exploits a single camera working in the visible range to acquire a digitized image of the structure being inspected. The software component of the system receives as input the single image framing the crack and gives as output an augmented image where the crack is highlighted as well as its average/max width and length. The measure of the crack width is performed perpendicularly to the crack central line with sub-pixel accuracy. The measurement system has been deployed on a smartphone for operator-based manual inspections as well on embedded systems for remote inspection with

robots or Unmanned Aerial Vehicles (Unmanned Aerial Vehicles (UAVs)). The strategies developed can be easily extended from concrete inspection applications to any other context where a surface quality control targeted to the identification of eventual damages/defects is required. The activity was triggered by an explicit need within the EnDurCrete project.

# Contents

<b>List of Symbols</b>	<b>1</b>
<b>Acronyms</b>	<b>3</b>
<b>1 General introduction and thesis outline</b>	<b>5</b>
<b>2 State of the Art: image-based Non-Destructive Testing (NDT) for crack severity analysis</b>	<b>9</b>
<b>3 Methods</b>	<b>17</b>
3.1 An optimized ridge detector algorithm for crack identification .	17
3.1.1 Crack identification via Ridge Detectors: theory . . . . .	17
3.1.2 Optimized ridge detector algorithm . . . . .	37
3.1.3 Open issues . . . . .	43
3.2 AI-based preprocessing for cracks isolation . . . . .	43
3.2.1 Convolutional Neural Network for image classification .	45
3.2.2 CNN sliding windows for image segmentation . . . . .	47
3.2.3 CNN to Fully Convolutional Neural Networks for image segmentation . . . . .	48
3.2.4 UNET for image segmentation. . . . .	52
3.3 A novel image-based NDT approach for measuring crack severity	62
3.3.1 Image acquisition and calibration . . . . .	64
3.3.2 Neural Network based semantic segmentation. . . . .	68
3.3.3 Automatic threshold parameters selection . . . . .	70
3.3.4 Filling Level Algorithm . . . . .	72
<b>4 Metrological characterization</b>	<b>75</b>
4.1 Working range definition . . . . .	75
4.1.1 Target rotated around x axis . . . . .	77
4.1.2 Target rotated around y axis . . . . .	77
4.1.3 Target rotated around x and y axis . . . . .	79
4.1.4 Manual acquisition by non-expert operator . . . . .	85
4.2 Measurement System Analysis: theory . . . . .	85
4.3 Metrological characterization of the measurement system with fiducial markers . . . . .	93

*Contents*

4.4	Metrological characterization of the measurement system with depth sensor . . . . .	96
4.5	Performance Comparison on reference specimen . . . . .	103
<b>5</b>	<b>Deployment on portable devices for in-field testing</b>	<b>111</b>
5.1	Measurement System design . . . . .	111
5.2	Software design . . . . .	112
5.3	Crack Analyser App . . . . .	115
<b>6</b>	<b>In-field testing results</b>	<b>121</b>
<b>7</b>	<b>Conclusions and future work</b>	<b>129</b>
7.1	Conclusions . . . . .	129
7.2	Future work . . . . .	131
	<b>Bibliography</b>	<b>133</b>
	<b>Appendix</b>	<b>145</b>
1	Gage RR study . . . . .	145

# List of Figures

2.1	Laser triangulation system's working principle [8] . . . . .	9
3.1	First derivative behaviour of edge filter is clearly shown through the 3D plot before and after filtration of a wall crack image . .	18
3.2	Original image (a), edge filter (b) and ridge filter (c) . . . . .	18
3.3	Parabolic line profile $f_p$ (a), bar-shaped line profile $f_b$ (b), and asymmetrical bar-shaped line profile $f_a$ (c), with $h = 1$ , $w = 1$ , and $a = 0.2$ . . . . .	19
3.4	1D Crack line profile from real image . . . . .	20
3.5	1D plot of the intensity of a row or column of an image along the perpendicular to a line. The plot shows how deriving the intensity function the noise increases. . . . .	21
3.6	1D plot of the intensity of a row or column of an image along the perpendicular to a line. The plot shows how deriving the convolution between the intensity function and the Gaussian profile, it is possible to locate salient points (local maxima) that allow the identification of the line. . . . .	22
3.7	Different levels in the scale-space representation of a crack image	22
3.8	Gaussian kernel (a), bar-shaped line profile (b) and the result of their convolution (c) with $\sigma = 0.2$ , $w = 1$ , $a = 1$ . . . . .	23
3.9	Scale-space behaviour of the bar-shaped line profile when convolved with the derivatives of Gaussian kernel for $x \in [-3, 3]$ , $\sigma \in [0.2, 2]$ , $a = 1$ and $w = 1$ . . . . .	24
3.10	Scale-space behaviour of the bar-shaped line profile when convolved with the derivatives of Gaussian kernel for $x \in 0$ , $\sigma \in [0.2, 2]$ , $a = 0$ and $w = 1$ . . . . .	25
3.11	Shaped assumed by the decomposition of the Hessian matrix varying $\lambda$ values. . . . .	27
3.12	Neighbours selection based on the angle perpendicular to the normal $n$ of the current line point. . . . .	29
3.13	Bresenham algorithm used to look for pixels perpendicular to the line until a distance of $2.5\sigma$ . . . . .	30
3.14	Artificial crack used for ridge detector algorithm testing on lab-scale scenario . . . . .	32

List of Figures

3.15	Workflow used for ridge detector algorithm testing on lab-scale scenario . . . . .	33
3.16	Steger algorithm used to measured artificial crack width varying $\sigma$ input parameters: comparison among Steger algorithm (green dots), G3F measurement system (red dots), and slider position (blue dots) for different $\sigma$ values ( $\sigma = 3$ (a), $\sigma = 7$ (b), $\sigma = 15$ (c), $\sigma = 23$ (d)). . . . .	34
3.17	Synthetic line with known width and asymmetry (left); synthetic line on the left blurred with blur filter to get a more realistic representation of the image (right). . . . .	35
3.18	Flow chart used for $\hat{w}(\sigma_i)$ behaviour examination in order to find a proper $\sigma$ value target line . . . . .	35
3.19	Flow chart used for $\hat{w}(\sigma_i)$ behaviour examination in order to find a proper $\sigma$ value target line . . . . .	36
3.20	The red circle identifies the $\hat{w}(\sigma_i)$ that best fit target line thickness (remember that line thickness is $2w$ ). . . . .	37
3.21	The red circle identifies the $\hat{w}(\sigma_i)$ that best fit target line thickness. Increasing line thickness, the intersection between $\sigma$ restriction (red line) and measured average width $\hat{w}(\sigma_i)$ trend (green dashed line) is no more representative of $\sigma^*$ location. . . . .	37
3.22	Five measurement with lines at different thickness are superimposed in order to observe how the knee position change according to the line width. . . . .	39
3.23	Linear interpolation of knee position for 45 lines images with thickness from 2 to 92 pixel. The obtained line localizes the $\sigma_{opt}$ at each line thickness. . . . .	39
3.24	Steps used in order to find $\sigma_{opt}$ using a few numbers of different $\sigma$ (5 in this case) based on the optimization line previously obtained. 40	
3.25	Circle line (a), v line (b) and diagonal line (c) used to validate the width estimation optimization methodology . . . . .	41
3.26	Performance of proposed algorithm compared to G3F triangulation system . . . . .	42
3.27	Ridge detector well detect cracks as curvilinear structures. . . . .	44
3.28	The ridge detector algorithm cannot distinguish between wall cracks, scratches and other disturbing elements that can be included in a real image. . . . .	44
3.29	VGG19 original architecture [92] . . . . .	46
3.30	Image segmentation through CNN sliding window . . . . .	48
3.31	Image segmentation results through CNN sliding window technique 48	
3.32	CNN architecture is converted to FCN replacing Fully Connected layers into Convolutional layers . . . . .	49

3.33	CNN architecture is converted to FCN replacing Fully Connected layers into Convolutional layers . . . . .	50
3.34	Crack ROI detection by a binary threshold on FCN output pixelwise probability value. The original image (a) and image deprived of non-cracked region (b) are reported in figure. . . .	51
3.35	Comparison between Steger algorithm (a) and FCN + Steger algorithm (b). Wall scratches and disturbing elements are correctly excluded from ridge detection by FCN VGG19 method. . . . .	53
3.36	Comparison between Steger algorithm (a) and FCN + Steger algorithm (b). Disturbing elements are correctly excluded from ridge detection by FCN VGG19 method. . . . .	54
3.37	UNET typical architecture. K classes segmentation of a 256x256 RGB input image. [97] . . . . .	54
3.38	On the left the RGB crack image, on the right the binary crack mask manually selected by an operator. . . . .	55
3.39	Mean histogram of the 5000 binary crack masks. . . . .	56
3.40	The model is trained with different train-validation-test ratio. The results confirms that the ratio that maximize IoU and F score is 0.80-0.20-0.20 . . . . .	60
3.41	The plot shows the IoU score curve over epochs for train and test dataset. Above 50, the IoU score stabilises. . . . .	61
3.42	The plot shows the f score curve over epochs for train and test dataset. Above 50, the f score stabilises. . . . .	61
3.43	The plot shows loss curve over epochs for train and test dataset. Above 50, the loss value stabilises. . . . .	62
3.44	Result of U-NET trained model inference on the various image from collected crack test dataset are shown in figure. On the left the input image, on the right the output binary mask. . . . .	63
3.45	4-points perspective correction using fiducial target of known size and shape. . . . .	66
3.46	To measure the severity of a crack it is necessary to switch between image pixels coordinates to real-world coordinates in mm. . . . .	66
3.47	The picture shows how is possible to obtain pixel-to-mm constant knowing the focal length and the distance from the target. . . . .	67
3.48	Crack ROI selection based on target 0 and target 2 positions. . . . .	68
3.49	On the left the acquired RGB input image, on the right the binary model prediction . . . . .	69
3.50	automated best-enveloping ROI selection based on model prediction . . . . .	69
3.51	Contrast $h$ evaluation workflow . . . . .	71

List of Figures

3.52	Line contrast $h$ is obtained as the absolute difference between the two local maxima of the cubic spline approximation histogram function . . . . .	71
3.53	Recognized portion of the crack varies changing $\sigma$ value . . . . .	72
3.54	Filling Level algorithm flowchart for $\sigma_{FL}$ identification . . . . .	74
3.55	FL over $\sigma$ plot. The data are collected within the Filling Level algorithm. . . . .	74
4.1	An anthropomorphic robot is used in order to define the working range of the developed device in a controlled environment. . . . .	76
4.2	The image acquisition system (1) it is placed at a known distance from the target (4). The parallelism between the sensor surface and the target surface is guaranteed through a system made of a laser (3) and a mirror (4). The two surfaces are to be considered parallel when the laser beam reflected by the mirror returns back to the source. . . . .	76
4.3	The figure shows the plot of the absolute error in millimetres, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every images are framed. . . . .	78
4.4	The figure shows the plot of the absolute error in pixel, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every image are framed. . . . .	78
4.5	The figure shows the plot of the absolute error in millimetres, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every images are framed. . . . .	79
4.6	The figure shows the plot of the absolute error in pixel, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every image are framed. . . . .	80
4.7	A controlled rotation is imposed to the target surface with a random angle between $-30^\circ$ to $+30^\circ$ (continuous uniform distribution). For each distances 360 images are framed. . . . .	81
4.8	Best-fit quadratic curve of measured points $p(R_x, R_y, \hat{w})$ for target distance of 70, 100, 130 and 240 cm. In all the cases examined, a fit error of 0.01 mm is obtained . . . . .	82
4.9	Rotations around the y $R_y$ axis tend to underestimate, while rotations around the x $R_x$ axis tend to overestimate the width measurement for perspective effects. . . . .	82



4.10	In red are highlighted all the acquisitions that fall within the angular range $-25^\circ < R_x < +25^\circ$ and $-25^\circ < R_y < +25^\circ$ . . .	83
4.11	Trend of the standard deviation in <i>mm</i> as the maximum rotation angle imposed on the target varies. . . . .	83
4.12	Trend of the standard deviation in <i>pixel</i> as the maximum rotation angle imposed on the target varies. . . . .	84
4.13	Dispersion of the values at each distance, normalized to the equivalent pixel width of the crack. . . . .	84
4.14	By curve-fitting the standard deviation data over imposed angular range, obtained in the previous chapter, a second degree function of equation $y = 0.00001x + 0.00003x^2 + 0.00221$ is obtained. A standard deviation of $0.013mm$ corresponds to a maximum angular range of $19^\circ$ . . . . .	85
4.15	Concrete wall crack used as target for reproducibility and repeatability tests . . . . .	86
4.16	Cross Gage R&R method for non-destructive testing for crack assessment. . . . .	87
4.17	Wenglor MLSL132 Optical data . . . . .	90
4.18	Measurement system for cracks assessment with WENGLOR laser profilometer . . . . .	91
4.19	LabVIEW VI flow chart for crack gap measurement . . . . .	92
4.20	On the left the detected laser line, on the right its gradient. . .	92
4.21	Distribution normalised with respect to the mean value of mean width values measured on 100 images by the same operator . .	93
4.22	Gage R&R (ANOVA) Report for mean width crack calculation. Metrological characterization of the measurement system with fiducial markers . . . . .	95
4.23	Lab-scale WENGLOR laser profilometer measurement setup. .	96
4.24	In order to make the measurement systems comparable, the starting point of the laser is manually positioned to the centre of maker ID 6 then move to centre of marker ID 8 with a step of $0.1\text{ mm}$ . . . . .	97
4.25	Crack gap measured between centre of marker ID 6 and ID 8 with a step of $0.1mm$ . . . . .	97
4.26	Intel Realsense RGBD D435i depth camera . . . . .	98
4.27	Relative position between sensor and target planes. . . . .	99
4.28	Depth frame can be used for estimating yaw and pitch angle between target and sensor planes. . . . .	100
4.29	Realsense acquisition loop for In order to assess the effect of yaw and pitch angles in terms of repeatability and reproducibility. .	101
4.30	Realsense live view for real-time feedback. . . . .	102

List of Figures

4.31 Repeatability and Reproducibility Standard Deviation increasing admitted yaw or pitch angles. . . . . 102

4.32 Total Gage R&R % Contribution of Variance increasing admitted yaw or pitch angles. . . . . 103

4.33 Total Gage R&R % Contribution of Study Variance increasing admitted yaw or pitch angles. . . . . 104

4.34 Number of Distinct categories increasing admitted yaw or pitch angles. . . . . 104

4.35 3D printed reference specimen. . . . . 105

4.36 Cross-shaped structure of OASIS. The piece to inspect is placed on the glass, between the two pairs of telecentric profilometers. [119] . . . . . 106

4.37 FOV comparison of a conventional and telecentric lens ([www.edmundoptics.com](http://www.edmundoptics.com))106

4.38 On the left, the reference specimen on OSASIS device. On the right, the resulting silhouette . . . . . 107

4.39 Wenglor MSL132 laser profilometer is used to perform measurement of the average groove width in the reference specimen. 108

4.40  $\hat{w}$  calculated using the Crack Analyser algorithm according to the distance of the camera from the target. In the graph there are superimposed the results of the measurements carried out with the laser and telecentric profilometers in terms of average value and uncertainty bounds. . . . . 109

4.41 Acquisition setup described in 4.3. On the left, a picture framed at a distance of 600mm. On the right a picture framed at a distance of 1650mm. . . . . 110

5.1 The hardware side of the measurement system consists of a system capable of recording and processing images able or not to detect distance between image sensor and target surface . . 112

5.2 BPMN model modules explanation. . . . . 113

5.3 BPMN model for software design . . . . . 114

5.4 Architecture design for Crack Analyser app. The front-end run on Flutter application, the back-end part is entirely managed by a python-based server. . . . . 115

5.5 Front-end application run on Android device. . . . . 117

5.6 Front-end application run on Android device. . . . . 118

5.7 Front-end application run on Android device. . . . . 119

6.1 Santa Barbara Foundation mining tunnels . . . . . 122

6.2 Corrosion and structural health monitoring of tunnel precast concrete segments for mining tunnel demonstrator in EnDurCrete Project . . . . . 122

6.3	Crack ruler used as reference measuring instrument. . . . .	123
6.4	The cracks present on the panels are barely visible, therefore it is necessary to use the high spatial resolution system. . . . .	124
6.5	Comparison between the proposed method, on the left, and the traditional method with a ruler, on the right. In both cases the device returns the measurement of 0.1mm. . . . .	124
6.6	Due to the high roughness it is impossible to apply targets on the surface, therefore it is necessary to use the device equipped with a depth sensor. . . . .	125
6.7	Complete calculation flow for the measurement of the average width of the crack in the surface in the tunnel. . . . .	126
6.8	The bridge chosen for the inspections in conditions of natural light is located on the "Centro Cadore" Lake . . . . .	127
6.9	Complete calculation flow for the measurement of the average width of the crack located in the lateral support structure of the bridge. . . . .	128



# List of Tables

- 3.1 Comparison among  $\sigma$  values obtained by knee location and Steger domain intersection . . . . . 38
- 3.2 Computational time for a  $\sigma$  span of 0.5 between 1 and 30 . . . 40
- 3.3 The table shows the variation of Knee and Optimization error due to the increase of  $\sigma_{delta}$  . . . . . 41
- 3.4 Optimization error for v-shape, diagonal-shape and circle-shape lines . . . . . 41
- 3.5 Optimization error for real crack case . . . . . 42
- 3.6 Comparison between Steger line detector and FCN VGG19 + Steger line detector in terms of Accuracy and Precision . . . . . 52
- 3.7 Models characteristics in terms of number of trainable weights and prediction computational time in seconds calculated as the average time for 100 predictions with different input images. . . 59
- 3.8 Models characteristics in terms of number of trainable weights and prediction computational time in seconds calculated as the average time for 100 predictions with different input images. . . 59
  
- 4.1 Gage R&R study assessed by asking 5 different operators to perform ten measurements each in 3 different target cracks. Metrological characterization of the measurement system with fiducial markers . . . . . 94
  
- 7 Example of ANOVA table . . . . . 157



# List of Symbols

$f_p$	Parabolic line profile
$f_b$	Bar-shaped line profile
$f_a$	Parabolic Bar-shaped line profile
$h$	Line Contrast
$w$	Line half local width
$a$	Line asymmetry
$\sigma$	Root of Variance of Gaussian Kernel, Steger algorithm parameter
$i$	Horizontal image coordinate
$j$	Vertical image coordinate
$p(i, j)$	Pixel intensity value at $i, j$
$g_\sigma$	2D Gaussian Kernel function
$r$	Convolution of Gaussian Kernel and line profile
$\phi_\sigma$	Integral of 2D Gaussian Kernel
$x_0$	1D line position
$H$	Hessian Matrix
$\lambda$	Eigenvalue of the Hessian matrix
$n$	Eigenvector of the Hessian matrix
$t$	Line point sub-pixel location
$d$	Distance between two neighbours line points
$\beta$	Angle between two neighbours line points
$e_l$	Left line edge
$e_r$	Right line edge
$l$	Lower threshold, Steger algorithm parameter
$u$	Upper threshold, Steger algorithm parameter
$\hat{w}$	Mean of $w$ calculated along all the detected line
$\sigma^*$	The $\sigma$ value from $\sigma$ array that minimize the difference between $\hat{w}$ and target line mean width
$K_l$	Knee location, represent the position of $\sigma^*$ on $\sigma$ array
$\sigma_{start}$	First value for $\sigma$ array
$\sigma_{end}$	Last value for $\sigma$ array

## List of Symbols

$\sigma_{delta}$	Spacing between single values on the equispaced $\sigma$ array
$e_k$	Optimization error, represent the difference between the target $\hat{w}$ and the $\hat{w}(\sigma^*)$
$e_S$	Steger error, represent the difference between the target $\hat{w}$ and the $\hat{w}$ calculated with Steger algorithm using $\sigma$ values corresponding to the intersection with Steger lower boundary
$e_o$	Optimization error, represent the difference between the target $\hat{w}$ and the $\hat{w}(\sigma_{opt})$
$\sigma_{FL}$	$\sigma$ corresponding to Filling Level position
$u_{FL}$	$u$ corresponding to Filling Level position
$l_{FL}$	$l$ corresponding to Filling Level position



# Acronyms

**AIAG** Automotive Industry Action Group

**ANOVA** Analysis of Variance

**APIs** Application Programming Interfaces

**ArUco** Augmented Reality library from the University of Cordoba [6, 7]

**CNN** Convolutional Neural Network

**COCO** Common Objects in COntext

**EnDurCrete** New Environmental friendly and Durable conCrete

**FCN** Fully Convolutional Neural Network

**FL** Filling Level

**FN** False Negative

**FOV** Field of View

**FP** False Positive

**G3F** Gap and Flush laser triangulation system [8]

**IOU** Intersection Over Union

**LIDAR** Laser Imaging Detection and Ranging

**MLP** Multi-layer Perceptron

**NDC** Number of Distinct Categories

**NDT** Non-Destructive Testing

**ROI** Region of Interest

**SCM** Supplementary Cementitious Materials

**SDK** Software Development Kit

**SHM** Structure Health Monitoring

*Acronyms*

**TN** True Negative

**ToF** Time-of-Flight

**TP** True Positive

**UAVs** Unmanned Aerial Vehicles

**UNIVPM** Università Politecnica delle Marche

# Chapter 1

## General introduction and thesis outline

The work presented in this thesis has been conducted within the research project New Environmental friendly and Durable conCrete (EnDurCrete). The project has received funding from the Horizon 2020 Programme under grant agreement No. 760639. "Ordinary Portland cement has been considered for many years the benchmark structural material for durable buildings and infrastructures, as well as for marine environments. However, its production process consumes significant mineral resources (good-quality limestone and clay), energy and fuel, and creates greenhouse gas emissions. Various efforts are underway to improve the environmental friendliness of concrete, by integrating Supplementary Cementitious Materials (SCM), such as by-products of industrial processes. Nevertheless, current state of the art sustainable solutions based on high substitution rate of Portland cement clinker by SCM occasionally lead to limited performance and durability of concrete structures, particularly critical when applied in harsh conditions. All these aspects affect also the resource consumption, waste production, environmental and economic costs of repair and replacement. In this framework, EnDurCrete Project aims to develop a new cost-effective sustainable reinforced concrete for long lasting and added value applications. The concept is based on the integration of novel low-clinker cement including high value industrial by-products, new nano and micro technologies and hybrid systems ensuring enhanced durability of sustainable concrete structures with high mechanical properties, self-healing and self-monitoring capacities" [9]. This four year project has been carried out by cooperation between two university Università Politecnica delle Marche (Italy) and Norges Teknisk-Naturvitenskapelige Universitet (Norway) and thirteen industrial partner: Vito (Belgium), Infra Plan (Croatia), Rina (Italy), Sika (Switzerland), Nuova Tesi System (Italy), i-Box (Spain), Advanced Management Solutions, KVAERNER (Norway), ZAG (Slovenia), Heidelberg Cement (Germany), Geonardo (Hungary), Fenix TNT (Czech Republic), Acciona (Spain) and CEA (France). In this project Università Politecnica delle Marche (UNIVPM) is leader in Work Package 5: Lab-scale

performance testing and development of monitoring and NDT tools for concrete components and structures. In particular, this thesis work is carried out into task 5.3: Advanced NDT tools for non-intrusive in-field inspection.

In many cases Structure Health Monitoring (SHM) must be complemented with full inspection of the structures, to optimize maintenance and durability. Here the aim is thus the development of advanced NDT tools for non-intrusive in-field inspection. In this task UNIVPM develop inspection tools targeted to the detection of sub-millimetre cracks in concrete. The activity is carried on by working both on the hardware side, by developing an integrated solution exploiting computer vision, and on the software side, by developing high performance processing algorithms for an autonomous crack identification.

The overall outcome of the project is the results of close cooperation among all the people involved in the project; however, this thesis specifically presents results related to task 5.3 of the project, which represents the main research focus of the author.

The thesis work is divided in 7 chapters.

Chapter 1 gives a general introduction and thesis outline with a focus on EnDurCrete project where the work is conducted within.

Chapter 2 presents an extensive literature review of the state-of-art existing techniques for the automatic detection of cracks on concrete surfaces exploiting an analytical and artificial intelligence inferential approach. All the limitations of both methods proposed by the literature are highlighted.

Chapter 3 is divided in 3 sections. Section 3.1 describes the novel image-based computer vision algorithm developed in this thesis work to identify and measure cracks on concrete structures. The section begins with a broad and new theoretical description of the chosen ridge detection algorithm, already known in the state-of-art. The algorithm is then tested on lab-scale scenario and all its limitations are highlighted. In the sub-section 3.1.2, an innovative optimization technique able to measures crack width without knowing a priori proper parameters is presented. Nevertheless, the developed optimization algorithm still presents several limitations that bind its use only in laboratory conditions (3.1.3). From these limitation arise the need of automatically locate the crack inside image. This will be overcome trough Deep Learning algorithms in next section. Section 3.2 describes all steps for the development and training of a model for the automatic discrimination of the target cracks from the background and disturbing elements. Starting from a Convolutional Neural Network architecture, a Fully Convolutional Neural Network able to perform semantic segmentation and locate with high accuracy cracks in input images is built and trained. Section 3.3 described the developed Crack Analyser algorithm. The model built in 3.2 is used for overcoming the limitation of the Sigma Optimization algorithm developed in 3.1.2 in order to assess cracks on a

real-scale scenario.

Chapter 5 explains the architecture design of the measurement system that integrates the developed Crack Analyser algorithm in order to make it usable and user-friendly even by non-expert operators for in-fields testing.

Chapter 4 is divided in 4 sections. Section 4.2 begins with a theoretical description of the techniques chosen to evaluate the metrological characteristics of the instrument. Sections 4.3 and 4.4 show the Measurement System Analysis (MSA) of the developed algorithm where the metrological performance of the measurement system is assessed in terms of precision and accuracy for two case scenarios: camera without depth sensor, that measure the conversion constant pixel-to-mm using fiducial markers, and camera equipped with depth sensor (i.e. depth cameras, LIDAR or Tof sensors), that measures the conversion constant pixel-to-mm without fiducial markers.

Chapter 6 shows the results of the in-field tests carried out in different scenarios.

Chapter 7 discusses the results and future developments of the thesis work carried out.



# Chapter 2

## State of the Art: image-based NDT for crack severity analysis

This chapter presents an extensive literature review of the state-of-art existing techniques for the automatic detection of cracks on concrete surfaces exploiting analytical and artificial intelligence inferential approaches.

Knowing the crack width is fundamental to verify the compliance with crack admissibility tables reported in the UNI EN 1992-1-1:2005 standard [5]. For the EnDurCrete project exposition classes the maximum admitted crack width is 0.3 [mm]. Such a tight boundary implies the use of measurement systems whose metrological characteristics ensure precision and accuracy at least one order of magnitude lower than the target dimension. In fact, a higher uncertainty would imply a misclassification of the crack severity that cannot be accepted. The typical instruments discussed in literature rely on optical triangulation that consists on a laser line projection onto a target surface while a camera, recording the scene from a known angle  $\theta$ , detects light scattered by the target surface (Figure 2.1).

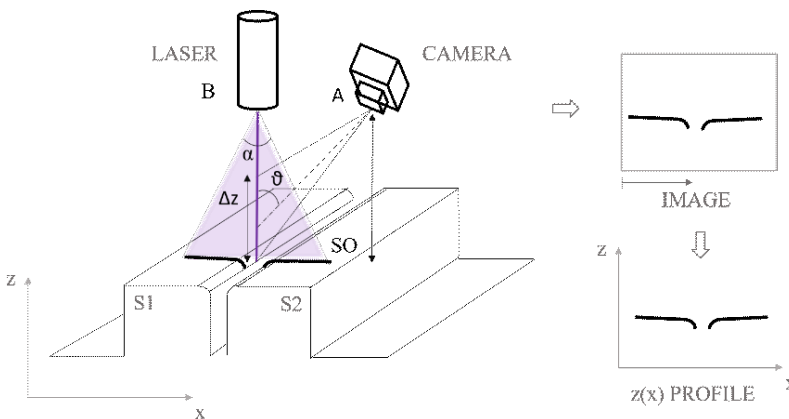


Figure 2.1: Laser triangulation system's working principle [8]

There are several works addressing surface crack detection through laser triangulation in literature. These papers report the use of this technique on several kind of surfaces like pavements, ranging from concrete to asphalt and chip seal. Through the combined use of high-speed cameras, laser line projector and 3D reconstruction algorithms this technique is able to well-detect crack line profile. In [13] Pavements inc., in collaboration with the National Optics Institute of Canada and the Minister des Transports du Quebec, develops and tests a new 3D technology called Laser Crack Measurement System (LCMS) proven to be 95% correct in general crack classification tasks. In [12] O'Brien et al. develop an automated pavement assessment using the LCMS that has been in extensive use in Ireland since 2012 to provide high speed pavement distress data and imagery for runways, taxiways and aprons. In both articles it is demonstrated that the LCMS system has an excellent 3D depth resolution (0.5 mm), a good resolution along the lateral direction of scanning (1mm) coupled with a field of view of 4 meters. This method appears to be extremely performing because, with an appropriate acquisition frequency, it would be possible to inspect very large areas in a short time. The downsides of this approach mainly concern the resolution and versatility of the system. In fact, this kind of instrumentation must be mounted on moving systems, such as vans or systems on rails, and as we know it is not always possible to scan cracks on surfaces in this way. Moreover the spatial resolution reached by the instrument for many applications could be limiting. The performance of this system is also very much subjected to how you process the data output from the system. In [14] Ouyang et al. introduce an automated pavement inspection system which uses cameras and laser to frame dense transverse profiles of a pavement lane surface when it carries a moving vehicle. Also in this case a system very similar to the previous one is introduced, with a transverse resolution of 1.56mm and a depth resolution of 0.5mm with the camera positioned at 1.4m. Also in this case there are the same problems found in the system previously introduced. As previously mentioned, the processing of images captured by linear cameras becomes of paramount importance in this type of application. In [15] a fully automated algorithm for segmenting and enhancing pavement crack on 3D pavement images is proposed, achieving an average precision of 88.38%, recall of 93.15%, and F-measure of 90.68% exploiting SMFB operation, Tensor Voting. Thanks to this technique it is possible to recognize the presence of cracks in the images acquired, to be then measured by traditional algorithms. Despite the excellent results achieved by the method, however, this type of processing is very expensive from a computational point of view (about 10.3s per image).

Although laser triangulation is a consolidated technique, it can be of difficult use in those applications where the target cannot be easily reached, like pylons of bridges, offshore structures, inaccessible surface of builds and all those scenarios



where is not possible to bring all the necessary equipment. From this point of view, other techniques based solely on computer vision can be more versatile, as they could be embedded in remotely controlled systems like UAVs (Unmanned Aerial Vehicles) [[17]]. Moreover, these approaches are also well suited to be adopted in mobile device to allow assisted inspection by human operators. As far as imaging is concerned, the use of threshold techniques (e.g. Niblack method [18], Bernsen method [19], Sauvola and Pietikäinen method [20], Wolf method [21], and Otsu method [22]) for image segmentation in order to detect surface crack or non-compliances is widely documented. In [28] an algorithm capable of automatically detect sealed cracks in pavement surface images via Heuristic Thresholding Approach is developed. In [29], Talab et al. present a new method for concrete crack detection based on Otsu method and multiple filtering technique. Despite the good performance of these two techniques both are primarily limited to crack detection and not to crack measurement. They also suffer greatly in distinguishing cracks from other disturbances. Oliveira and Correia [33] perform a crack segmentation using entropy and image dynamic threshold. The main disadvantage of using thresholds lies in the fact that it does not allow for complete generalization of the algorithm, always requiring human input in assessing its value. Cheng et al. in [32] try to overcome this limitation performing a real-time automated detection of pavement cracks through space reduction and interpolation approach for automatic determination of greyscale threshold values. This method has the advantage of being able to automatically derive the threshold value on the gray scale ensuring the use of the method even in real-time processing tasks. This method has the advantage of being able to automatically derive the threshold value on the gray scale ensuring the use of the method even in real-time processing tasks. Nevertheless, working on the grayscale can be extremely risky as a crack can be easily confused with other disturbing elements. In [26], Rimkus et al. exploit Digital Image Correlation (DIC) technique for crack localization using the Agglomerative Hierarchical Clustering Technique for identifying pixels belonging to cracks. Alam et al. [30] determine crack width and spacing through digital image correlation and acoustic emission. Both of the proposed methods prove to be very effective in automatically detecting cracks even of very complex shapes, but they face all the limitations introduced by the use of DIC-based techniques. In fact, although the use of DIC brings with it a number of important advantages, such as the possibility of extracting the deformation field and from this the stress state through inverse approaches, necessarily implies the acquisition of sequences of images and it is therefore usable only where the generation of the fracture is predictable and controllable. The quality of the acquired images is of fundamental importance in the localization and measurement of cracks. An incorrect illumination could compromise the measurement. L. Ying and E. Salari

[31] developed an algorithm to correct the non-uniform background illumination by calculating the multiplicative factors that eliminate the background lighting variation. Yu et al. [34] overcome challenging issues such as low contrast, uneven illumination, and severe noise pollution that generally exist in a tunnel lining image through infrared images. The use of infrared images for improving crack detection is also reported in [34].

All of the image-based techniques just described have the advantage of being versatile and fast. For these reasons, the possibility to exploit UAVs and mobile devices is the background idea that drove the development of the crack severity measurement system reported in this work.

The techniques presented so far mostly involve the use of analytical computer vision algorithms; in recent years numerous possibilities have been explored involving the use of artificial intelligence techniques for the identification of cracks on structures of various kinds. Among the most used techniques we must certainly mention those exploiting Convolutional Neural Networks (CNNs) and their evolution over time. Convolutional Neural Networks, introduced by LeCun in 1989 for computer vision applications [36, 37], are a category of Neural Networks that have proven to be very effective in areas such as image classification [42]. CNNs have been successful in identifying faces, objects, and traffic signs apart from powering vision in robots and autonomous applications. [43–50] In [44], Zang et al. present a new approach to detect traffic signs based on cascaded convolutional neural networks. In [43], Abdel-Salam et al. introduces Real-Time Image Enhanced CNN (RIECNN) for traffic sign recognition, a real-time novel approach that tackles multiple, diverse traffic sign datasets, and out-performs the state-of-the-art architectures in terms of recognition rate and execution time. In [45], Memon et al. explore the system to helps the driver about recognition of road signs to avoid road accidents using CNNs achieving the 95% of accuracy in classification tasks. In [48] an hybrid neural-network for human face recognition is presented for the first time. In [46] six commercial face recognition CNNs are tested. Hancock et al. found that they outperform typical human participants on standard face-matching tasks. In [47] Syafeeza et al. presents a robust 4-layer CNN architecture for face recognition problems, with a solution that is capable of handling facial images that contain occlusions, poses, facial expressions and varying illumination. In [50] Lou et al. develops a multi-level information fusion model based on the VGG16 fully connected neural network for face image recognition leading reducing computational time and increasing the recognition rate of the image.

CNN is a Deep Learning algorithm which usually takes images as input and returns a vector, usually used for classification, as output. The effectiveness of using CNNs for the automatic identification of cracks on images is amply demonstrated in the literature by the presence of several works. [56] proposes

a deep learning framework, based on a CNN and a Naive Bayes data fusion scheme, called NB-CNN, to analyze individual video frames for crack detection of nuclear power plant components. This method is not only limited to the classification of single frames of a video, but uses the information obtained from each frame to increase the overall performance and robustness of the proposed system. In this way, through Naive Bayes decision making, the number of false positives is lowered, achieving an accuracy of 98.5%. [57] proposed a customized CNN for crack detection in concrete structure and evaluate the performance of the proposed model through the comparison with pretrained networks, i.e., the VGG-16, VGG-19, ResNet-50, and Inception V3 models, on eight datasets of different sizes. This work is particularly useful because it tests the performance of known models in the literature from both a classification and a localization perspective, exploiting a "sliding windows" approach. At the same time it analyzes the performance of the models as the variability and numerosity of the input dataset increase, showing that performance does not always increase as the numerosity of the data used for training increases. By analyzing the variance of the data used for training, we also conclude that all of the models tested can achieve good accuracy if good variance is ensured among the data. Shengyuan et al. design a CNN classifier architecture through fine-tuning an existed CNN. The results illustrate that the proposed method shows high accuracy and robust performance and can indeed detect crack on real concrete surface [60]. In this work the authors integrate the system into a smartphone making it easy and immediate to use.

The main limitation of CNNs is that they act as a classifier. In the literature there are several articles, already cited above, where "sliding windows" approaches are adopted in order to perform a semantic segmentation starting from CNN networks. However, this brings numerous disadvantages from the point of view of resolution and in terms of computational times. For all applications where semantic segmentation is most required, such as in the case of the identification of cracks, more advanced networks were soon adopted, capable of implementing convolution logics of the same CNNs: from Regional-CNN to Fully Convolutional Neural Networks (FCNs), up to the most advanced UNET. The UNET was developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation [97]. The UNET is largely used for image segmentation with great success in different fields: seabed segmentation in mineral images, nuclei, tumor, retina vessel and skin lesion segmentation in medical images, and rice lodging segmentation in UAVs images. In [64], Zhang et al. discuss the advantages and disadvantages of the Dice loss function for lesion segmentation in medical images. In [65], Zeng et al. present a new method called RIC-Unet (residual-inception-channel attention-Unet) for nuclei segmentation tasks in histology images. In [66], Li et al. propose a novel hybrid densely

connected UNet (H-DenseUNet), which consists of a 2-D DenseUNet for efficiently extracting intra-slice features and a 3-D counterpart for hierarchically aggregating volumetric contexts for liver and tumor segmentation. In [69], Chen et al. propose a novel Separable 3D U-Net architecture using separable 3D convolutions overcoming the problem that 2D convolutions cannot make full use of the spatial information of volumetric medical image data, while 3D convolutions suffer from high expensive computational cost and memory demand. Xiao et al. proposed a U-Net-like model with the weighted attention mechanism and the skip connection scheme in order to increase segmentation performance for retina vessel segmentation which is particularly complicated due to the extreme variations in the morphology of the vessels against the noisy background [70]. In [71] Skin Lesion Segmentation (SLS) method based on the separable-UNet with stochastic weight averaging is proposed by Tang et al. In [72], Huang et al. propose a novel architecture called UNET 3+, which takes advantage of full-scale skip connections and deep supervisions. The full-scale skip connections incorporate low-level details with high-level semantics from feature maps in different scales while the deep supervision learns hierarchical representations from the full-scale aggregated feature maps. In [73] Zhao et al. proposed a new method for rice lodging assessment based on a deep learning UNET exploiting UAVs and multi-spectral imaging.

The effectiveness of using R-CNN, FCN and U-NET for the automatic semantic segmentation of cracks on images is widely demonstrated in the literature by the presence of several works [77–83]. In [74] Dung et al. propose a vision-based method for concrete crack detection exploiting FCN using VGG16 architecture as backbone. This architecture proves to perform better than InceptionV3 and ResNet for crack image classification, reaching 90% of F-1 score. In [75], Attard et al. demonstrate that Mask R-CNN can be used successfully in localization and segmentation tasks of cracks on concrete surfaces. Tests on the trained model achieved a precision value of 94%. As explained by the authors of both articles, these methods, although very performant in segmenting the crack, does not solve the problem of accurately measuring its geometric dimensions automatically.

Nevertheless, all the methods described above are limited to the simple identification of the crack on the image. What is missing is a complete and exhaustive technique for identifying and measuring the main characteristics of the crack: length, average width, maximum width. This measurement is in fact fundamental in practice in order to understand the degree of severity of the crack and monitor its evolution over time. Thus, the literature review shows that there is a need to develop a strategy that can automatically detect and measure a crack on concrete surfaces. Not only will this technique need to be metrologically appropriate, but it will also need to be versatile and usable in

complex situations. Fundamental will be therefore the possibility to embed this technology on remote systems, such as robots or UAVs, or in portable systems. This system shall be used during inspection processes even by non-expert operators, with the feature of not having to resort to bulky, expensive and difficult to use equipment.



# Chapter 3

## Methods

### 3.1 An optimized ridge detector algorithm for crack identification

#### 3.1.1 Crack identification via Ridge Detectors: theory

As described in chapter 2, line identification techniques in image processing is typically addressed by two main image filtering approaches: edge detection and ridge detection. Despite similar at first impression, they embed important differences that prevent the use of one approach with respect to the other in certain applications. An Edge filter is typically a first derivative operator, which measures the speed with which the intensity level changes in the whole image. On the one hand, these filters (e.g. Sobel, Canny, Roberts, etc.) well detect borders between areas of high and low grey values; on the other hand, the output generates a double-line, one on each side of the line. This is clearly visible in the concrete crack example image in figure 3.1. While Edge detectors are typically first derivative operators, Ridge detector algorithms detect thin lines darker or brighter than their neighbour pixels and work as second derivative operators (figure 3.2). This characteristic made ridge detectors very suitable for not only for surface defects detection, but also for width estimation because it allows to detect the central line position with high accuracy.

For these reasons, in order to detect the central line of concrete cracks, ridge detector algorithms are better to be used. There are different ridge detection algorithms in computer vision literature. A well-established ridge detection algorithm able to locate, link, and measure line width is the well-known Steger line detector algorithm [84]. For this work we use the Steger's second-derivative hysteresis-threshold algorithm. The basic idea is that ridge in a surface will have high curvature in the direction perpendicular to the ridge, this means an high second derivative of the image function  $p(i, j)$ . Furthermore, the first derivative will be zero at the top of the ridge since it is a local maximum. Thus, ridges are identified as contiguous sets of points at which  $p(i, j)$  has a local high curvature. The algorithm involves four main different steps:

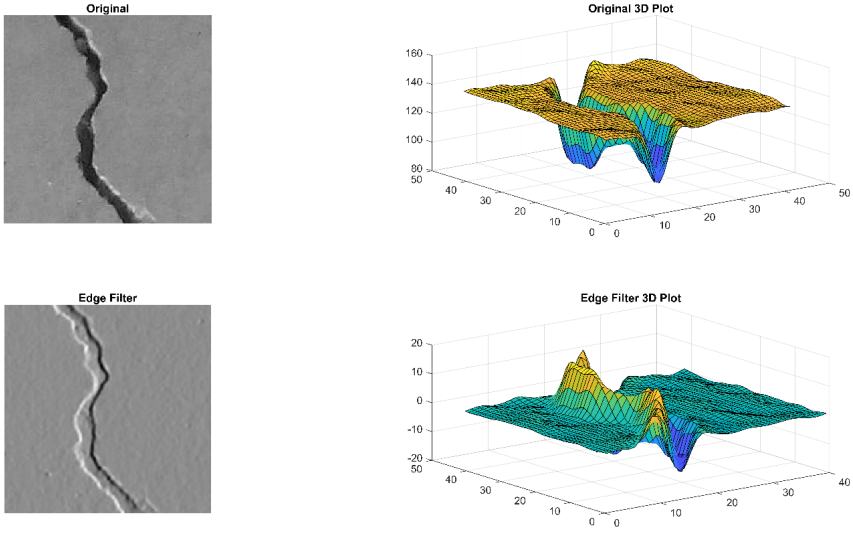


Figure 3.1: First derivative behaviour of edge filter is clearly shown through the 3D plot before and after filtration of a wall crack image

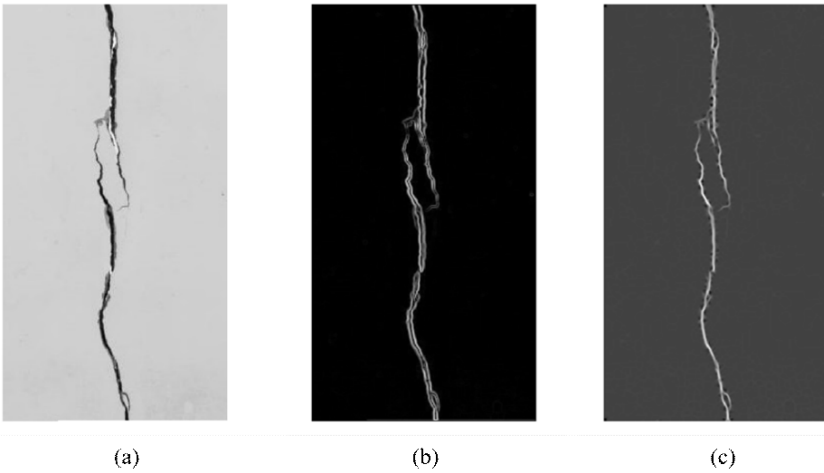


Figure 3.2: Original image (a), edge filter (b) and ridge filter (c)



- line points detection,
- line points joining,
- line width calculation,
- line bias removal.

### Line Points Detection

The line point detection step defines which pixels of the image are possible candidates to belong to line and defines the final line with sub-pixel accuracy. It is possible to consider cracks, or surface defects, as darker lines in a bright surrounding, as shown in Figure 3.2. If we take a 2D greyscale image containing cracks (ridges) and cross-cut perpendicularly to one crack, three different types of line profile can be obtained (Figure 2.4):

- parabolic line profile (3.1),
- bar-shaped line profile (3.2),
- asymmetrical Bar-Shaped line profile (3.7).

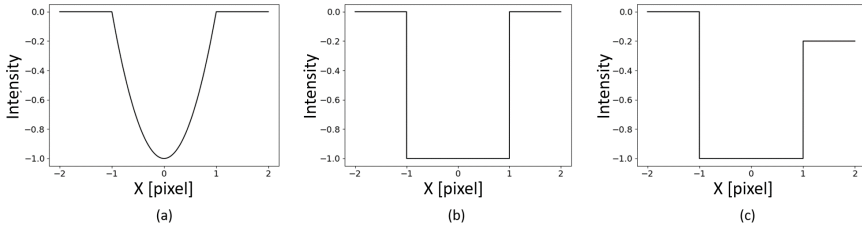


Figure 3.3: Parabolic line profile  $f_p$  (a), bar-shaped line profile  $f_b$  (b), and asymmetrical bar-shaped line profile  $f_a$  (c), with  $h = 1$ ,  $w = 1$ , and  $a = 0.2$

$$f_b(x, w, h) = \begin{cases} -h, & |x| \leq w \\ 0, & |x| \geq w \end{cases} \quad (3.1)$$

$$f_p(x, w, h) = \begin{cases} -h \left( 1 - \left( \frac{x}{w} \right)^2 \right), & |x| \leq w \\ 1, & |x| \geq w \end{cases} \quad (3.2)$$

$$f_a(x, w, a, h) = \begin{cases} 0, & x < w \\ -h, & |x| \leq w \\ -ha, & x > w \end{cases} \quad (3.3)$$

where  $2w$  is the crack width,  $h \in [0, 1]$  is the line depth (grey value), and  $a \in [0, 1]$  represents the asymmetry of all the lines which have different surrounding heights. The  $a$  value will be zero if there is no asymmetry. As shown in Figure 3.3, it is clear that the asymmetrical bar-shaped profile (Figure 3.3 (c)) is the more general and can best fit 1D crack profile in real images (e.g. Figure 3.4). From this point forward, the term  $h$  is neglected from all the calculations because the asymmetrical bar-shaped line profile can be scaled as  $hf_a(x)$ .

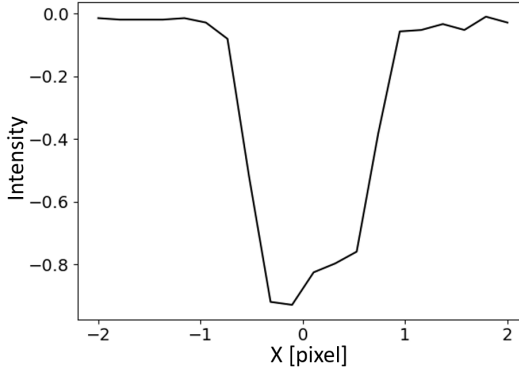


Figure 3.4: 1D Crack line profile from real image

### Gaussian kernel convolution

Noise can be an obstacle to identify line points and on a framed image it will always be present. In the previous chapter it has seen how, through the first or second derivative, it was possible to identify curvilinear structures in an image. But an image, which itself contains noise, cannot be directly derived. Figure 3.5 shows the graph of the 1D profile of a curvilinear structure with added noise, as it could be the profile of a crack on surface. By deriving the function, noise increases without showing any salient points for identifying the line.

One possible solution to this problem is to convolve the image with a Gaussian kernel in order to smooth the image according to a  $\sigma \in \mathbb{R}^+$  parameter. In this way, it is possible to reduce noise contribution and neglect it as plausible candi-

### 3.1 An optimized ridge detector algorithm for crack identification

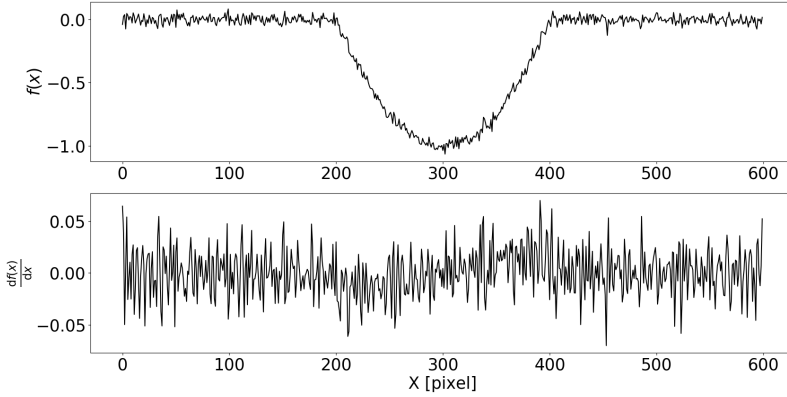


Figure 3.5: 1D plot of the intensity of a row or column of an image along the perpendicular to a line. The plot shows how deriving the intensity function the noise increases.

date to constitute line points. Figure 3.6 shows how deriving the convolution between the intensity function and the Gaussian profile, it is possible to locate the salient points (local maxima) that allow the identification of the line.

Choosing the  $\sigma$  parameter involves the definition of a low threshold, because it will cut out line under a certain width. For this reason, in this discussion,  $\sigma$  is used as scale-space parameter, in order to decide the range of line width of interest. Scale-space theory is a well-known framework in computer vision to handle the images multi-scale nature. Indeed, the best value of a parameter is difficult to be known a priori, hence it is more reasonable to represent input data at multiple scales. Figure 3.7 shows different levels of scale-space representation of a crack image at scale level  $\sigma = 1, 2, 3, 4, 5, 6$ . The Gaussian kernel  $g_\sigma$  and its derivatives are given by the following equations:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.4)$$

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.5)$$

$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}} \quad (3.6)$$

$$g''_\sigma(x) = \frac{-x^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{x^2}{2\sigma^2}} \quad (3.7)$$

The Gaussian kernel is then convolved with the asymmetrical bar-shape line

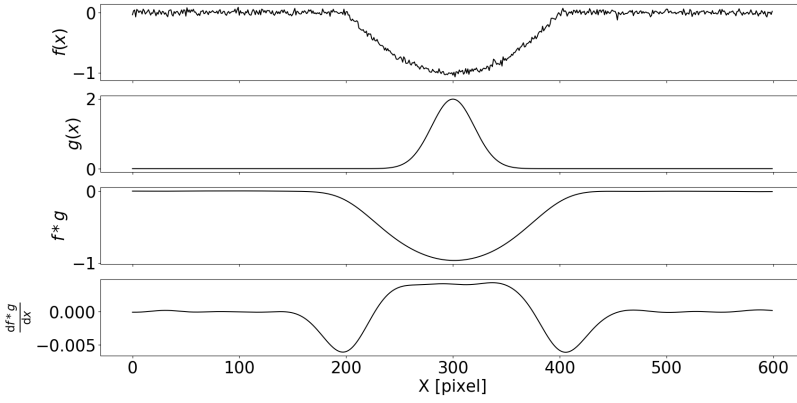


Figure 3.6: 1D plot of the intensity of a row or column of an image along the perpendicular to a line. The plot shows how deriving the convolution between the intensity function and the Gaussian profile, it is possible to locate salient points (local maxima) that allow the identification of the line.

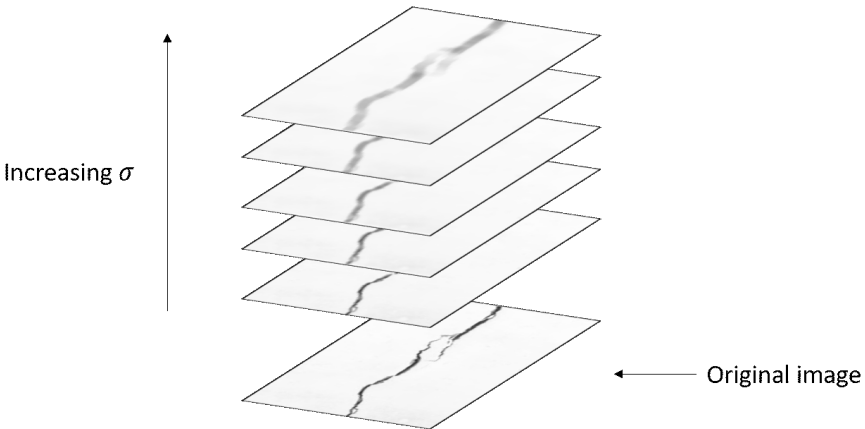


Figure 3.7: Different levels in the scale-space representation of a crack image

### 3.1 An optimized ridge detector algorithm for crack identification

profile function. Convolution of two functions produces a third function expressing how the shape of one is modified by the other. It is defined as the integral of the product of the two functions, one of which is reversed and shifted. It is an integral transform given by the following expression:

$$(g * f)(x) = \int_{-\infty}^{+\infty} g(\tau)f(x - \tau)d\tau \quad (3.8)$$

The convolution,  $r(x, \sigma, w, a)$ , of the Gaussian function and asymmetrical bar-shape line function, is given by the following expressions:

$$r(x, \sigma, w, a) = g_{\sigma}(x) * f_a(x, w, a) = \phi_{\sigma}(x + w) + (a - 1)\phi_{\sigma}(x - w) \quad (3.9)$$

$$r'(x, \sigma, w, a) = g'_{\sigma}(x) * f_a(x, w, a) = g'_{\sigma}(x + w) + (a - 1)g'_{\sigma}(x - w) \quad (3.10)$$

$$r''(x, \sigma, w, a) = g''_{\sigma}(x) * f_a(x, w, a) = g''_{\sigma}(x + w) + (a - 1)g''_{\sigma}(x - w) \quad (3.11)$$

The integral function,  $\phi_{\sigma}(x)$ , of the Gaussian kernel is reported as follow:

$$\phi_{\sigma}(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dt \quad (3.12)$$

Unfortunately, Equation 3.12 cannot be solved analytically. As shown in Figure 3.8, after convolving the Gaussian kernel to bar-shaped line profile, the profile becomes smoother and fit better a real crack profile. This also leads to the

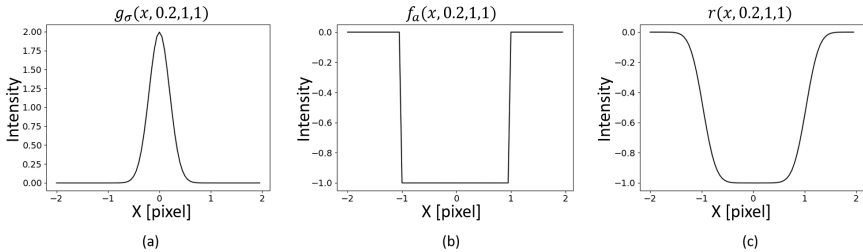


Figure 3.8: Gaussian kernel (a), bar-shaped line profile (b) and the result of their convolution (c) with  $\sigma = 0.2, w = 1, a = 1$ .

detection of the centre of the curve using the zero point of the first derivative  $r'(x, \sigma, w, a)$ . The  $x$  position where the first derivative become zeros is given by:

$$r' = (x_0, \sigma, w, a) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_0+w)^2}{2\sigma^2}} + (a-1) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_0-w)^2}{2\sigma^2}} \quad (3.13)$$

$$e^{-\frac{(x_0-w)^2}{2\sigma^2}} = (1-a) e^{-\frac{(x_0-w)^2}{2\sigma^2}} \quad (3.14)$$

$$-\frac{(x_0+w)^2}{2\sigma^2} = -\frac{(x_0-w)^2}{2\sigma^2} + \ln(1-a) \quad (3.15)$$

$$-\frac{(2x_0)(-2w)}{2\sigma^2} = \ln(1-a) \quad (3.16)$$

$$x_0 = -\frac{\sigma^2}{2w} \ln(1-a) \quad (3.17)$$

Equation 3.17 clearly demonstrates that, if we use the zero point of the first derivative  $r'(x, \sigma, w, a)$  to locate the line position,  $x_0$  will be affected by a significant error if the line is asymmetrical. The correction of this bias is one of the key points of the Steger's algorithm and will be discussed in detail later in the next chapter. If we look at Figure 3.9, which shows the scale-space behaviour of a bar-shaped line profile ( $h = 1, a = 1, w = 1$ ) when convolved with the derivatives of Gaussian kernel, it is easy to note that the second derivatives do not have the maximum on  $x = 0$  for small sigma values. In order to select

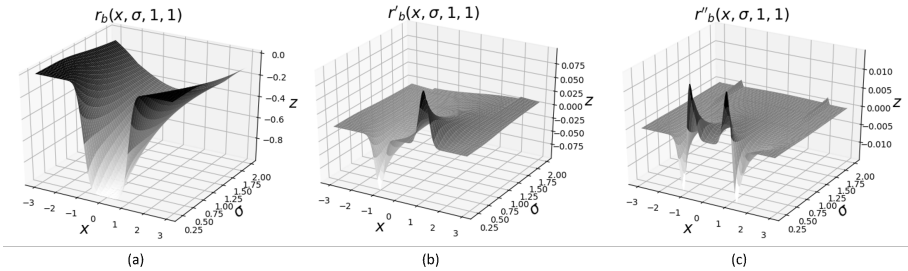


Figure 3.9: Scale-space behaviour of the bar-shaped line profile when convolved with the derivatives of Gaussian kernel for  $x \in [-3, 3], \sigma \in [0.2, 2], a = 1$  and  $w = 1$

salient line points, it is desirable that the second derivative of the response  $r''(x, \sigma, w, a)$  presents a maximum at  $x = 0$  (line centre), that is the true line position. As shown in Figure 2.9, this happens only after a certain value of  $\sigma$ . Once this has been assessed, line point detection can be performed. The value

### 3.1 An optimized ridge detector algorithm for crack identification

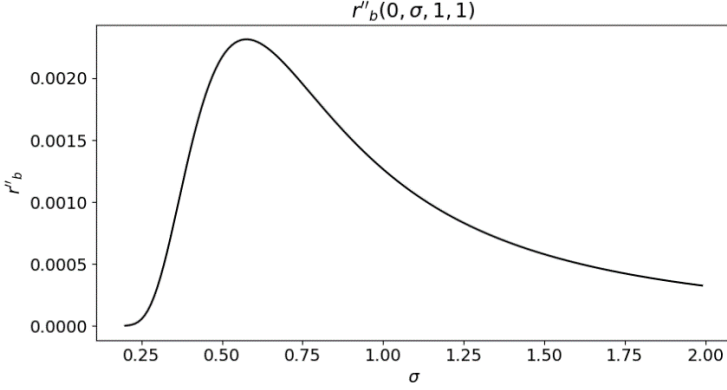


Figure 3.10: Scale-space behaviour of the bar-shaped line profile when convolved with the derivatives of Gaussian kernel for  $x \in 0, \sigma \in [0.2, 2], a = 0$  and  $w = 1$

of  $\sigma$  where the function  $r''(0, \sigma, w, 0)$  takes its maximum can be found by:

$$\frac{\partial}{\partial \sigma} r''_b(0, \sigma, w, 0) \leq 0 \quad (3.18)$$

$$\frac{\partial}{\partial \sigma} r''_b(0, \sigma, w, 0) = \frac{\partial}{\partial \sigma} [g''_\sigma(0) * f_a(0, w, 0)] = \frac{\partial}{\partial \sigma} [g'_\sigma(w) + g'_\sigma(w)] \quad (3.19)$$

$$g'_\sigma(w) + g'_\sigma(w) = -\frac{-w}{\sqrt{2\pi}\sigma^3} e^{-\frac{-w^2}{2\sigma^2}} = e^{-\frac{-w^2}{2\sigma^2}} \left( -\frac{-2w}{\sqrt{2\pi}\sigma^3} \right) \quad (3.20)$$

$$\frac{\partial}{\partial b} (g'_\sigma(w) + g'_\sigma(w)) = w^2 \frac{1}{\sigma^3} e^{-\frac{w^2}{2\sigma^2}} \left( -\frac{-2w}{\sqrt{2\pi}\sigma^3} \right) + e^{-\frac{w^2}{2\sigma^2}} \left( \frac{2w}{\sqrt{2\pi}} \right) (3) \left( \frac{1}{\sigma^4} \right) \quad (3.21)$$

$$w^2 \frac{1}{\sigma^3} e^{-\frac{w^2}{2\sigma^2}} \left( -\frac{-2w}{\sqrt{2\pi}\sigma^3} \right) + e^{-\frac{w^2}{2\sigma^2}} \left( \frac{2w}{\sqrt{2\pi}} \right) (3) \left( \frac{1}{\sigma^4} \right) \geq 0 \quad (3.22)$$

$$\frac{e^{-\frac{-w^2}{2\sigma^2}}}{\sqrt{2\pi}} \left( \frac{-2w^3}{\sigma^6} + \frac{6w}{\sigma^4} \right) \geq 0 \quad (3.23)$$

$$2w(2\sigma^2 - w^2) \geq 0 \quad (3.24)$$

$$\sigma \geq \frac{w}{\sqrt{3}} \quad (3.25)$$

This imply that  $\sigma$  not only imposes a lower limit due to the smoothing of the

Gaussian kernel, but also an upper limit where the function  $r''_b(0, \sigma, w, 0)$  reach its maximum. From now on this will be referred to be the Steger limitation. This upper limit depends on the line width value  $2w$ . The next step is to determinate the ridge location with sub-pixel accuracy. At first, we will illustrate the 1D lines case, then we will move this concept for 2D lines. The second order Taylor polynomial of the piecewise image function is given by:

$$p(x) = r + r'x + \frac{1}{2}r''x^2 \quad (3.26)$$

where  $r$ ,  $r'$  and  $r''$  are the previously obtained results of convolving with Gaussian kernel. The ridge location will be where the first derivative of the function becomes zero.

$$p'(x) = r' + r''x = 0 \quad (3.27)$$

$$x = \frac{r'}{r''} \quad (3.28)$$

For 2D images, it is possible to approximate locally the pixel value by a second order polynomial. If the image function is called  $p(i, j)$ , for small values of  $i$  and  $j$  it is possible to express  $p(i, j)$  as:

$$p(i, j) \approx \frac{1}{2}i^2 \frac{\partial^2 p}{\partial i^2} + ij \frac{\partial^2 p}{\partial i \partial j} + \frac{1}{2}j^2 \frac{\partial^2 p}{\partial j^2} + i \frac{\partial p}{\partial i} + j \frac{\partial p}{\partial j} + p(0, 0) \quad (3.29)$$

In matrix form, it is possible to write:

$$p(i, j) \approx \frac{1}{2} \begin{pmatrix} i \\ j \end{pmatrix} \begin{pmatrix} \frac{\partial^2 p}{\partial i^2} & \frac{\partial^2 p}{\partial i \partial j} \\ \frac{\partial^2 p}{\partial i \partial j} & \frac{\partial^2 p}{\partial j^2} \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} i \\ j \end{pmatrix} \begin{pmatrix} \frac{\partial p}{\partial i} & \frac{\partial p}{\partial j} \end{pmatrix} + p(0, 0) \quad (3.30)$$

The second order derivative matrix is the Hessian Matrix  $H$  and it describes ridges structure:

$$H(i, j) = \begin{pmatrix} \frac{\partial^2 p}{\partial i^2} & \frac{\partial^2 p}{\partial i \partial j} \\ \frac{\partial^2 p}{\partial i \partial j} & \frac{\partial^2 p}{\partial j^2} \end{pmatrix} \quad (3.31)$$

It is possible to decompose this matrix into the sum of two parabolic functions according to the following equation:

$$H(i, j) = \lambda_1 i^2 + \lambda_2 j^2 \quad (3.32)$$

where  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of  $H(i, j)$  matrix. This function can assume different shapes depending on  $\lambda_1$  and  $\lambda_2$  values. The plot of the function with different  $\lambda_1$  and  $\lambda_2$  values are shown in Figure 3.11.

To detect ridges, it is easy to note that the function  $H(i, j)$  must look like the graph relating to  $\lambda_1 = 1$  and  $\lambda_2 = 0$ , i.e. bottom right image of Figure



### 3.1 An optimized ridge detector algorithm for crack identification

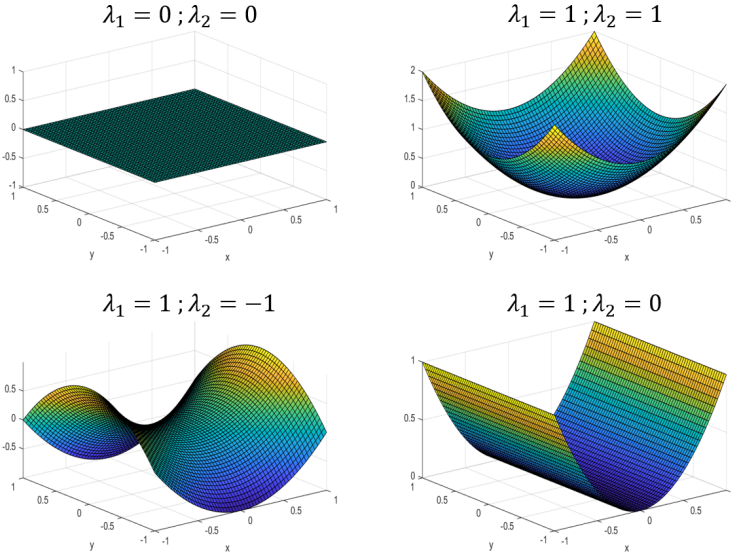


Figure 3.11: Shaped assumed by the decomposition of the Hessian matrix varying  $\lambda$  values.

3.11. If we want to detect ridges, we must look for all  $p(i, j)$  pixels where the absolute difference between eigenvalues of the Hessian matrix is maximum. Then, it is possible to calculate the eigenvector corresponding to the eigenvalue of  $H(i, j)$  with maximum absolute value  $(n_i, n_j)$  with  $|(n_i, n_j)|_2 = 1$ . This vector represents the direction perpendicular to the line. For 2D images, the first directional derivatives  $r'$  and  $r''$  along  $(n_i, n_j)$  are given by:

$$r' = \frac{\partial p}{\partial i} n_i + \frac{\partial p}{\partial j} n_j \quad (3.33)$$

$$r'' = \frac{\partial^2 p}{\partial i^2} n_i^2 + \frac{\partial^2 p}{\partial j^2} n_j^2 + \frac{\partial^2 p}{\partial i \partial j} n_j n_i \quad (3.34)$$

From Equation 3.17, the line point sub-pixel location  $t$  can be determined as:

$$t = -\frac{\frac{\partial p}{\partial i} n_i + \frac{\partial p}{\partial j} n_j}{\frac{\partial^2 p}{\partial i^2} n_i^2 + \frac{\partial^2 p}{\partial j^2} n_j^2 + \frac{\partial^2 p}{\partial i \partial j} n_j n_i} \quad (3.35)$$

Therefore, given the longest eigenvector  $n = (n_i, n_j)$ , the directional derivative along  $n$  vanishes in the point given by:

$$p = (p_i, p_j) = (tn_i, tn_j) \quad (3.36)$$

In conclusion, the line point detection phase works as follows:

- Calculation of the local Hessian Matrix for each pixel storing eigenvectors and eigenvalues.
- Application of an hysteresis threshold on pixels eigenvalues: any pixel that has a value above the higher threshold will be marked as possible line pixel, and any pixel whose value lies between the higher and the lower thresholds, but is connected to a pixel whose value is above the higher threshold, will also be marked as possible line pixel. All the other pixels of the image are rejected.
- For each of the selected line pixels, the local point  $(tn_i, tn_j)$  is calculated. The global line point coordinates will be given by  $(c + tn_i, r + tn_j)$ , where  $c, r$  are the number of column and row of the target pixel, respectively.

### Line points joining

From the line point detection step, for each possible line point pixel, the following information is obtained:

- Direction of the longest eigenvector  $n = (n_i, n_j)$ , which represents the direction perpendicular to the line.
- Subpixel position where the directional derivative along  $n$  vanish  $(c + tn_i, r + tn_j)$ .

The next step will join all these points into lines, when possible. All the possible points are examined based on their position on the image, from top left to bottom right. At this point, Steger assumes that line does not have sharp corners. For this reason, as shown in Figure 3.12, it is sufficient to examine only three of the eight neighbours of each pixel based on the direction perpendicular to the normal  $n$  founded in the previous step.

Once three neighbours are selected, the algorithm examines only those points belonging to a line. For each line point, the distance  $d$  and the angle difference  $\beta$  are measured. The point that minimizes the value of  $d + c\beta$  is chosen as point to join to the current point, where  $c$  is set to 1. This step is repeated until no line points fall into examinable neighbours. If a point belonging to another line is selected as point to join with, that point became a junction, and the line of the junction point is split into two parts obtaining three lines in total.

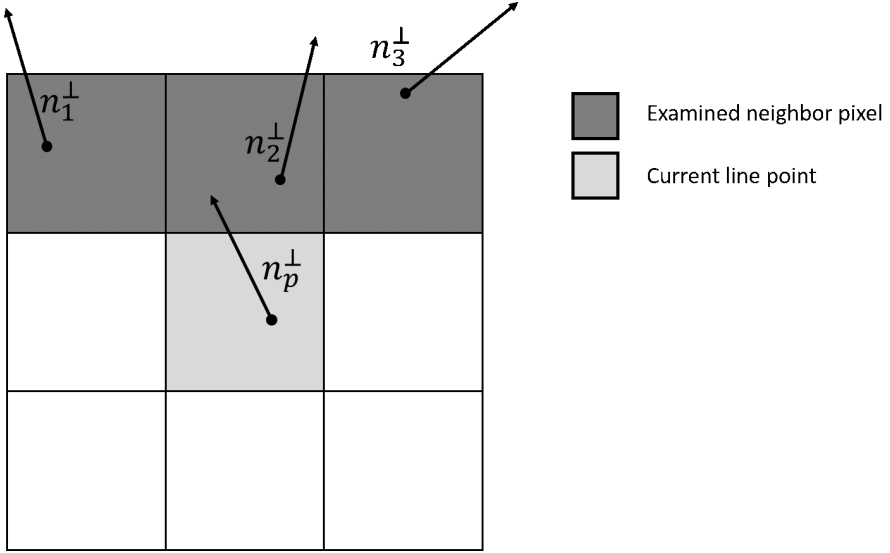


Figure 3.12: Neighbours selection based on the angle perpendicular to the normal  $n$  of the current line point.

### Line width calculation

Once the lines in the image are located with sub-pixel accuracy is possible to measure their width depending on the  $\sigma$  value. For line width determination, we must look for edges along the direction perpendicular to the line. To determine the neighbour pixel perpendicular to the line, Steger implements in the script a modified Bresenham algorithm [85], which calculates all intermediate points over the interval between start and end points, implemented entirely with integer numbers, in order to select which pixel belong to the desired line. From the previous section, we know that it is sufficient to look at a distance of  $\sqrt{3}\sigma$ , but the algorithm refers to a larger value of  $2.5\sigma$  (Figure 3.13), in order to ensure that almost all the edge points are detected and stay on safe side. Once all of the points belonging to the normal of the line are founded, the algorithm looks for the point where the absolute value of the gradient is maximum and it uses an approach similar to the previous step used to find line points, in order to find where  $r''$  vanishes. As shown in Figure 3.9,  $r''$  will vanish at the edge location of the line, so we will find two edge points, namely  $e_l$  on the left and  $e_r$  on the right side, where  $r''(e_l, \sigma, w, a) = r''(e_r, \sigma, w, a) = 0$ . This equation cannot be solved analytically, so they are computed by a root finding algorithm. If for some reason the line edges are not located, the line width is linearly interpolated to the closest line points width.

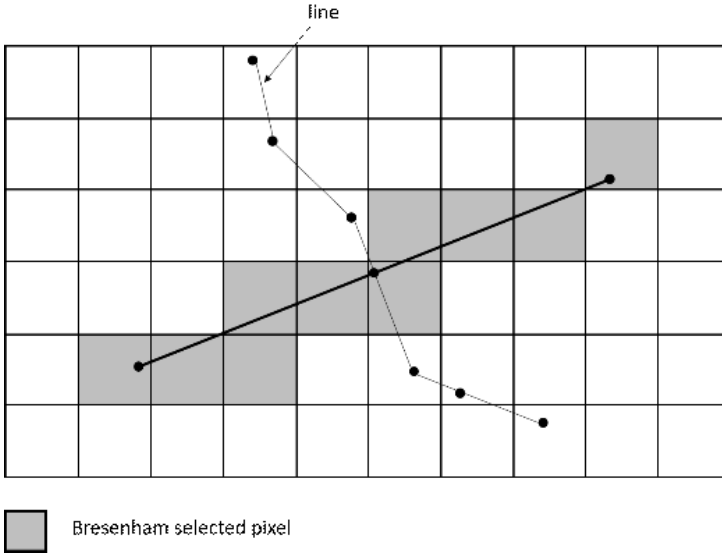


Figure 3.13: Bresenham algorithm used to look for pixels perpendicular to the line until a distance of  $2.5\sigma$

### Bias removal for asymmetrical lines

Inverting the expression  $r''(e_l, \sigma, w, a) = r''(e_r, \sigma, w, a) = 0$  founded in the previous step, it is possible to obtain the true value of  $w$  and  $a$ , assuming that  $w$  and  $\sigma$  are not independent. For this reason,  $\sigma$  value can be fixed to 1 and use values normalized to  $\sigma$ . In order to determine the optimal  $w$  and  $a$  value, more information is needed to solve the equation. Other information can be given by the ratio  $r = |r'(e_l, \sigma, w, a)|/|r'(e_r, \sigma, w, a)|$  of the gradient at detected edges  $e_l$  and  $e_r$ , which can be observed in the image. Then, since the equations are not analytically solved, a root finding algorithm is used to create an inverse function in order to determine  $w$  and  $a$ , once given the edge location and the gradient ratio. Steger in his algorithm pre-calculate some selected values of this function and interpolation is used to obtain other values during the bias removal step [84].

### Ridge detector algorithm testing on lab-scale crack scenario

As largely described in chapter 3.1.1, Steger algorithm can measure line pixel width; therefore, it is possible to use this algorithm for detecting and evaluating crack width in images. The evaluation of the width  $w$  depends on several parameters, including Lower  $l$  and Upper  $u$  threshold (hysteresis threshold) and  $\sigma$  value of the convolved Gaussian kernel. While the first two parameters depend on the line contrast  $h$  that can be estimated through the histogram associated

### 3.1 An optimized ridge detector algorithm for crack identification

to the intensity distribution of the image, the  $\sigma$  value strongly depends on the crack width itself. As described in 3.1.1, salient lines are defined by their second directional derivative:

$$r''(x, \sigma, w, a, h) = h(g''_{\sigma}(x) * f_a(x, w, a)) = h(g'_{\sigma}(x+w) + (a-1)g'_{\sigma}(x-w)) \quad (3.37)$$

Assuming zero asymmetry, Steger limitation 3.25 and remembering that  $w$  is the half of line width, is possible to write up the following expression:

$$r''(w, \sigma, h) = h(g'_{\sigma}(2w)) \quad (3.38)$$

$$r''(w, \sigma, h) = h \left( \frac{-2w}{\sqrt{2\pi}\sigma^3} e^{-\frac{2w^2}{2\sigma^2}} \right) \quad (3.39)$$

$$r''(\sigma, h) = h \left( \frac{-2\sqrt{3}\sigma}{\sqrt{2\pi}\sigma^3} e^{-\frac{2(\sqrt{3}\sigma)^2}{2\sigma^2}} \right) \quad (3.40)$$

$$r''(\sigma, h) = h \left( \frac{-2\sqrt{3}}{\sqrt{2\pi}\sigma^2} e^{-3} \right) \quad (3.41)$$

Then  $\sigma$  value and the line contrast  $h$  can be plugged into the above equation to yield an upper threshold  $u$  for the algorithm. The lower threshold can be estimated as a fraction of upper threshold [84]. The  $\sigma$  value cannot be known a-priori, nevertheless the  $\sigma$  value can be estimated.

The algorithm is tested on lab-simulated cracks through a dedicated setup (Figure 3.14). A camera is fixed on a tripod and an artificial crack is realized with two plastic plates on the mobile and fixed elements of a micro-metric stage (Newport 3-axis motion controller ESP300, uncertainty  $\pm 0.01mm$ ). The setup is illuminated homogeneously, for avoiding illumination gradient problems. The distance between camera and crack was fixed to 500 mm. The camera used in this setup is a 24mpx Nikon D7200 equipped with 60mm 2.8 Nikon Macro Lens. The width of the artificial crack is varied through the sliding stage and 30 images were acquired varying the crack width between 0.3 and 3.5 mm, with 0.1 mm steps in one direction and then in the other.

Two pose estimation markers were used to find pixel-to-mm conversion constant. The process of pose estimation is based on looking for correspondences between points in the real environment and their 2D image projection were considered and binary square fiducial markers were used for this purpose; in particular, the ArUco markers were employed. The Augmented Reality library from the University of Cordoba [6, 7] (ArUco) module is based on the ArUco library, which was developed by Rafael Muñoz and Sergio Garrido [6, 7]. They are synthetic square markers with a wide black border (to detect the markers

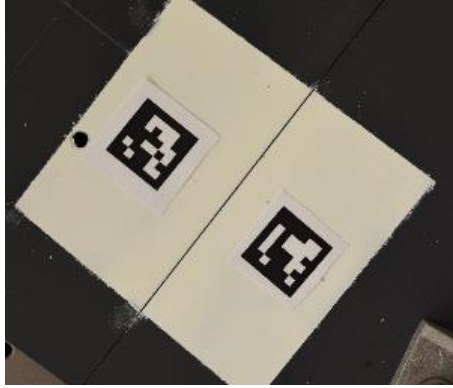


Figure 3.14: Artificial crack used for ridge detector algorithm testing on lab-scale scenario

position in the image) and an inner binary matrix (to identify each marker). Two ArUco markers were applied to the setup of the artificial crack, two for each side. Since the marker's dimensions were known, the pixel-to-mm conversion constant of the image was derived, measuring the average length of each side of the markers. The workflow used for the ridge detector algorithm testing on lab-scale scenario is described in Figure 3.15. As starting input for Steger's line detector, a fixed set of parameters (lower and upper thresholds of pixel eigenvalues and  $\sigma$  value) was considered. At first, the two ArUco markers were detected in order to compute the pixel-to-mm conversion constant. The laser triangulation measurement system Gap and Flush laser triangulation system [8] (G3F) for measuring artificial crack gap is used as reference [8]. All these steps were repeated with all image set that show an increasing and decreasing crack aperture values in order to evaluate the algorithm performances at different crack apertures. The artificial crack can be detected by the Steger's algorithm, which also can measured its mean width in pixel that can be converted in mm with the conversion constant found before.

Steger's algorithm was applied to the 30 collected images following the procedures explained in Figure 3.15. While lower and upper threshold parameters can be calculated with 3.41 because depend only on image contrast  $h$ , that is constant in each image, the choose of the correct  $\sigma$  parameter depend only on crack aperture. For this reason, the algorithm is tested on the 30 images multiple times using different  $\sigma$  values ( $\sigma = 3, 7, 15, 23$ ). Figure 3.16 shows the comparison between width measured with Steger algorithm (green dots), G3F measurement system (red dots), and slider position (blue dots). Lower values of  $\sigma$  are suitable for thinner cracks, and higher values of  $\sigma$  are suitable for thicker cracks. It is easy to note that it is not possible to well approximate any crack mean width using a fixed  $\sigma$  value without knowing a priori the order

### 3.1 An optimized ridge detector algorithm for crack identification

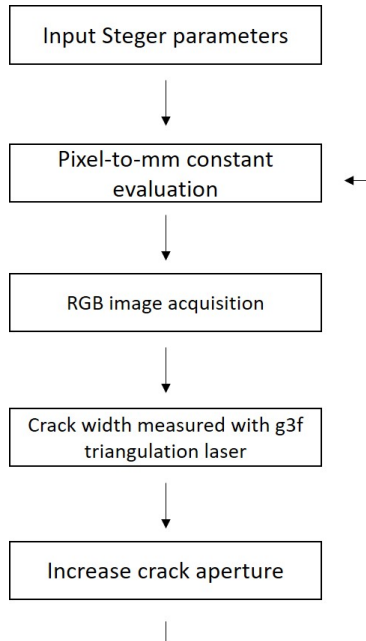


Figure 3.15: Workflow used for ridge detector algorithm testing on lab-scale scenario

of magnitude of the crack.

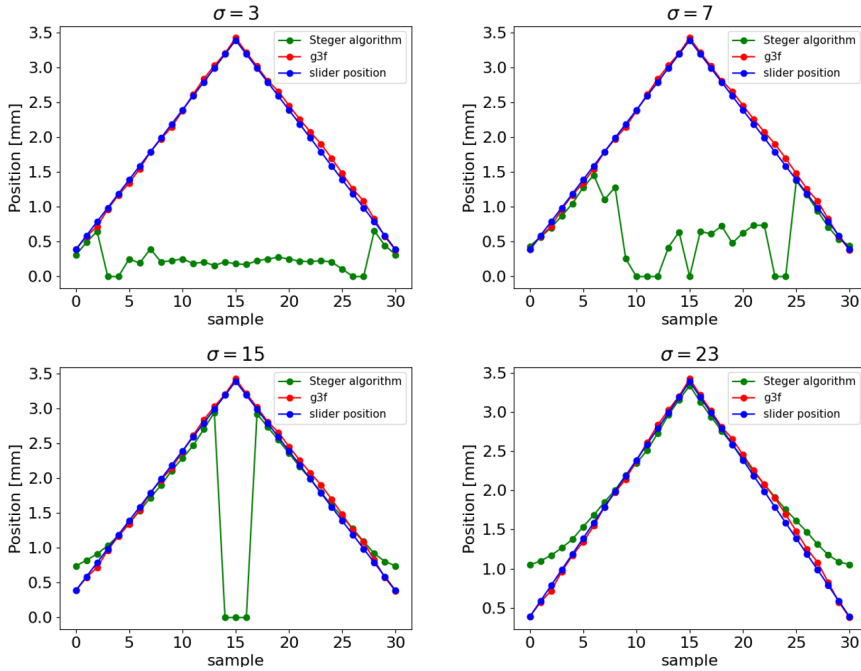


Figure 3.16: Steger algorithm used to measured artificial crack width varying  $\sigma$  input parameters: comparison among Steger algorithm (green dots), G3F measurement system (red dots), and slider position (blue dots) for different  $\sigma$  values ( $\sigma = 3$  (a),  $\sigma = 7$  (b),  $\sigma = 15$  (c),  $\sigma = 23$  (d)).

In order to better understand its behaviour, the algorithm was applied to synthetic lines with known width and asymmetry. Blur filter was applied to the generated synthetic lines, hence adding noise and making the image more realistic (3.17). Steger's algorithm was applied at the same image by varying  $\sigma$  value and fixing upper and lower thresholds, according to the workflow shown in Figure 3.18, using  $\sigma$  as space-scale parameter.

A synthetic line is generated with known width and blur. The line contrast  $h$  is known, so the upper and lower threshold parameters can be calculated analytically with 3.41 at each  $\sigma$ . Starting from  $\sigma = 1$ , Steger line detector is applied to the image and the mean half width  $\hat{w}$  of the detected line is stored. These steps are repeated increasing each time the  $\sigma$  value until  $\sigma = \sigma_{max}$ , where  $\sigma_{max}$  is determined by the operator. The mean line width  $\hat{w}$  value calculated by Steger's algorithm by varying  $\sigma$  value from 1 to 30 is shown in Figure 3.19. The calculated  $\hat{w}$  at each step  $\hat{w}(\sigma_1), \hat{w}(\sigma_2), \hat{w}(\sigma_3), \dots, \hat{w}(\sigma_{30})$ , is represented by the blue dots. The blue line represents the target width value (the algorithm



### 3.1 An optimized ridge detector algorithm for crack identification

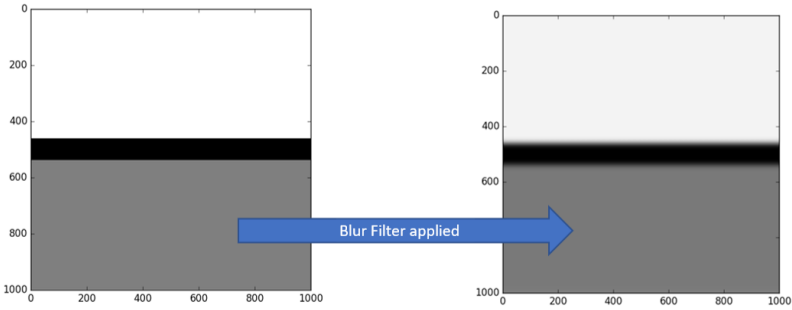


Figure 3.17: Synthetic line with known width and asymmetry (left); synthetic line on the left blurred with blur filter to get a more realistic representation of the image (right).

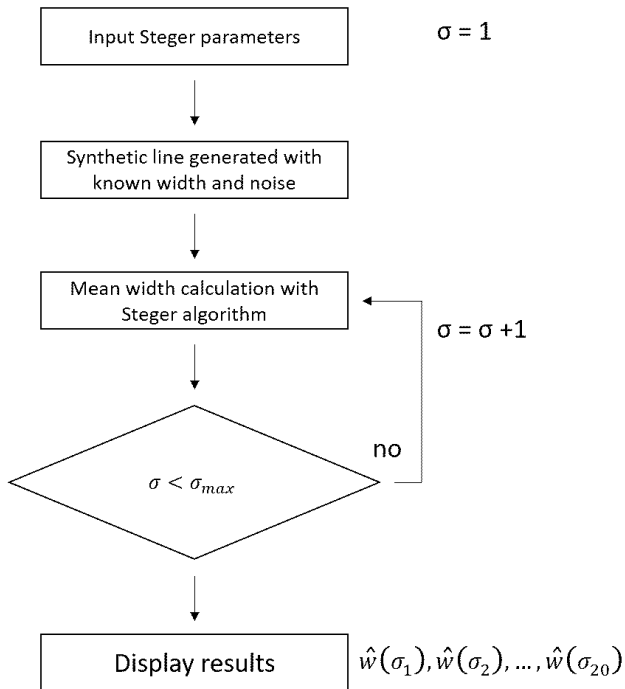


Figure 3.18: Flow chart used for  $\hat{w}(\sigma_i)$  behaviour examination in order to find a proper  $\sigma$  value target line

calculates  $w$ , that is half line thickness). The red line determines the Steger’s algorithm upper boundary given by 3.25, where the second derivative should take its maximum; under that line the  $\sigma$  boundary is respected. It easy to notice that the measured  $\hat{w}(\sigma_i)$  closer to this line is also the closest to the target values. In this way, it is possible to determine line width without knowing the line thickness a priori, applying the algorithm several times on the same image using  $\sigma$  value as a scale-space parameter. As shown in Figure 3.19, by varying blur noise condition or increasing synthetic line width, the previous considerations do not change.

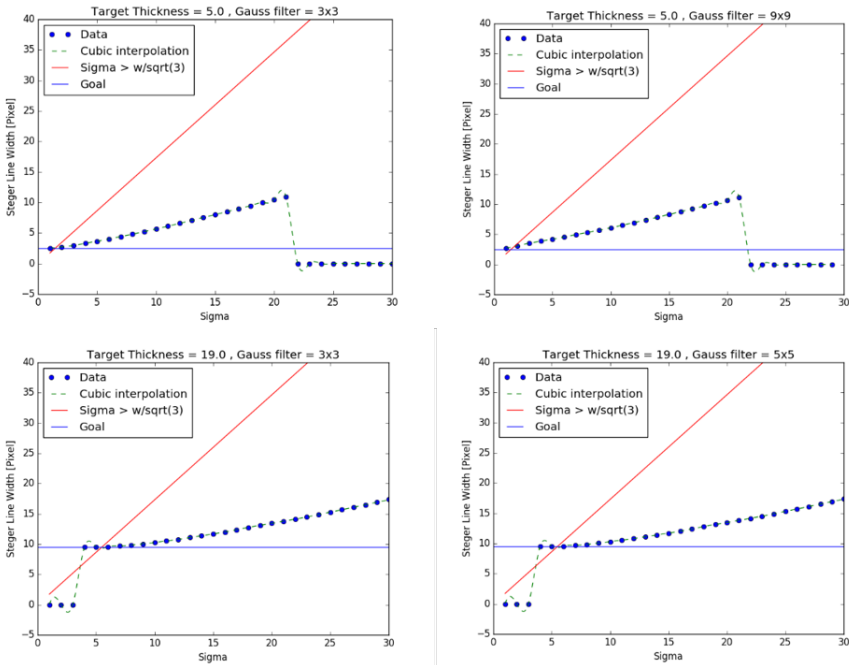


Figure 3.19: Flow chart used for  $\hat{w}(\sigma_i)$  behaviour examination in order to find a proper  $\sigma$  value target line

Observing the obtained results, it is easy to notice that increasing  $\sigma$  value will make  $\hat{w}(\sigma_i)$  increase as well. The red circle in Figure 3.20 identifies the  $\hat{w}(\sigma_i^*)$  that well approximates crack width (i.e. half of the target, 11.5 in this example) where  $\sigma^*$  is the value that led Steger algorithm to obtain the  $\hat{w}(\sigma_i)$  that better approximates the target line width (i.e.  $\sigma^* = 5$  in the considered example).

Looking at Figure 3.21, it is possible to make two main observations:

Therefore, it is necessary to find a methodology suitable to find  $\sigma^*$  based on  $\hat{w}(\sigma_i)$  trend that can led good results whatever the thickness value is. The

### 3.1 An optimized ridge detector algorithm for crack identification

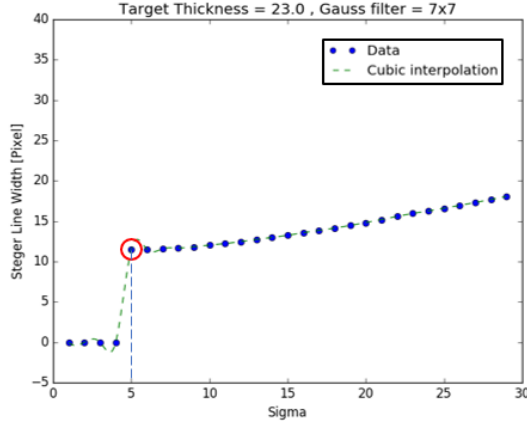


Figure 3.20: The red circle identifies the  $\hat{w}(\sigma_i)$  that best fit target line thickness (remember that line thickness is  $2w$ ).

identification of a correct  $\sigma$  value is of key importance in assessing the correct line width. A novel methodology was defined to well estimate the optimal  $\sigma^*$  value. The  $\sigma$  value that minimizes the error between  $\hat{w}(\sigma)$  and target  $w$  is always located on the Knee Location  $K_l$  of the  $\hat{w}(\sigma) = \hat{w}$  curve, where the width changes fast its value (Figure 3.21).

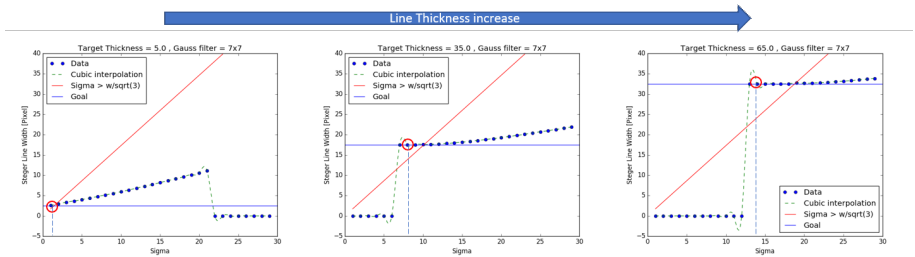


Figure 3.21: The red circle identifies the  $\hat{w}(\sigma_i)$  that best fit target line thickness. Increasing line thickness, the intersection between  $\sigma$  restriction (red line) and measured average width  $\hat{w}(\sigma_i)$  trend (green dashed line) is no more representative of  $\sigma^*$  location.

#### 3.1.2 Optimized ridge detector algorithm

In section 3.1.1 has been proven as the identification of a correct  $\sigma$  value is of key importance in assessing the correct crack width. This section describes a novel methodology to well estimate the optimal  $\sigma^*$  value. From the previous examples, it is easy to notice that the  $\sigma$  value that minimizes the error between  $\hat{w}(\sigma)$  and target  $\hat{w}$  is always located on the Knee of the  $\hat{w}(\sigma) = \hat{w}$  curve, where

$\sigma_{start}$	$\sigma_{end}$	$\sigma_{delta}$	$e_k$ [px]	$e_S$ [px]
1	30	1	0.017	0.202
1	30	2	0.018	0.203
1	30	3	0.034	0.250
1	30	4	0.059	0.250
1	30	5	0.099	0.250
1	30	6	0.153	1.188
1	30	7	0.211	1.200

Table 3.1: Comparison among  $\sigma$  values obtained by knee location and Steger domain intersection

the width changes fast its value. For this purpose, a set of synthetic images are generated with line width from 2 to 92 pixels with a step of 2 pixel (45 images in total) and fixed 5 X 5 gaussian blur. In Table 3.1 the comparison in terms of mean absolute error (45 cases in total), use of  $\sigma$  values corresponding to Knee Location  $K_l$  and  $\sigma$  values corresponding to the intersection with Steger lower boundary is shown. Seven cases are considered varying the  $\sigma_{span}$  values from 1 to 7. The use of the  $\sigma_{knee}$  values corresponding to the Knee Location strongly minimizes the mean absolute error in all the considered cases.

Figure 3.22 shows 5 measures superimposed at different thickness (2, 12, 22, 32 and 42 pixels); the  $\sigma$  value is varied from 1 to 30 in a step of 0.5. Blue lines represent the true  $\hat{w}$  values, red lines are the linear interpolation of the measured  $\hat{w}(\sigma_1)$ ,  $\hat{w}(\sigma_2)$ ,  $\hat{w}(\sigma_3)$ , ...,  $\hat{w}(\sigma_{60})$ , and green dots are the knee position at each case considered. It is easy to note that the knee position, that is also the minimum error location, seems to follow a linear trajectory.

Figure 3.23 shows the linear interpolation of the knee positions for 45 images with different line thicknesses. The resulting line can be considered as an optimization line hosting those  $\sigma$  values minimizing the error between the actual and the estimated width value. Now it is possible to determine the optimum  $\sigma$  value,  $\sigma_{opt}$ , providing the minimum absolute error.

For computational time reason, it is not practical to apply Steger algorithm 60 times on the same image (one time for each  $\sigma$  on the set). As shown in Table 3.2, the usage of a  $\sigma$  span of 0.5 between 1 and 30 (60 repetitions of Steger algorithm) means approximately 60 s of computational time (Core i5-655K).

The number of different  $\sigma$  was therefore reduced to 5 and the optimisation curve found in the previous step was employed to find  $\sigma_{opt}$ . The Steger's algorithm was applied 5 times (Figure 20) with a  $\sigma_{span} = 6$  ( $\sigma = 1, 7, 13, 19, 25$ ). The  $\hat{w}(\sigma_i)$  values obtained are plotted (red dotted line); then, the knee is located (Point 1). Obviously, the knee location cannot be part of optimization line previously found because before a  $\sigma$  span of 0.5 was used. The knee location

### 3.1 An optimized ridge detector algorithm for crack identification

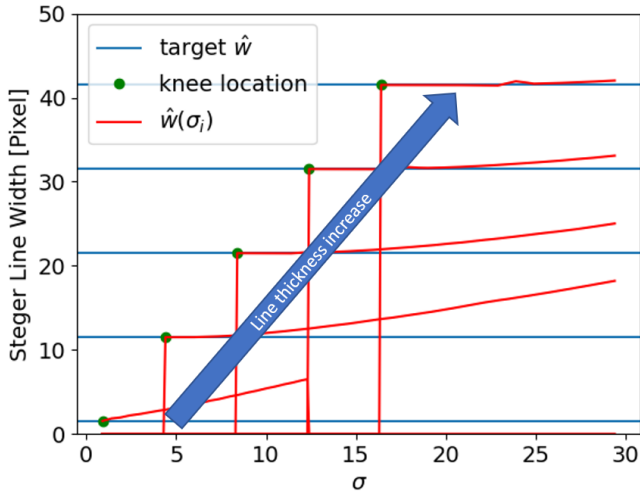


Figure 3.22: Five measurement with lines at different thickness are superimposed in order to observe how the knee position change according to the line width.

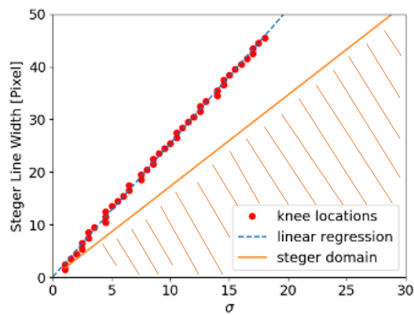


Figure 3.23: Linear interpolation of knee position for 45 lines images with thickness from 2 to 92 pixel. The obtained line localizes the  $\sigma_{opt}$  at each line thickness.

$\sigma_{start}$	$\sigma_{end}$	$\sigma_{delta}$	Computational Time [s]
1	30	0.5	60
1	30	1	34
1	30	2	19
1	30	3	13
1	30	4	9
1	30	5	8
1	30	6	7
1	30	7	6

Table 3.2: Computational time for a  $\sigma$  span of 0.5 between 1 and 30

is then projected on the optimization line (point 2) and the  $\sigma_{opt}$  is obtained. In the end, Steger algorithm is applied to the image with  $\sigma_{opt}$  parameter.

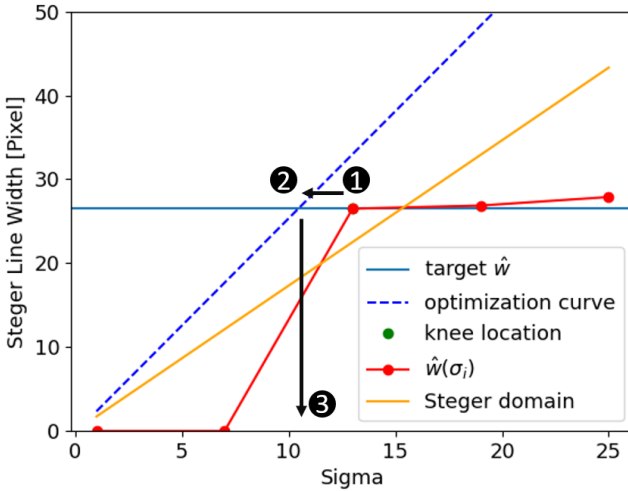


Figure 3.24: Steps used in order to find  $\sigma_{opt}$  using a few numbers of different  $\sigma$  (5 in this case) based on the optimization line previously obtained.

The introduction of this new methodology involves significant improvements in terms of absolute error, as shown in Table 3.3. The  $e_o$  associated with  $\sigma_{span} = 7$  is three times lower than the  $e_k$  at  $\sigma_{span} = 1$  ( $e_k = 0.017[px]$  at  $\sigma_{span} = 1, e_o = 0.006[px]$  at  $\sigma_{span} = 7$ ).

This procedure is then extended to other types of lines besides the horizontal one: V-shaped line, diagonal line, and circle line (Figure 3.25). Results of this experiment are reported in Table 3.4. Even in this case, the method correctly locates line position and well estimates its width.

Finally, the method was applied to the real artificial crack used in the previous

### 3.1 An optimized ridge detector algorithm for crack identification

$\sigma_{start}$	$\sigma_{end}$	$\sigma_{delta}$	$e_k$ [px]	$e_o$ [px]
1	30	1	0.017	0.001
1	30	2	0.018	0.001
1	30	3	0.034	0.001
1	30	4	0.059	0.002
1	30	5	0.099	0.002
1	30	6	0.153	0.004
1	30	7	0.211	0.006

Table 3.3: The table shows the variation of Knee and Optimization error due to the increase of  $\sigma_{delta}$

	$\sigma_{start}$	$\sigma_{end}$	$\sigma_{delta}$	$e_k$ [px]	$e_o$ [px]
V shaped	1	30	5	0.13	0.09
Diagonal	1	30	5	0.14	0.07
Circle	1	30	5	0.13	0.07

Table 3.4: Optimization error for v-shape, diagonal-shape and circle-shape lines

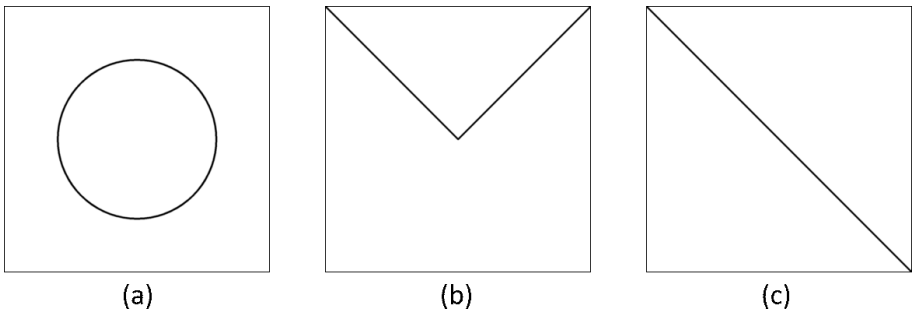


Figure 3.25: Circle line (a), v line (b) and diagonal line (c) used to validate the width estimation optimization methodology

$\sigma_{start}$	$\sigma_{end}$	$\sigma_{delta}$	$e_k$ [mm]	$e_o$ [mm]
1	30	2	0.011	0.007

Table 3.5: Optimization error for real crack case

chapter. The width of the artificial crack was varied through the sliding stage using the same procedure described in the previous chapter. The results are reported in terms of absolute error with a reference crack mean width of 0.40 mm in 3.5.

The width of the artificial crack was varied through the sliding stage and 30 images were acquired varying the crack width between 0.3 and 3.5 mm with 0.1 mm steps in one direction and then in the other. For each step, an image was acquired, and the optimization procedure was applied (Chapter 3.1.1). The comparison between the crack mean width measured by the proposed algorithm (green line) and G3F laser triangulation system is reported in Figure 22; the mean absolute error was equal to 0.02 mm and 0.06 mm with and without the proposed optimization technique, respectively.

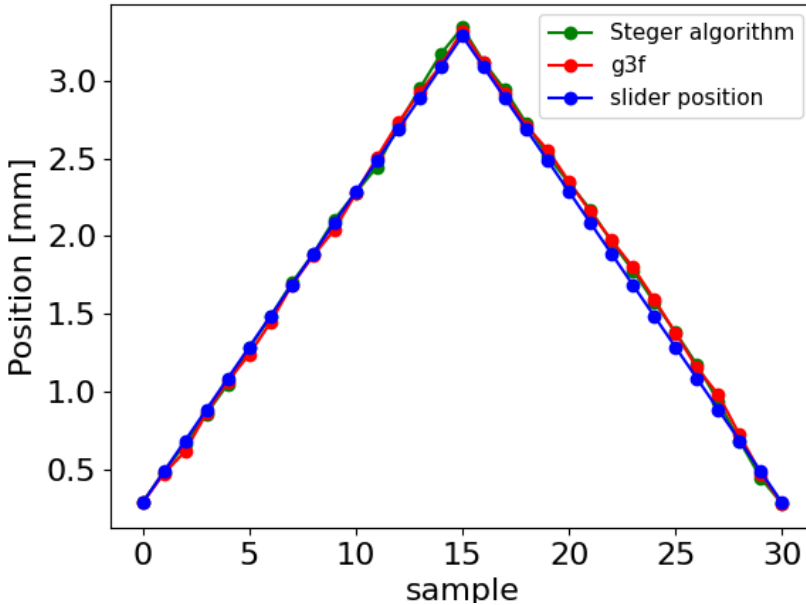


Figure 3.26: Performance of proposed algorithm compared to G3F triangulation system



### 3.1.3 Open issues

In conclusion, This chapter discussed the use of a ridge detector approach to locate a crack and measure its width. It has been proved that the Steger Ridge Detector, combined with the new developed parameters optimization technique, is able to locate and measure lines and crack on lab-scale scenario with high accuracy and low computational time comparable to more established measurement system based on laser triangulation. But unresolved issues remain:

1. In the Sigma Optimization procedure is taken for granted that Steger algorithm is able to located the entire line with every value of  $\sigma$ . In a real-scale scenario this may not happen due to the complexity of the crack and the variation of with  $w$  and contrast  $h$  along the line.
2. In a real-scale scenario is not possible to know a-priori line contrast  $h$  because the histogram can be easily affected by disturbing elements.
3. In a real-scale scenario there is no guarantee that during the Sigma Optimization procedure the algorithm focuses on the same crack: varying  $\sigma$  value it is easy for the located crack to change making the entire procedure pointless.
4. In a real-scale scenario there is no guarantee that the ridge detector will focus only on target crack. The presence of disturbing element or other cracks will make the entire procedure pointless.
5. The established procedure is applied to the entire image resulting in high processing times. Even a computational time of 6s ( $\sigma_{span} = 7$ ) may be too long for several applications.
6. The ridge detector is not able to distinguish cracks from scratches or other disturbing elements.

From all these observation comes the need for a technique able to narrow the working area around the target crack in order to reduce computational time and avoiding wrong detections. In this way there will be also possible to calculate precisely the line contrast based on local image histogram. It can be concluded that the developed Sigma Optimization Technique as it is brings remarkable improvements in terms of line width measurements and resolves the problem of algorithm parameters selection in lab-scale scenarios. Nevertheless the technique presents strong limitations if applied to a real-scale scenario.

## 3.2 AI-based preprocessing for cracks isolation

Chapter 3.1 proves that Ridge Detectors are valid algorithms to identify and measure crack width (3.27). For highly featured images in which cracks as well

as other elements (e.g. objects, scratches on the wall, ...) are present (see, for instance, 3.28 a), ridge detection algorithms fail to detect cracks location and width (3.28 b), hence it is necessary to find an autonomous technique to isolate the areas containing the crack directly on the image.

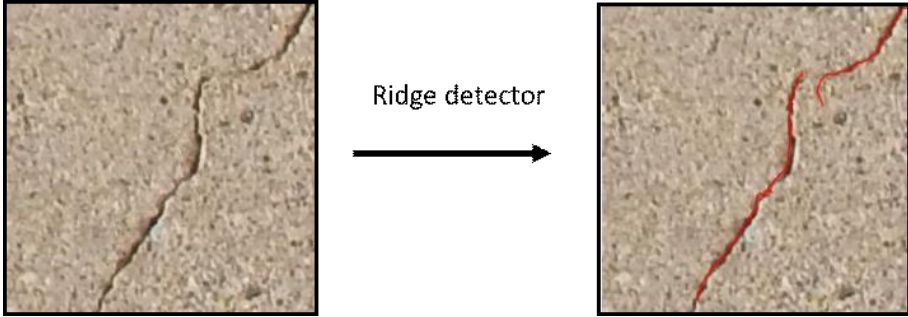


Figure 3.27: Ridge detector well detect cracks as curvilinear structures.

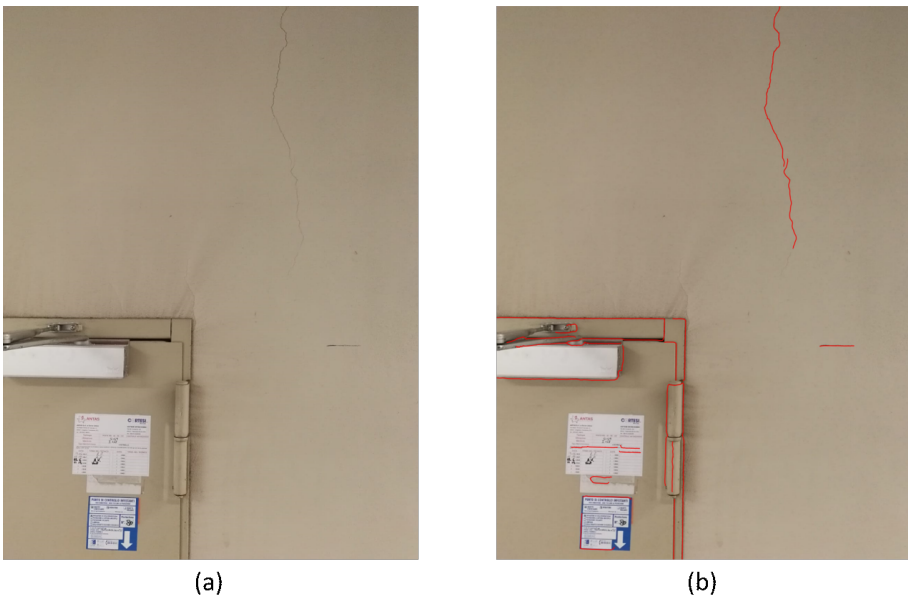


Figure 3.28: The ridge detector algorithm cannot distinguish between wall cracks, scratches and other disturbing elements that can be included in a real image.

The key idea discussed in the following is to exploit Neural Networks to create a procedure able to improve crack detection. After proper training, the network can autonomously detect regions of interest (Region of Interest (ROI)s) containing only real defects, i.e. cracks. Once the crack is roughly isolated in

the image, its correct position can be identified by the ridge detector algorithm presented in the previous section. This strategy avoids measurement errors and optimizes computational costs, given the reduced size of the image on which ridge detection is performed. Four different neural approaches were investigated:

1. Convolutional Neural Networks (CNNs) for image classification.
2. Convolutional Neural Network (CNN) sliding windows for image segmentation.
3. CNN to Fully Convolutional Neural (Fully Convolutional Neural Network (FCN)) Networks for image segmentation.
4. U-NET for image segmentation.

### 3.2.1 Convolutional Neural Network for image classification

CNN architecture is very similar to classic Neural Networks. Like the more ordinary Multi-layer Perceptron (Multi-layer Perceptron (MLP)) [86] is a supervised learning algorithm that learns a function  $f(\cdot) = R^i \rightarrow R^j$  by training on a dataset, where  $i$  is the number of dimensions for input (image dimension) and  $j$  for the output (number of classes). The basic component of the network is the Artificial Neuron, a simple operation unit with weighted input and biases, which produces output through an activation function that fixes the output value boundaries and determines the neuron activation threshold. All these Artificial Neurons are arranged into layers of neurons and all the Neurons are connected to all the Neurons of the next layer as Fully Connected layers. In addition to this, in the CNN architecture, convolutional and pooling operations are applied to the input image to generate an output to transfer to one or more fully connected layers. The output prediction is then obtained at the end of the network, where an activation function (e.g. SoftMax [87]) is used. A CNN algorithm typically works in two main phases. Starting from a complex input (e.g. images), passing through several pooling and convolutional layers, features are extracted. Then a classification is performed based on the previous feature learning phase. A CNN mainly involves the following operation: convolution layers, pooling layers, fully connected layers [88]. The convolutional layers convolve inputs with learnable filters (kernels), generating a feature map for each applied filter. Each feature map is joined together into a volume obtaining the output of the convolutional layer. The kernel has specific dimensions, stride, and padding (typical zero padding) and determines the output dimension. The stride is the parameter that controls how the filter convolves around the input volume. The convolution output volume dimension follows the expression 3.42

$$O = \frac{W - K + 2P}{S} + 1 \quad (3.42)$$

where  $O$  is the output volume dimension,  $W$  is the input dimension,  $K$  is the filter size,  $P$  is the padding, and  $S$  is the stride. After each convolutional layer, there is an elementwise activation function (e.g. Rectified Linear Unit,  $\max f(x, 0)$ ) in order to add non-linearity in the feature map [89].

The most frequent pooling technique involves max and average pooling. The pooling layer always reduces the dimension of the input feature map from the convolutional layer. The input is divided into subregions and each is replaced by its average (average pooling) or max (max pooling) value. Pooling layers lead a spatial subsampling of the input feature, making the results more robust to changes in the position of the feature in the image (e.g. crack location) [90]. The pooling layer adds the so-called propriety of invariance to local translation to the model. In the end, the last layer is flattened into a Fully Connected layer that will feed the classifier. The most famous classifier is the Soft-max function, that returns the probability value of belonging to a certain class given a certain input [91].

For the purpose of this work, the CNN VGG-19 architecture is chosen as classifier. VGG-19, presented in [92], is a convolutional neural network that is 19 layers deep, with an image input size of 224-by-224. This type of architecture has been shown to be particularly efficient for classification tasks in terms of accuracy, size, and scalability. In fact, its architecture makes it possible to convert the model into more complex networks (e.g., semantic segmentation tasks) as we will see in later chapters. The network includes 13 convolutional layers with filter size of 3 X 3 and fully connected layers. The padding and the stride are fixed to 1 pixel for all the convolutional operations. As shown in 3.29, each group of Convolutional layers is followed by a max pooling layer, and the Fully Connected layers end with the Soft-max activation function. The last VGG19 FC layer size is then modified from 1000 to 2, in order to perform a binary classification (cracked and non-cracked). In order to train the

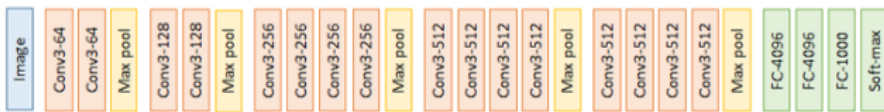


Figure 3.29: VGG19 original architecture [92]

model, a large dataset of images containing/not-containing cracks is needed. A set of 56000 images in several different scenarios (cracked and non-cracked concrete bridge decks, walls, and pavements) were collected from SDNET2018 [93, 94]. SDNET2018 is an annotated image dataset for training and validation of artificial intelligence-based crack detection algorithms for concrete. The database crack width ranges from 0.06 mm to 25 mm. The dataset includes a variety of obstructions: shadows, surface roughness, scaling, edges, holes, and

background debris [94]. This database has never been tested in the literature with the configuration described in this thesis work. The databank is split into a training set and a validation set, with a ratio of 5:1. The entire train process can be summarized as follows. The model training starts initializing all the filters and weights with random values. Then, it takes the first image from the training dataset and performs the forward computation through convolutional, pooling, and fully connected layers, returning the class output that encodes the probability for the input image of belonging to a particular class. At the first iteration, the output probability will be random. The error of the output layer is then calculated through a loss function. Since a binary classifier is modelled (cracked or non-cracked), the binary cross-entropy loss between true label and predicted label is calculated as 3.43.

$$BCE = - \sum_{i=1}^C t_i \log(s_i) \quad (3.43)$$

Where  $BCE$  is the Binary Cross-Entropy loss,  $C$  is the number of classes (2 in this case),  $t_i$  is the true label, and  $s_i$  is the predicted label [95]. The next steps are Backpropagation followed by parameter updates. The Backpropagation step calculates the gradient of the error with respect to all network filters/weights and uses the gradient descent in order to perform the parameter update minimizing the output error, by adjusting the filters/weights proportionally to their contribution to the total error. Successively, all the steps are repeated: the next image is used as input for the forward computation using the weight found with the previous image, and so on, until the training set is finished. Several hyperparameter configurations (e.g. learning rate, number of epochs, batch size, weight initialization, ...) were tested, in order to improve network performances in terms of loss, accuracy, and false positives, simultaneously avoiding overfitting [96]. The training was performed in a Python-Caffe environment obtaining 0.924 validation set accuracy and 0.213 validation set loss. While the loss is defined by the loss function, the accuracy is a metric that can be applied only to classification tasks and describes just what percentage of the test data is classified correctly (e.g. if the binary classification has 1000 test samples and the model is able to classify 900 of them correctly, then the model's accuracy will be 90.0%).

#### 3.2.2 CNN sliding windows for image segmentation

The model obtained in the section 3.2, based on VGG19 CNN, is able to predict if an image contains cracks or not. Since the model was trained with 224x224 images which contain only a small portion of concrete surfaces in different scenarios, it is not able to predict and detect the presence of defects in real

large images. One might think of applying the CNN repeatedly in a 224x224 sliding window with a target stride and a proper padding 3.30. Then, all the

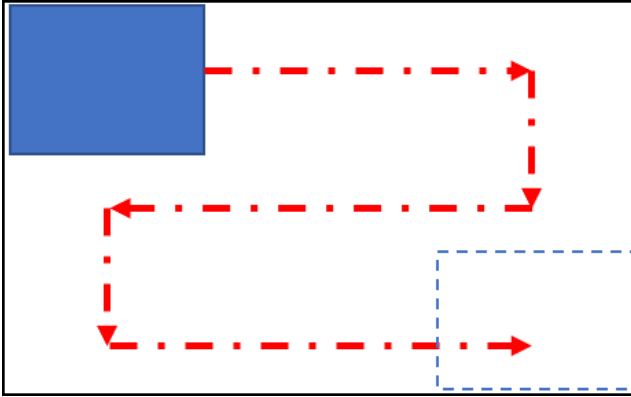


Figure 3.30: Image segmentation through CNN sliding window

windows predicted as ‘cracked’ can be used to create a mask, obtaining an image that contains only cracked regions 3.31, in order to ensure that the ridge detector works only in the proper region of interest, avoiding local disturbing elements. The problem of this approach is the high computational cost: the

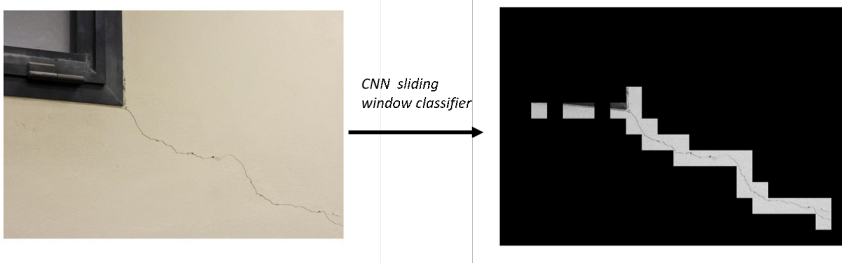


Figure 3.31: Image segmentation results through CNN sliding window technique

CNN is inferred  $N$  times, where  $N$  is the number of windows slid on the image multiplying, in this way, the time consumption. This problem can be solved through an image semantic segmentation by a convolutional implementation of the sliding window algorithm. This process allows to pass from CNN to Fully Convolutional Neural Networks.

### 3.2.3 CNN to Fully Convolutional Neural Networks for image segmentation

The CNN classification architecture has different downsides:

1. The input image dimensions are fixed by the problem.

2. The output is a discrete probability distribution that represents the dominant class in the input image.
3. The location information of the predicted class cannot be obtained except through the sliding window approach that lead with high computational time.

Model architecture that practises semantic image segmentation eliminates all these drawbacks. In order to perform a semantic segmentation with the obtained CNN weights, the Fully Connected layers are replaced with convolutional ones, converting the model into a Fully Convolutional Neural Network model. This option is preferred to CNN sliding window for computational time reasons. The entire model conversion process can be summarized as follows (Figure 3.32). Starting from a simple CNN, all the Fully Connected layers are replaced by a convolutional 1X1 kernel; in this way, 3 convolution kernels are added to the network architecture. Figure 3.32 shows an example of a simplified version of VGG19 binary classifier that, like in the case of this work, takes as input one 224 X 244 X 3 (RGB) image and returns a classification vector of length 2 (number of classes probability). At first glance nothing seems to be

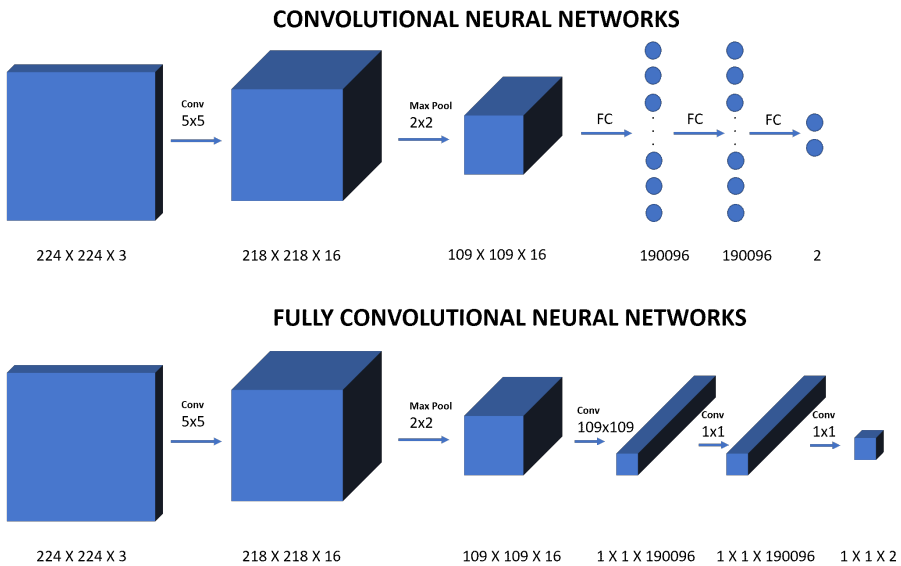


Figure 3.32: CNN architecture is converted to FCN replacing Fully Connected layers into Convolutional layers

changed; however, thanks to this alteration of the architecture, it is possible to take as input bigger size images. Applying a sliding window with a stride of 2 in a set image of size 2240 X 2240 X 3 would mean applying the CNN

1007 times ( $\lceil (2240-224)/2 \rceil - 1$  according to the previous formula), because the window will assume 1007 different positions along the image. Figure 3.33 shows that, instead of applying the sliding windows algorithm, if 2240 X 2240 X 3 image is given to the FCN network, with no change in terms of convolutional kernel dimensions, a vector 1007 X 1007 X 2 is obtained. Each element of the final vector represents the class probability obtainable by a hypothetical sliding algorithm. For example, the first red element of the output vector describes the probability value of belonging to a certain class of the region 224 X 224 X 3 which would cover the first window in the input image, the second green one to the second (with a stride of 2 due to the max pooling layer 2x2), and so on (Figure 3.33). The convolutional implementation of sliding window leads

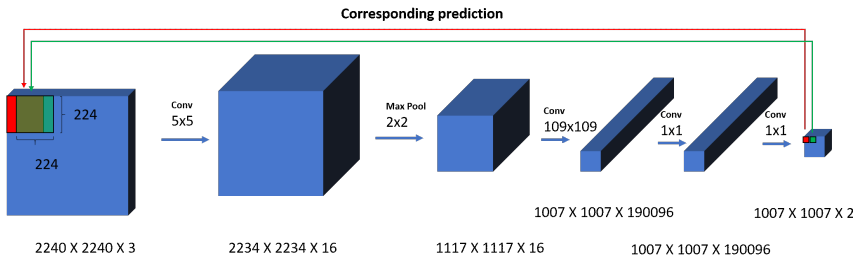


Figure 3.33: CNN architecture is converted to FCN replacing Fully Connected layers into Convolutional layers

to perform all the predictions at the same time, decreasing dramatically the computational time, hence making the CNN much more efficient. All these concepts are adopted in order to convert the modified VGG19 CNN into FCN architecture. The output tensor will be a tensor of  $N \times SV \times SH$ , where  $N$  is the number of classes and  $SH$  and  $SV$  are the number of sliding windows along vertical and horizontal axes, whose stride depends strictly on max pooling layers. When an image passes through the convolutional VGG19 model, coarse features are extracted. Since the purpose of the semantic segmentation is to perform pixel-wise prediction, all these coarse features obtained have to be reconstructed through the addition of a deconvolutional layer, which is then put on the top of FCN VGG19 model. In this way, the down-sampled response maps from the new architecture are up-sampled through this deconvolution layer, producing the feature that can be used to predict class labels at all the pixel locations. A value  $P$  from 0 (non-cracked) to 1 (cracked) is assigned to each pixel of the input image. A binary threshold on pixel-wise class label is then applied: all pixels with  $P < 0.8$  are turned into black pixels. In this way, ROIs are automatically selected in an easy way, where only cracked concrete regions are included (Figure 3.34). A number of 50 real cracked wall images were collected. For each image, two different methods were applied to the images:



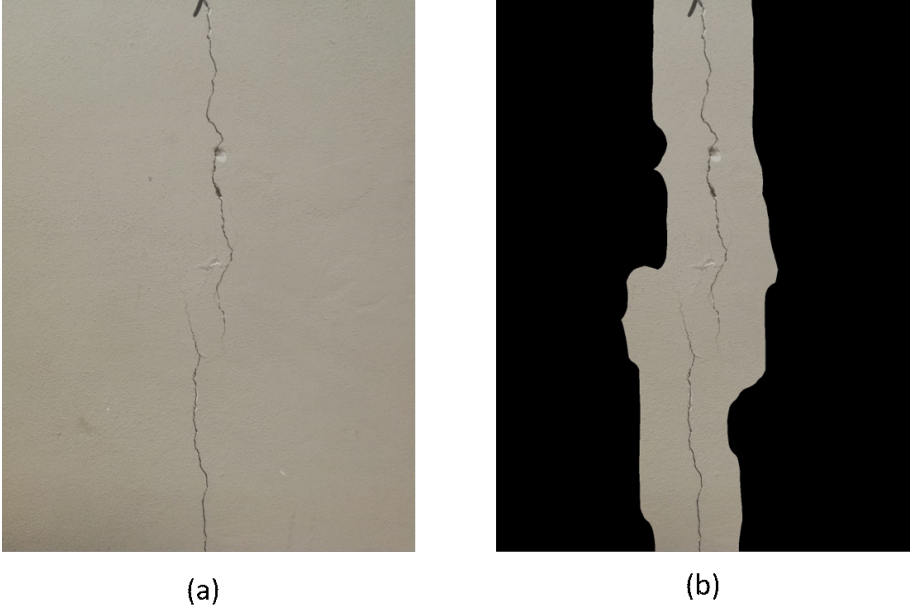


Figure 3.34: Crack ROI detection by a binary threshold on FCN output pixelwise probability value. The original image (a) and image deprived of non-cracked region (b) are reported in figure.

1. Steger line detector algorithm, where the ridge detector is applied directly to the entire image.
2. FCN VGG19 + Steger line detector algorithm, where, at first, ROI containing only cracked concrete regions are autonomously detected, then the ridge detector is applied to the masked image.

In order to evaluate the methods performances, the Accuracy and Precision are calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.44)$$

$$PRE = \frac{TP}{TP + FP} \quad (3.45)$$

where:

1. True Positive True Positive (TP) value is the number of cracks correctly individuated by the algorithm.
2. True Negative True Negative (TN) value is set to zero (i.e. non-cracked areas), because in a problem like this, the value 1 makes no sense.

Method	ACC	PRE
Steger	16%	17%
FCN + Steger	78%	85%

Table 3.6: Comparison between Steger line detector and FCN VGG19 + Steger line detector in terms of Accuracy and Precision

3. False Positive False Positive (FP) value is the number of false cracks individuated by the algorithm.
4. False Negative False Negative (FN) value is the number of cracks that the algorithm was unable to recognize.

In binary classification, the Accuracy describes how comfortable the method is with detecting the positive and negative classes, whereas the Precision describes the probability of making a correct positive classification. Accuracy and Precision are calculated for all the considered 50 cases, obtaining the results reported in Table 3.6.

As clearly shown by the results of the comparison of the Accuracy and Precision of both methods, the introduction of the automatic ROI detection increases from 16% to 78% the Accuracy and from 17% to 85% the Precision. In fact, the ridge detector used alone is not able to distinguish crack from other disturbing elements and the FCN makes up for this lack. Figure 3.35 and Figure 3.36 report a clear example where FCN is able to well-detect crack regions, even avoiding eventual wall scratches. The usage of ridge detector alone led 32 false positives in the example in Figure 3.35, and 25 false positive in the example in Figure 3.36.

### 3.2.4 UNET for image segmentation.

The model architecture involved in this step is the U-NET, an evolution of Convolutional Neural Network and stems from the Fully Convolutional Neural networks. As shown in Figure 3.37, the architecture consists contracting path (encoder), typically a series of convolution and max pooling operation that reduces spatial information and increase the feature information, and expanding path (decoder) where an up-sampling is performed allowing the network to propagate context information to high resolution layer. This architecture can accept as encoder side different model architectures. UNET encodes the image passing it through a CNN as it gets downsampled and then decode it back or upsample it to obtain the segmentation mask. According to the choose backbone, the decoder side is built by concatenating the previous output on the decoder side. Since the model does not contain Dense layers, it can accept

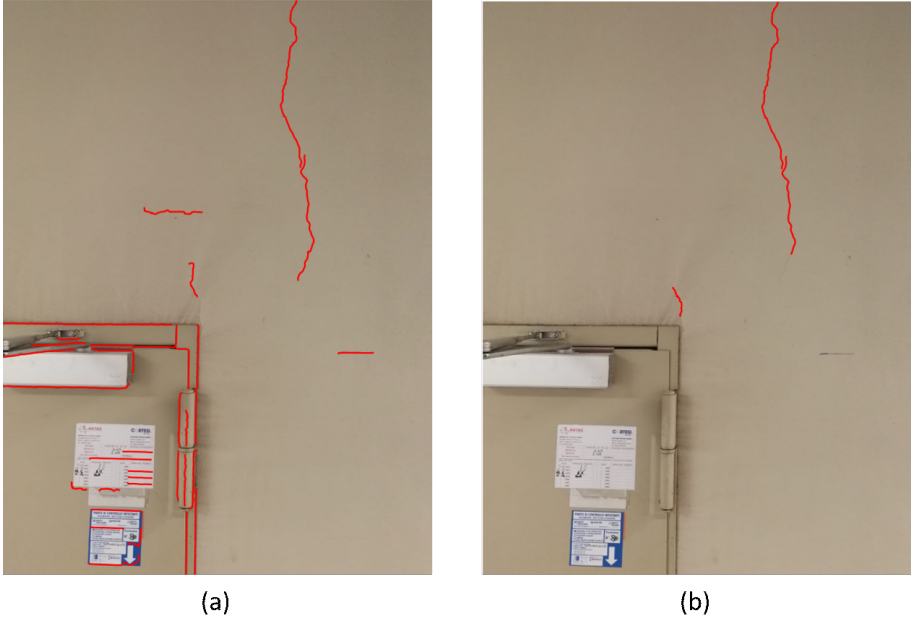


Figure 3.35: Comparison between Steger algorithm (a) and FCN + Steger algorithm (b). Wall scratches and disturbing elements are correctly excluded from ridge detection by FCN VGG19 method.

input of any size. [97]. A total of 5000 crack images in different scenarios and with different cameras (reflex and smartphones) have been taken from different operators to build an on-purpose crack database. For each crack image the corresponding binary target crack mask is realized (Figure 3.38) with the help of a graphics tablet and a specific software written in Python. This mask will represent the input ground truth for the segmentation model. Five different operators were asked to perform this operation to avoid that the model fits too much the behaviour of a single operator.

The obtained dataset is randomly divided in train, validation and test dataset. The training dataset is the sample of data used to fit the model and the validation dataset is used to provide an unbiased evaluation of a model fit to training dataset while tuning model weights. While train and validation dataset are used in the training phase, the test dataset is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset, for this reason the test dataset is never in training phase. [98, 99] analyse the influence of validation-training-test-set size ratio. Since the database of classified images is relatively small, a ratio of 0.80-0.20-0.20 of train, validation and test is chosen.

Data augmentation techniques are used in order to increase training performances. Data augmentation is a data-space solution to the problem of limited

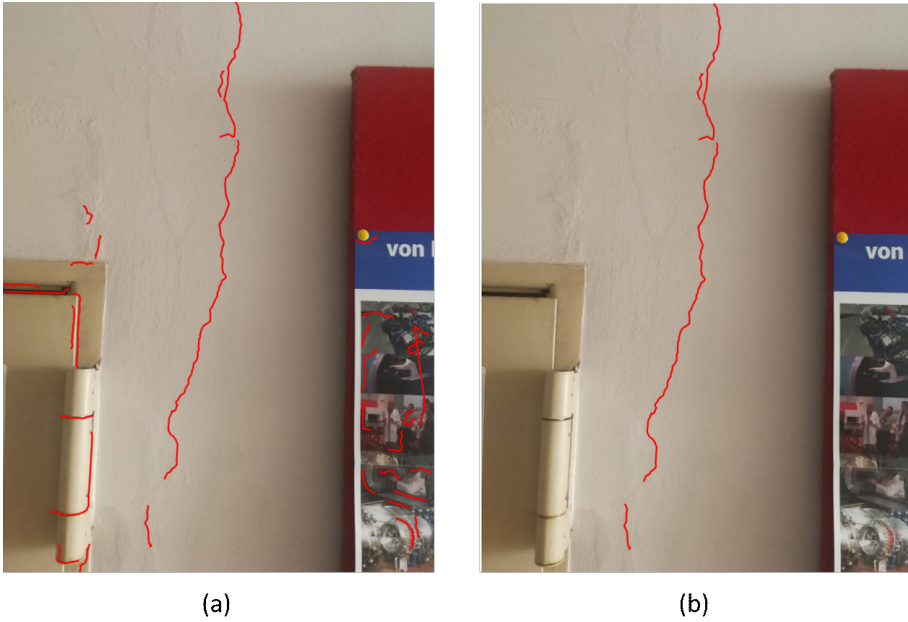


Figure 3.36: Comparison between Steger algorithm (a) and FCN + Steger algorithm (b). Disturbing elements are correctly excluded from ridge detection by FCN VGG19 method.

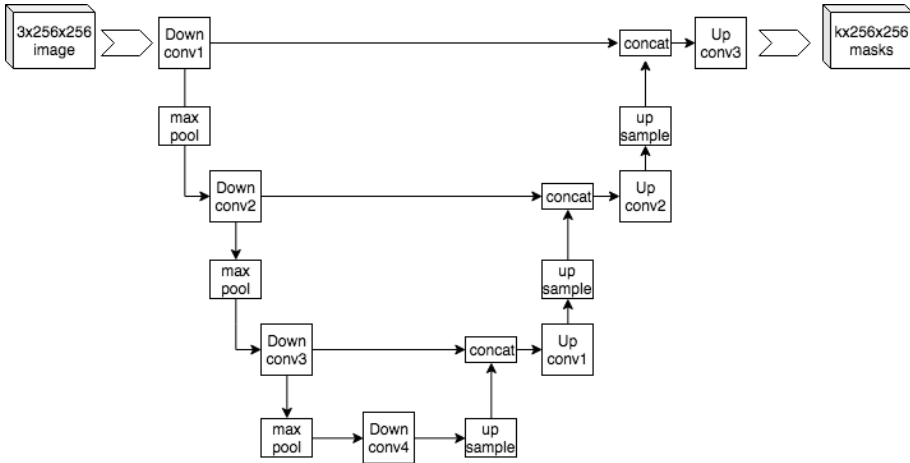


Figure 3.37: UNET typical architecture. K classes segmentation of a 256x256 RGB input image. [97]



Figure 3.38: On the left the RGB crack image, on the right the binary crack mask manually selected by an operator.

availability of data that enhance the size and quality of training datasets avoiding over-fitting, that is the situation where a network makes predictions that corresponds too closely or exactly to a particular set of data, and may therefore fail to make prediction outside training dataset [100]. It has been largely proven that a correct data augmentation can improve dramatically the performances in classification and semantic segmentation problems [101–103]. The following techniques are used for creating additional training samples:

1. Horizontal Flip, where the image is randomly flipped along horizontal axis.
2. Vertical Flip, where the image is randomly flipped along vertical axis.
3. Shift, the image is randomly shifted, the image dimensions are kept the same.
4. Rotation, the image is randomly rotated around a random point, the image dimensions are kept the same.
5. Crop, the image is randomly cropped, the image dimensions are kept the same.
6. Gaussian Noise, apply gaussian noise to the input image from 0 to 20%.
7. Perspective transform, perform a random four point perspective transform of the input, the image dimensions are kept the same.
8. Blur, blur the input image using a random-sized kernel.c
9. CLAHE, random applying Contrast Limited Adaptive Histogram Equalization [104] to the input image.
10. Brightness, randomly change brightness of the input image.

Figure 3.39 shows the mean histogram of the 5000 binary crack masks manually selected. The crack region, with assigned value of 1, represents the 0.004% of the mask where the background has an assigned value of 0. Since this segmentation problem is strongly imbalanced, the loss function must be chosen properly. Since the data-imbalanced nature of the segmentation problem

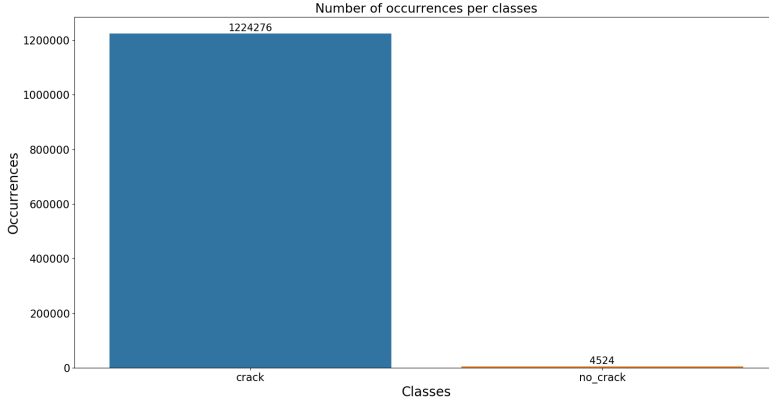


Figure 3.39: Mean histogram of the 5000 binary crack masks.

the Dice + Focal loss function is chosen for training the model [105],[106],[107]. The Dice coefficient is typically used in computer vision application where it is necessary to calculate the similarity between two images and it is largely used as metric to evaluate segmentation results. The Dice loss function is inspired by the Dice coefficient given by the equation 3.46

$$D = \frac{2 \sum_i^N y_i p_i}{\sum_i^N y_i^2 + \sum_i^N p_i^2} \quad (3.46)$$

Where  $p$  represents the pixel predicted value and  $y$  represents the ground truth. In this specific case of binary classification problem,  $p$  and  $y$  varies from 0 to 1. While the denominator is the sum of total crack pixels and always grow as the crack grow, the numerator increments its value only when the prediction and the ground truth match. The Dice Loss is defined by the equation 3.47

$$DiceLoss(y, p) = \frac{2yp + 1}{y + p + 1} \quad (3.47)$$

Where the 1 is added to ensure that the function works well even if  $y$  and  $p$  are both null. The Focal Loss derives from the Binary Cross Entropy Loss function. Its purpose is to increase the contribution and focus the model on hard example, such as find pixels belonging to crack regions, and down-weight

### 3.2 AI-based preprocessing for cracks isolation

the contribution of easy examples, such as find no-crack pixels. The Binary Cross Entropy Loss function can be written by:

$$BCE = -\log(p_t) \quad (3.48)$$

Where  $p$  is the estimated probability of a class, and it is defined by:

$$p_t = \begin{cases} p, & y = 1 \\ 1 - p, & otherwise \end{cases} \quad (3.49)$$

The Focal Loss is then defined by:

$$FocalLoss(p_t) = -\alpha_t(1 - p_t)^\delta \log(p_t) \quad (3.50)$$

Where the term  $-\alpha_t(1 - p_t)^\delta$  is called Modulating Factor and up-weights the contribution of hard samples.  $\alpha$  and  $\delta$  can be considered model hyper-parameters. Focal and Dice Loss can be summed in order to take the advantages of both techniques [105]:

$$TotalLoss = DiceLoss + FocalLoss \quad (3.51)$$

in this way is possible to train UNET model to the hard-negative examples instead of overlooking that class and getting an arbitrarily high accuracy. In order to select the proper backbone and hyper-parameters, 60 different training are performed with the following parameters:

- Backbone = ['efficientNET', 'vgg19', 'resNET', 'mobileNET', 'inceptionv3']
- batch size = [2,4,8,16]
- Learning Rate = [0.01,0.001,0.0001,0.00001]

Different CNN architecture backbones are tested.

The efficientNET was developed by Mingxing Tan et al. in 2019. the idea behind this model is simple: CNNs are developed at fixed resource budget, based on availability of resources are then scaled up for better performance. In their work is proven that balancing network depth, width and resolution can lead to better accuracy without increasing the scale of the model [108]. Efficientnet achieve state-of-art accuracy on ImageNet [109] while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet.

The resNET was developed by Kaiming He et al. in 2015. They present a residual learning framework to ease the training of networks that are substantially deeper than those used previously [110]. The network won the 1st places on the tasks of ImageNet detection, ImageNet localization, Common Objects in COntext (COCO) [111] detection, and COCO segmentation in 2015.

The mobileNET was developed by Andrew G. et al. in 2017. The model is designed for mobile and embedded vision applications and it is based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks [112].

The Inception model was developed by Christian Szegedy et al. in 2014. The main hallmark of this architecture is the improved utilization of the computing resources inside the network [113]. The network won the 1st places on ImageNet Large-Scale Visual Recognition Challenge in 2014.

When evaluating a CNN classification model the predictions is usually classified into four categories: false positives, true negatives, false negatives and true positives. This is not possible with pixel-wise predictions in segmentation models. The Jaccard Index, also know as Intersection Over Union (IOU) can be used as metric for segmentation model evaluation. This metric quantifies the percent overlap between the target mask (ground truth) and the output prediction measuring the number of pixels common between the ground truth and predicted masks divided by the total number of pixels present across both masks 3.52.

$$IOU = \frac{Ground\ Truth \cap Predicted}{Ground\ Truth \cup Predicted} \quad (3.52)$$

Another metric used for evaluation segmentation model is the Dice Coefficient, also known as F-score, already discussed in [105]. Both metrics range from 0 (no overlap) to 1 (perfect overlap). While F score tends to measure something closer to average performance, the IOU score measures evaluate the model relating to the worst case performance.

Adam algorithm is used as model weights optimizer. The Adam optimizer is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [114]. The model is trained on a machine equipped with Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and NVIDIA GPU GeForce RTX 2070 SUPER. The 60 configuration are trained with the following hyper-parameters:

- Model = According to the case, 'efficientNETb3', 'vgg19', 'resNET152', 'mobileNETv2' or 'inceptionv3'.
- Optimizer = Adam with learning rate according to the case, 0.01,0.001,0.0001 or 0.00001
- Batch Size = according to the case, 2,4,8 or 16
- Number of Classes = 2, crack and no crack.
- Epochs = 50.
- Loss Function = Dice Loss + Focal Loss



### 3.2 AI-based preprocessing for cracks isolation

model	weights	time [s]
efficientnetb3	17,867,833	0.06
vgg19	29,061,969	0.07
resnet152	67,295,194	0.10
mobilenetv2	8,047,441	0.05
inceptionv3	29,933,105	0.07

Table 3.7: Models characteristics in terms of number of trainable weights and prediction computational time in seconds calculated as the average time for 100 predictions with different input images.

Top	Batch Size	LR	model	Loss	IoU-score	f1-score
1	4	0.0001	efficientnetb3	0.011	0.992	0.996
2	4	0.0001	mobilenetv2	0.012	0.992	0.996
3	4	0.0001	inceptionv3	0.010	0.992	0.996
4	4	0.0001	resnet152	0.012	0.991	0.996
5	2	1.00E-05	resnet152	0.018	0.991	0.996
6	2	0.0001	efficientnetb3	0.011	0.991	0.996
7	2	0.0001	inceptionv3	0.012	0.991	0.995
8	2	1.00E-05	inceptionv3	0.015	0.991	0.995
9	2	0.0001	resnet152	0.013	0.991	0.995
10	4	0.001	mobilenetv2	0.013	0.990	0.995
11	4	0.0001	vgg19	0.013	0.990	0.995
12	2	1.00E-05	efficientnetb3	0.033	0.989	0.995
13	2	1.00E-05	mobilenetv2	0.023	0.989	0.995
14	4	1.00E-05	efficientnetb3	0.023	0.989	0.995
15	2	0.0001	mobilenetv2	0.015	0.989	0.994

Table 3.8: Models characteristics in terms of number of trainable weights and prediction computational time in seconds calculated as the average time for 100 predictions with different input images.

- Metrics = IOU and F score

Table 3.7 shows models characteristics in terms of number of trainable weights and prediction computational time in seconds calculated as the average time for 100 predictions with different input images. Table 3.8 shows top 15 best models ordered by IOU and F scores. The backbone mobileNETv2 with batch size of 4 and learning rate of 0.0001 is chosen due to its performance in terms of computational time under the same IoU and F scores.

In summary, the following hyper-parameters are used for the training phase:

- Optimizer = Adam, with a learning rate of 0.0001.
- Backbone = 'mobilenetv2'

- Batch Size = 4
- Number of Classes = 2, crack and no crack.
- Loss Function = Dice Loss + Focal Loss
- Metrics = IOU and F score

The model is also trained with different train database split ratio. The result, shown in figure 3.40, confirms that the ratio that maximize IoU and F score is 0.80-0.20-0.20 of train, validation and test. The model is then trained for 50 epochs obtaining a IoU score of 0.99 and f1 score of 0.99. Training the model over 50 epochs does not improve model performance 3.40 3.42 3.41. Examples of the U-NET trained model inference results on the various image from collected crack test dataset are shown in figure 3.44.

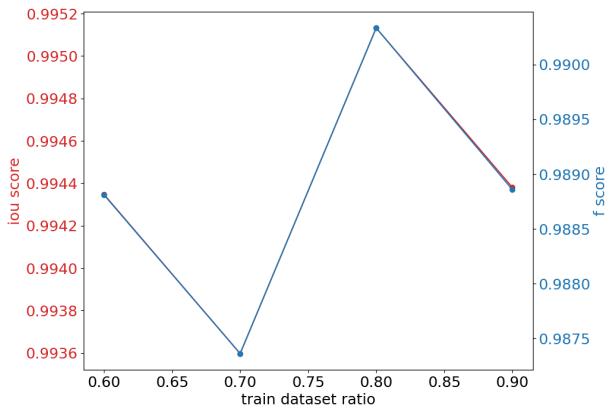


Figure 3.40: The model is trained with different train-validation-test ratio. The results confirm that the ratio that maximize IoU and F score is 0.80-0.20-0.20

In this chapter it has been seen how it is possible to identify the area of an image containing a crack surface. Through the use of most recent neural networks trained with databases collected during the thesis work, it is possible to extract the mask containing only the crack with extreme precision even in noisy images containing disturbing elements (shadows, external objects, ...). This technology will be the basis for development of a new methodology that will lead to the resolution of all the limitations of Sigma Optimization algorithm presented in chapter 3.1.2.

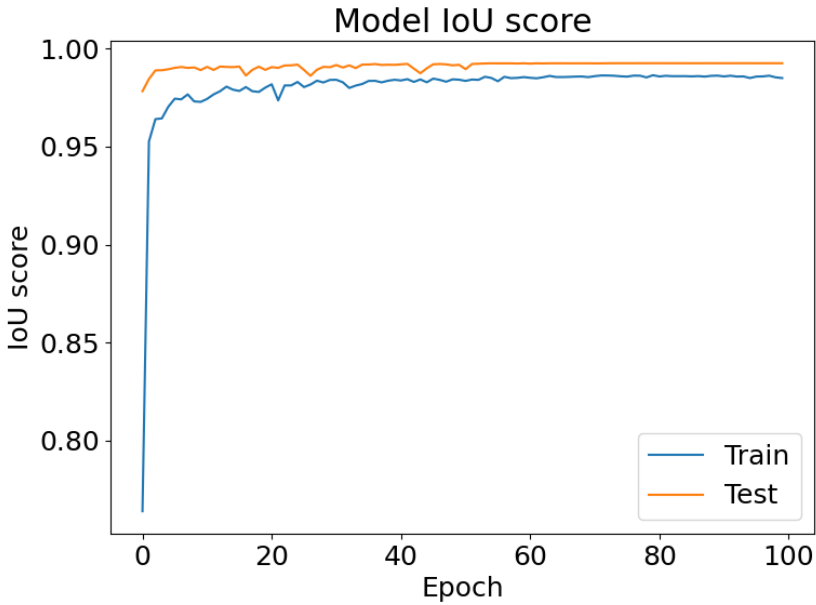


Figure 3.41: The plot shows the IoU score curve over epochs for train and test dataset. Above 50, the IoU score stabilises.

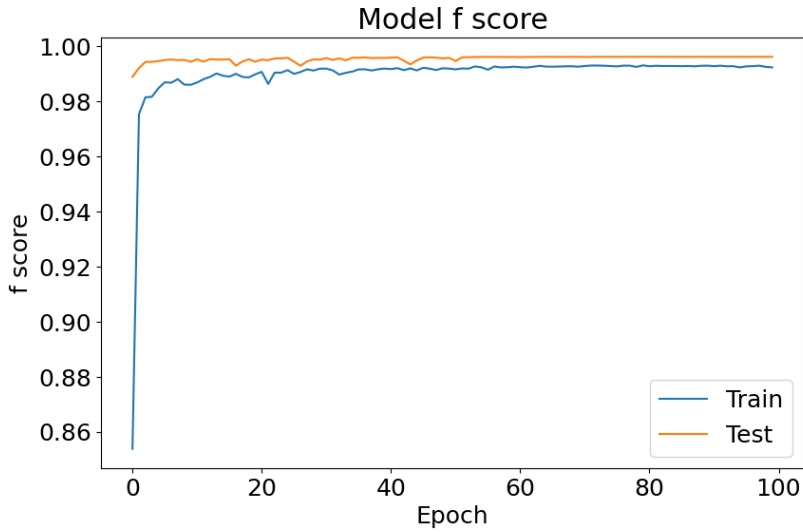


Figure 3.42: The plot shows the f score curve over epochs for train and test dataset. Above 50, the f score stabilises.

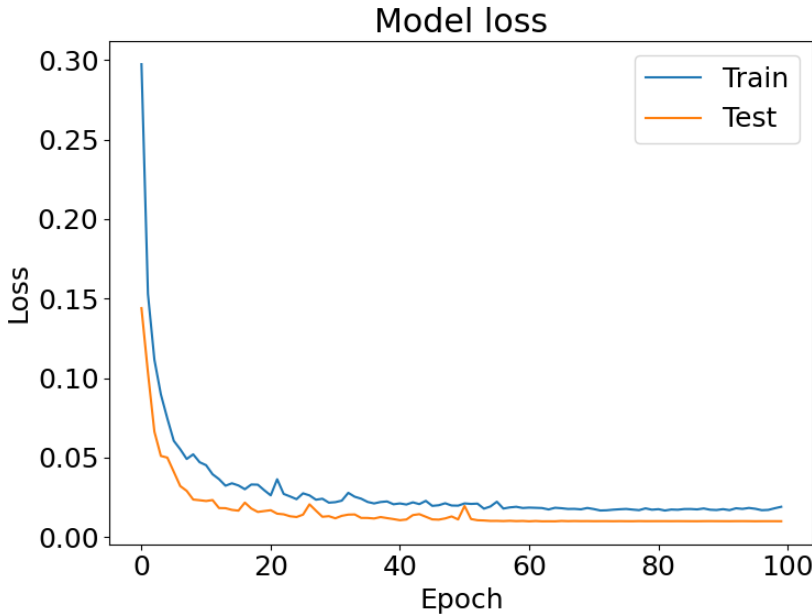


Figure 3.43: The plot shows loss curve over epochs for train and test dataset. Above 50, the loss value stabilises.

### 3.3 A novel image-based NDT approach for measuring crack severity

In section 3.1.1 the use of ridge detectors for line location and measurement is discussed in terms of lab-scale crack scenario. From chapter 3.1.1 unresolved issues remain:

- the Sigma Optimization procedure is taken for granted that Steger algorithm is able to locate the entire line with every value of  $\sigma$ . In a real-scale scenario this may not happen due to the complexity of the crack and the variation of width  $w$  and contrast  $h$  along the line,
- in a real-scale scenario it is not possible to know a-priori line contrast  $h$  because the histogram can be easily affected by disturbing elements,
- in a real-scale scenario there is no guarantee that during the Sigma Optimization procedure the algorithm focuses on the same crack: varying  $\sigma$  value the located crack could change, making the entire procedure pointless,
- in a real-scale scenario there is no guarantee that the ridge detector will

### 3.3 A novel image-based NDT approach for measuring crack severity

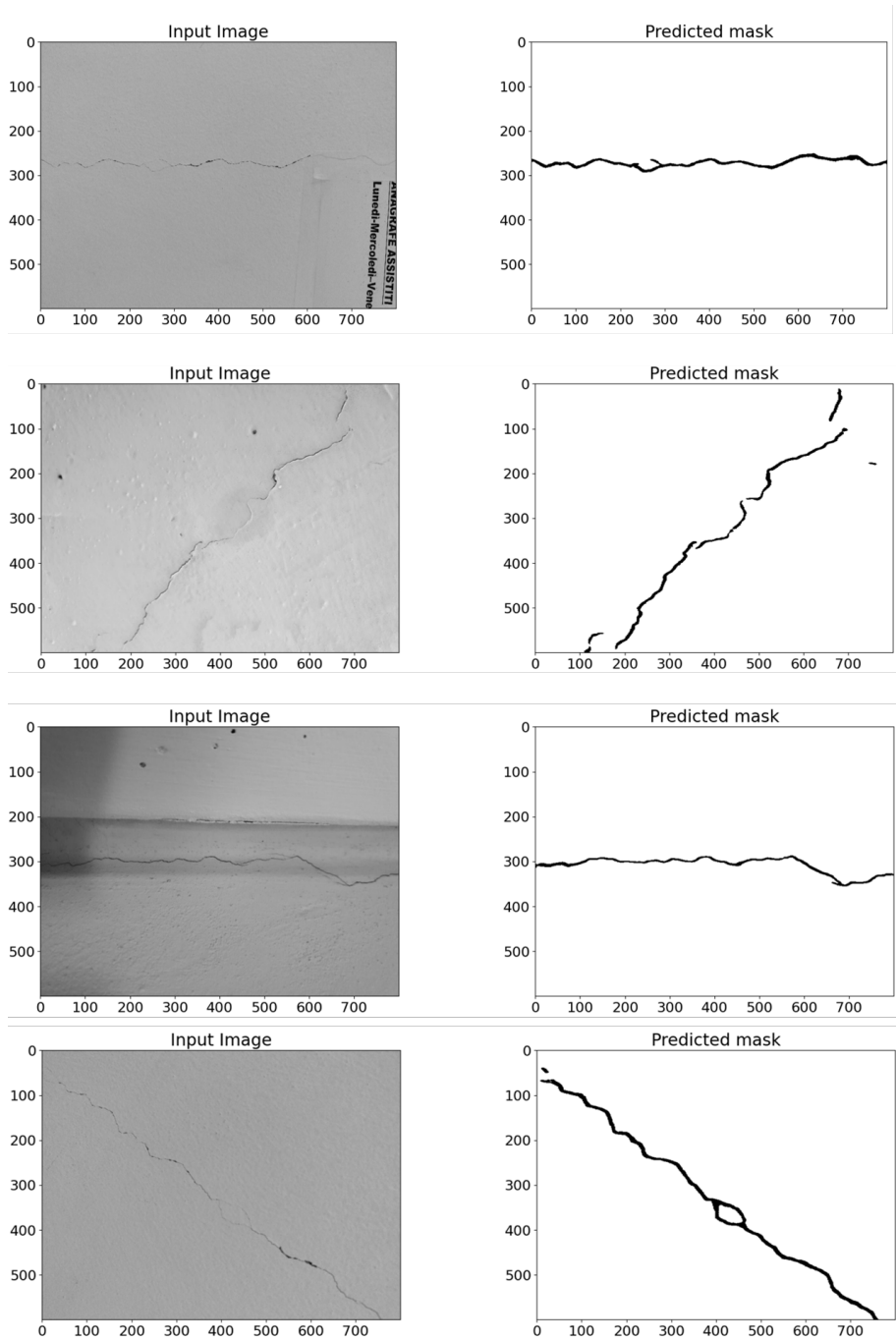


Figure 3.44: Result of U-NET trained model inference on the various image from collected crack test dataset are shown in figure. On the left the input image, on the right the output binary mask.

focus only on target crack. The presence of disturbing element or other cracks will make the entire procedure pointless,

- the established procedure is applied to the entire image resulting in high processing times. A computational time of 6s with  $\sigma_{span} = 7$  on 12MP image may be too long for several applications,
- the ridge detector is not able to distinguish cracks from scratches or other disturbing elements.

In this section the use of the semantic segmentation developed in 3.2.4 for resolving these issues is discussed. The Sigma Optimization algorithm will be modified in order to be used in a real-scale scenario. In this chapter will be described the developed Crack Analyser algorithm that rapidly locate and measure crack width in a real-scale crack scenario. The algorithm involves different steps:

1. image acquisition and calibration,
2. neural Network based semantic segmentation,
3. automatic threshold parameters selection,
4. filling Level algorithm

### 3.3.1 Image acquisition and calibration

The first step of the crack Analyser algorithm involves the acquisition of the image containing defects or cracks. The image can be acquired with any camera. The accuracy of the algorithm strongly depends on lens distortion, that can be mitigated with a custom system calibration, and the spatial resolution of the camera sensor. As is usually the case in computer vision application, the higher the spatial resolution the higher the accuracy and the precision. The performance of the algorithm increases significantly when using high resolution cameras combined with low perspective distortion lenses such as macro or telephoto lens.

In most cases, it is possible to assume that the target surface is a plane. As parallelism between the sensor plane and the surface plane cannot be guaranteed, a perspective correction of the image will always be necessary.

Equation 3.53 shows the camera matrix, a 3x4 matrix that transform from 3D scene points  $(X, Y, Z)$  to 2D scene points  $(u, v, 1)$ .

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.53)$$

### 3.3 A novel image-based NDT approach for measuring crack severity

If the 2D scene points lies on a plane (e.g., target surface) they all have a Z coordinate of zero in the reference system of the plan under consideration. This imply that the equation 3.53 can be simplified since the third column of the camera matrix will be multiplied by zero. As shown in the equation 3.54, the camera matrix become a 3x3 matrix, that can be referred as a planar homography.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.54)$$

Just as for the camera matrix, even the homography matrix has an arbitrary scale factor and it can be normalized. It is possible to estimate the homography matrix if four world points coordinates and the corresponding position of those point on the image plane of the camera are known [115]. The homography transformation can be easily used for perspective removal. Therefore, to eliminate the defects due to the perspective in the acquired image, it is necessary to know the coordinates of at least 4 points in the image. Those four points can be obtained in different ways, including:

- the use of fiducial targets of known size and shape,
- the use of external sensors capable of measuring the depth map and therefore the position in the real world of each point of the image. Examples of these sensors can be: LIDAR, Time of Flight, etc.

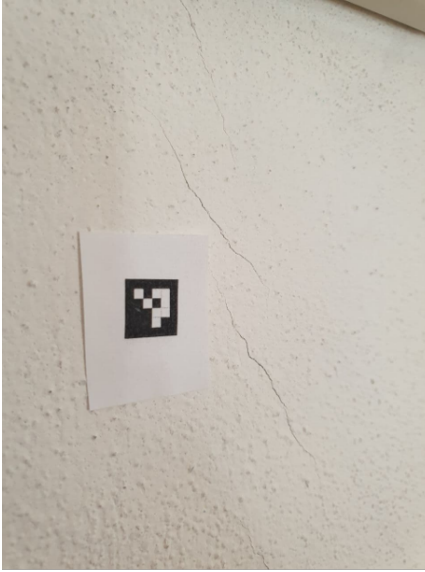
Figure 3.45 shows the use of the 4-points perspective correction technique applied to an input image exploiting a fiducial target of known size and shape.

In order to be able to measure the severity of a crack it is necessary to switch between image pixels coordinates to real-world coordinates in mm (Figure 3.46). This step can be performed in several ways using different technologies:

- depth cameras,
- Laser Imaging Detection and Ranging (LIDAR) sensors,
- Time-of-Flight (ToF) sensors,
- fiducial markers.

As shown in Figure 3.47, given the focal length in multiple of the pixel size  $F = (f_x, f_y)$  and the distance between the camera and the target, is possible to obtain the pixel-to-mm conversion constants based on similarity criterion of triangles:

$$X_{pix-to-mm} = \frac{depth}{f_x} \quad (3.55)$$



framed image



4-points perspective correction

Figure 3.45: 4-points perspective correction using fiducial target of known size and shape.

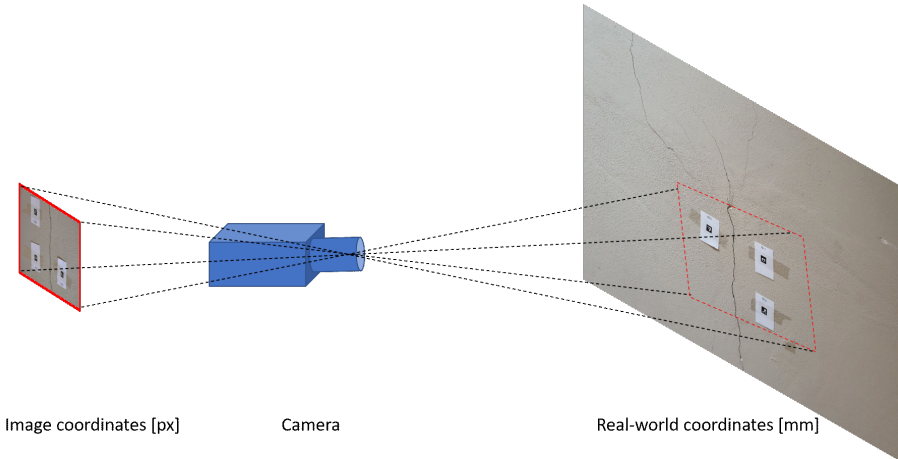


Figure 3.46: To measure the severity of a crack it is necessary to switch between image pixels coordinates to real-world coordinates in mm.



### 3.3 A novel image-based NDT approach for measuring crack severity

$$Y_{pix-to-mm} = \frac{depth}{f_y} \quad (3.56)$$

Assuming a pixel aspect ratio 1:1 is possible to write  $Y_{pix-to-mm} \approx X_{pix-to-mm}$ . While the focal length is a camera intrinsic parameter, the depth value varies based on reciprocal position between camera and target. Sensors like ToF, LIDAR or depth cameras can easily obtain depth value for each image pixel in real time obtaining the pixel-to-mm conversion constant where 1 is the normalized dimension of the pixel. The usage of depth cameras, LIDARs or ToF

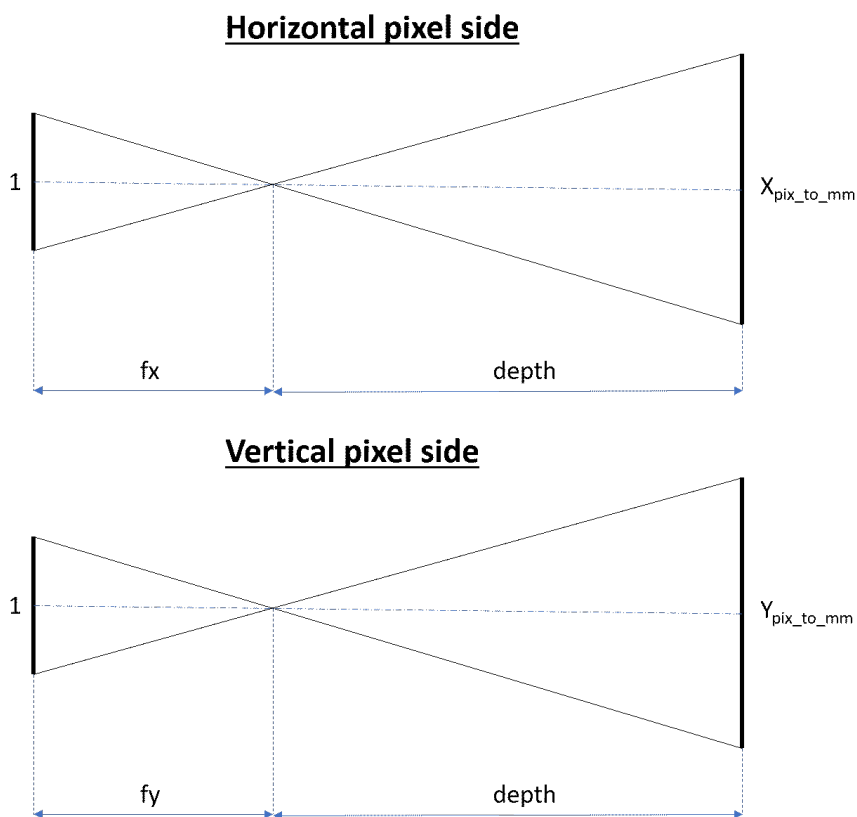


Figure 3.47: The picture shows how is possible to obtain pixel-to-mm constant knowing the focal length and the distance from the target.

sensors allows the user not to have to physically reach the target to be measured. Several applications, like pylons of bridges, offshore structures, inaccessible surface of builds are inaccessible. LIDAR, ToF sensor and depth cameras can be easily embedded in remotely controlled systems like UAVs and allows the user to obtain instantly pixel-to-mm conversion constant.

If it is possible to reach the target, a cheaper solution can be the use of

fiducial markers. It is enough that the user prints and tapes 3 ArUco markers [6, 7] on the wall around the crack to be measured. The algorithm detects the markers and assign them a number from zero to two in clockwise direction. The script measures, for each of them, the length of the sides of the markers in terms of pixels. Then, the average value of the 12 measured sides is calculated. Knowing that the side of each marker is 10 mm (printer rescaling during print operation must be avoided), the pixel-to-mm conversion constant is obtained, and the picture is calibrated. As shown in Figure 3.48 ROI can be selected then based on the centres of marker 0 and marker 2 in order to reduce image dimension and focus on the target.

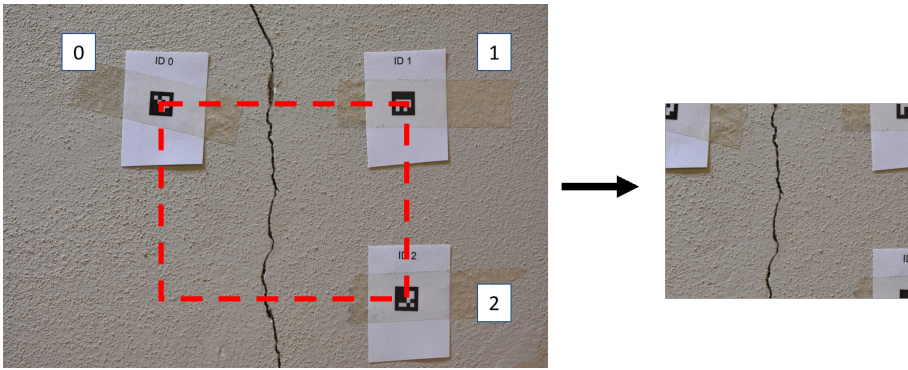


Figure 3.48: Crack ROI selection based on target 0 and target 2 positions.

### 3.3.2 Neural Network based semantic segmentation.

Once the pixel-to-mm constant is obtained, the image is inferenced by a neural network model in order to create a mask that contains only the target crack of defect. The model architecture involved in this step is the U-NET, an evolution of Convolutional Neural Network and stems from the Fully Convolutional Neural networks already describes and trained in 3.2.4. The model is trained obtaining a of IOU score of 0.99 and f1 score of 0.99. Figure 3.49 shows an example of the model prediction. The output binary mask is used for the selection of the automated best-enveloping ROI that envelop the predicted crack (Figure 3.50). The image is now limited to what is strictly necessary which leads to the complete elimination of disturbing elements and significantly increases the performance of the ridge detector algorithm in terms of computational burden and accuracy. This turns out to be a key point of the Crack Analyser algorithm as it minimizes the error and focus the Steger algorithm only where it is necessary avoiding false positives and it increases the computational speed by several orders of magnitude.

3.3 A novel image-based NDT approach for measuring crack severity



Figure 3.49: On the left the acquired RGB input image, on the right the binary model prediction

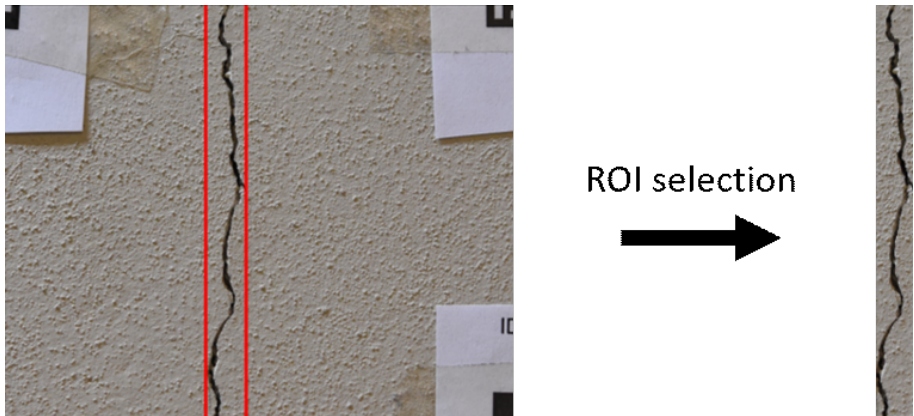


Figure 3.50: automated best-enveloping ROI selection based on model prediction

### 3.3.3 Automatic threshold parameters selection

Once the automated best-enveloping ROI containing the target crack is selected, the ridge detector algorithm can be applied to the image. From the previous chapter 3.1.1 is known that Steger ridge detector depends on 3 key parameters:

1. Lower threshold  $l$
2. Upper threshold  $u$
3. Sigma  $\sigma$ , and consequently line width  $w$

As described in 3.1.1, salient lines are defined by their second directional derivative:

$$r''(\sigma, h) = h \left( \frac{2\sqrt{3}}{\sqrt{2\pi}\sigma^2} e^{-3/2} \right) \quad (3.57)$$

In the equation 3.57 it is assumed that  $\sigma = w/\sqrt{3}$ , but if values greater than 3 below the root are used, it simply remain more conservative in the choice of threshold parameters. Then  $\sigma$  value and the line contrast  $h$  can be plugged into the above equation to yield an upper threshold for the algorithm. The  $\sigma$  value cannot be known a-priori, nevertheless the  $h$  value can be estimated. The workflow for calculating the line contrast is shown in Figure 3.51. Taking the advantage of the tailored selection of the essential ROI containing only the crack region explained in section 3.3.2, there is the possibility to study its histogram and determine with good accuracy the medium pixel intensity of the crack and the medium pixel intensity of the surrounding framed area. First of all, the same ROI is selected on both RGB input image and on the binary mask (a). An erosion filter is applied to the mask to enlarge the area that immediately surround the crack (b). The mask is multiplied to the RGB image: all the RGB pixel not belonging to the mask therefore became zero (c). The histogram of the resulting image is performed (d), null intensity values are excluded from the calculation. The crack line contrast is finally obtained as the absolute difference between the two local maxima of the cubic spline approximation histogram function (Figure 3.52). The first maxima (peak) represents pixels belonging crack region, the second peak represent pixels belonging background area (e). After  $h$  is obtained, the upper and lower thresholds  $u, l$  can be obtained as:

$$u = h \left( \frac{2\sqrt{3}}{\sqrt{2\pi}\sigma^2} e^{-3/2} \right) \quad (3.58)$$

$$l = 0.1h \left( \frac{2\sqrt{3}}{\sqrt{2\pi}\sigma^2} e^{-3/2} \right) \quad (3.59)$$

The lower threshold can be considered the 10% of upper threshold using a more conservative assumption than Steger paper. As described in the Steger

3.3 A novel image-based NDT approach for measuring crack severity

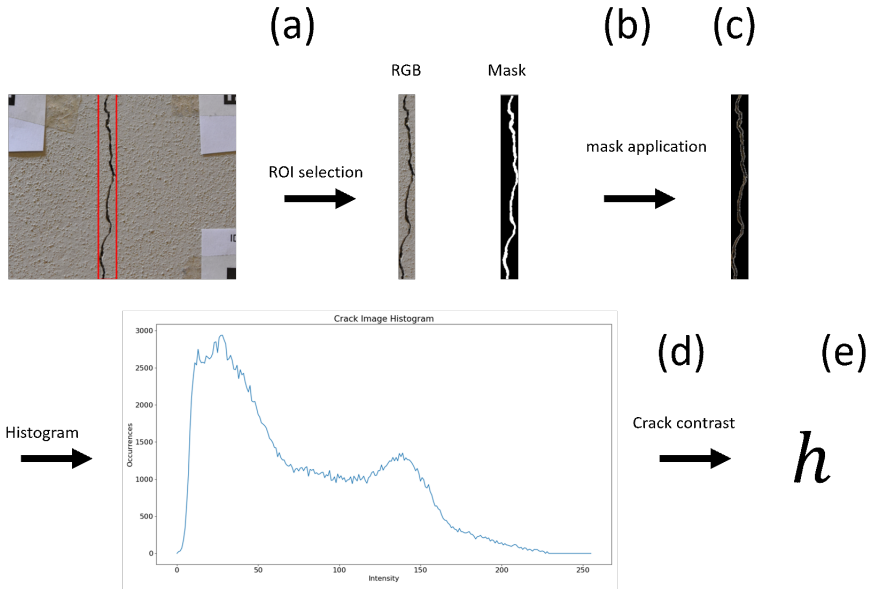


Figure 3.51: Contrast  $h$  evaluation workflow

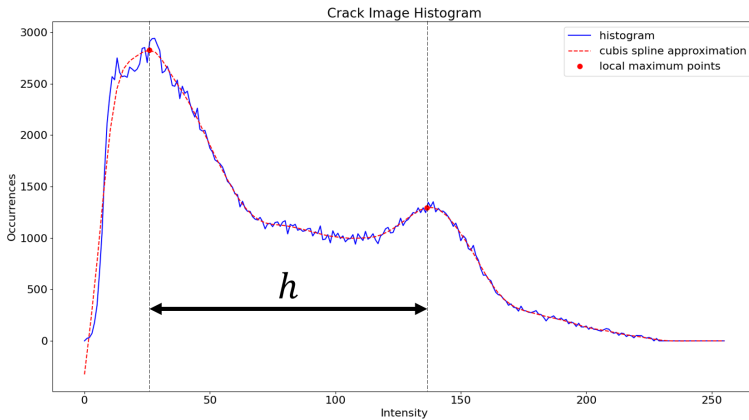


Figure 3.52: Line contrast  $h$  is obtained as the absolute difference between the two local maxima of the cubic spline approximation histogram function

algorithm any pixel that has a value above the upper threshold will be marked as possible line pixel, and any pixel whose value lies between the higher and the lower thresholds, but is connected to a pixel whose value is above the higher threshold, will also be marked as possible line pixel. All the other pixels of the image are rejected [84].

### 3.3.4 Filling Level Algorithm

At this point, the only input to be chosen for applying Steger algorithm to the target RGB image, is  $\sigma$  value. Assuming a  $\sigma$  value is possible to obtain line width and line location with sub-pixel accuracy. Is not possible to known a-priori the correct  $\sigma$  value because it depends on line width. In section 3.1.2 the Sigma Optimization Algorithm is presented but the procedure is taken for granted that Steger algorithm is able to located the entire line with every value of  $\sigma$ . In a real-scale scenario this may not happen due to the complexity of the crack and the variation of width  $w$  and contrast  $h$  along the line. As shown in Figure 3.53, changing  $\sigma$  value, the recognized portion of the crack varies. It is possible to notice how, after a certain value, the crack is always well-recognized by the algorithm. From Sigma Optimization algorithm chapter is known that, overestimating  $\sigma$  value leads to an overestimation of the line thickness. The idea is to establish a procedure able to determine the minima  $\sigma$  value that best fit the target curve. Figure 3.54 shows the Filling Level

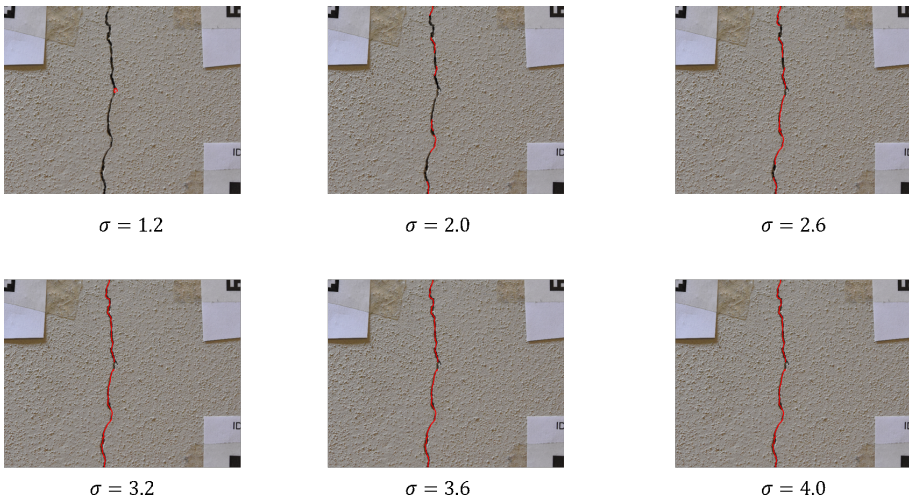


Figure 3.53: Recognized portion of the crack varies changing  $\sigma$  value

algorithm procedure in order to select  $\sigma_{FL}$ , that is the minima  $\sigma$  value able to well-locate the target crack. First of all, an array of  $\sigma$  is defined ranging from 0.6 to 6 with a step of 0.2 (a). The initial value of 0.6 is a limitation

imposed by Steger line detector. According to the algorithm, with  $\sigma = 6$  is possible to locate line with a maximum thickness of about 20.8 pixel; since the application is meant to be used by standard resolution cameras, it is logical to assume that a crack will never be thicker than this size. For high-resolution cameras, it will be sufficient to raise the limit in accordance with its resolution and proximity to the crack. For low-resolution cameras (around 1MP) it will be possible to reduce this value in order to increase computational speed since crack thickness will never exceed a few pixels. Starting from the first value of  $\sigma_i$ , the hysteresis threshold parameters  $u_i$  and  $l_i$  are calculated with equations 3.58 and 3.59 basing crack contrast following the procedure explained in section 3.3.3(b). Given  $\sigma_i, u_i$  and  $l_i$  is possible to apply Steger line detector to the target RGB crack image (c). All the lines that do not fall within crack mask are automatically discarded. Basing on the binary crack mask and the detected line location, the Filling Level determines how much the detected lines fills the mask and it is measured in percentage terms (d). The operations (b), (c) and (d) are iterated until  $\sigma_i > \sigma_N$ . The final step consists on finding  $\sigma_{FL}$  (e) with an automatic procedure basing on Filling Level (FL) over  $\sigma$  plot (Figure 3.55). The  $\sigma$  selected will be the first value of FL plateau, that is the point at which the FL stops growing as the  $\sigma$  the increases. Once  $\sigma_{FL}, u_{FL}, l_{FL}$  are determined is possible to apply Steger Line detector to obtain the line width and position for each line point with sub-pixel accuracy.

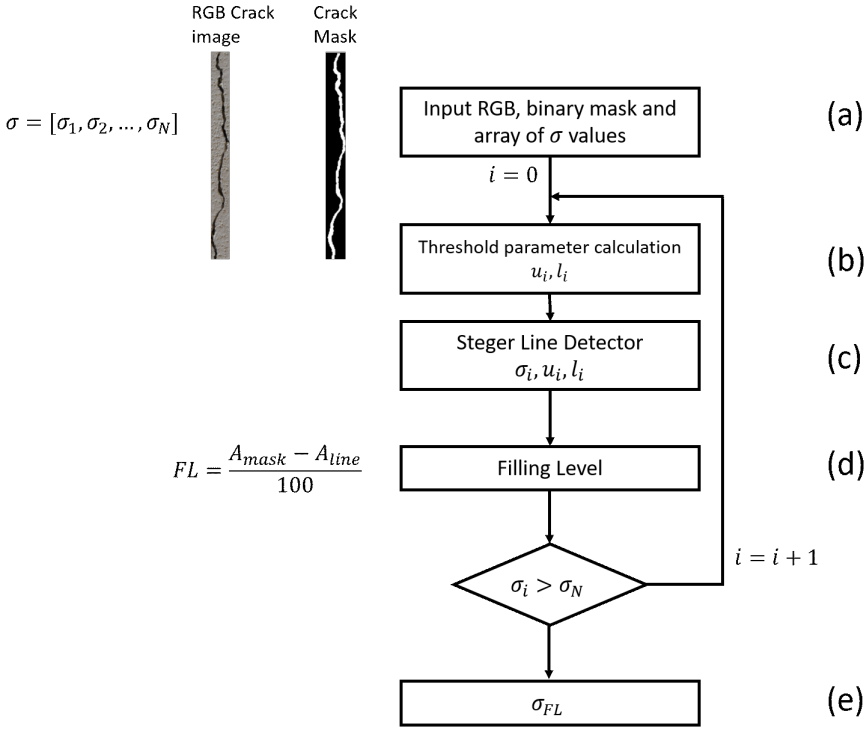


Figure 3.54: Filling Level algorithm flowchart for  $\sigma_{FL}$  identification

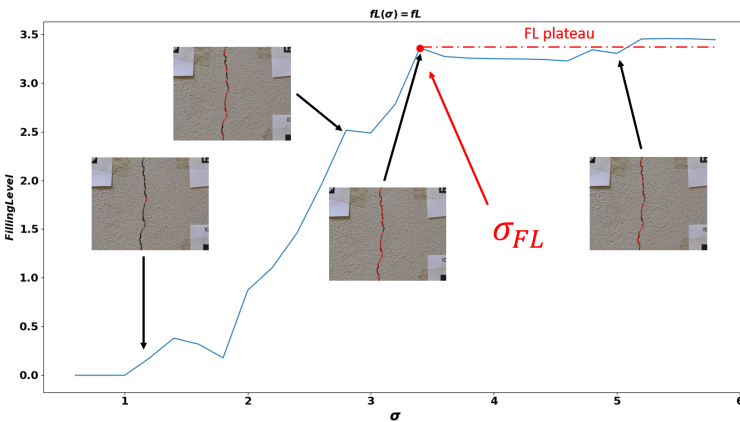


Figure 3.55: FL over  $\sigma$  plot. The data are collected within the Filling Level algorithm.



# Chapter 4

## Metrological characterization

The metrological characterization of the proposed system is performed through a structured test, where for each acquired image the relative angles between the sensor surface and the target surface are known 4.1, and a test that does not adopt any type of control, thus simulating a real-scale application of the measuring device 4.2. In the first type of test, an anthropomorphic robot is used to acquire images at known angles and define the working range of the developed device. Subsequently the results obtained from these tests will be compared with the results obtained through uncontrolled measurements.

### 4.1 Working range definition

The working range of the device is determined through the analysis of cracks in a controlled environment. To this end, an anthropomorphic robot is used. On the surface of the end-effector a plate is placed on which a crack with a constant width of 1.25 mm was printed. 3 fiducial markers have been placed around the crack in order to calibrate the acquired image. The reference system of the end-effector is shown in figure 4.1, with the Z axis coming out of the plane, with origin of the axes fixed at the center of the crack. Nikon D7200 24MP camera, equipped with Nikon NIKKOR 60mm f2.8 macro lens, is used as acquiring system. The reflex is placed on a tripod at a known distance from the target. The parallelism between the sensor surface and the target surface is guaranteed through a system of a laser and a mirror 4.2: a laser is made integral with the target surface (3) and a mirror (2) has been affixed to the front surface of the lens. Through manual positioning, the end-effector is positioned in such a way that the laser beam, reflecting on the mirror surface, tarnished back to the emission point. Four types of acquisitions are made:

1. controlled rotation of the target surface from  $-30^\circ$  to  $+30^\circ$ , with steps of  $0.5^\circ$ , rotating around the x axis, for a total of 60 images acquired,
2. controlled rotation of the target surface from  $-30^\circ$  to  $+30^\circ$ , with steps of  $0.5^\circ$ , rotating around the y axis, for a total of 60 images acquired,

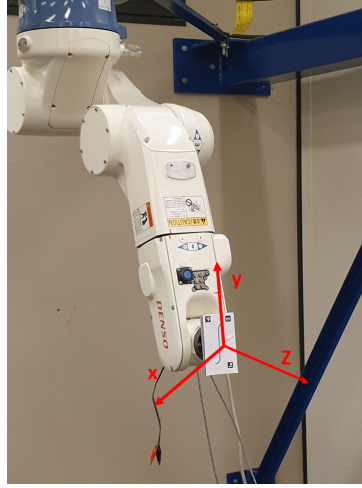


Figure 4.1: An anthropomorphic robot is used in order to define the working range of the developed device in a controlled environment.

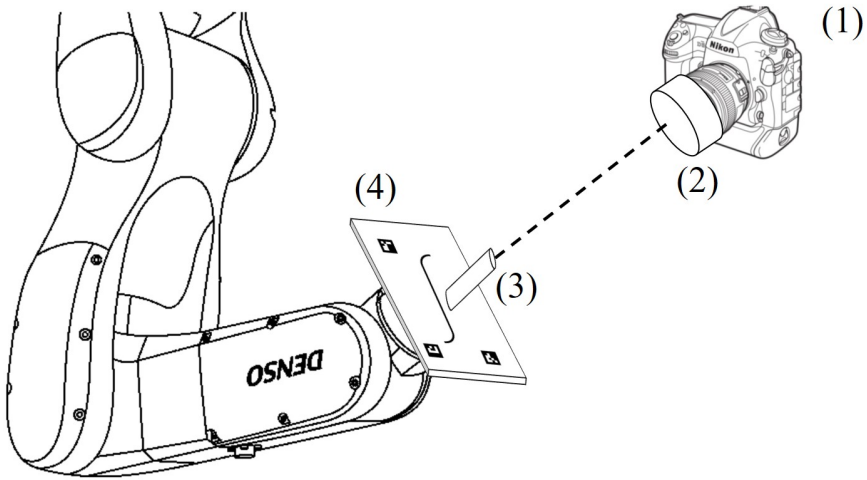


Figure 4.2: The image acquisition system (1) it is placed at a known distance from the target (4). The parallelism between the sensor surface and the target surface is guaranteed through a system made of a laser (3) and a mirror (4). The two surfaces are to be considered parallel when the laser beam reflected by the mirror returns back to the source.

3. controlled rotation of the target surface with random rotation angles from  $-30^\circ$  to  $+30^\circ$  along the x and y axes, for a total of 360 images acquired,
4. 100 images acquired manually by a non-expert operator without any type of control over the exposure angle.

The first two types of acquisition are carried out in such a way as to separate the single contribution due to the relative laying angle between the surfaces. The third type of test, on the other hand, is used to estimate the type A uncertainty of the system as the pose angles vary. All these measurements were repeated at different distances: 70cm, 100cm, 130cm and 240cm. The Crack Analyser algorithm is then applied to all the acquired images.

### 4.1.1 Target rotated around x axis

A controlled rotation is imposed to the target surface from  $-30^\circ$  to  $+30^\circ$ , with steps of  $0.5^\circ$ , rotating around the x axis. The Crack Analyser algorithm is applied to the 60 images framed at each distance from the target. At each iteration, the measured average width value and the pixel-mm conversion factor are saved. Figure 4.3 shows the plot of the absolute error over the angle at which every image is framed. The error is calculated as the absolute difference between mean width measured by the Crack Analyser algorithm and the target width (1.25 mm). It is easy to notice the measurement result does not depend on the distance of acquiring system from the target crack. This is most likely made possible by the nature of the ridge detection algorithm used. In fact, once the crack position has been detected, the algorithm measures its width perpendicular to the centre of the line, identifying the transitions of the edges, therefore the fact that the crack is represented by more or less pixels inside the sensor it matters less. This concept is clear from the figure 4.4, where the thickness of the crack converted into pixels is shown and where, for each acquisition distance, the actual thickness in number of pixels is indicated. As it is logical to expect, moving away from the target the number of pixels of crack thickness decrease as well as the absolute error in pixels due to the lack of correct parallelism between the measurement and crack planes.

### 4.1.2 Target rotated around y axis

A controlled rotation is imposed to the target surface from  $-30^\circ$  to  $+30^\circ$ , with steps of  $0.5^\circ$ , rotating around the y axis. The Crack Analyser algorithm is applied to the 60 images framed at each distance from the target. At each iteration, the measured average width value and the pixel-mm conversion factor are saved. Figure 4.5 shows the plot of the absolute error over the angle at which every image is framed. The absolute error is calculated as the difference

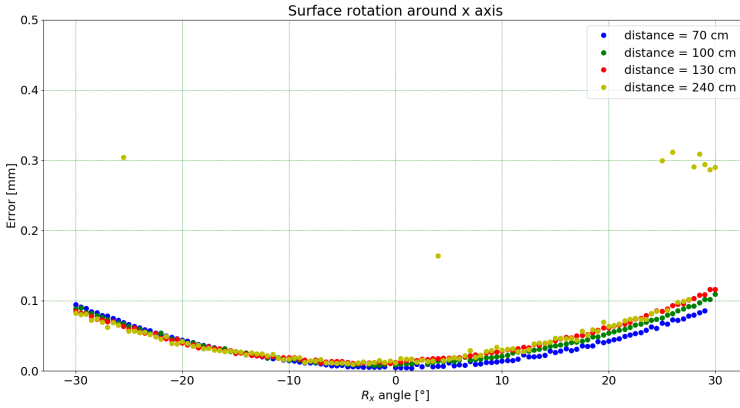


Figure 4.3: The figure shows the plot of the absolute error in millimetres, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every images are framed.

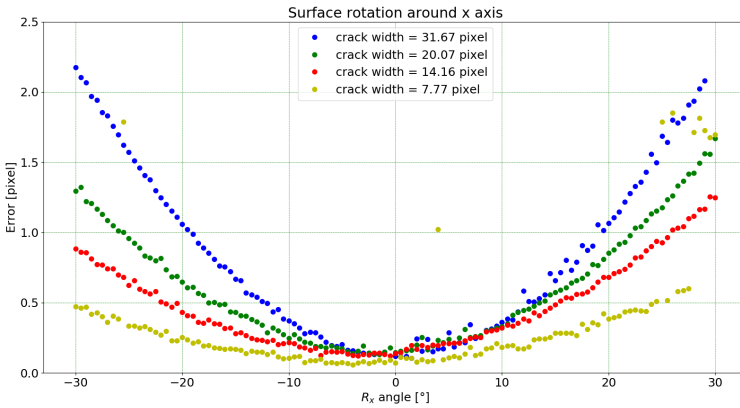


Figure 4.4: The figure shows the plot of the absolute error in pixel, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every image are framed.

between mean width measured by the Crack Analyser algorithm and the target width (1.25 mm). The same considerations of the previous section 4.1.1 can be made.

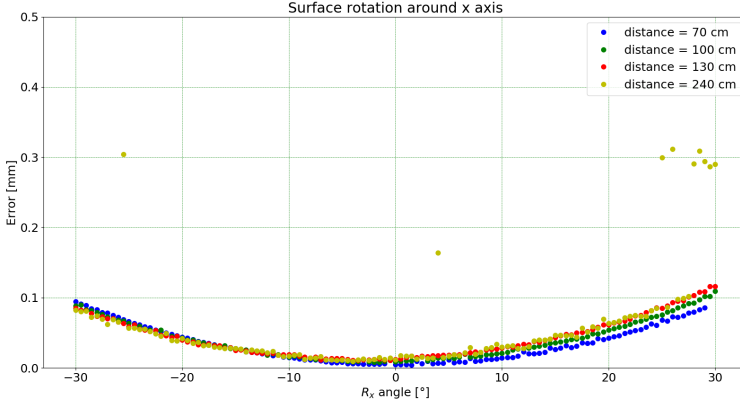


Figure 4.5: The figure shows the plot of the absolute error in millimetres, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every images are framed.

### 4.1.3 Target rotated around x and y axis

The camera, placed parallel to the surface at the predetermined distance by means of the method described above. acquires an image and then the surface randomly rotates around its centre of gravity. A controlled rotation is imposed to the target surface with a random angle between  $-30^\circ$  to  $+30^\circ$  (continuous uniform distribution). A total of 360 images are collected with different angular positions for each distance considered. The angles assumed by the surface are shown in figure 4.7. The Crack Analyser algorithm is then applied to each acquired image to calculate the average width of the detected crack. The results for target distance are shown in figure 4.8. The plots show the measured points  $p(R_x, R_y, \hat{w})$  for target distance of 70, 100, 130 and 240 cm. In all the cases examined, a fit error of 0.01 mm is obtained. As it is easy to see, all the interpolating curves are of the hyperbolic paraboloid type  $\hat{w} = \frac{R_x^2}{a^2} - \frac{R_y^2}{b^2}$ , where  $a$  and  $b$  are positive real numbers. All this was easy to expect since, as displayed in figure 4.9, rotations around the y-axis  $R_y$  axis tend to underestimate, while rotations around the x-axis  $R_x$  axis tend to overestimate the width measurement for perspective effects. Subsequently, a study was carried out on the trend of the standard deviation as the rotation angle changed. Starting from  $1^\circ$ ,

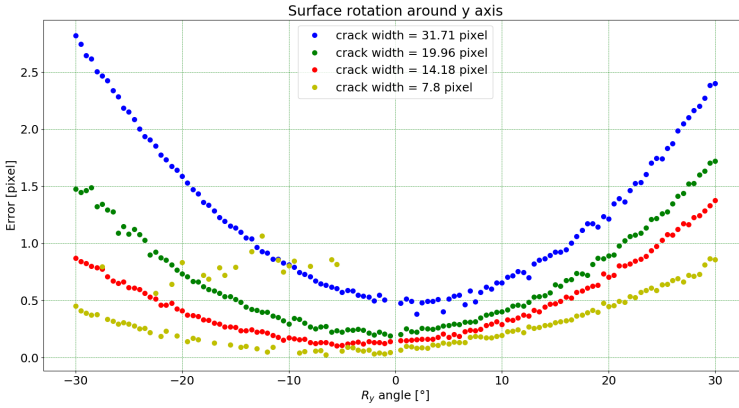


Figure 4.6: The figure shows the plot of the absolute error in pixel, as difference between mean width measured by the Crack Analyser algorithm and the target width, for each angle at which every image are framed.

up to  $30^\circ$  with a step of  $1^\circ$ , the crack width standard deviation is calculated considering only the measurements carried out in that angular range. For example in figure 4.10 all the acquisitions that fall within the angular range  $\pm 25^\circ$  are highlighted in red. The graphs in figures 4.11 and 4.12 show the trend of the standard deviation as the imposed angle varies. The first figure shows the standard deviation in mm, the second in pixels. From the graphs it can be seen how, similarly to what happened with fixed rotation on  $R_x$  and  $R_y$  angles, the distance from the target does not affect the trend of the standard deviation of the measurement. On the contrary, the increase in the angle pose leads to a significant increase in the standard deviation and the measurement error. Although these acquisitions are made with a reflex with high spatial resolution and fiducial markers, the concept exposed can also be extended to acquisition systems with a lower number of megapixels and other image calibration methods simply by making the appropriate proportions based on the spatial resolution.

A further interesting aspect is linked to the variability of the sigma value, obtained through the filling level algorithm, at each framing distance. Figure 4.13 shows the dispersion of the values at each distance, normalized to the equivalent pixel width of the crack. For example, if we consider a crack about 7 pixels thick, the optimal sigma value selected by the algorithm varies from 1.0 to 2.4 for images taken at the same distance, but with different angles. The range of variability reaches from 5.8 to 8.4 in the case in which the crack has an equivalent thickness of about 30 pixels (image taken from a closer distance). In the latter case, the sigma can therefore assume 13 different values, the range being equal to 2.6 with the filling level algorithm set at a step of 0.2. All this

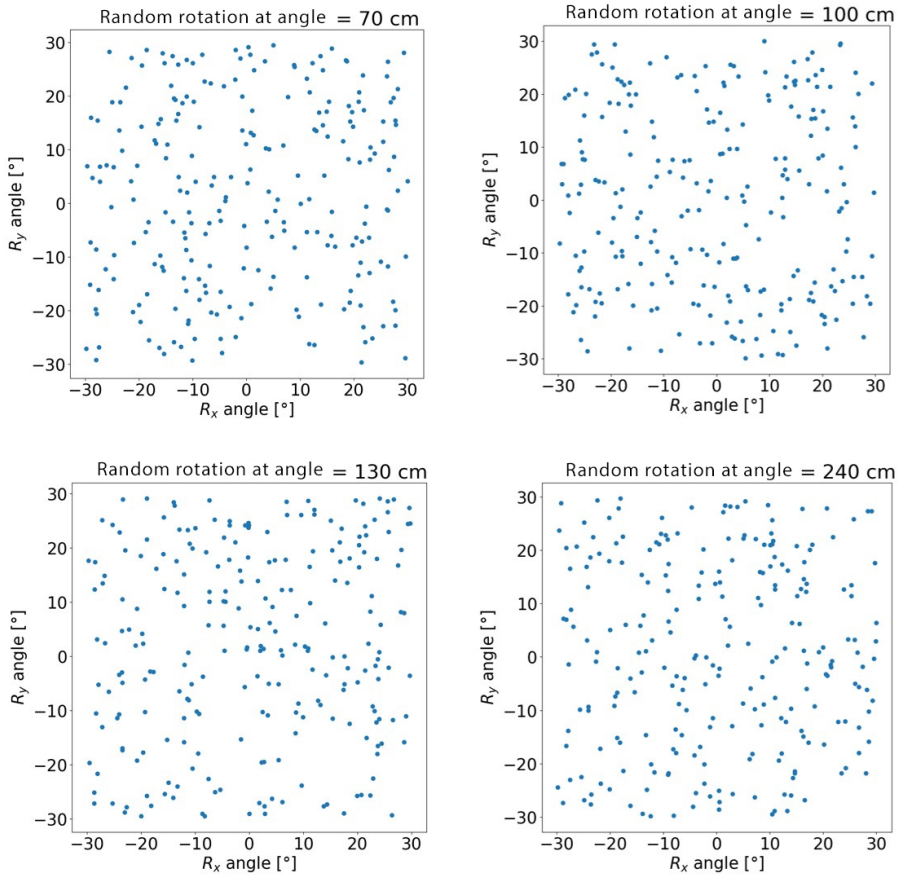


Figure 4.7: A controlled rotation is imposed to the target surface with a random angle between  $-30^\circ$  to  $+30^\circ$  (continuous uniform distribution). For each distances 360 images are framed.

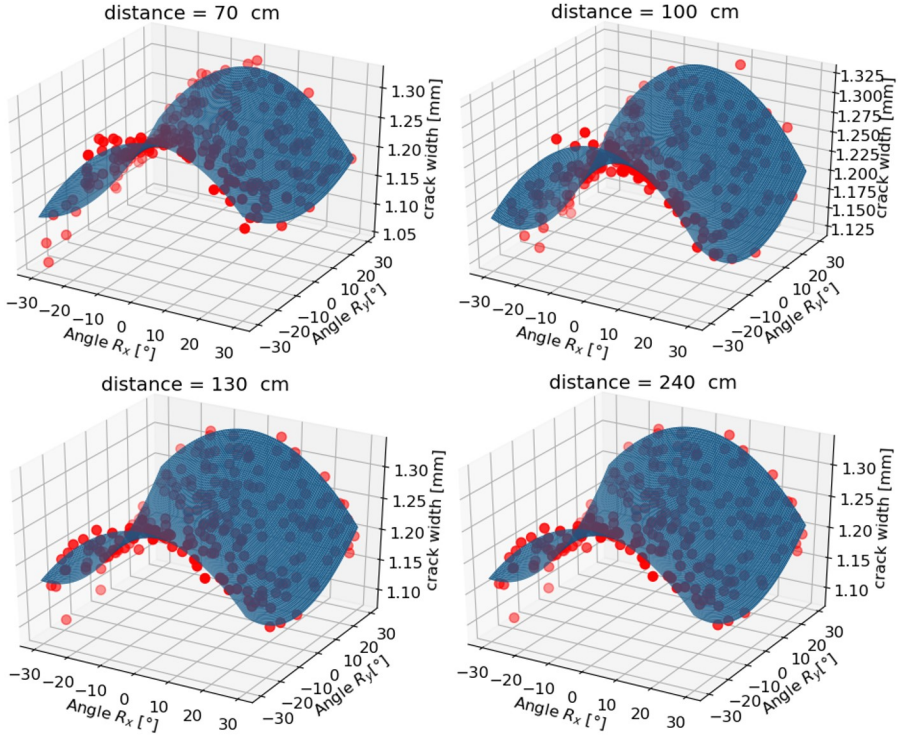


Figure 4.8: Best-fit quadratic curve of measured points  $p(R_x, R_y, \hat{w})$  for target distance of 70, 100, 130 and 240 cm. In all the cases examined, a fit error of 0.01 mm is obtained

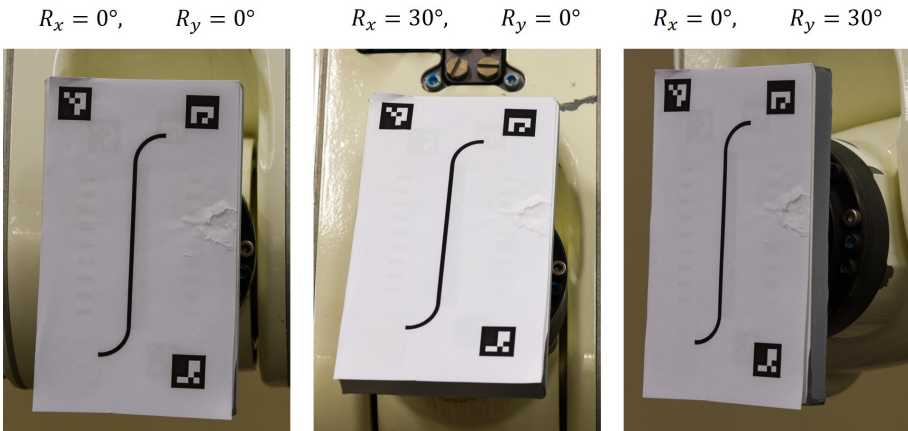


Figure 4.9: Rotations around the y  $R_y$  axis tend to underestimate, while rotations around the x  $R_x$  axis tend to overestimate the width measurement for perspective effects.



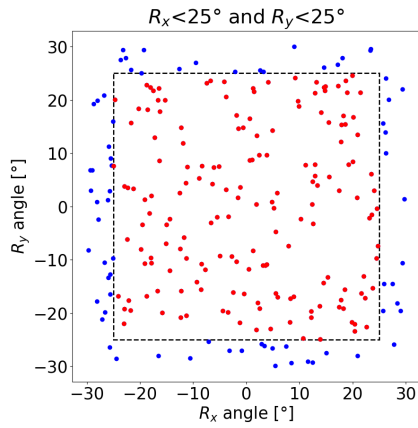


Figure 4.10: In red are highlighted all the acquisitions that fall within the angular range  $-25^\circ < R_x < +25^\circ$  and  $-25^\circ < R_y < +25^\circ$

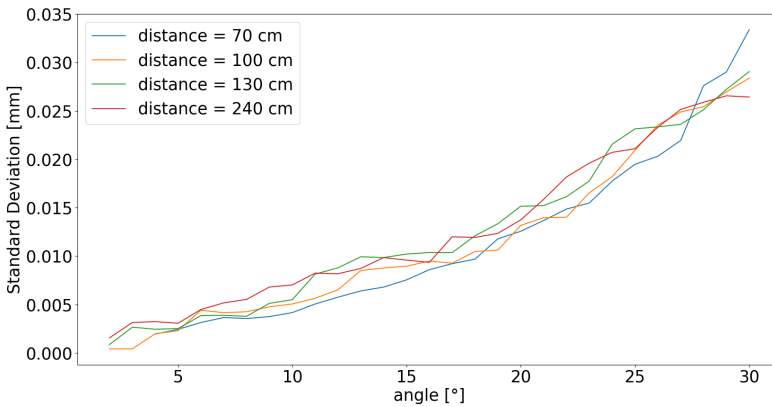


Figure 4.11: Trend of the standard deviation in  $mm$  as the maximum rotation angle imposed on the target varies.

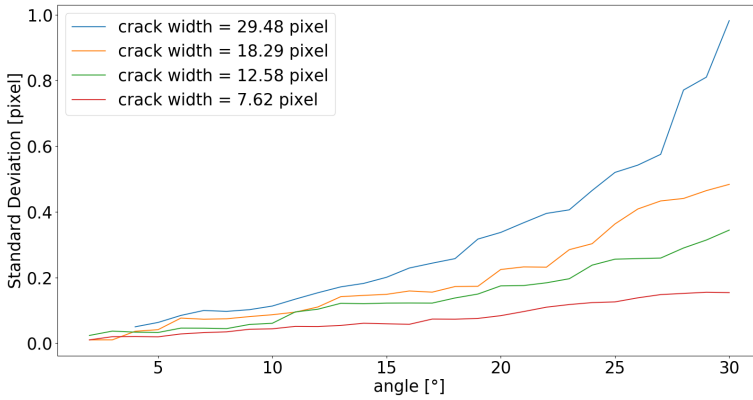


Figure 4.12: Trend of the standard deviation in *pixel* as the maximum rotation angle imposed on the target varies.

immediately makes us understand how fundamental the choice of a correct sigma is and how much this can vary a lot even in images taken at the same distance from the target. The Steger algorithm, without the proposed automatic optimization technique, could never have returned the results obtained by not knowing a-priori the optimal sigma value.

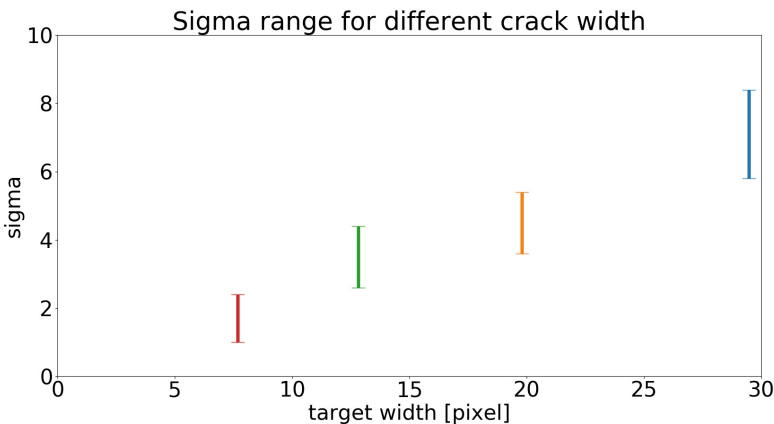


Figure 4.13: Dispersion of the values at each distance, normalized to the equivalent pixel width of the crack.

#### 4.1.4 Manual acquisition by non-expert operator

For this test 100 images are framed manually by a non-expert operator. The purpose of this test is to understand, through the study of the standard deviation of the results obtained, what is the equivalent pose angle by comparing it with the study in a controlled environment of the previous section 4.1.3. For each of the 100 images the average crack width is measured by applying the Crack Analyser algorithm. The results show an average value of average crack width of 1.23 mm and a standard deviation of 0.13 mm. By curve-fitting the standard deviation data over imposed angular range, obtained in the previous chapter, a second degree function of equation  $y = 0.00001x + 0.00003x^2 + 0.00221$  is obtained (figure 4.14). From this it can be deduced that, through the manual acquisition of images, a result, in terms of standard deviation, equal to the controlled acquisition with a maximum angle set of  $19^\circ$  can be obtained.

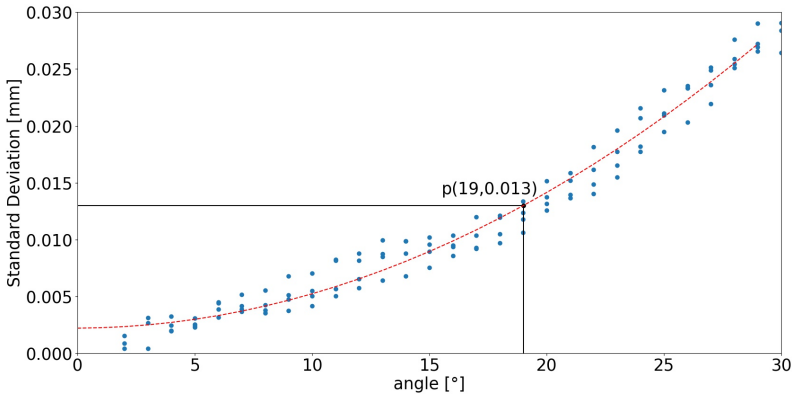


Figure 4.14: By curve-fitting the standard deviation data over imposed angular range, obtained in the previous chapter, a second degree function of equation  $y = 0.00001x + 0.00003x^2 + 0.00221$  is obtained. A standard deviation of  $0.013mm$  corresponds to a maximum angular range of  $19^\circ$ .

## 4.2 Measurement System Analysis: theory

In the previous chapter a metrological analysis was carried out through a structured test and the use of a printed crack with known dimensions. In the following chapters, on the other hand, the metrological analysis will be carried out in a context of real and uncontrolled environment. The metrological performance of the Crack Analyser algorithm in real-scale environment is assessed in terms of precision and accuracy [116] for two case scenarios:

1. Camera without depth sensor, that measure the conversion constant pixel-to-mm and calibrate the image using fiducial markers.
2. Camera equipped with depth sensor (i.e. depth cameras, LIDAR or ToF sensors), that measures the conversion constant pixel-to-mm and calibrate the image without fiducial markers.

Three different target crack on concrete surface are chosen for repeatability and reproducibility tests (Figure 4.15). The Type A uncertainty is used for

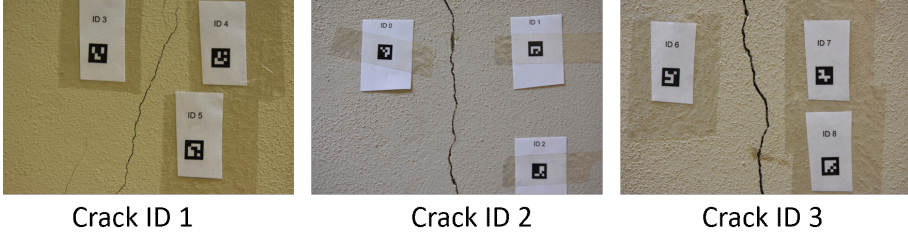


Figure 4.15: Concrete wall crack used as target for reproducibility and repeatability tests

the system characterization. The Type A uncertainty is a method of assessing uncertainty based on the statistical analysis of series of measures. The evaluation of a Type A contribution is based on any statistically valid method. An example is the calculation of the standard deviation of the mean of a series of independent observations and performing an analysis of variance (Analysis of Variance (ANOVA)) [117] to analyse and quantify the random effects in certain types of measures. If is considered a quantity  $X_i$  whose value is estimated from  $n$  independent observation  $X_{i,k}$  of  $X_i$  obtained with identical measurement conditions, in this case the estimation  $x_i$  is defined by the sample mean:

$$x_i = \bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{i,k} \quad (4.1)$$

And the standard uncertainty  $u(x_i)$  associated with  $x_i$  is the standard deviation of the sample mean  $x_i$ :

$$u(x_i) = s(\bar{X}_i) = \frac{1}{n(n-1)} \sum_{k=1}^n (X_{i,k} - \bar{X}_i)^2 \quad (4.2)$$

Although the standard uncertainty  $u$  is used to express the uncertainty of many measurement results, what is often required is an uncertainty measure that define an interval, around the result of the measurement  $y$ , that can reasonably contain the true value of the measurand  $Y$ . The measure of uncertainty used to

satisfy this requirement is the expanded measurement uncertainty  $U$  obtained multiplying  $u$  by a coverage factor  $k$ . The true value of the measurand is considered to be in the range  $x_i \pm U$ . The coverage factor  $k$  is chosen on the basis of the required confidence level, typically  $k = 2$  or  $3$ . When the normal distribution of the  $n$  independent observation  $X_{i,k}$  of  $X_i$  is valid and  $u(x_i)$  is a reliable estimation of the standard deviation of  $y$ ,  $k = 2$  defines an interval with a confidence level of approximately 95 %, and  $k = 3$  defines an interval with a confidence level greater than 99%.

Gage Repeatability and Reproducibility (Gage R&R) method is performed in order to assess repeatability and reproducibility of the measurement system. While the repeatability is defined as the variation between successive measurements by the same operator, on the same target and using the same tool, the Reproducibility studies the variation in measurements taken by different operators on the same target using the same system. The Gage R&R method compares the variability caused by the system itself with the total variability in order to assess the actual variability of the entire measurement system. Basing on the system type there are several types of Gage R&R. The best that can fit this work is the Crossed Gage R&R that is used for non-destructive testing and fit all cases where all the operators can repeatedly measure the same targets (Figure 4.16). Crossed Gage R&R can be performed with the ANalysis Of Vari-

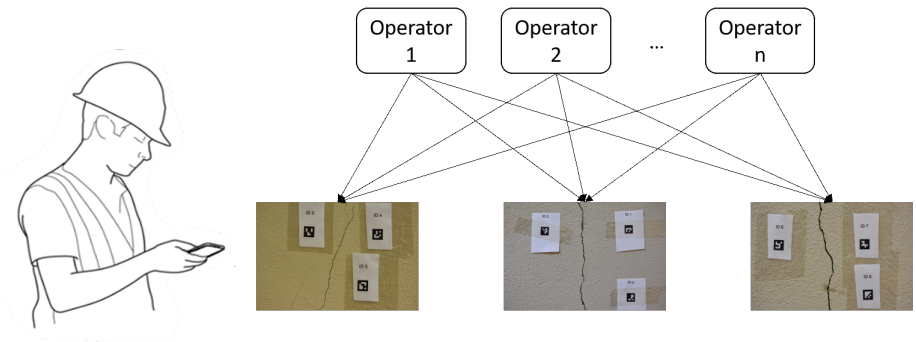


Figure 4.16: Cross Gage R&R method for non-destructive testing for crack assessment.

ance (ANOVA) that is also able to measure the variability of the interaction between the operator and the different targets measured. The ANOVA test typically works as follow. A number  $o$  of operators, minimum required is  $o = 2$ , perform  $m$  measurements, minimum required is  $m = 3$ , on  $p$  different parts. The total Sum of Square of deviations is given by sum of the contribution of operators, target parts, their interaction and the equipment used:

$$SS_{total} = SS_{operator} + SS_{parts} + SS_{operator*parts} + SS_{equipment} \quad (4.3)$$

The operator sum of squares is calculated as:

$$SS_{operators} = \sum_{i=1}^o p \cdot m (X_i - \bar{X}_i)^2 \quad (4.4)$$

Where  $X_i$  is the average of the measurements made by the  $i$  operator on all parts and  $\bar{X}$  is the average of the measurements made by all operator on all parts. The target part sum of square is similarly obtained as:

$$SS_{parts} = \sum_{k=1}^p o \cdot m (X_k - \bar{X}_i)^2 \quad (4.5)$$

Where  $X_k$  is the average of the measurements made by all operator on  $j$  part. The total sum of squares is obtained as the squared deviation of each individual result from overall average:

$$SS_{total} = \sum_{i=1}^o \sum_{k=1}^p \sum_{j=1}^m (X_{i,j,k} - \bar{X}_i)^2 \quad (4.6)$$

Where  $X_{ijk}$  is the  $j$  measure made on  $k$  part by  $i$  operator. Then the equipment sums of squares, that represent the repeatability of the system, is obtained as the deviation of all trials for a given part and given operator from the average for that part and that operator  $\bar{X}_{ik}$

$$SS_{equipment} = \sum_{i=1}^o \sum_{k=1}^p \sum_{j=1}^m (X_{i,j,k} - \bar{X}_{ik})^2 \quad (4.7)$$

Now it is easy to separate the single contribution of the interaction between target parts and operators:

$$SS_{operator*parts} = SS_{total} - (SS_{operator} + SS_{parts} + SS_{equipment}) \quad (4.8)$$

The ANOVA Table can be created as shown in Table 7. F-statistic asses if the variance between two population (operators and parts) are significantly different and determines the p-value that is the probability the results could have happened by chance. The F-statistic determine if a group of variables are jointly statistically significant. After choosing the so-called  $\alpha$  level, typically  $\alpha = 0.05$ , that is the Significance threshold Level, is possible to reject of not the Null Hypothesis. Reject the Null Hypothesis is possible when p-value is smaller than the  $\alpha$  level. The Null Hypothesis states there is no difference between certain characteristics of a groups. Rejecting the Null Hypothesis means that the differences between groups are significant and the test is statistically significant.

For p-value calculation the Excel FDIST function is used:

$$p_{value} = FDIST(F, DoF1, DoF2) \quad (4.9)$$

This function calculates the F probability Distribution, which measures the degree of diversity between two data sets given the F-statistic where the function is to be evaluated and the degree of freedom of the involved groups. If the  $p_{value}(interaction) > \alpha$  is not possible to consider the repeatability with interaction values, the interaction row of the ANOVA table has to be ignored. After the p-value study, the variances of repeatability, operator, part, and interaction has to be measured as follow:

$$\sigma_{rep}^2 = MS_{rep} \quad (4.10)$$

$$\sigma_{int}^2 = \frac{MS_{int} - \sigma_{rep}^2}{m} \quad (4.11)$$

$$\sigma_{part}^2 = \frac{MS_{part} - MS_{int}}{o \cdot m} \quad (4.12)$$

$$\sigma_{op}^2 = \frac{MS_{op} - MS_{int}}{p \cdot m} \quad (4.13)$$

The Gage R&R, Repeatability, Reproducibility, Part to Part and Total Variation are obtained as follow:

$$GAGE\ R\&R = \sigma_{rep}^2 + \sigma_{op}^2 \quad (4.14)$$

$$Repeatability = \sigma_{rep}^2 \quad (4.15)$$

$$Reproducibility = \sigma_{op}^2 + \sigma_{int}^2 \quad (4.16)$$

$$Part\ to\ Part = \sigma_{part}^2 \quad (4.17)$$

$$Total\ Variation = \sigma_{op}^2 + \sigma_{int}^2 + \sigma_{rep}^2 + \sigma_{part}^2 \quad (4.18)$$

If the  $GAGE\ R\&R\ Variance\% < 1\%$  it is considered to be an acceptable measurement system, if it is between 1% and 9% it may be acceptable according on the specific application, over it is considered to be unacceptable. Then the Study Variation is the amount of variation caused by the measurement system

and by the differences between parts. It is calculated as:

$$\text{Study Variance} = 6 \cdot \sigma \tag{4.19}$$

According to the Automotive Industry Action Group (AIAG) if the GAGE R&R  $\text{StudyVariance}\% < 10\%$ , the measurement system is acceptable, if it is between 10% and 30%, it may be acceptable according on the specific application, over it is considered to be unacceptable [118]. Finally, the Number of Distinct Categories (NDC) can be measured:

$$\text{Number of Distinct Categories} = \frac{\sigma_{part}}{\sigma_{GAGE\ R\&R}} \sqrt{2} \tag{4.20}$$

This is a metric that is used in GAGE R&R studies to identify a measurement system's ability to detect a difference in the measured characteristic. According to AIAG this metric should be greater than 5 for an adequate measuring system. In summary:

- P-value of interaction between operator and part should be less than the  $\alpha$ , that is typically fixed at 0.05.
- The % Variance contribution of GAGE R&R should be less than 1% to be considered as an acceptable measurement system.
- The % Study Variance contribution of Gage R&R should be less than 10% to be considered as an acceptable measurement system.
- The NDC should be greater than 5% to be considerate an adequate measuring system.

While the precision of the measurement system is assessed by Gage R&R studies, the accuracy is assessed with measurement comparison with 2D profilometer laser Wenglor MSL132. The optical characteristics of the laser profilometer are shown in Figure 4.17. The profilometer is able to project a laser line onto

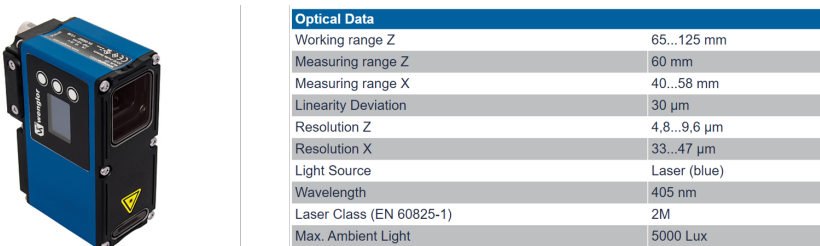


Figure 4.17: Wenglor MSL132 Optical data

the object to be detected and generate an accurate, linearised height profile



with an internal camera which is set up at a triangulation angle. The resolution  $X$  of  $33 \mu m$  at working distance of  $65 mm$  from the target make this instrument suitable for measuring surface cracks on concrete once mounted on a motorized micro-metric trail. As shown in Figure 4.18, the Wenglor MLSL132 (7) is mounted on an electronically controlled micro-metric positioning system (6) which is integral with the tripod (5). The profilometer is connect to the laptop (1) with GigE Vision standard (4). The trail is connected to the laptop (1) with GigE Vision standard (4). The trail is connected with serial cable (3) to the trail controller (2) that is piloted by the laptop with serial interface. A

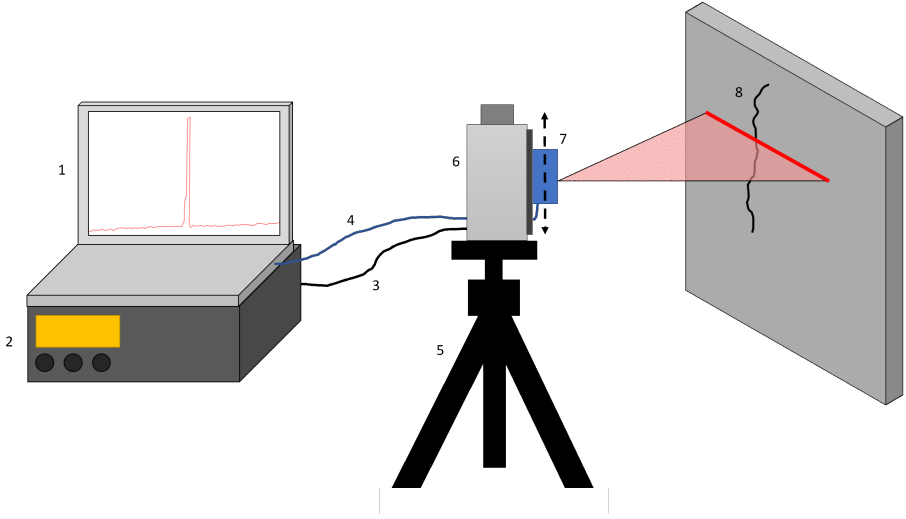


Figure 4.18: Measurement system for cracks assessment with WENGLOR laser profilometer

LabVIEW Virtual Instrument (VI) is realized in order to control the micro-metric positioning system following the flowchart reported in Figure 4.19. The system is positioned manually to the starting position. The total displacement  $Y$  and the  $mm$  per step  $\Delta y$  are defined based on target crack. The first laser profile is acquired then the positioning system moves  $\Delta y$   $mm$ , wait 2 second in order to stabilize and acquire the laser line profile until the current position  $y = Y$ . Finally, the gap for all stored laser line profiles, can be calculated. Several gap measurement algorithms are available in the literature. The gap algorithm used in this scenario measures the Euclidean distance between points  $p_1, p_2 \in z$ , where  $z$  is the measured laser line profile. The laser line profile gradient  $z'$  is computed using second order accurate central differences in the interior points and either first or second order accurate one-sides differences at the boundaries (Figure 4.20). Point  $p_1$  is the point of line profile where the gradient exceeds a certain threshold (e.g. threshold = 0.2 mm). The order of elements of laser line profile is then reversed, the shape of the array is preserved,

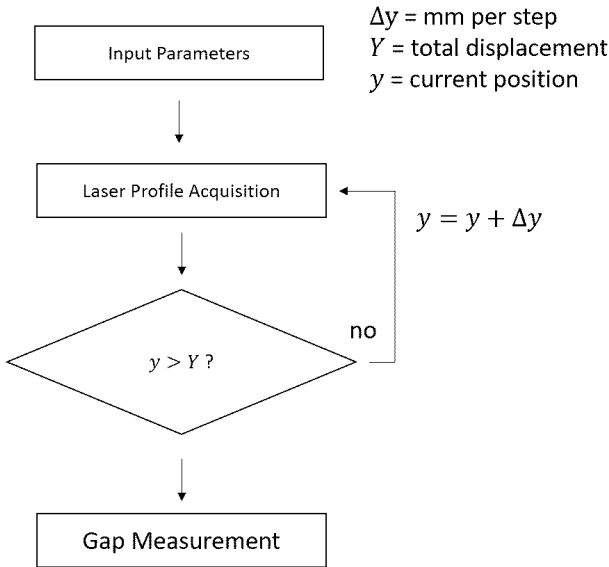


Figure 4.19: LabVIEW VI flow chart for crack gap measurement

but the elements are reordered. Point  $p_2$  is the point of line profile where the gradient of this new array exceeds a certain threshold (e.g. threshold = 0.2 mm). The mean crack width is calculated as the mean gap measured in the laser lines recorded between  $y_0$  and  $y_0 + \Delta Y$ .

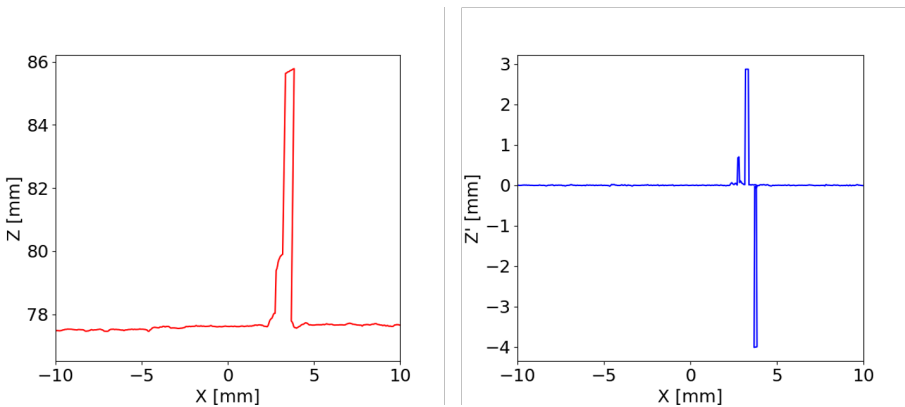


Figure 4.20: On the left the detected laser line, on the right its gradient.

### 4.3 Metrological characterization of the measurement system with fiducial markers

Nikon D7200 24MP camera, equipped with Nikon NIKKOR 60mm f2.8 macro lens, is used for the metrological characterization of measurement system with fiducial markers. The Type A uncertainty is assessed by taking 100 pictures of the same crack while the reciprocal pose between operator and target is varying (i.e. moving the camera near and away from the wall and taking the picture from different height). The distribution, normalised with respect to their mean value, related to the values measured in the intra-operator analysis is reported in Figure 4.21. The Type A uncertainty associated to the measurements is estimated to be 0.0019 mm. If considered a coverage factor of  $k = 2$ , and expanded uncertainty value of 0.0038 mm is identified. The Gage R&R study is assessed by asking

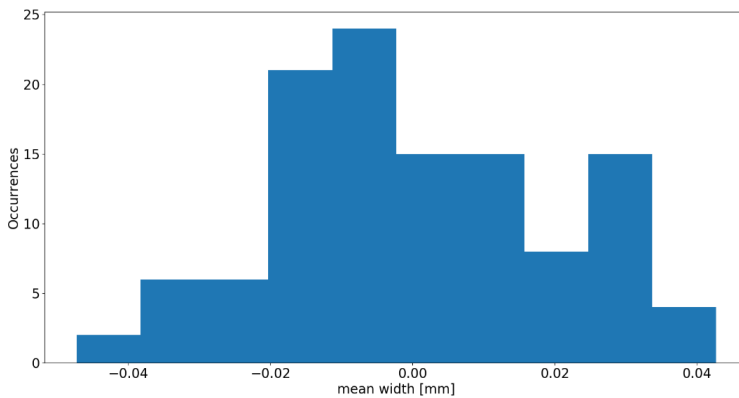


Figure 4.21: Distribution normalised with respect to the mean value of mean width values measured on 100 images by the same operator

5 different operators to perform ten measurements each in 3 different target cracks. A two-way ANOVA is run on sample of 150 measurements to examine the effect of operator and the type of crack, defined by CRACK ID (Table 4.1).

The results show that there is a significant interaction between the crack and the operator ( $p < 0.05$ ), the null hypothesis can be rejected, the differences between groups are significant and the test is statistically significant. The Variance analysis shows that the repeatability variance is quite small ( $0.000055mm^2$ ) if compared with the reproducibility ( $0.000925mm^2$ ). The contribution % of Variance of Total Gage R&R minor than 1%, that is acceptable. According to AIAG it is considered to be an acceptable measurement system because the contribution % of Study Variance is 9.23% and the NDC is 15. As shown in the

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	11.412	5.706	753.39	0
Operator	4	0.051	0.013	1.684	0.25
Interaction	8	0.061	0.008	137.863	0
Repeatability	135	0.007	0		
Total	149	11.531			

Source	Variance	% Contr
Total Gage R&R	0.00098	0.85
Repeatability	0.000055	0.05
Reproducibility	0.000925	0.8
Operator	0.000173	0.15
Interaction	0.000752	0.65
Part to Part	0.113967	99.15
Total Variation	0.114947	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.031298	0.188	9.23
Repeatability	0.007412	0.044	2.19
Reproducibility	0.030408	0.182	8.97
Operator	0.013145	0.079	3.88
Interaction	0.02742	0.165	8.09
Part to Part	0.33759	2.026	99.57
Total Variation	0.339038	2.034	100

Number of Distinct Categories = 15

Table 4.1: Gage R&R study assessed by asking 5 different operators to perform ten measurements each in 3 different target cracks. Metrological characterization of the measurement system with fiducial markers

### 4.3 Metrological characterization of the measurement system with fiducial markers

Gage R&R report (Figure 4.22) several considerations can be drawn:

- As it is reasonable to expect, the much variation is between Part-to-Part, but not due to Gage R&R (Components of Variation chart)
- The difference between operators is quite small (average width by operator chart).
- The operators measure is mostly consistently, all the points except one fall within the control limits. (S chart by operator).
- There is no significant difference between the operators and also there is no interaction between parts and operator. (Crack ID\*operator Interaction chart).

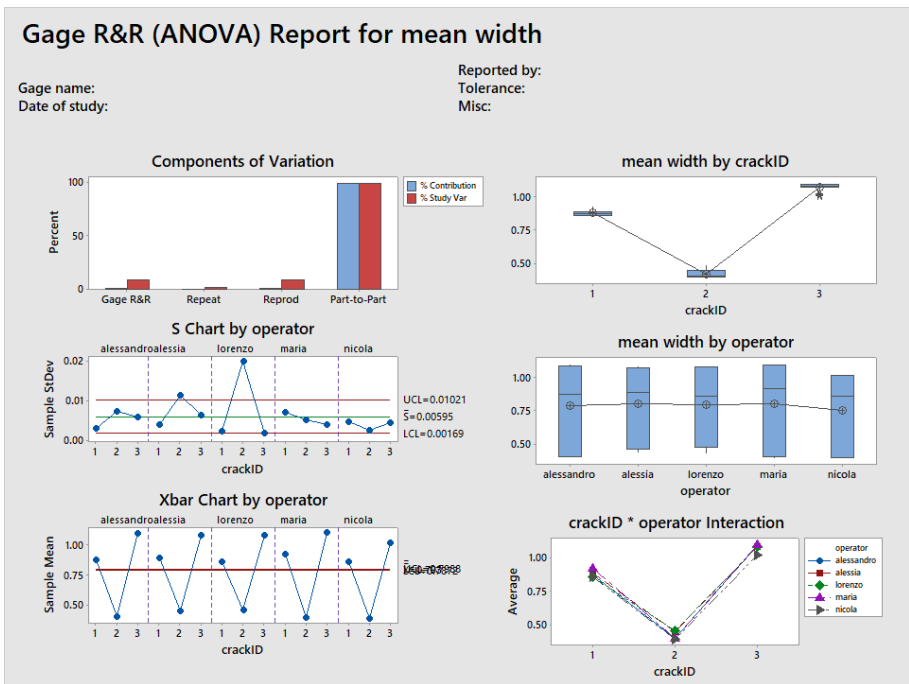


Figure 4.22: Gage R&R (ANOVA) Report for mean width crack calculation. Metrological characterization of the measurement system with fiducial markers

While the precision of the measurement system is assessed by Gage R&R studies, the accuracy is assessed with measurement comparison with 2D profilometer laser Wenglor MSL132 following the procedure described in 4.2. The measure is performed at the minimum working distance (65mm) in order to maintain X resolution of 33  $\mu m$ . Measurements are performed on Crack ID

3 with the 2D laser profilometer WENGLOR. The realised setup is shown in figure 4.23. In order to make the measurement systems comparable, the starting point of the laser is manually positioned to the centre of marker ID 6 (Figure 4.24). A  $\Delta Y = 0.1mm$  is set as minimum step for the electronically controlled micro-metric positioning system (figure 4.19).



Figure 4.23: Lab-scale WENGLOR laser profilometer measurement setup.

The gap is measured in each point, and results are reported in figure 4.24. The same measure is performed with Nikon D7200 and Crack Analyser algorithm. The average gap measured by the profilometer is 1.11 mm and the mean gap measured with the developed algorithm is 1.08 mm. The difference between the two measures falls within the X resolution of the profilometer ( $33 \mu m$ ).

## 4.4 Metrological characterization of the measurement system with depth sensor

Intel Realsense D435i depth camera, largely used in applications such as robotics or augmented and virtual reality, is used to assess the metrological characterization of measurement system without fiducial markers (Figure 4.26). The camera is equipped with two monochromatic sensors (1280x720pixels) and one IR projector for estimating depth map through Active IR Stereo technology. The RGB model for image acquisition has a  $64^\circ \times 41^\circ \times 77^\circ (\pm 3^\circ)$  Field of View (FOV) and 1920x1080 pixels spatial resolution. Through Active IR

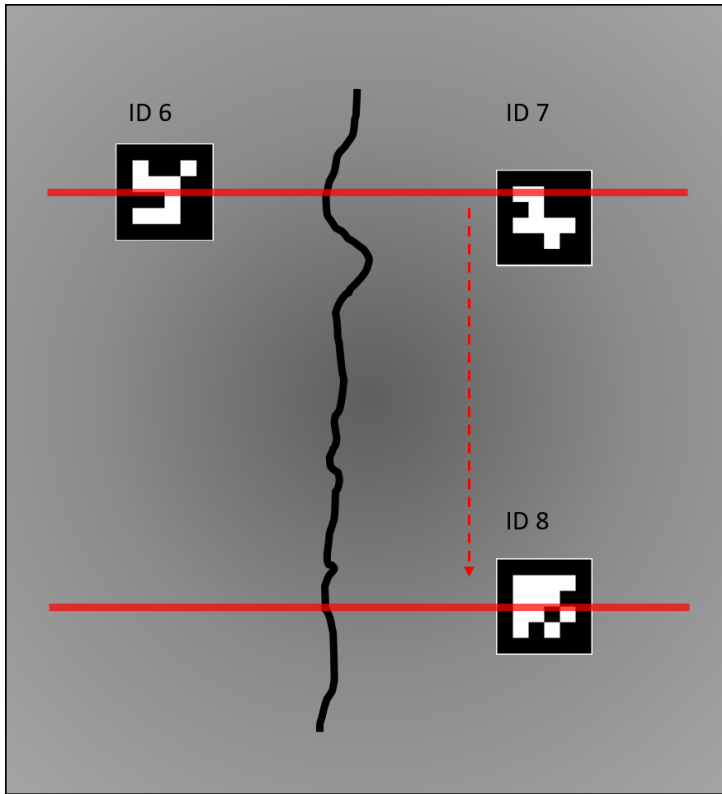


Figure 4.24: In order to make the measurement systems comparable, the starting point of the laser is manually positioned to the centre of marker ID 6 then move to centre of marker ID 8 with a step of 0.1 mm

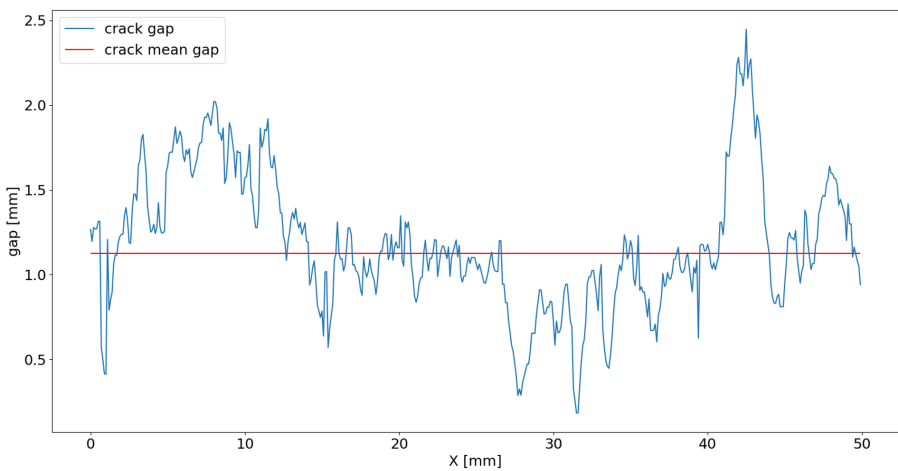


Figure 4.25: Crack gap measured between centre of marker ID 6 and ID 8 with a step of 0.1mm

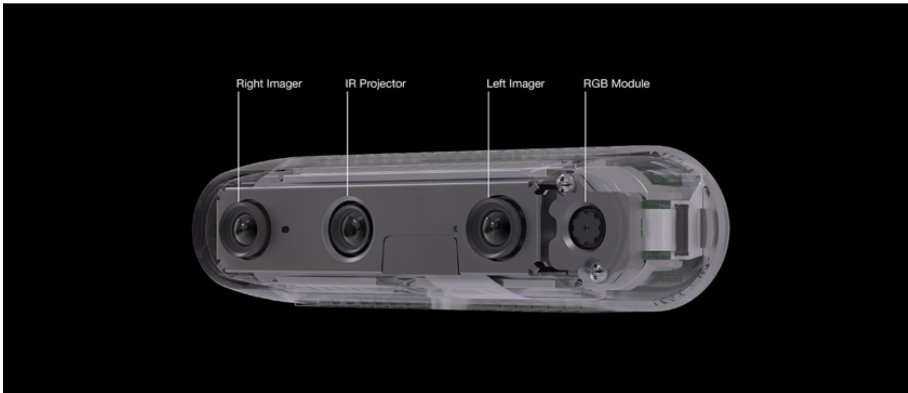
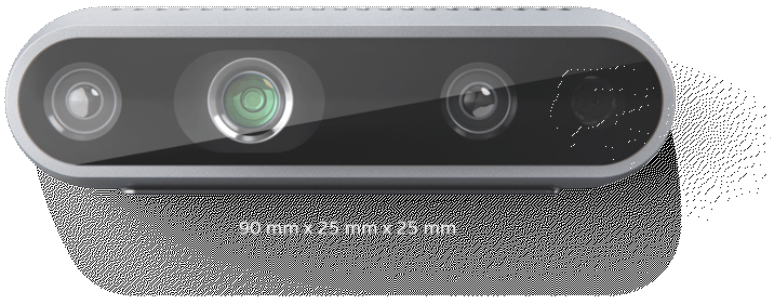


Figure 4.26: Intel Realsense RGBD D435i depth camera



#### 4.4 Metrological characterization of the measurement system with depth sensor

Stereo technology, after a proper calibration with a custom calibration target provided by Intel, a calibrated depth layer 1280x720 can be obtained. Each single pixel value represents the distance in millimetres between that pixel and projected real-world surface with a depth accuracy of  $< 2\%$  in the range of  $0.1 - 2m$ . With proper internal algorithm the RGB and Depth layer are matched obtaining a RGBD image. In all that cases where the target surface is plane (e.g. concrete wall), the Depth layer can be used for checking the parallelism between sensor and target surface plane (Figure 4.27). In this way

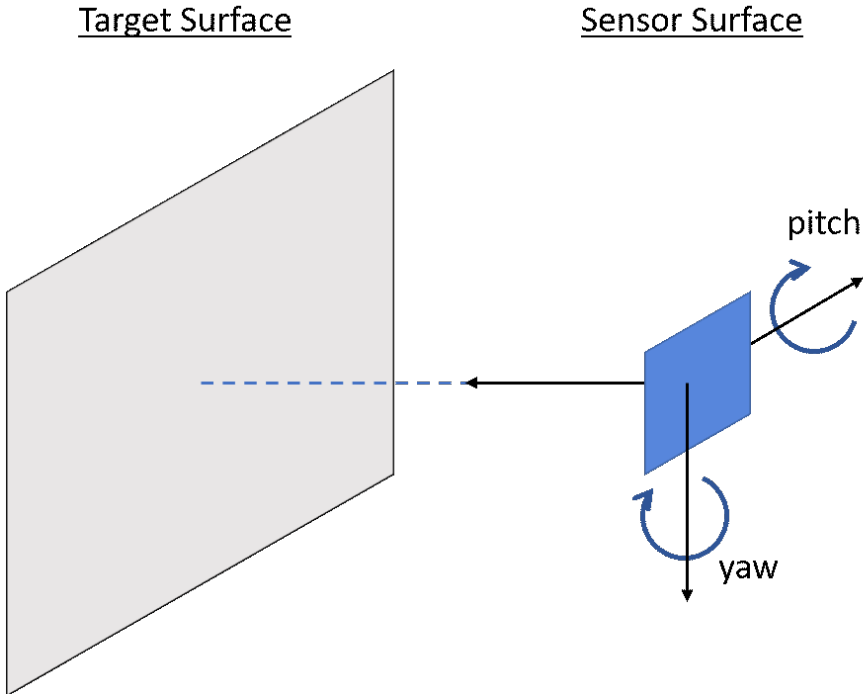
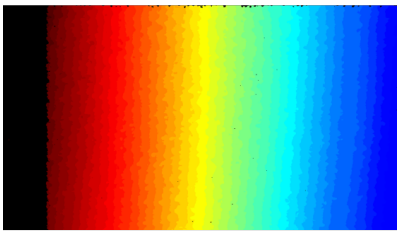


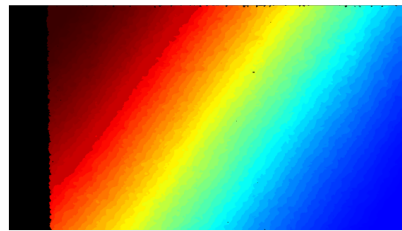
Figure 4.27: Relative position between sensor and target planes.

the operator could be forced to capture the image as parallel as possible to the target surface under a certain threshold angle. In Figure 4.28 is shown 4 different acquisition where is angle between target and sensor planes range from  $0^\circ$  to  $10^\circ$  for yaw and pitch. In order to assess the effect of yaw and pitch angles in terms of repeatability and reproducibility 5 different acquisition scenario are performed following the scheme in Figure 4.29. The operator approaches the target surface and the camera start looping acquisition (a). Each acquired frame is used for estimating yaw and pitch angles between target and sensor planes, if one of them is higher than target angular threshold the frame is rejected and the camera continue the acquisition loop (b) until an RGBD frame with yaw and pitch angles lower than target angular threshold are acquired (c). The

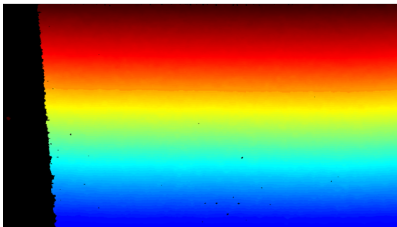
Depth frames



Yaw = 10°, pitch = 0°



Yaw = 10°, pitch = 10°



Yaw = 0°, pitch = 10°



Yaw = 0°, pitch = 0°

Figure 4.28: Depth frame can be used for estimating yaw and pitch angle between target and sensor planes.

#### 4.4 Metrological characterization of the measurement system with depth sensor

loop continues until 10 RGBD frame are collected (d), then the acquisition is automatically stopped. Figure 4.30 shows the live view application provided to the operator in order to have real-time feedback on what is captured by the camera. The Gage R&R study is assessed by asking 3 different operators to

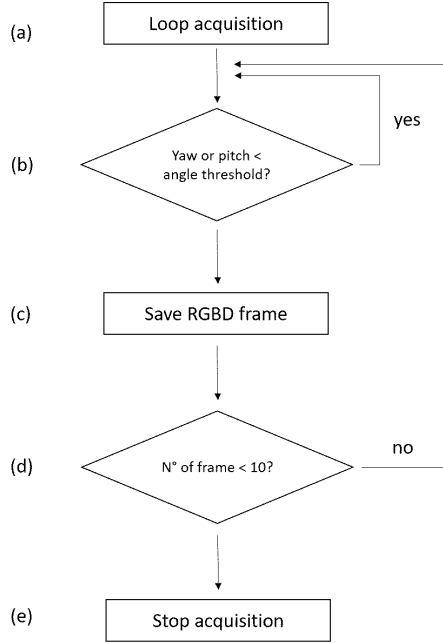


Figure 4.29: Realsense acquisition loop for In order to assess the effect of yaw and pitch angles in terms of repeatability and reproducibility.

perform 10 measurements each in 3 different target cracks following the scheme reported in Figure 4.29. These acquisitions are repeated for six different angular thresholds of  $5^\circ$ ,  $10^\circ$ ,  $15^\circ$ ,  $20^\circ$ ,  $25^\circ$  and  $30^\circ$ . A two-way ANOVA is run on sample of 90 measurements in the 6 different angular scenarios to examine the effect of operator, the type of crack and the angular threshold (Appendix 1). The results show there is a significant interaction between the crack ID and the operator ( $p < 0.05$ ), the null hypothesis can be rejected, the test is statistically signification considering all case scenarios. The Variance analysis shows that the repeatability standard deviation is quite small (0.0074 mm) if compared with the reproducibility (0.0151 mm) where the angles admitted are lower than  $5^\circ$ . As shown in Figure 4.31 this difference grows as the admitted angles increases, increasing the admitted angles the standard deviation associated to the reproducibility increases as well significantly. The contribution % of Variance associated to the Total Gage R&R is minor than 1% where the angles admitted are lower than  $5^\circ$ , that makes it an acceptable measurement system. As shown in Figure 4.32 increasing the admitted angles, the contribution % of

Angular threshold = 10°

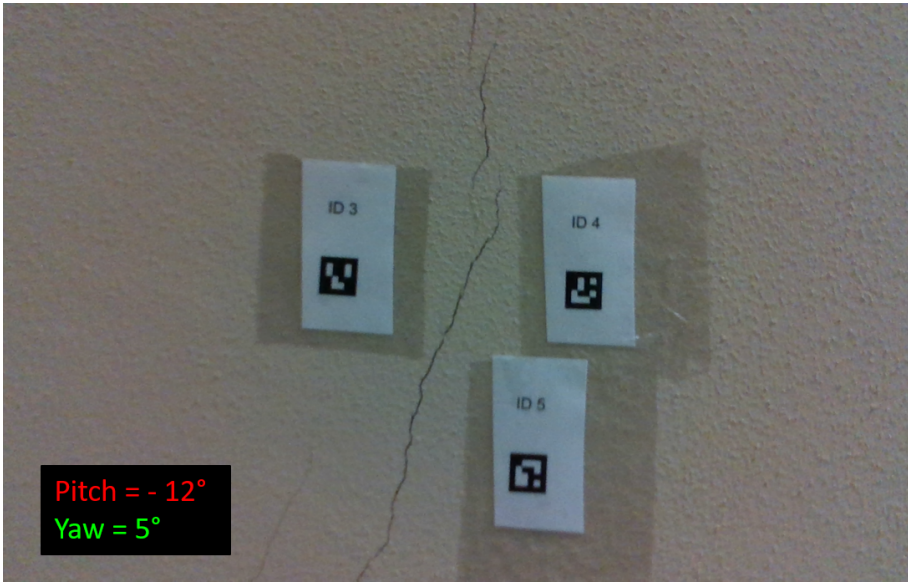


Figure 4.30: Realsense live view for real-time feedback.

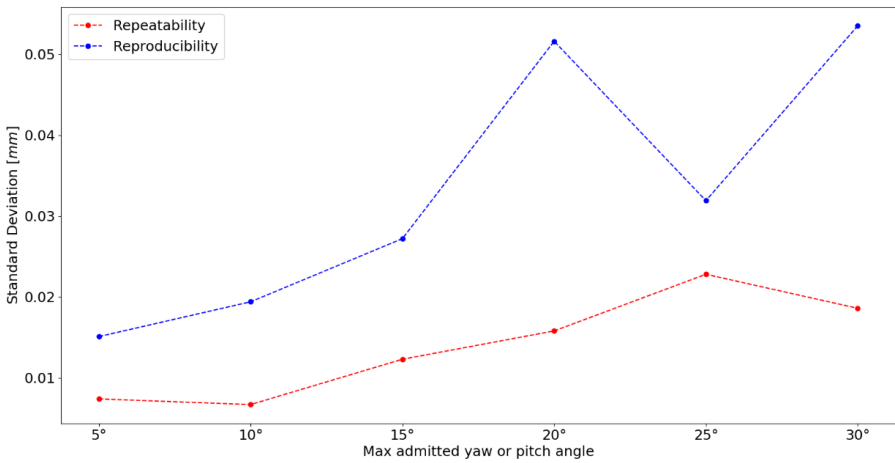


Figure 4.31: Repeatability and Reproducibility Standard Deviation increasing admitted yaw or pitch angles.

#### 4.5 Performance Comparison on reference specimen

Variance associated to the Total Gage R&R increases significantly making the measurement system unacceptable. Such a thing happens with the contribution % of Study Variance associated to the Total Gage R&R (Figure 4.33) and the number of distinct categories (Figure 4.34). According to AIAG, the

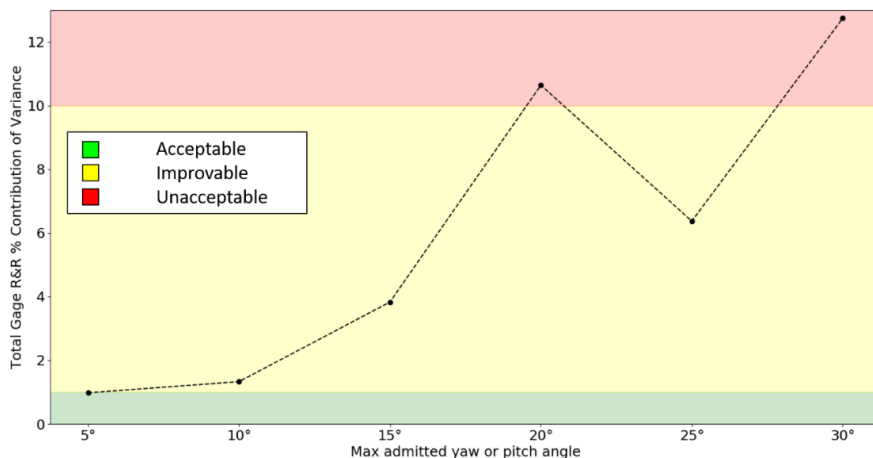


Figure 4.32: Total Gage R&R % Contribution of Variance increasing admitted yaw or pitch angles.

scenario with maximum admitted yaw or pitch angle of 5° is the only that can be considered acceptable. These results underline the importance of assessing the parallelism between sensor and target surface plane. Comparing the gauge R&R test in Table 7 and Table 4.1 it is important to note how a limited system like the 2MP RGBD Intel RealSense is able to obtain performances comparable to those of a much better performing system such as a 24MP DSLR (Nikon D7200) with aspherical lens with very limited distortion (AF-S Micro NIKKOR 60mm f/2.8G ED), all this thanks to the parallelism between sensor and target surface plane.

### 4.5 Performance Comparison on reference specimen

The performance of the algorithm is tested by comparing the measurement between the system described in section 4.3 and 2 different high-precision optical-based measurement systems. For the purposes of comparison, a reference specimen is realized exploiting Fused Deposition Modelling (FDM) 3D printing technology. As shown in figure 4.35, the specimen consists of a central groove with a design thickness of 5mm. Around the groove there are 9 cavities, with a thickness of 0.5mm, on each side, equally spaced by 5mm. The cavities will be used to ensure perpendicularity between the laser and the central line of the

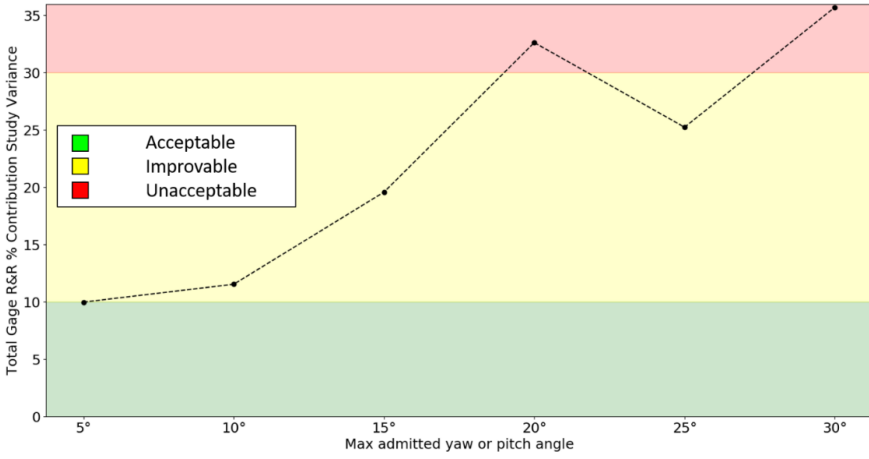


Figure 4.33: Total Gage R&R % Contribution of Study Variance increasing admitted yaw or pitch angles.

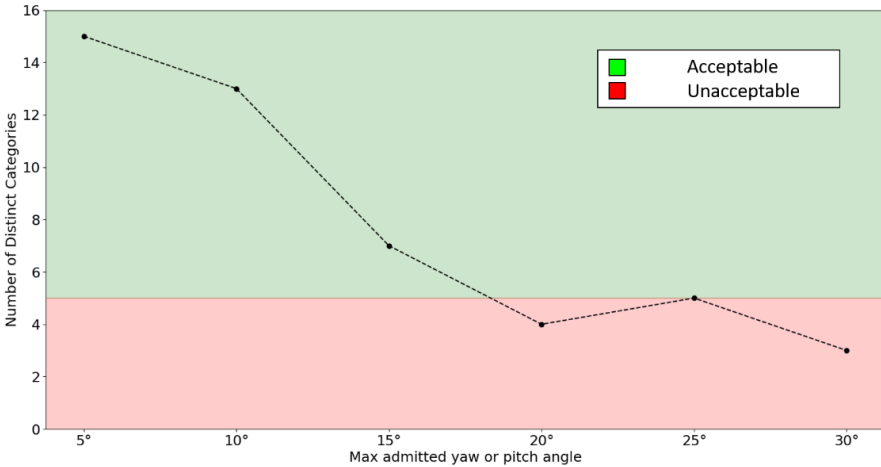


Figure 4.34: Number of Distinct categories increasing admitted yaw or pitch angles.

groove. The measurement systems chosen for the test are:

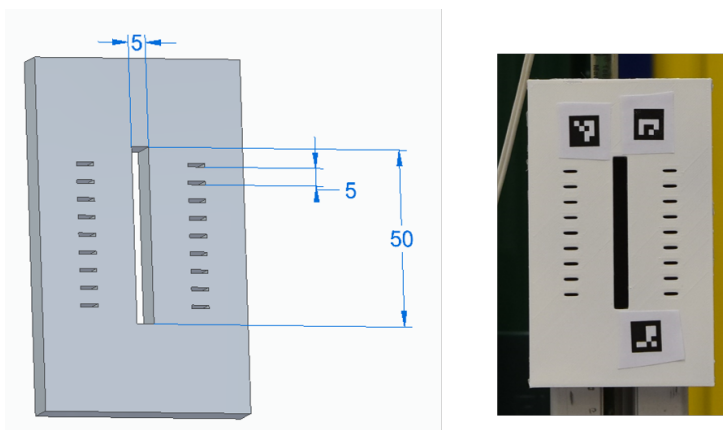


Figure 4.35: 3D printed reference specimen.

- OASIS, a custom telecentric profilometer designed by Zannini and Z4Tech with a declared uncertainty of  $25.5\mu\text{m}$ ,
- Wenglor MSL132, a laser profilometer, already described in section 4.2, with a declared uncertainty of  $33\mu\text{m}$  at a working distance of  $15\text{cm}$ .

The OASIS telecentric profilometer used for performance comparison on reference specimen is a custom measurement system developed by Zannini and Z4Tech. OASIS is based on telecentric optical profilometry for in-production dimensional quality control. Short inspection time, the possibility to simultaneously measure several dimensions on different sides of the part and suitability for automated part handling are key specifications of the device. Figure 4.36 shows the cross-shaped mechanical structure of the instrument. The OASIS instrument is composed of:

- a pair of 61 mm telecentric lens equipped with 5Mpx sensors,
- a pair of  $520\mu\text{m}$  telecentric illuminator.

Traditional lenses are affected by parallax, the perspective error can decrease significantly the measurement accuracy. Telecentric lenses, through their particular optical scheme, overcome this problem. As shown in figure 4.37 at any distance from the lens, a telecentric lens will always have the same field of view. As happens for the lenses, the telecentric illuminators have the ability to emit light rays that are collimated and parallel to the optical axis. A telecentric illuminator increases edge contrast and measurement accuracy by decreasing diffuse reflections from the object. In this way, from the combination

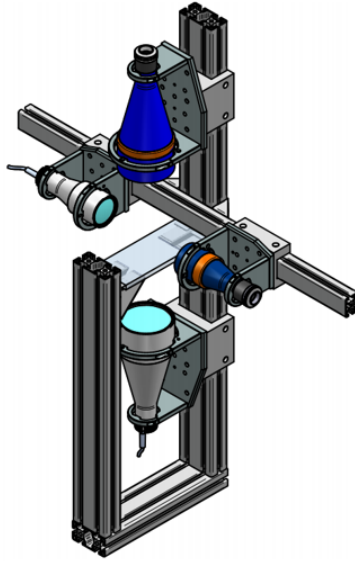


Figure 4.36: Cross-shaped structure of OASIS. The piece to inspect is placed on the glass, between the two pairs of telecentric profilers. [119]

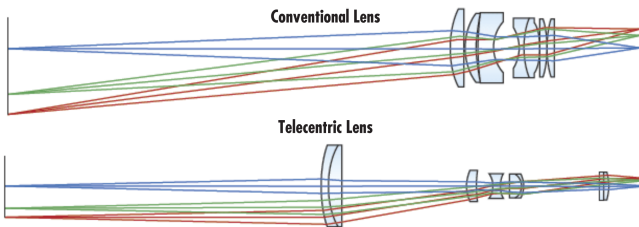


Figure 4.37: FOV comparison of a conventional and telecentric lens (www.edmundoptics.com)



#### 4.5 Performance Comparison on reference specimen

of telecentric lenses and illuminators, profilometer are obtained that are able to return a perfect silhouette of the sample to be measured. OASIS device exploit NI LabVIEW Vision library in order to perform image-based measurements. This device is used to perform measurement of the average groove width in the reference specimen (Figure 4.38). As shown in figure, the same measure-

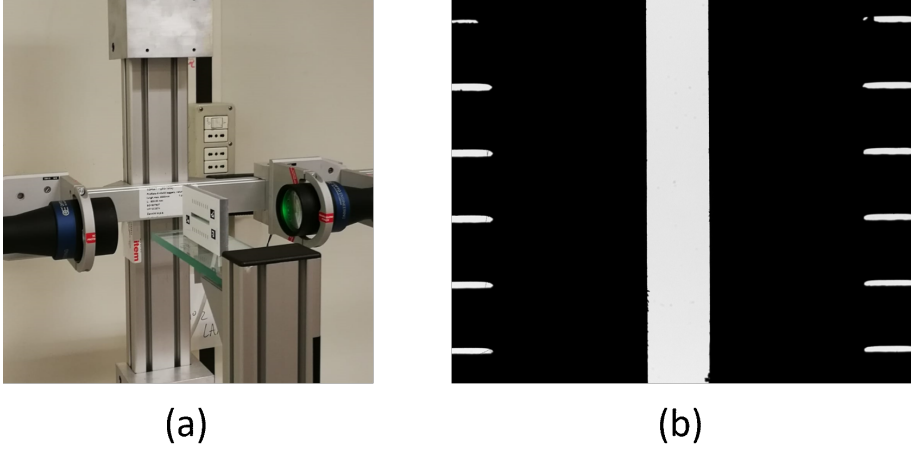


Figure 4.38: On the left, the reference specimen on OASIS device. On the right, the resulting silhouette

ment setup and procedure already described in section 4.2 are used to perform measurement of the average groove width in the reference specimen with the Wenglor MLSL132 laser profilometer. The Crack Analyser algorithm is tested according the same method already described in 4.3. The reference specimen and the camera are they are fixed in an optical bench. Starting from a distance of  $40\text{cm}$ , the camera acquires an image and moves away from the target by  $2\text{cm}$  at each photo, for a total of 30 images collected. From the measurements carried out, it is obtained an extended standard uncertainty  $u(x_i)k = U = 0.018$  (coverage factor  $k = 2$ ) and a sample mean  $x_i = 4.657$ . The extended standard uncertainty appears to be compatible with that obtained in the section 4.3. The results of the measurements of the groove average width of the reference specimen are:

- Crack Analyser device:  $\hat{w} = 4.657 \pm 0.018$ , working distance from 600mm to 1650mm,
- laser profilometer device:  $\hat{w} = 4.631 \pm 0.033$ , working distance of 150mm,
- telecentric profilometer device:  $\hat{w} = 4.660 \pm 0.025$ , working distance of 102mm.

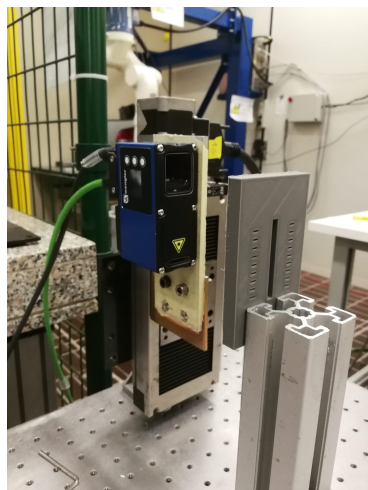


Figure 4.39: Wenglor MSL132 laser profilometer is used to perform measurement of the average groove width in the reference specimen.

The graph in the figure 4.40 shows the  $\hat{w}$  calculated using the Crack Analyser algorithm according to the distance of the camera from the target. From this analysis several considerations can be drawn:

- all the  $\hat{w}$  measured with Crack Analyser method at each distance fall within the measurement uncertainty calculated with the same methodology described in section 4.3,
- the proposed system is able to obtain results compatible with reference measuring instruments known in the state of the art even at very variable working distances. The figure 4.41 shows two images framed respectively at a distance of 700 mm and 1650 mm: despite the enormous difference between the two acquisitions, the results of  $w$  fall within the previously calculated uncertainty interval. This type of behaviour is also in line with what was observed in the chapter 4.1 in controlled scenario environment.
- the proposed system, in addition to achieving the same result, is much more versatile than competing systems. In fact, both for one and for the other, it is necessary to guarantee the perpendicularity between the measured piece surface and the optical axis. This results in a much shorter tool set-up time than the competition,
- for the purposes of the test, it was necessary to guarantee the perfect perpendicularity the axis of symmetry of the groove to be measured, as neither the laser profilometer nor the telecentric one are able to measure the thickness of the groove perpendicularly to the center of the line. The

#### 4.5 Performance Comparison on reference specimen

proposed system, on the other hand, is totally indifferent to the inclination of the groove as it measures the thickness point by point perpendicularly to the axis of symmetry of the groove.

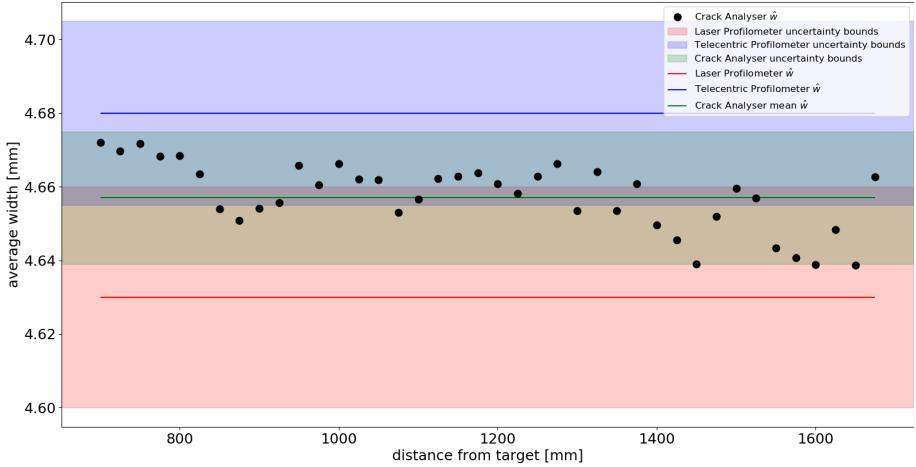
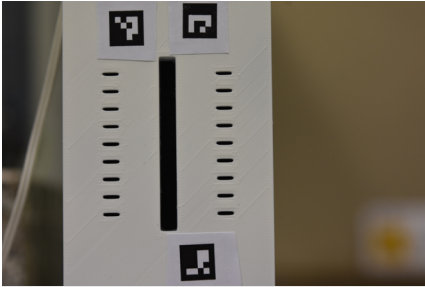


Figure 4.40:  $\hat{w}$  calculated using the Crack Analyser algorithm according to the distance of the camera from the target. In the graph there are superimposed the results of the measurements carried out with the laser and telecentric profilometers in terms of average value and uncertainty bounds.



Distance = 700 mm



Distance = 1675 mm

Figure 4.41: Acquisition setup described in 4.3. On the left, a picture framed at a distance of 600mm. On the right a picture framed at a distance of 1650mm.

# Chapter 5

## Deployment on portable devices for in-field testing

In chapter 3.3 the Sigma Optimization algorithm is modified in order to be used in a real-scale scenario. The result of this work is the Crack Analyser algorithm, a novel automatic image-based approach able to locate and measure crack respecting the metrological constraint imposed by the UNI EN 1992-1-1:2005 standard. Using only one image, this method is able to rapidly locate and measure crack average width and length without requiring any user input or parameters selection in order to reduce the bias introduced by the user that performs the operation. The algorithm is equipped with two different systems: a standard DSLR reflex and the Intel Real Sense depth camera. The Measurement System Analysis proves that both systems have a standard deviation associated to reproducibility and repeatability tests under 0.03 mm. The precision of both system is assessed with the state-of-art gap measurement system: the laser profilometer. In this chapter is explained the architecture design of the system and the application that integrate the Crack Analyser algorithm in order to make it usable and user-friendly even by non-expert operators for in-field testing.

### 5.1 Measurement System design

The measurement system which is the subject of this document consists of a hardware component and a software component. The hardware consists of a system capable of recording and processing images (for example smartphones, dedicated vision systems, . . . ). The measurement set-up (hardware component) is schematically represented in figure 5.1. The system may or may not be able to measure the distance between the camera sensor and target surface. As explained in chapter 3.3, if the device does not have a depth measurement system (such as LIDAR sensors, Time of Flight or depth cameras), in order to calibrate the image, fiducial markers have to be applied to the surface. If the measurement system is equipped with depth sensor, an image acquisition device

(3), equipped on board a UAV (5) or in the hand of an operator, observes a crack on a surface that is not physically reachable (1) and carries out the measurement by calibrating the image acquired through the depth sensor (4). In the other case, an image acquisition device (3), in the hand of an operator, observes a lesion (1) and carries out the measurement by calibrating the acquired image through the use of applied markers on the surface (2). The algorithm (software

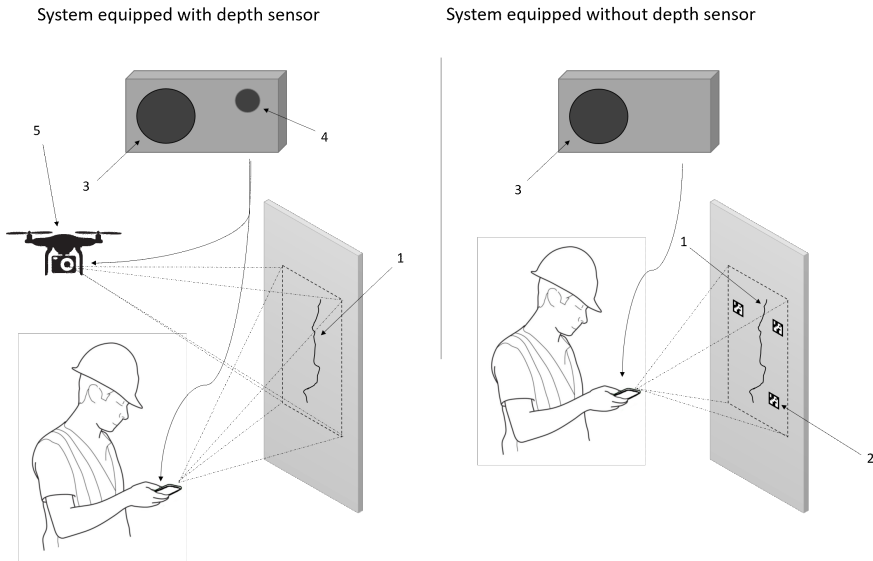


Figure 5.1: The hardware side of the measurement system consists of a system capable of recording and processing images able or not to detect distance between image sensor and target surface

component) for the identification and measurement of the lesion performs the following flow of operations:

1. Acquisition of an RGB image containing the crack.
2. Applying of the Crack Analyser algorithm.
3. Displaying result for user feedback.

## 5.2 Software design

Business Process Modelling Notation (BPMN) is adopted to model the software design of the proposed method from end to end. Figure 5.2 shows the BPMN model modules explanation. In figure 5.3 is shown the BPMN model that shows the software architecture.









BPMN components	Description
 <p>Start      Intermediate      End</p>	<p>These symbols represent the trigger that starts, modifies, or completes a process.</p>
 <p>Task</p>	<p>The Task represents a particular activity performed by a person or a system. The task is the basic unit of the BPMN. It can be a looped task, a user task or a service task based on symbols applied. It can be detailed by adding sub-processes.</p>
 <p>Gateway</p>	<p>A gateway is represented with a diamond shape and represents decision point that can adjust the path based on condition and events. It can be inclusive, exclusive, parallel, and based on data depending on the contained symbol.</p>
 <p>Sequence flow</p>	<p>The sequence flow row shows the order of activities/tasks to be performed.</p>
 <p>Message flow</p>	<p>The message flow row depicts messages that flow across organization boundaries. It should not connect events or activities.</p>
 <p>Database</p>	<p>BIM2TWIN centralized database for storing and updating the produced data.</p>
 <p>Data Object</p>	<p>This icon symbolizes either input data or output data. A data object shows what data is necessary for an activity.</p>
 <p>Pool and lanes</p>	<p>A pool contains many tasks, triggers, and gateway. A lane represents a sub-part of the pool.</p>

Figure 5.2: BPMN model modules explanation.

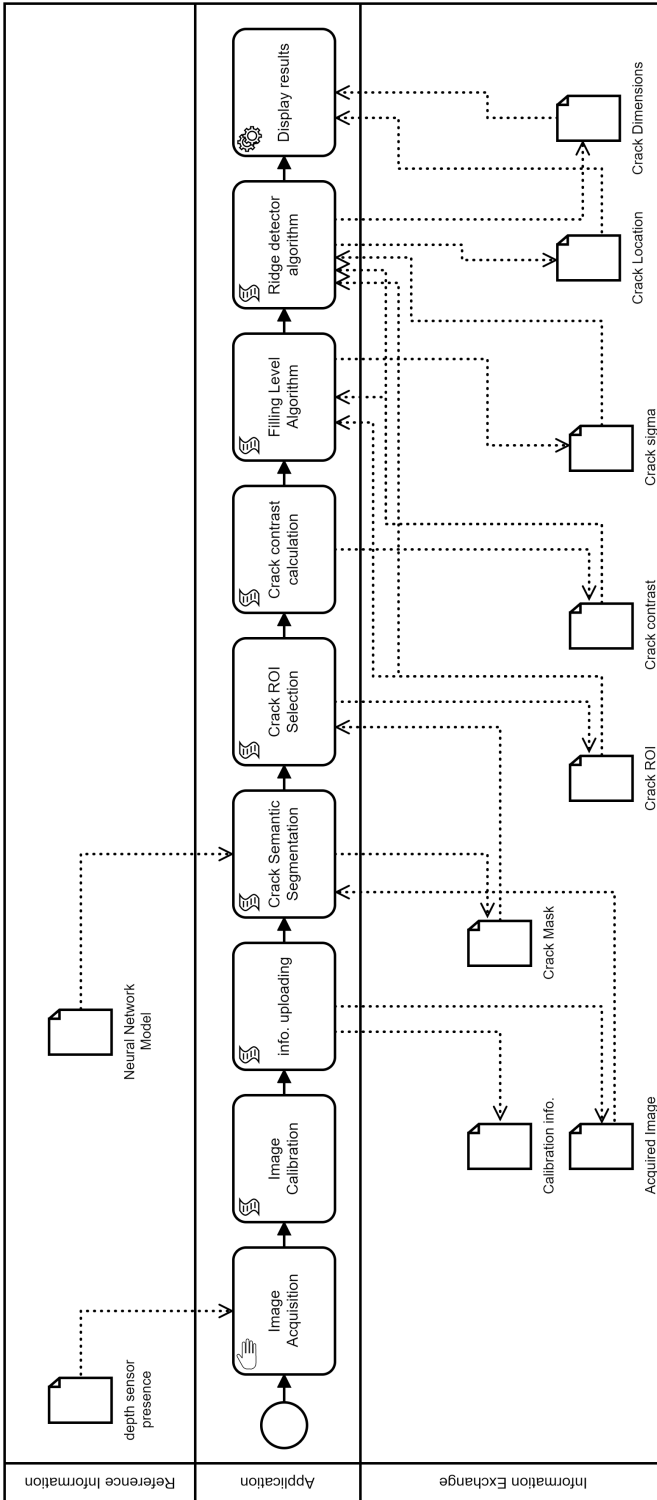


Figure 5.3: BPMN model for software design



### 5.3 Crack Analyser App

The Crack Analyser algorithm is integrated in a Flutter application, an open-source user interface Software Development Kit (SDK) by Google for cross-platform development. It can be used for writing applications for Android, iOS, Linux, Mac, Windows, and web-app from a single codebase. Flutter apps are based Dart language [120] and make use of many of the language's more advanced features. [121]. Figure 5.4 shows the architecture design of the application: the front-end side run on Flutter application mounted on the device, the back-end side is entirely managed by a python-based server where the Crack Analyser algorithm run. The front-end application can be easily embedded on any Android/iOS smartphone or dedicated Linux/Windows-based hardware since the cross-platform nature of Flutter SDK. The front-end and the back-end server communicate through HTTP requests with Application Programming Interfaces (APIs) managed by Flask, a micro web framework written in Python [122]. The APIs allows this two systems to communicate each other providing

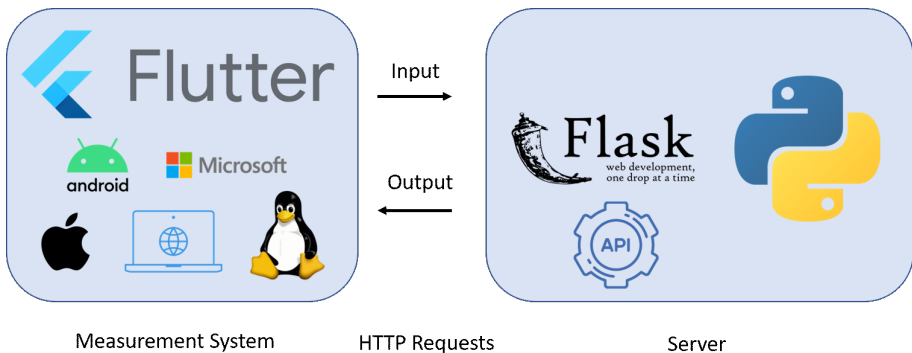


Figure 5.4: Architecture design for Crack Analyser app. The front-end run on Flutter application, the back-end part is entirely managed by a python-based server.

the language and contract for how the two systems interact and determines how information can be transferred. The exposed API use Http requests to receive and give informations. The points of entry in a communication channel when two systems are interacting are the API endpoints. In particular, the developed APIs used comply with the REST (REpresentational State Transfer) protocol, which is why they are called REST APIs [123]. The designed API consists of a list of endpoints:

1. `http : //SERVER_IP/image` : this endpoint receives as input the acquired image, the user credentials, the camera configuration. The image is uploaded to the server in a dedicated directory based on user

credentials. The outputs are the image location in the server and a boolean that represents the feedback of uploading procedure.

2. `http://SERVER_IP/detect_line` : this endpoint receives no inputs from front-end application. It gives the command to server to applying the Crack Analyser algorithm to the image previously uploaded. If the image comes from a camera not equipped with depth sensor the algorithm will look for fiducial marker in order to calibrate the input image. The resulting image and the Filling Level plot are saved on server according to user credentials previously received. Returns as output the processed image, the crack masked with semantic segmentation, the Filling level plot image and the value of measured crack average width.

Figures 5.5, 5.6 and 5.7 shows the Flutter front-end application developed on Android device. The application is composed of three screens: the login/register screen, the home screen, and the measure screen. The starting page is the login/register screen (a). In this page are present two buttons: the login button (1) where the user can insert its credentials, and the register button (2) where the user can register a new account and log in. The login and register data are stored and verified through Google Firebase Authentication. Once the user is register or logged in, the app move to home screen (b). In this page are present four buttons: Camera, Gallery, Detect Line and RealSense. With the Camera button (3) the application will open the default device camera in order to acquire an image. With the Gallery button (4) the application will open the default device image gallery in order to use photos stored on the device as input for the measurement. By default the RealSense and DetectLine buttons are greyed out. A Intel RealSense camera can be plugged in to the device through USB OTG cable. If a camera RealSense camera is detected on the device, the RealSense button became green and can be used for acquiring RGBD images. The image, regardless of how it was acquired, is then shown on the screen (c). Through `/image` HTTP request the device automatically uploads the input image to the back-end server sharing infos about the camera and the user. If the server communicates that the upload is been done, the DetectLine button (5) become green and if pressed the user land on Detect Line screen (d). The Detect Line page contains three buttons: Measure, Graph FS and Mask buttons. By default the Graph FS and Mask buttons are greyed out. Pushing on Measure button (7) the `/detect_line` HTTP request is sent to the server. The server receive as input the acquired image and data regarding the user and camera infos. Once the image is processed by the python-based back-end, the server give as output the processed image, the crack masked with semantic segmentation, the Filling level plot image and the value of measured crack average width. Once the data are received with positive feedback from

the server the processed image and the crack average width are displayed on the screen and the Graph FS and Mask buttons are activated (e). The graph FS button (8) allows the user to have a visual feedback about the Crack Analyser algorithm, the red dot on the plot represents the  $\sigma_{FL}$  used as input for Steger algorithm (f). The Mask button (9) allows the user to have a visual feedback about the semantic segmentation of the crack, incorrect segmentation can lead to errors in estimating average crack width.

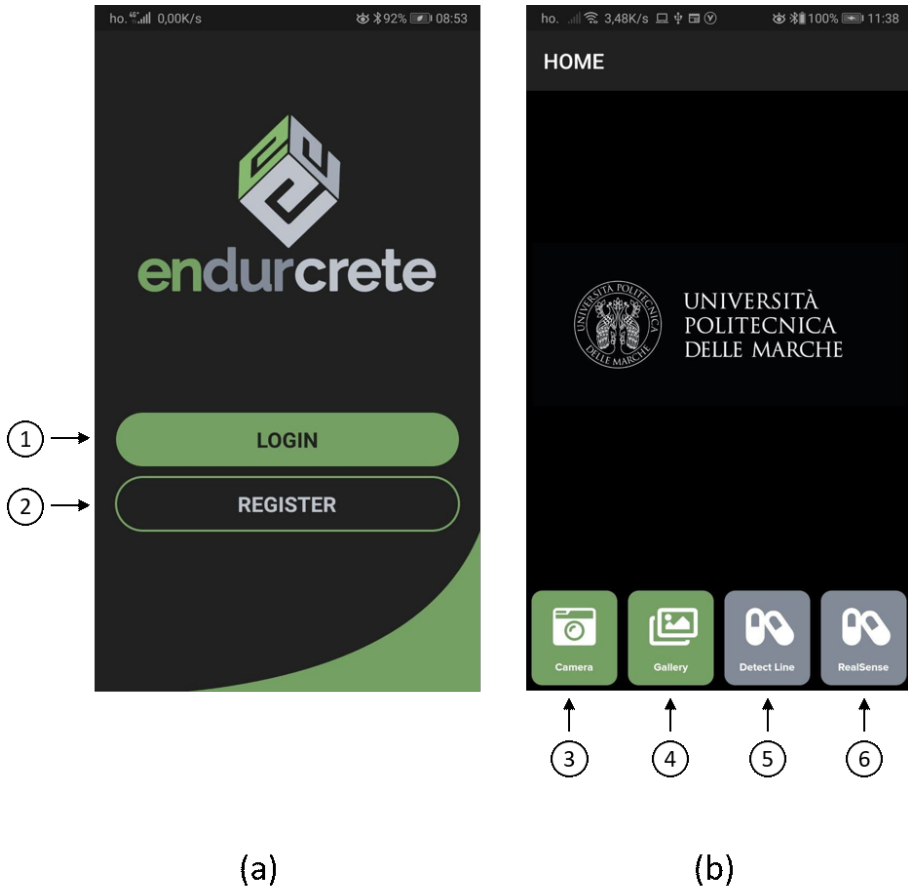
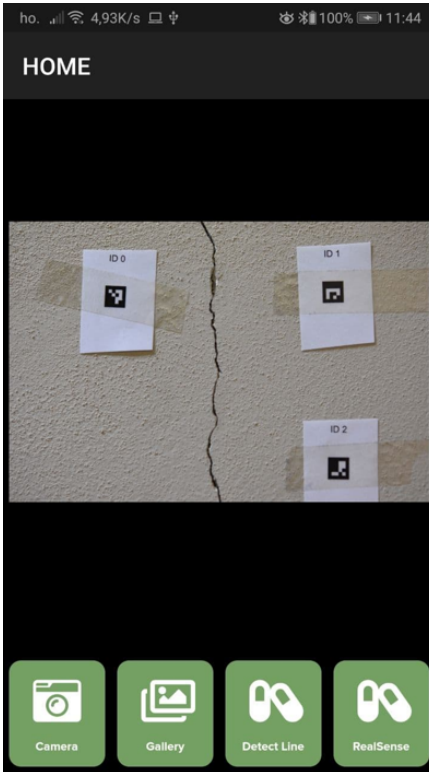
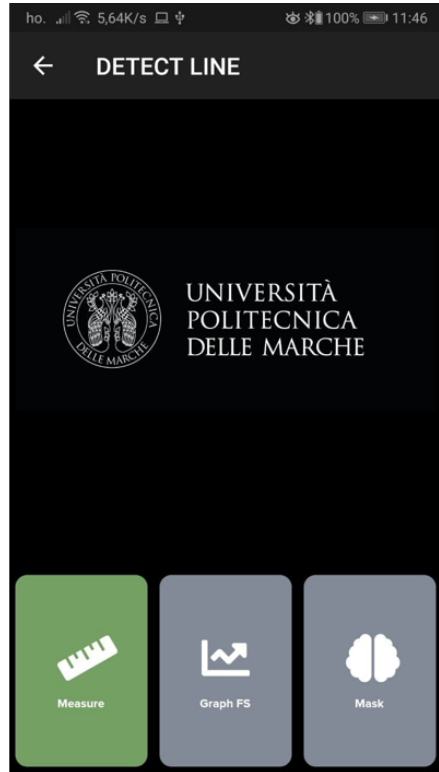


Figure 5.5: Front-end application run on Android device.

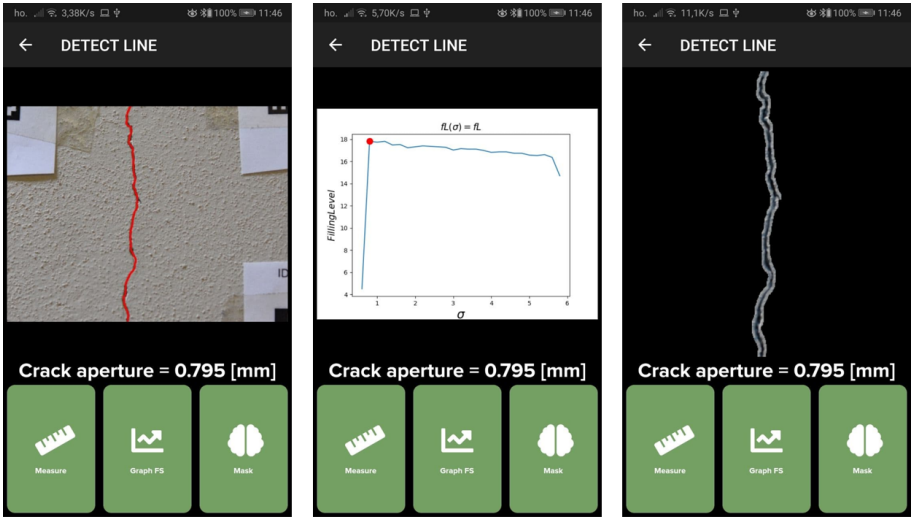


(c)



(d)

Figure 5.6: Front-end application run on Android device.



(e)

(f)

(g)

Figure 5.7: Front-end application run on Android device.



# Chapter 6

## In-field testing results

The Crack Analyser algorithm is tested through the application developed in chapter 5 in the facilities of an EnDurCrete project partner, the Santa Barbara Foundation. The Santa Barbara Foundation is a Public Foundation of indefinite duration and non-profit nature, attached to the Ministry of Economy and Finance of the Junta de Castilla y León through the General Directorate of Energy and Mines that collaborates with the social welfare and the promotion of those cultural activities aimed at promoting knowledge and scientific and technical research. Today, the Santa Barbara Foundation is a consolidated entity, which works on training tasks, technological tests and research, development and innovation projects, always acting within the field of applied technology, security, and technological progress, developing its activity in its centres of El Bierzo Labor School, located in the municipalities of Folgoso de la Ribera and Torre del Bierzo (León). Santa Barbara Foundation offers professional training courses for employment and specialized continuous training in various training areas: Underground Construction, Earthworks, Electromechanics, Renewable Energies and the Environment. Its specialized staff and facilities allow it to provide eminently practical training, with full-scale pitches suitable for adequate learning aimed at insertion and professional improvement (Figure 6.1). FSB mining tunnels facilities were used within the EnDurCrete project for corrosion and structural health monitoring of tunnel precast concrete segments for tunnel lining and mortar grouting formulations and non-destructive inspection techniques for detection of delamination after installation (Figure 6.2). The system developed in 5 is used to inspect and measure cracks on the tunnel precast concrete panels and mining tunnel surface. This type of environment seems to be ideal for testing the measurement system developed in this thesis work as it is the closest to a real use case, with the addition of being in a hostile, dirty and not very illuminated. The classic crack ruler (range from 0.010 to 2.50 mm) was used as the reference measuring instrument, designed to provide inspectors for determining the width of a crack in concrete or other building materials (Figure 6.3).

In total, two cracks were found on the panels and one crack on the external



Figure 6.1: Santa Barbara Foundation mining tunnels



Figure 6.2: Corrosion and structural health monitoring of tunnel precast concrete segments for mining tunnel demonstrator in EnDurCrete Project



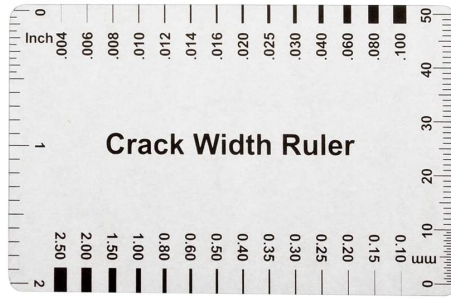


Figure 6.3: Crack ruler used as reference measuring instrument.

surface of the mine. In the first case the cracks were barely visible to the naked eye, so it was necessary to use a high spatial resolution system (tablet camera with 18MPx) applying fiducial markers on the concrete surface (Figure 6.4). Despite the reduced thickness, the algorithm was able to identify the crack and measure its thickness in both cases considered, reporting a width of 0.1 mm. This result is validated through the use of a crack ruler (Figure 6.5).

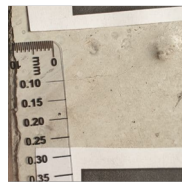
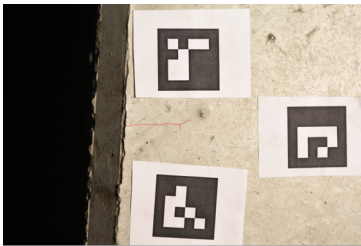
The most difficult case is the measurement of the cracks on the internal surface of the mining tunnel. In this case, in fact, it is a question of analysing a crack on an optically non-cooperative surface with very high roughness and made by dark material that minimises the contrast between target crack and surrounding surface. Due to the high roughness it is impossible to apply targets on the surface, therefore it was necessary to use the device equipped with a depth sensor, the Intel RealSense depth camera (Figure 6.6).

The figure shows the complete calculation flow for the measurement of the average width of the crack in the surface in the tunnel. The calibrated RGBD image is acquired (a), then AI is exploited for the binary semantic segmentation in order to find the region of interest that contain the crack (b). The mask obtained (c) is used to find analyse the crack image histogram (d) and calculate the crack contrast  $h$  (e) by calculating the distance between histogram peaks. Then the developed Filling Level algorithm is applied to the image and  $\sigma_{FL}$  is calculated. In the end, Steger algorithm is applied to the image with optimal parameters and the  $\hat{w}$  is obtained. The  $\hat{w}$  turns out to be 2 mm, exactly like what the crack ruler reads. Also in this case the developed system proved to be suitable for the measurement of cracks in a working environment and was able, through the acquisition of a single image, to localize and measure the crack.

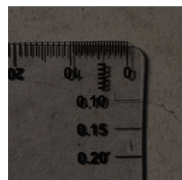
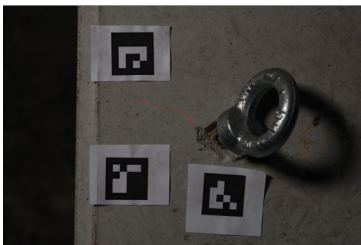
In order to test the device in uncontrolled lighting conditions, the analysis of cracks in bridge inspections was considered. The bridge chosen for the inspection is located on the "Centro Cadore" Lake (46.44909, 12.40585). An approximately 1.5 mm lesion measured through the crack ruler was found in the lateral support structure of the bridge (Figure 6.8).



Figure 6.4: The cracks present on the panels are barely visible, therefore it is necessary to use the high spatial resolution system.



(1)



(2)

Figure 6.5: Comparison between the proposed method, on the left, and the traditional method with a ruler, on the right. In both cases the device returns the measurement of 0.1mm.



Figure 6.6: Due to the high roughness it is impossible to apply targets on the surface, therefore it is necessary to use the device equipped with a depth sensor.

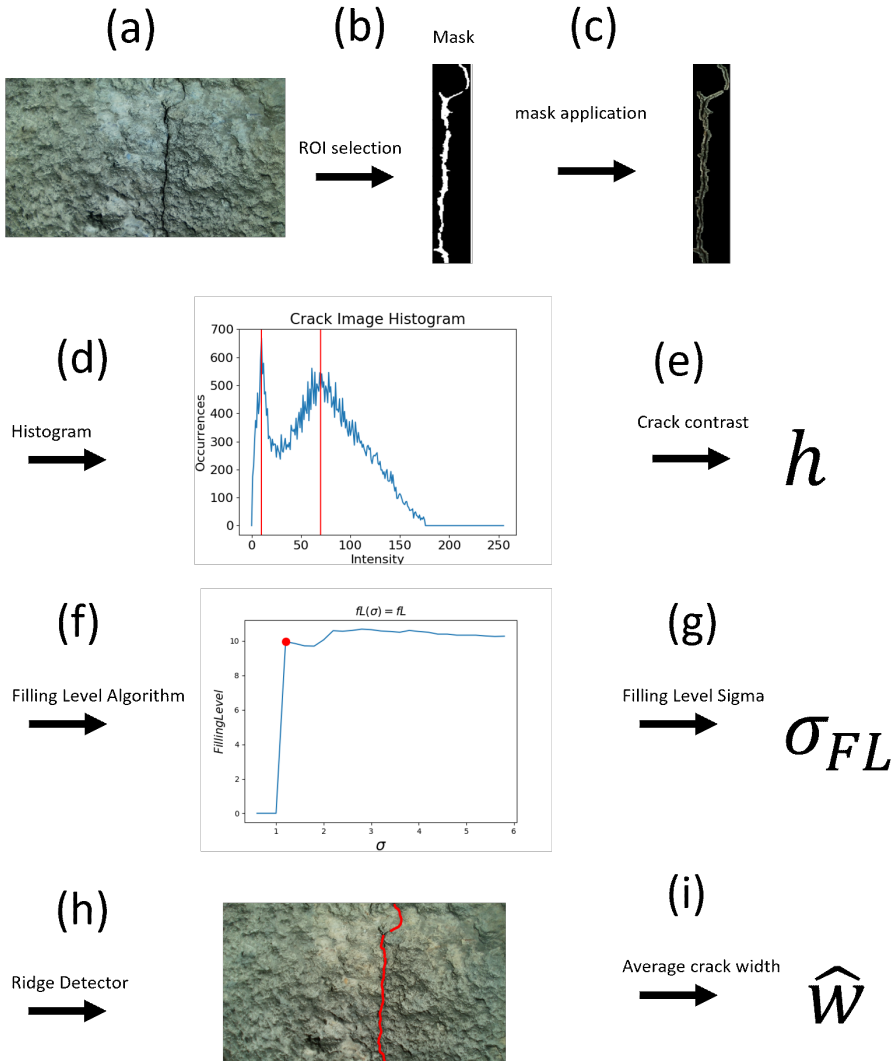


Figure 6.7: Complete calculation flow for the measurement of the average width of the crack in the surface in the tunnel.



Figure 6.8: The bridge chosen for the inspections in conditions of natural light is located on the "Centro Cadore" Lake

Analogously to the previous case, the figure 6.9 shows the complete calculation flow for the measurement of the average width of the crack located in the lateral support structure of the bridge. Unlike what was acquired in the tunnel, however, in this case, it was possible to carry out the acquisition through the system with and without the use of a depth sensor. In both cases, an average thickness along the crack of 1.4mm was measured.

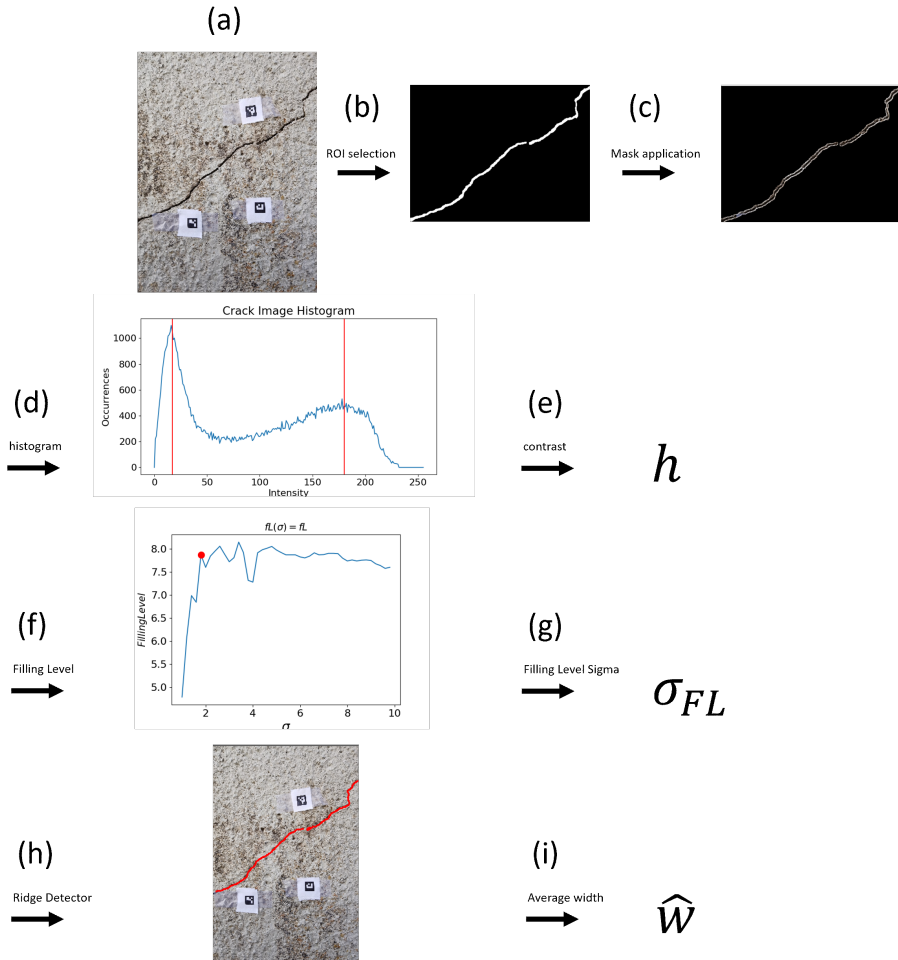


Figure 6.9: Complete calculation flow for the measurement of the average width of the crack located in the lateral support structure of the bridge.

# Chapter 7

## Conclusions and future work

### 7.1 Conclusions

This thesis work proposes a novel automatic image-based approach able to locate and measure cracks on concrete surfaces respecting the metrological constraint imposed by UNI EN 1992-1-1:2005 standard. Using only one image, the developed method is able to automatically and rapidly locate and measure the average width and length of a crack in an existing concrete structure. The measurement system developed exploits a single camera working in the visible range to acquire a digitized image of the structure being inspected. The software component of the system receives as input the single image framing the crack and gives as output an augmented image where the crack is highlighted as well as its average/max width and length. The measure of the crack width is performed perpendicularly to the crack central line with sub-pixel accuracy. The measurement system has been deployed on a smartphone for operator-based manual inspections as well on embedded systems for remote inspection with robots or Unmanned Aerial Vehicles (UAVs). The methodology is validated and characterized from the metrological point of view by means of multi-operator analysis of variance and type-A uncertainty estimation. The metrological characterization of the proposed system is performed through a structured test, where for each acquired image the relative angles between the sensor surface and the target surface are known 4.1, and a test that does not adopt any type of control, thus simulating a real-scale application of the measuring device 4.2. In the first type of test, an anthropomorphic robot is used to acquire images at known angles and define the working range of the developed device. This type of test made it possible to demonstrate that:

- the error in calculating the crack width does not depend on the distance, and therefore on the number of pixels representing the crack in the framed image.
- the error increases as the angular misalignment between the measuring plane and the sensor plane increases.

- By varying the angle of installation between  $\pm 30^\circ$  the maximum error is always below 0.1 mm, regardless of the distance and therefore of the number of pixels that represent the crack in the framed image.
- The standard deviation of the measurement increases as the angle of misalignment increases in a quadratic manner. With a maximum angle of misalignment of  $5^\circ$  a standard deviation of 0.0025mm is obtained, while with  $30^\circ$  you get a standard deviation of 0.03mm.
- Based on the thickness of the crack in pixels, it is possible to establish sigma ranges of optimal functioning of the algorithm within which the optimal sigma falls when the laying angle varies, with a maximum range allowed between  $\pm 30^\circ$ .
- The standard deviation associated with the 100 measurements carried out on the same reference sample by an operator is equal to 0.013mm.

Subsequently the results obtained from these tests will be compared with the results obtained through uncontrolled measurements in a real-case scenario. The tests carried out on a real crack demonstrate that the standard deviation associated with the 100 measurements carried out on the same reference sample by an operator is equal to 0.02mm.

Furthermore, the Sigma Optimization algorithm, developed in chapter 3.1.2, led to the creation of an innovative optimization technique for C. Steger's line detection algorithm for lab-scale controlled environments. However, there are some aspects to be improved:

- the algorithm that identifies the plateau in the Filling Level trend is currently based on interpolation and gradient concepts. The use of a neural network that, through an inferential approach, leads to the accurate selection of the  $\sigma_{FL}$  position could lead to improvements and reduce the uncertainty associated to the estimation of this parameter.
- The algorithm that calculates the contrast of the crack is based on the calculation of the distance of the peaks in the histogram of the image. Although its proven effectiveness, in some particular cases may fails. Exactly as in the case of the identification of the  $\sigma_{FL}$ , it would be possible to exploit an inferential approach as a further control and prediction of the parameter to be selected.
- The algorithm developed cannot handle large variations in contrast along the detected line. This means that if the crack involves several materials of different colours, or is crossed by different contrast scenes on the same material, there would be an error in the estimation of the line width.



This problem could be overcome through different hardware and software approaches. From the software point of view, a logic could be implemented for which the identified line is considered as the union of several lines with different contrast, in this way the thickness would be calculated separately for each line and then joined subsequently. From the hardware point of view, one could think of an approach based on hyper-spectral imaging. Thanks to the use of hypercubes, in fact, the effect of the variation in contrast of the image could be greatly attenuated and concentrated solely on the target crack by excluding extraneous elements from the scene.

- Although the trained neural network has proved to be extremely effective in all situations, in a perspective of large-scale use, it would be necessary to develop a control procedure on semantic segmentation step. In the event of an error in the segmentation, a manual correction could be simply made by the operator who is performing the measurement via the touch screen of the device.

## 7.2 Future work

Given the ease of use of the developed system, it has stimulated interest in the device also within the BIM2TWIN research project where UNIPM is partner and leader of work package 4 "Progress and quality monitoring of surface/textural work". From a survey conducted as part of the project, which involved more than 300 companies in the construction sector from 15 different countries from all over the world, the importance of having measurement systems characterized by high ease of use, rapidity of execution and possibility of remote access to data. It also emerged that the use of smartphones within the construction site to collect data and carry out reports in relation to reporting defects and deviations from the initial project has become increasingly widespread. For this reasons, the next step of this work will certainly be to integrate the entire developed method on portable devices with integrated LIDAR (e.g., iPhone 13 pro or iPad pro). In this way you can guarantee maximum performance and all the necessary sensors in a single device.

The strategies developed in this thesis work can be easily extended from concrete inspection applications to any other context where a surface quality control targeted to the identification of eventual damages/defects is required. The method has proved to be particularly valid in the automatic detection of defects, scratches and non-conformities on surfaces of various kinds, such as for example the surfaces of car frames or rims. The developed algorithm will be used within the European research project BIM2TWIN for the inspection of masonry constructions, in particular the assessment of mortar joint thickness deviation,

mortar joint position deviation and mortar joint alignment deviation. In this project the possibility of increasing the algorithm performance through hyper-spectral imaging technology will also be investigated. In fact, it is thought that exploiting particular wavelengths it is possible to investigate other properties of the crack. BIM2TWIN project will try to take a step forward implementing the use of hyper-spectral imaging to detect chemical species or moisture penetration that can reduce the service life of the building.

# Bibliography

- [1] K. Gkoumas. Research and innovation in bridgemaintenance, inspection and monitoring. *A European perspective based on the Transport Research and Innovation Monitoring and Information System (TRIMIS)*, 2019.
- [2] Portland Cement Association. Types and causes of concrete deterioration. *CONCRETE Information*.
- [3] ACI Committee 224. Causes, prevention, and repair of cracks in concrete. *American Concrete Institute, Farmington Hills, Michigan*, 1993.
- [4] DOUGLAS J. HAAVIK. Evaluating concrete cracking by measuring crack width. *LA HABRA, CALIFORNIA*, 1990.
- [5] *UNI EN 1992-1-1:2005 Eurocodice 2 - Progettazione delle strutture di calcestruzzo - Parte 1-1: Regole generali e regole per gli edifici*.
- [6] S. Garrido-Jurado; R. Munoz-Salinas; F. J. Madrid-Cuevas; and R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 2016.
- [7] F. J. Romero-Ramirez; R. Munoz-Salinas; and R. Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 2018.
- [8] E. Minnetti. A smartphone integrated hand-held gap and flush measurement system for in line quality control of car body assembly. *Sensors*, 2020.
- [9] <http://www.endurcrete.eu/>.
- [10] S. Nabulsi S. Yehia, O. Abudayyeh and I. Abdelqader. Detection of common defects in concrete bridge decks using nondestructive evaluation techniques. *Journal of Bridge Engineering*, 2007.
- [11] J. S. Popovics-R. W. Arndt S.-H. Kee, T. Oh and J. Zhu. Nondestructive bridge deck testing with air-coupled impact-echo and infrared thermography. *Journal of Bridge Engineering*, 2012.

## Bibliography

- [12] David O'Brien B. Mulry, M. Jordan. Automated pavement condition assessment using laser crack measurement system (lcms) on airfield pavements in ireland. presentation structure. *International Conference on Managing Pavement Assets*, 2015.
- [13] J. Laurent. Using 3d laser profiling sensors for the automated measurement of road surface conditions. *RILEM Bookseries*, 2012.
- [14] W. Ouyang and B. Xu. Pavement cracking measurements using 3d laser-scan images. *Measurement Science and Technology*, 2013.
- [15] Baoxian Li ;Kelvin C. P. Wang; Allen Zhang ;Yue Fei, , and Giuseppe Sollazzo. Automatic segmentation and enhancement of pavement cracks based on 3d pavement images. *Journal of advanced transportation*, 2019.
- [16] M. Abdellatif T. H. Awad H. M. Shehata, Y. S. Mohamed. Depth estimation of steel cracks using laser and image processing techniques. *Alexandria Engineering Journal*, 2018.
- [17] I. H. Kim; H. Jeon; S. C. Baek; W. H. Hong; and H. J. Jung. Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. *Sensors*, 2018.
- [18] W. Niblack. An introduction to digital image processing. *Strandberg Publishing Company*, 1985.
- [19] J. Bernsen. Dynamic thresholding of gray-level images. 1986.
- [20] J. Sauvola and M. Pietikak. Adaptive document image binarization. 2000.
- [21] C. Wolf; J-M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Analysis and Applications*, 2003.
- [22] N. Otsu. A threshold selection method from grey level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979.
- [23] K. Davis. Activity recognition based on inertial sensors for ambient assisted living. *19th International Conference on Information Fusion (FUSION)*, 2016.
- [24] K. Bose and S. Kumar Bandyopadhyay. Crack detection and classification in concrete structure. *International Journal for Scientific Research and Development*, 2016.
- [25] H. Kim; E. Ahn; S. Cho; M. Shin; and S. H. Sim. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement and Concrete Research*, 2017.

- [26] A. Rimkus; A. Podvieszko; and V. Gribniak. Processing digital images for crack localization in reinforced concrete members. *Procedia Engineering*, 2015.
- [27] A. Ebrahimkhanlou; A. Farhidzadeh; and S. Salamone. Multifractal analysis of crack patterns in reinforced concrete shear walls. *Structural Health Monitoring*, 2016.
- [28] M. Kamaliardakani; L. Sun; and M. K. Ardakani. Sealed-crack detection algorithm using heuristic thresholding approach. *Journal of Computing in Civil Engineering*, 2016.
- [29] A. M. A. Talab; Z. Huang; F. Xi; and L. Haiming. Detection crack in image using otsu method and multiple filtering in image processing techniques. *Optik*, 2016.
- [30] Syed Yasir Alam; Saliba Jacqueline ; Ahmed Loukili. Fracture examination in concrete through combined digital image correlation and acoustic emission techniques. *Construction and Building Materials*, 2014.
- [31] L. Ying and E. Salari. Beamlet transform-based technique for pavement crack detection and classification. *Computer-Aided Civil and Infrastructure Engineering*, 2010.
- [32] H. D. Cheng; X. J. Shi; and C. Glazier. Real-time image thresholding based on sample space reduction and interpolation approach. *Journal of Computing in Civil Engineering*, 2003.
- [33] H. Oliveira and P. L. Correia. Automatic road crack segmentation using entropy and image dynamic thresholding. *17th European Signal Processing Conference*, 2009.
- [34] T. Yu; A. Zhu; and Y. Chen. Efficient crack detection method for tunnel lining surface cracks based on infrared images. *Journal of Computing in Civil Engineering*, 2017.
- [35] Y. J. Cha; W. Choi; G. Suh; S. Mahmoudkhani;. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 2018.
- [36] Yann LeCun; Yoshua bengio. Convolutional networks for images, speech, and time-series. 1995.
- [37] Y. LeCun; B. Boser; J. S. Denker; D. Henderson; R. E. Howard; W. Hubbard and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.

## Bibliography

- [38] Yann LeCun; J. S. Denker; S. Solla; R. E. Howard and L. D. Jackel. Optimal brain damage. *Advances in Neural Information Processing Systems*, 1990.
- [39] Y LeCun; B Boser; JS Denker; D Henderson; RE Howard; W Hubbard. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 1989.
- [40] Y LeCun; L Bottou; Y Bengio; P Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- [41] P Sermanet; D Eigen; X Zhang; M Mathieu; R Fergus; Y LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *International Conference on Learning Representations (ICLR 2014)*, 2014.
- [42] M. Serrhini; C. Silva; and S. Aljahdali. Innovation in information systems and technologies to support learning research. *Springer International Publishing*, 2020.
- [43] Alexander Shustanova; Pavel Yakimova. Cnn design for real-time traffic sign recognition. *Procedia Engineering*, 2017.
- [44] Yihui Wu; Yulong Liu; Jianmin Li; Huaping Liu; Xiaolin Hu. Traffic sign detection based on convolutional neural networks. *The 2013 International Joint Conference on Neural Networks*, 2003.
- [45] S.; Memon M.; y Rahoo A. Narejo, S.; Talpur. An automated system for traffic sign recognition using convolutional neural network. *Glosas de innovacion aplicadas a la pyme*, 2020.
- [46] Convolutional neural net face recognition works in non-human-like ways. Peter j. b. hancock; rosyl s. somai and viktorija r. mileva. *Royal Society Open Science*, 2020.
- [47] A. R. Syafeeza; M. Khalil-Hani ; S. S. Liew ; R. Bakhteri. Convolutional neural network for facerecognition with pose and illumination variation. *International Journal of Engineering and Technology*, 2014.
- [48] Steve Lawrence; C Lee Giles; Ah Chung Tsoi; Andrew D Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 1997.
- [49] Pranav KB; Manikandan J. Design and evaluation of a real-time face recognition system using convolutional neural networks. *Third International Conference on Computing and Network Communications*, 2020.

- [50] M. Coskun; A. Ucar; O. Yildirim and Y. Demir. Face recognition based on convolutional neural network. *International Conference on Modern Electrical and Energy Systems*, 2017.
- [51] Soad Almadby ; and Lamiaa Elrefaei. Deep convolutional neural network-based approaches for face recognition. *Applied Sciences*, 2019.
- [52] Baohua Qiang ; Ruidong Chen ; Mingliang Zhou ; Yuanchao Pang ; Yijie Zhai and Minghao Yang. Convolutional neural networks-based object detection algorithm by jointing semantic segmentation for images. *sensors*, 2020.
- [53] Object Detection with Deep Learning: A Review. Zhong-qiu zhao; peng zheng; shou-tao xu; and xindong wu. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 2019.
- [54] Kaidong Li; Wenchi Ma; Usman Sajid; Yuanwei Wu; Guanghui Wang. Object detection with convolutional neural networks. *Computer Vision and Pattern Recognition*, 2019.
- [55] Luis Tobias; Aurelien Ducournau; Francois Rousseau; Gregoire Mercier; Ronan Fablet. Convolutional neural networks for object recognition on mobile devices: A case study. *International Conference on Pattern Recognition*, 2016.
- [56] F. Chen and M. R. Jahanshahi. Nb-cnn: Deep learning-based crack detection using convolutional neural network and naive bayes data fusion. *IEEE Transactions on Industrial Electronics*, 2018.
- [57] Luqman Ali; Fady Alnajjar; Hamad Al Jassmi ; Munkhjargal Gocho; Wasif Khan; M. Adel Serhani. Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures. *Sensors*, 2021.
- [58] N.; Sri Preethaa K.R. Kim, B.; Yuvaraj. Surface crack detection using deep learning with shallow cnn architecture for enhanced computation. *Neural Computing and Application*, 2021.
- [59] H. M. La S. Gibb and S. Louis. A genetic algorithm for convolutional network structure optimization for concrete crack detection. *IEEE Congress on Evolutionary Computation*, 2018.
- [60] Xuefeng Zhao Shengyuan Li. Convolutional neural networks-based crack detection for real concrete surface. *Proceedings Volume 10598, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 2018.

## Bibliography

- [61] Xuefeng Zhao; Shengyuan Li. A method of crack detection based on convolutional neural networks. *Proceedings for structural health monitoring*, 2017.
- [62] Shengyuan Li ; Xuefeng Zhao. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, 2019.
- [63] Young-Jin Cha; Wooram Choi. Vision-based concrete crack detection using a convolutional neural network. *Dynamics of Civil Structures*, 2017.
- [64] Yue Zhang ;Shijie Liu ;Chunlai Li and Jianyu Wang. Rethinking the dice loss for deep learning lesion segmentation in medical images. *Journal of Shanghai Jiaotong University*, 2021.
- [65] Zitao Zeng; Weihao Xie; Yunzhe Zhang; Yao Lu. Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images. *IEEE Access*, 2019.
- [66] Xiaomeng Li; Hao Chen; Xiaojuan Qi; Qi Dou; Chi-Wing Fu; Pheng-Ann Heng. H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE Transactions on Medical Imaging*, 2018.
- [67] Yu Weng; Tianbao Zhou; Yujie Li; Xiaoyu Qiu. Nas-unet: Neural architecture search for medical image segmentation. *IEEE Access*, 2019.
- [68] USA Steven Guan; Amir A. Khan; Siddhartha Sikdar; Parag V. ChitnisGeorge Mason University, Fairfax. Fully dense unet for 2-d sparse photoacoustic tomography artifact removal. *IEEE Journal of Biomedical and Health Informatics*, 2020.
- [69] Wei ChenBoqiang; Liu Suting; Peng Jiawei; Sun Xu Qiao. S3d-unet: Separable 3d u-net for brain tumor segmentation. *International MICCAI Brainlesion Workshop*, 2018.
- [70] Xiao Xiao; Shen Lian; Zhiming Luo; Shaozi Li. Weighted res-unet for high-quality retina vessel segmentation. *9th International Conference on Information Technology in Medicine and Education*, 2018.
- [71] Xintong Yan Shao Xiang Wei Sun Dan Zhang Gianmarc Coppola Peng Tang, Qiaokang Liang. Efficient skin lesion segmentation using separable-unet with stochastic weight averaging. *Computer Methods and Programs in Biomedicine*, 2019.



- [72] Huimin Huang; Lanfen Lin; Ruofeng Tong; Hongjie Hu; Qiaowei Zhang; Yutaro Iwamoto; Xianhua Han; Yen-Wei Chen. Unet 3+: A full-scale connected unet for medical image segmentation. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.
- [73] Xin Zhao ;Yitong Yuan ;Mengdie Song ;Yang Ding ;Fenfang Lin ;Dong Liang ;Dongyan Zhang. Use of unmanned aerial vehicle imagery and deep learning unet to extract rice lodging. *Sensors*, 2019.
- [74] Cao Vu Dunga ;Le Duc Anh. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 2019.
- [75] L. Attard; C. J. Debono; G. Valentino; M. Di Castro; A. Masi and L. Scibile. Automatic crack detection using mask r-cnn. *11th International Symposium on Image and Signal Processing and Analysis*, 2019.
- [76] J. Yang; W. Wang; G. Lin; Q. Li; Y. Sun and Y. Sun. Infrared thermal imaging-based crack detection using deep learning. *IEEE Access*, 2019.
- [77] Nasim ; Mohammed Yahya M. Tan, Chenjun ; Uddin. Deep learning-based crack detection using mask r-cnn technique. *9 th International Conference on Structural Health Monitoring of Intelligent Infrastructure*, 2019.
- [78] J. Tang; Y. Mao; J. Wang and L. Wang. Multi-task enhanced dam crack image detection based on faster r-cnn. *IEEE 4th International Conference on Image, Vision and Computing*, 2019.
- [79] Fen Fang; Liyuan Li; Hongyuan Zhu; Joo-Hwee Lim. Combining faster r-cnn and model-driven clustering for elongated object detection. *IEEE Transactions on Image Processing*, 2020.
- [80] Kemal Haciefendioglu; Hasan Basri Basaga. Concrete road crack detection using deep learning-based faster r-cnn method. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 2021.
- [81] Jin-Hwan Lee; Sung-Sik Yoon; In-Ho Kim; Hyung-Jo Jung. Diagnosis of crack damage on structures based on image processing techniques and r-cnn using unmanned aerial vehicle (uav). *Proceedings Volume 10598, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 2018.
- [82] Wenzhe Wang; Bin Wu; Sixiong Yang; Zhixiang Wang. Road damage detection and classification with faster r-cnn. *IEEE International Conference on Big Data*, 2018.

## Bibliography

- [83] Fu-Chen Chen; Mohammad R. Jahanshahi. Nb-fcn: Real-time accurate crack detection in inspection videos using deep fully convolutional network and parametric data fusion. *IEEE Transactions on Instrumentation and Measurement*, 2020.
- [84] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [85] J. E. Bresenham. Algorithm for computer control of a digital plotter. 1965.
- [86] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 1991.
- [87] Yoshua; Courville Aaron Goodfellow, Ian; Bengio. Softmax units for multinoulli output distributions. *Deep Learning*, 2016.
- [88] Elnaz Habibi Aghdam, Hamed; Jahani Heravi. *Guide to Convolutional Neural Networks*. Springer, 2017.
- [89] I. Sutskever A. Krizhevsky and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2017.
- [90] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [91] Y. Bengio. Deep learning of representations for unsupervised and transfer learning. *Workshop and Conference Proceedings*, 2012.
- [92] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *Computer Vision and Pattern Recognition*, 2015.
- [93] S. Dorafshan; R. J. Thomas;and M. Maguire. Sdnet2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in Brief*, 2018.
- [94] M. Maguire; S. Dorafshan; and R. Thomas. Sdnet2018: A concrete crack image dataset for machine learning applications. 2018.
- [95] Peter Sadowski. Notes on backpropagation.
- [96] Y. Bengio. *Practical Recommendations for Gradient-Based Training of Deep Architectures*. Springer, 2012.
- [97] Olaf Ronneberger; Philipp Fischer; Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Computer Vision and Pattern Recognition*, 2015.

- [98] Quang Hung;Ly Hai-Bang;Ho Lanh Si;Al-Ansari Nadhir;Le Hiep Van;Tran Van Quan;Prakash Indra;Pham Binh Thai Shen, Yu-Sheng; Nguyen. Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Artificial Intelligence for Civil Engineering*, 2021.
- [99] Isabelle Guyon. A scaling law for the validation-set training-set size ratio. *ATeT Bell Laboratories, Berkeley, California*.
- [100] Connor Shorten;Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019.
- [101] Agnieszka Mikolajczyk; Michal Grochowski. Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop*, 2020.
- [102] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. December 2017.
- [103] Sebastien C. Wong; Adam Gatt; Victor Stamatescu; Mark D. McDonnell. Understanding data augmentation for classification: When to warp? *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016.
- [104] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems IV*, 1994.
- [105] Carole H Sudre; Wenqi Li; Tom Vercauteren; Sebastien Ourselin; M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Computer Vision and Pattern Recognition*, 2017.
- [106] Shruti Jadon. A survey of loss functions for semantic segmentation. *IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology*, 2020.
- [107] WEI SONG ;NAN ZHENG ;XIANGCHUN LIU; LIRONG QIU ; RUI ZHENG. An improved u-net convolutional networks for seabed mineral image segmentation. *IEEE Access SPECIAL SECTION ON ADVANCED OPTICAL IMAGING FOR EXTREME ENVIRONMENTS*, 2012.
- [108] Mingxing Tan; Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, 2019.

## Bibliography

- [109] Jia Deng; Wei Dong; Richard Socher; Li-Jia Li; Kai Li; Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [110] Kaiming He; Xiangyu Zhang; Shaoqing Ren; Jian Sun. Deep residual learning for image recognition. 2015.
- [111] Tsung-Yi Lin; Michael Maire; Serge Belongie; Lubomir Bourdev; Ross Girshick; James Hays; Pietro Perona; Deva Ramanan; C. Lawrence Zitnick; Piotr Dollar. Microsoft coco: Common objects in context. 2014.
- [112] Andrew G. Howard; Menglong Zhu; Bo Chen; Dmitry Kalenichenko; Weijun Wang; Tobias Weyand; Marco Andreetto; Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *Google Inc.*, 2017.
- [113] Christian Szegedy; Wei Liu; Yangqing Jia; Pierre Sermanet; Scott Reed; Dragomir Anguelov; Dumitru Erhan; Vincent Vanhoucke; Andrew Rabinovich. Going deeper with convolutions. 2014.
- [114] Diederik P. Kingma; Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations, San Diego*, 2015.
- [115] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 2011.
- [116] *BIPM - Guide to the Expression of Uncertainty in Measurement (GUM)*, (2017). <http://www.bipm.org/en/publications/guides/gum.html>.
- [117] *R.A. Fisher, Statistical Methods for Research Workers, in: S. Kotz, N.L. Johnson (Eds.), Break. Stat., Springer Series in Statistics (Perspectives in Statistics), Springer, New York, NY, 1992.*
- [118] *MEASUREMENT SYSTEMS ANALYSIS Reference Manual Fourth Edition June 2010 Copyright 1990, 1995, 2002, 2010 Chrysler Group LLC, Ford Motor Company, General Motors Corporation.*
- [119] Alessia Baleani; Paolo Castellini; Paolo Chiariotti; Nicola Paone; Saverio Zitti. Dimensional measurements in production line: a comparison between a custom-made telecentric optical profilometer and on-the-market measurement systems. *IEEE International Workshop on Metrology for Industry 4.0 and IoT*, 2021.
- [120] Dart.dev. <https://dart.dev/>.

- [121] Flutter technical overview. <https://flutter.dev/docs/resources/architectural-overview>, 2021.
- [122] Flask web development. <https://flask.palletsprojects.com/en/1.1.x/>.
- [123] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures. Chapter 5*. PhD thesis, University of California, Irvine, 2000.



# Appendix

## 1 Gage RR study

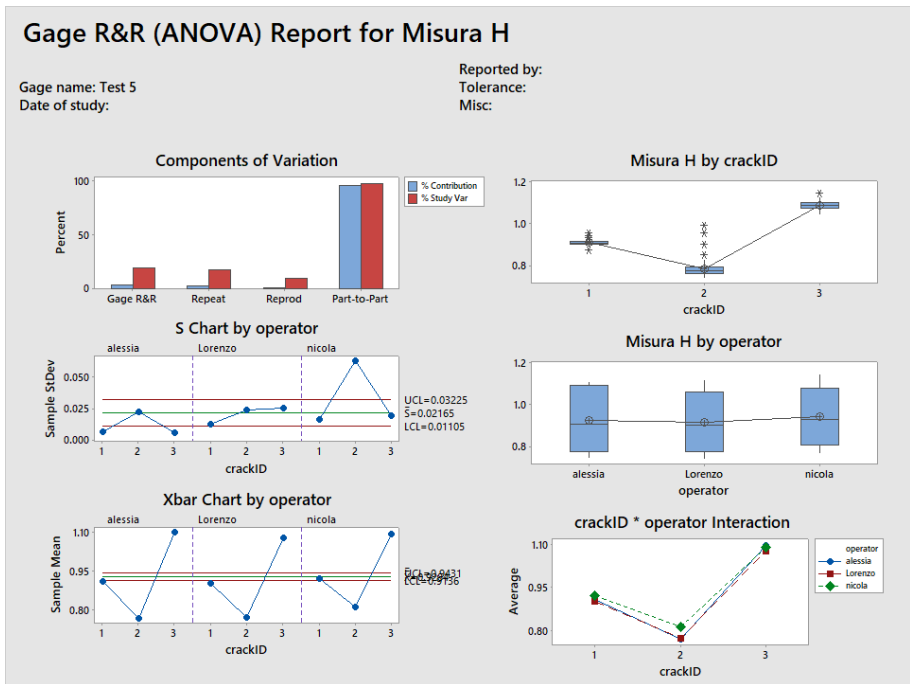


Figure 1: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to  $5^\circ$ .

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.46	0.73	654.103	0
Operator	2	0.009	0.005	4.274	0.1
Interaction	4	0.004	0.001	20.523	0
Repeatability	81	0.004	0		
Total	89	1.479			

Source	Variance	% Contr
Total Gage R&R	0.0004	0.98
Repeatability	0	0.22
Reproducibility	0.0004	0.93
Operator	0.0003	0.5
Interaction	0.0001	0.43
Part to Part	0.0311	98.85
Total Variation	0.0315	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0168	0.1008	9.99
Repeatability	0.0074	0.0443	4.7
Reproducibility	0.0151	0.0906	9.63
Operator	0.011	0.0662	7.04
Interaction	0.0103	0.0618	6.57
Part to Part	0.1559	0.9353	99.42
Total Variation	0.1568	0.9408	100

Number of Distinct Categories = 15

Table 1: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 5°.



ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.87	0.935	799.031	0
Operator	2	0.018	0.009	7.743	0.04
Interaction	4	0.005	0.001	26.053	0
Repeatability	81	0.004	0		
Total	89	1.897			

Source	Variance	% Contr
Total Gage R&R	0.0004	1.33
Repeatability	0	0.14
Reproducibility	0.0004	1.19
Operator	0.0003	0.83
Interaction	0.0001	0.36
Part to Part	0.0311	98.67
Total Variation	0.0315	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0205	0.123	11.54
Repeatability	0.0067	0.0402	3.77
Reproducibility	0.0194	0.1163	10.91
Operator	0.0162	0.0973	9.13
Interaction	0.0106	0.0637	5.97
Part to Part	0.1764	1.0586	99.33
Total Variation	0.1776	1.0657	100

Number of Distinct Categories = 13

Table 2: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to  $10^\circ$ .

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.35	0.675	122.811	0
Operator	2	0.023	0.012	2.119	0.24
Interaction	4	0.022	0.005	36.601	0
Repeatability	81	0.012	0		
Total	89	1.407			

Source	Variance	% Contr
Total Gage R&R	0.0009	3.83
Repeatability	0.0002	0.65
Reproducibility	0.0007	3.19
Operator	0.0002	0.88
Interaction	0.0005	2.3
Part to Part	0.0223	96.17
Total Variation	0.0232	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0298	0.179	19.58
Repeatability	0.0123	0.0735	8.04
Reproducibility	0.0272	0.1632	17.85
Operator	0.0143	0.0859	9.4
Interaction	0.0231	0.1387	15.18
Part to Part	0.1494	0.8962	98.06
Total Variation	0.1523	0.9139	100

Number of Distinct Categories = 7

Table 3: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 15°.

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.498	0.749	46.354	0
Operator	2	0.096	0.048	2.983	0.16
Interaction	4	0.065	0.016	64.894	0
Repeatability	81	0.02	0		
Total	89	1.679			

Source	Variance	% Contr
Total Gage R&R	0.0029	10.64
Repeatability	0.0002	0.91
Reproducibility	0.0027	9.73
Operator	0.0011	3.91
Interaction	0.0016	5.82
Part to Part	0.0244	89.36
Total Variation	0.0273	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0539	0.3235	32.62
Repeatability	0.0158	0.0947	9.54
Reproducibility	0.0516	0.3094	31.19
Operator	0.0327	0.1961	19.77
Interaction	0.0399	0.2393	24.12
Part to Part	0.1563	0.9377	94.53
Total Variation	0.1653	0.992	100

Number of Distinct Categories = 4

Table 4: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 20°.

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.368	0.684	93.356	0
Operator	2	0.035	0.017	2.367	0.21
Interaction	4	0.029	0.007	14.077	0
Repeatability	81	0.042	0.001		
Total	89	1.475			

Source	Variance	% Contr
Total Gage R&R	0.0015	6.37
Repeatability	0.0005	2.16
Reproducibility	0.001	4.21
Operator	0.0003	1.39
Interaction	0.0007	2.83
Part to Part	0.0226	93.63
Total Variation	0.0241	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0392	0.2351	25.24
Repeatability	0.0228	0.1369	14.7
Reproducibility	0.0319	0.1911	20.52
Operator	0.0183	0.1096	11.77
Interaction	0.0261	0.1566	16.81
Part to Part	0.1502	0.9013	96.76
Total Variation	0.1552	0.9314	100

Number of Distinct Categories = 5

Table 5: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 25°.

ANOVA TABLE					
Source	DoF	SS	MS	F	p
Crack ID	2	1.368	0.684	28.075	0
Operator	2	0.077	0.038	1.571	0.31
Interaction	4	0.097	0.024	70.439	0
Repeatability	81	0.028	0		
Total	89	1.57			

Source	Variance	% Contr
Total Gage R&R	0.0032	12.74
Repeatability	0.0003	1.37
Reproducibility	0.0029	11.37
Operator	0.0005	1.84
Interaction	0.0024	9.53
Part to Part	0.022	87.26
Total Variation	0.0252	100

Source	SD	SV (6 sigma)	% SV
Total Gage R&R	0.0567	0.34	35.7
Repeatability	0.0186	0.1116	11.72
Reproducibility	0.0535	0.3212	33.72
Operator	0.0215	0.1292	13.56
Interaction	0.049	0.2941	30.87
Part to Part	0.1483	0.8898	93.41
Total Variation	0.1588	0.9525	100

Number of Distinct Categories = 3

Table 6: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to  $30^\circ$ .

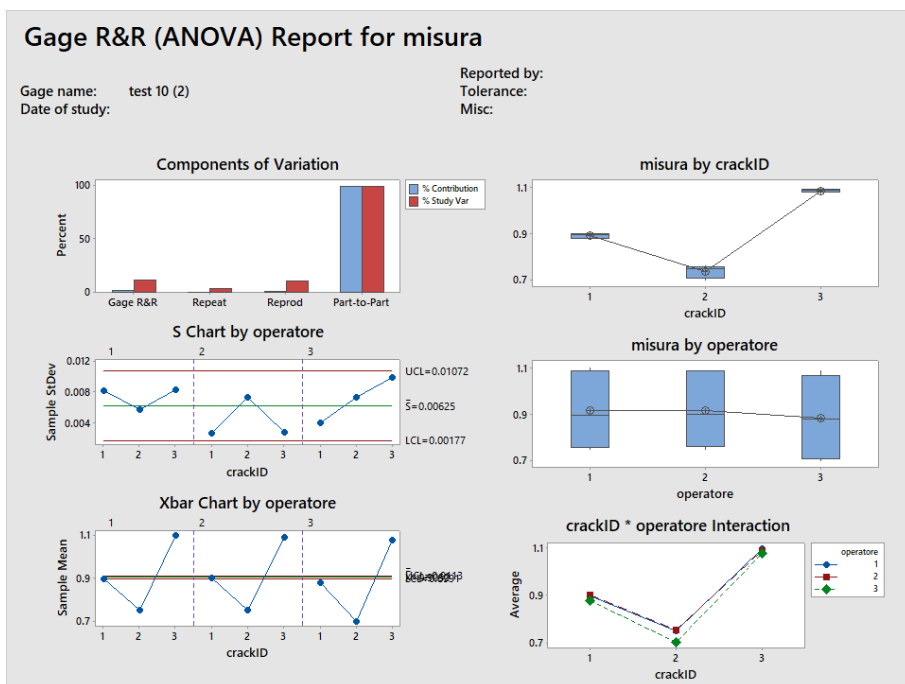


Figure 2: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 10°.

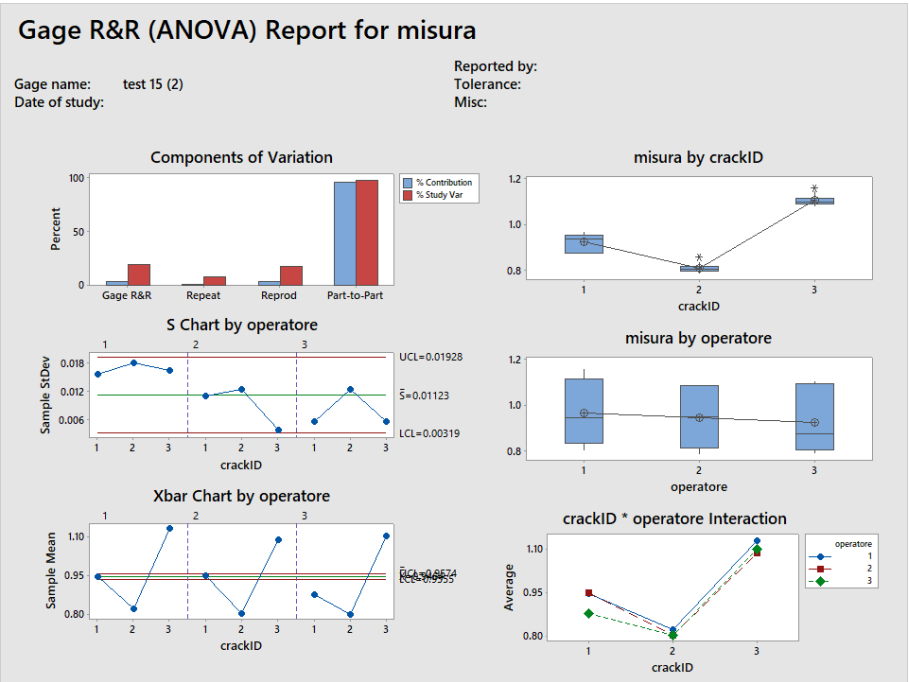


Figure 3: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 15°.

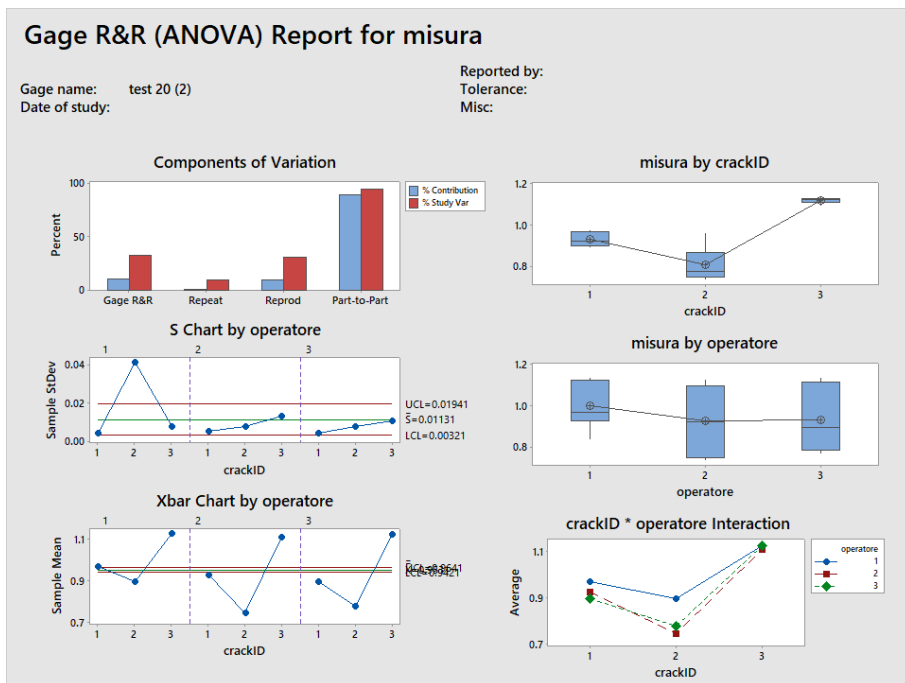


Figure 4: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 20°.



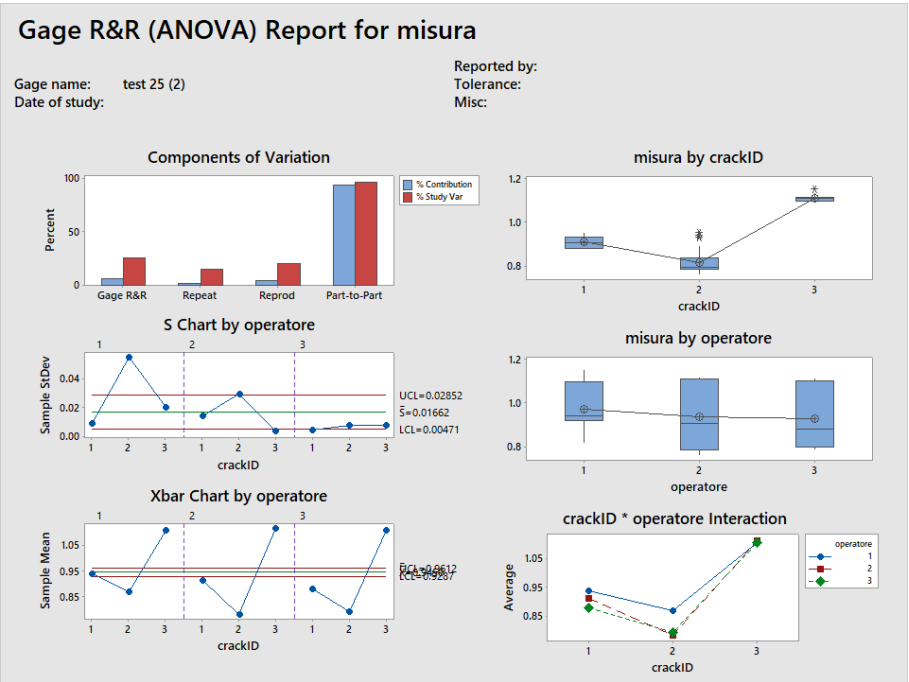


Figure 5: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 25°.

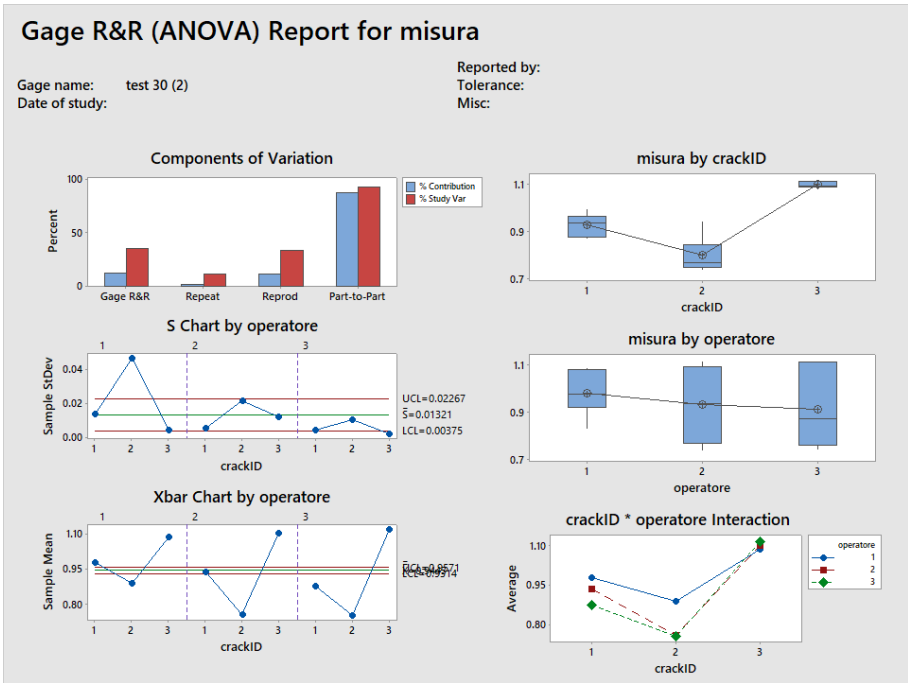


Figure 6: Gage R&R study assessed by asking 3 different operators to perform 10 measurements each in 3 different target cracks with maximum angle admitted equal to 30°.

ANOVA TABLE			
Source	DoF	Sum of Squares	Mean Square
CrackID	p-1	SS par	SS par/(p-1)
Operator	o-1	SS op	SS op/(o-1)
Interaction	(p-1)(o-1)	SS int	SS int/(p-1)(o-1)
Repeatability	op(m-1)	SS eq	SS eq/op(m-1)
Total	omp-1	SS total	

ANOVA TABLE			
Source	F-stat	p-value	
CrackID	MS part/MS int	Fdist(F par,p-1,(p-1)(o-1))	
Operator	MS op/MS int	Fdist(F op,p-1,(p-1)(o-1))	
Interaction	MS int/MS rep	Fdist(F int,(p-1)(o-1),op(m-1))	
Repeatability			
Total			

Table 7: Example of ANOVA table