UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

BEATRIZ SALES DA CUNHA

# DEVELOPMENT OF COMPUTER VISION BASED MODELS FOR AUTOMATED CRACK DETECTION

Recife
2020

BEATRIZ SALES DA CUNHA

# DEVELOPMENT OF COMPUTER VISION BASED MODELS FOR AUTOMATED CRACK DETECTION

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

**Área de concentração**: Pesquisa Operacional.

**Orientador**: Prof. Dr. Márcio José das Chagas Moura.

Recife

2020

Catalogação na fonte
Bibliotecário Gabriel Luz, CRB-4 / 2222

C972d Cunha, Beatriz Sales da.

Development of computer vision based models for automated crack detection / Beatriz Sales da Cunha – Recife, 2020.

69 f.: figs., tabs.

Orientador: Prof. Dr. Márcio José das Chagas Moura.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia de Produção, 2020.

Inclui referências e apêndice.

1. Engenharia de Produção. 2. Detecção de trincas. 3. Processamento de imagens. 4. Segmentação. 5. Análise de textura. 6. Aprendizagem de máquina. I. Moura, Márcio José das Chagas (Orientador). II. Título.

UFPE

658.5 CDD (22. ed.) BCTG / 2020-154

BEATRIZ SALES DA CUNHA

# DEVELOPMENT OF COMPUTER VISION BASED MODELS FOR AUTOMATED CRACK DETECTION

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

Aprovada em: 17/02/2020.

## BANCA EXAMINADORA

____________________________________________

Prof. Dr. Márcio José das Chagas Moura (Orientador)
Universidade Federal de Pernambuco

____________________________________________

Profa. Dra. Isis Didier Lins (Examinadora Interna)
Universidade Federal de Pernambuco

____________________________________________

Prof. Dr. Carlos Alexandre Barros de Mello (Examinador Externo)
Universidade Federal de Pernambuco

# ACKNOWLEDGMENTS

If I could thank everyone that supported me throughout my path until here the way they really deserve, that would be no pages left for my dissertation. Although it takes a lot of willpower to apply the concept of "continuous improvement", me and my fellow Production Engineers colleagues know so much about, to build an academic career, I do think that some special people played an important role in getting me where I am and, hence, need to be highlighted.

First, I would like to thank my parents, that inspire me to be a better person, encourage me to take my own risks and support me to go further. I finally began to understand and agree with all the yes's and no's! My brother, for being so different from me and teaching me to respect and appreciate that. I love seeing you grow up! I also would like to thank my husband-to-be, João, for his respect, attention, kindness and for starting to dream my own dreams.

To Professor Márcio Moura, I will be forever grateful for all the opportunities you have been giving me since you started advising me in 2016. It has been a long journey, one that I learned a lot, but there is still so much to learn from you. I also would like to thank Professor Isis Lins, that always played a role in advising me as well and inspires me with her confident, intelligent and sweet attitude. I also would like to thank my colleagues at CEERMA-UFPE for helping one another and for the research collaborations, especially Ana Cláudia, Caio, July, and Rafael.

Finally, I thank the Programa de Pós Graduação em Engenharia de Produção (PPGEP-UFPE) for the opportunity to enrich my academic experience, expanding my horizons and introducing me to very talented people. I also appreciate the financial support provided by *Fundação de Amparo a Ciência e Tecnologia de PE* (FACEPE), *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) and *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq).

# ABSTRACT

Systems subjected to continuous operation in harsh conditions are exposed to different failure mechanisms (e.g., corrosion, fatigue, and temperature-related defects). In this context, inspection and health monitoring have become crucial to prevent system, environment, and users from severe damage. However, visual inspection strongly depends on human' experience, having its accuracy influenced by the physical and cognitive state of the inspector (i.e., human factors). Particularly, infrastructures need to be periodically inspected, which is costly, time-consuming, hazardous and biased. Nowadays, the increase in computer power allows for analyzing a considerable number of images in a shorter time and use more robust algorithms. Advances in Computer Vision (CV) and Machine Learning (ML) provide the means to the development of automated, accurate, non-contact and non-destructive inspection methods. Therefore, this dissertation proposes and compares the adoption of different CV approaches to extract features for crack detection. In fact, we applied texture-based and region-based methods to a real concrete crack image database, and then the results fed four ML models to identify crack existence, namely Support Vector Machine (SVM), Multilayer Perceptron (MLP), Adaboost (AB), and Random Forest (RF). Results show the potential of data preprocessing to improve methods' performance in reaching a balanced accuracy above 97%.

Keywords: Crack detection. Image processing. Segmentation. Texture analysis. Machine learning.

# RESUMO

Sistemas sujeitos à operação contínua em condições adversas são expostos a diferentes mecanismos de falha (por exemplo, corrosão, fadiga e problemas relacionados à temperatura). Nesse contexto, a inspeção e o monitoramento da saúde desses sistemas tornaram-se cruciais para evitar danos graves ao sistema, ao ambiente e aos usuários. No entanto, a inspeção visual depende fortemente da experiência humana, tendo sua precisão influenciada pelo estado físico e cognitivo do inspetor (ou seja, fatores humanos). Particularmente, edificações precisam ser inspecionadas periodicamente, o que é caro, demorado, perigoso e tendencioso. Atualmente, com o aumento da capacidade computacional, é possível analisar um número considerável de imagens em menos tempo e usar algoritmos mais robustos para tal. Os avanços em Visão Computacional (CV) e Aprendizagem de Máquina (ML) fornecem os meios para o desenvolvimento de métodos de inspeção automatizados, precisos, sem contato e não destrutivos. Portanto, esta Dissertação propõe e compara a adoção de diferentes abordagens envolvendo CV para extrair características de imagens, com o objetivo de detecção de trincas. De fato, foram aplicados métodos baseados em textura e região em um banco de dados real de imagens de trincas em concreto e, em seguida, os resultados alimentaram quatro modelos de ML para identificar a existência de trincas, a saber: *Support Vector Machine* (SVM), *Multilayer Perceptron* (MLP), Adaboost (AB) e *Random Forest* (RF). Os resultados mostram o potencial do pré-processamento de dados para melhorar o desempenho dos métodos ao atingir uma *balanced accuracy* acima de 97%.

Palavras-chave: Detecção de trincas. Processamento de imagens. Segmentação. Análise de textura. Aprendizagem de máquina.

## FIGURE LIST

# TABLE LIST

# CONTENT

# 1 INTRODUCTION

Engineering structures, such as bridges, buildings, dams, and pipelines, are essential for the economy and the routine of society (FENG; FENG, 2018). They age and deteriorate as they are often exposed to fatigue stress and cyclic loading (MOHAN; POOBAL, 2018), affecting their reliability (ASTORGA; DROGUETT; MERUANE, 2018). Therefore, regulation in many countries (QINGGUO *et al.*, 2019b) enforces they are periodically inspected to ensure their safety and serviceability (KOCH *et al.*, 2015; SANKARASRINIVASAN *et al.*, 2015) and to prevent long-term damage accumulation (DAVOUDI; MILLER; KUTZ, 2018b; DORAFSHAN *et al.*, 2018), which could lead to the structure's collapse.

In this context, inspection and Structural Health Monitoring (SHM) have become crucial to prevent system, environment, and users from severe damage. As civil infrastructures suffer from aging and deterioration (SULEIMAN; NELSON; NEHDI, 2019), SHM aims at identifying changes in the geometry of structures at early stages (CHOUDHURY *et al.*, 2018), and, thus, reduce maintenance costs. Recent researches have focused on crack detection, as they are the most frequent defect in civil infrastructures, to support the implementation of preventive measures, such as reinforcement, and reconstruction, which have a high potential to prevent its expansion (BAYAR; BILIR, 2019; ZOU *et al.*, 2019).

## 1.1 CONTEXT

Various materials are subjected to the occurrence of cracks. In particular, due to the influence of self-weight, fatigue stress, structural loading, environmental exposure, and physical/chemical reactions (BAYAR; BILIR, 2019; QINGGUO *et al.*, 2019b; SULEIMAN; NELSON; NEHDI, 2019), concrete structures are exposed to various types of damages (LUO; GE; TIAN, 2019), such as concrete cracks, steel delamination, steel corrosion, and bolt corrosion. The earliest stage of degradation is expressed in the form of surface cracks, and, as its continuous exposure may lead to severe damages, the presence of cracks trigger maintenance actions and even evacuation (MOHAN; POOBAL, 2018; RUIZ *et al.*, 2018) to avoid continuous exposure that may lead to more critical damage and even the collapse of the structure. This is of special interest in aging structures, such as buildings and bridges, especially because of the risk it imposes to the public. For instance, about 40% of Canada's bridges are older than 50 years (ADHIKARI; MOSELHI; BAGCHI, 2014a). In fact, a bridge in Latchford

(Ontario, Canada), which was built in 1960, partially failed in 2003 due to fatigue fractures (PIPINATO, 2016). More recently, in 2018, the cause of the collapse of a Miami (Florida, US) university pedestrian bridge is also accredited to inattentive inspections and poor maintenance (NTSB BLAMES FATAL MIAMI BRIDGE FALL ON DESIGN, LACK OF OVERSIGHT, 2019). Moreover, in 2019, a building in Ceará (Fortaleza, Brazil) collapsed causing at least three deaths and injuring seven people (ALMEIDA, 2019).

In the same way, although pavements are exposed to different kinds of distresses triggered mainly by traffic loads, temperature oscillation, and aging (HOANG; NGUYEN, 2019), pavement cracks are especially critical since they affect both vehicles (speed limit and wear out) and safety (VARONA; MONTESERIN; TEYSEYRE, 2019; ZHANG, Dejin *et al.*, 2017a). Moreover, it only takes a rainy night to transform a crack into a hole, imposing risks to high-speed vehicles (ZOU *et al.*, 2019). For instance, state transportation agencies are responsible for the periodic inspection of road conditions and its conservation (ZHANG, Yuming *et al.*, 2018), which entails the inspection of more than a hundred thousand kilometers in a country like China or the US (ZOU *et al.*, 2019). Reducing costs and ensuring roads' condition is of particular interest for those agencies, especially in times of economic recession. In fact, improving roads' inspection effectiveness must be the foremost concern before talking about smart cities, as roads need to support the operation of autonomous vehicles (MEI; GÜL, 2019).

Since the identification of early stages of cracks in infrastructures can avoid significant future problems, the development of reliable and robust techniques to recognize them has great applicability. In fact, researches are exploiting the potential of Information and Communication Technologies (ICT) to provide innovative methodologies to improve the identification of civil infrastructures' defects (TEDESCHI; BENEDETTO, 2017).

## 1.2 JUSTIFICATION

SHM is primarily assessed in terms of visual changes, such as cracks and corrosion (CHA; CHOI; SUH; MAHMOUDKHANI, 2018), to monitor assets and plan maintenance policies. This inspection is generally conducted by skilled personnel, and, generally, it is costly, time-consuming, hazardous, and biased, aside from being even more challenging when conducted under poor lighting conditions and on rusted or irregular surfaces (DORAFSHAN *et al.*, 2017). Hence, results of inspections may exhibit some fluctuation (PAYAB; ABBASINA;

KHANZADI, 2019) and “may give only comparative results rather than standardized ones” (BAYAR; BILIR, 2019).

Then, it is essential to balance the system’s safety and the cost of an expert when scheduling inspections (ASTORGA; DROGUETT; MERUANE, 2018). Sometimes, this analysis is impracticable since inspections can be imposed by federal law, which is the case of the US, where bridges must be physically inspected every two years (DORAFSHAN *et al.*, 2018). In Brazil, although inspections are not regulated by federal law, the *Associação Brasileira de Normas Técnicas* (ABNT) determines (but does not enforce) the standards for inspection methods and maintenance performance. Aside from mobilizing resources, impacting the traffic, and involving accessibility issues (LIANG, Dong *et al.*, 2019), these inspections also pose safety risks, both to the inspectors and the public.

Therefore, there has been an increasing demand for non-destructive and non-contact inspections, which provided a stage for the use of Unmanned Aerial Vehicle (UAV) implementations, such as inspection, maintenance, and structures monitoring using image analysis (DORAFSHAN *et al.*, 2018). Indeed, the use of images to identify automatically cracks has been applied to different structures, such as buildings, transmission lines, refineries towers, and wind turbines (LIANG, Dong *et al.*, 2019; MOHAN; POOBAL, 2018).

The automation of the inspection process is expected to improve efficiency, reduce cost, and lead to more frequent inspection cycles (ZOU *et al.*, 2019). Moreover, it is also expected an improvement in accuracy compared to conventional manual methods (MOHAN; POOBAL, 2018). However, the application of this kind of technology requires acquiring and dealing with a significant number of images. Other existing concerns are the demand for expensive equipment, like UAVs and high-resolution cameras (BAYAR; BILIR, 2019), and the time restriction, imposed by UAVs’ batteries, for example.

Nevertheless, emerging technologies bring opportunities to support decision-makers on structure inspection, by improving the capacity, time, and accuracy of these activities. Specifically, advances in Computer Vision (CV) and Machine Learning (ML) techniques have been providing the means to address these issues by the development of automated, accurate, non-contact, and non-destructive inspection methods (DAVOUDI; MILLER; KUTZ, 2018b; DORAFSHAN *et al.*, 2017), which enables performing inspection remotely [21]. CV studies the automated extraction of information from images and videos (TEDESCHI; BENEDETTO, 2017), gathering knowledge from fields such as image processing, pattern recognition, mathematics, and artificial intelligence. On the other hand, ML techniques focus on learning patterns and making a prediction based on data. In this sense, it can improve both the efficiency and robustness of CV methods (CHA; CHOI; SUH; MAHMOUDKHANI; *et al.*, 2018).

The development of CV provides new opportunities for non-intrusive real-time monitoring in the context of reliability engineering for fault detection and diagnosis purposes based on image characteristics (i.e., color, structure, texture, and morphology). Nevertheless, noisy images lead to poor continuity of the crack and low contrast between the crack and the background, posing a challenge for automatic crack detection (MEI; GÜL, 2019).

According to (MOHAN; POOBAL, 2018), difficulties in concrete image crack detection are mostly based on the irregular shape and size of cracks, noisy background, quality of the image (i.e., resolution, illumination, shading, low contrast, reflection) and presence of irrelevant objects (e.g. oil spots, road signs) (LIANG, Sun; JIANCHUN; XUN, 2018; ZHANG, Dejin *et al.*, 2017a). Hence, a suitable alternative to deal with these issues is the usage of image pre-processing and segmentation techniques (MOHAN; POOBAL, 2018).

The possibility of addressing the detection, localization, and quantification of damage with appropriately captured inspection images can add substantial improvement to qualitative inspection systems. An automated structural inspection system should retrieve damage's information to a level likely to be impossible for a human inspector and remove much of the uncertainty associated with an inspector's judgment of the severity of structural damage. With this technology, the health status of infrastructures could be systematically monitored and tracked, and, thus, effective inspection policies developed and implemented. Therefore, there is a need to diagnose damages preventively through advanced monitoring technologies tailored for implementing systematic, permanent, and real-time monitoring of the infrastructures' state, generating the information necessary to develop and implement a more efficient and effective inspection and maintenance plan.

## 1.3 OBJECTIVES

This dissertation aims at developing an accurate and reliable inspection method for concrete infrastructures. The focus of this work is on proposing a method for the automatic identification of cracks from images of the structure under inspection, using CV do extract information from the images and ML techniques to learn patterns from this information and make predictions. The proposed approach aims at overcoming the problem caused by the variety of both crack and background characteristics by determining appropriated and discriminative features for the images.

### 1.3.1 Specific Objectives

In order to achieve the general objective, some specific targets are defined:

- Literature review and background study about infrastructure inspection (purpose, existing methodologies, and opportunities) and techniques applied in the context of inspection (image processing, CV, and ML);
- Development and validation of a model to identify cracks on images;
- Analysis of real concrete crack image dataset;
- Test the model on a different image dataset and evaluate results.

## 1.4 DISSERTATION STRUCTURE

The content of the following chapters of this dissertation are briefly described below:

Chapter 2 presents the theoretical background and literature review of essential concepts related to crack detection;
Chapter 3 gives a detailed description of the proposed methodology for crack detection;
Chapter 4 shows the results of the application of the proposed method on a concrete crack dataset, as well as a test on a different concrete crack dataset;

Chapter 5 concludes the dissertation and provides some perspectives towards limitations and future works.

# 2 THEORETICAL BACKGROUND AND LITERATURE REVIEW

Applications of crack detection algorithms can be found in the literature with a focus either on the material or in the structure itself. Examples of the materials studied are: pavement (AMHAZ *et al.*, 2016; FAN, Z. *et al.*, 2018; HOANG; NGUYEN, 2019, 2018; HU; ZHAO; WANG, 2010; INZERILLO; DI MINO; ROBERTS, 2018; LIU, Wenjun *et al.*, 2019; NHAT-DUC; NGUYEN; TRAN, 2018; SALMAN *et al.*, 2013; TEDESCHI; BENEDETTO, 2017; WENG; HUANG; WANG, 2019; YANG, Fan *et al.*, 2019; ZHANG, Dejin *et al.*, 2017a; ZHANG, Yuming *et al.*, 2018; ZOU *et al.*, 2012), concrete (ALAM *et al.*, 2015; BAYAR; BILIR, 2019; CHO *et al.*, 2018a; DAVOUDI; MILLER; KUTZ, 2018a; DORAFSHAN; THOMAS; MAGUIRE, 2018a; DUNG; ANH, 2019; FUJITA, Y.; HAMAMOTO, 2011; KIM; CHO, 2018; LEE *et al.*, 2019; LIU, Zhenqing *et al.*, 2019; NGUYEN; KAM; CHENG, 2014; OLIVEIRA SANTOS; VALENÇA; JÚLIO, 2019; PAGLINAWAN *et al.*, 2018; PAYAB; ABBASINA; KHANZADI, 2019; QINGGUO *et al.*, 2019a; QU *et al.*, 2019; SHAN; ZHENG; OU, 2016; WANG, P.; HUANG, 2010; YAMAGUCHI; HASHIMOTO, 2010; YANG, Y. *et al.*, 2018), glass (GLUD *et al.*, 2016; YIYANG, 2014), composite materials (ABUOBAID; HEIDER; YARLAGADDA, 2019; HWANG *et al.*, 2019), and steel (KONG; LI, 2018). Some of the structures are: solar cells (CHAWLA; SINGAL; GARG, 2018; CHEN *et al.*, 2018), tunnels (ATTARD *et al.*, 2018; PANELLA *et al.*, 2018; PROTOPAPADAKIS *et al.*, 2019), roads (FAN, Rui *et al.*, 2019; MAEDA *et al.*, 2018; SHI *et al.*, 2016; ZHANG, L. *et al.*, 2016), bridges (ADHIKARI; MOSELHI; BAGCHI, 2014b; DUNG *et al.*, 2019; LEI *et al.*, 2018; LI, Yundong *et al.*, 2018; PRASANNA *et al.*, 2016; REAGAN; SABATO; NIEZRECKI, 2018; YANG, Yuan-Sen; YANG; HUANG, 2015), dams (VALENÇA; JÚLIO, 2018), buildings (WANG, Niannian *et al.*, 2019), wind turbines (WU *et al.*, 2019; ZHANG, Huiyi; JACKMAN, 2014), underground pipeline (HEO; JEON; SON, 2019; IYER; SINHA, 2005; SINHA; FIEGUTH, 2006), magnetic tile (LI, Xueqin; JIANG; YIN, 2014), welding joint (WANG, Xin; WU, 2019), and rail (CHOTZOGLOU *et al.*, 2019; ZHUANG *et al.*, 2018).

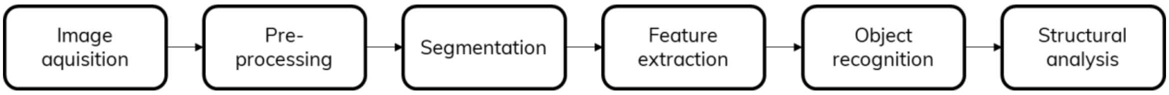
The assumptions related to the development of automatic crack detection methods using digital images indicate that cracks: (i) are darker than the background, (ii) are continuous regions, (iii) consist of connected pixels, and (iv) have various widths (PAYAB; ABBASINA; KHANZADI, 2019). According to (LIANG, Dong *et al.*, 2019; LIANG, Sun; JIANCHUN; XUN, 2018; MOHAN; POOBAL, 2018; ZHANG, Dejin *et al.*, 2017b), difficulties in image crack detection are mostly based on the irregular shape and size of cracks, noisy background (e.g. shadows, water stains, graffiti, spider web), quality of the image (e.g. resolution, illumination, shading, low contrast, reflection) and presence of irrelevant objects (e.g. oil spots, road signs).

A suitable alternative to deal with these issues is the usage of image processing and CV techniques, which have been evolving since the 1960s when the first powerful computers appeared (GONZALEZ; WOODS, 2006). Concepts of line detection, which is a classical CV problem, can be applied in the context of crack detection (ZOU *et al.*, 2019). Roughly speaking, early works have focused on methods based on intensity-thresholding, saliency detection, texture analysis, wavelet transform, minimal path, and ML (SHI *et al.*, 2016)**.**

## 2.1 IMAGE PROCESSING

The general architecture of crack detection methods based on image processing and CV, shown in Figure 1, is composed of the following steps: image acquisition, pre-processing, segmentation, feature extraction, object recognition, and structural analysis (TEDESCHI; BENEDETTO, 2017). The image acquisition step is the process of collecting images from the structure under inspection, which can be classified into active or passive. In the former, the acquisition is triggered by an exciting source, while in the latter, the acquisition period is pre-determined, which is more common for the inspection of structures (QINGGUO *et al.*, 2019b). Also, various types of images can be acquired, such as digital, ultrasonic, or laser (MOHAN; POOBAL, 2018). Given that images have been acquired, the pre-processing step consists of filtering the image to enhance its quality and, hence, the efficiency of the detection process.

Figure 1 – General architecture of the crack detection methods based on CV investigated in this Dissertation.



Source: adapted from (TEDESCHI; BENEDETTO, 2017)

The segmentation step aims to create a new image containing only the regions of interest, i.e., the parts that resemble a crack. Segmentation methods can be roughly divided into edge detection and threshold segmentation (LIANG, Dong *et al.*, 2019; ZOU *et al.*, 2019), but some papers also propose hybrid methodologies. Among threshold segmentation methods, the ones based on mathematical morphology and set theory, are the most used ones across the literature (PAYAB; ABBASINA; KHANZADI, 2019).

The idea of this type of segmentation is to enhance specific elements from the image that represent the shape of the region of interest by using structuring elements (SE), which specify the nature of the operation (PARKER, 2011). It is essential to understand that SE is determined by two parameters: format and size. For instance, the format can be of a disk, a line, or a rectangle. The combination of these parameters enables various applications. Basic operations are erosion and dilatation, and, from them, some algorithms are derived, such as hole filling, extraction of connected components, thinning, thickening, and skeletons (GONZALEZ; WOODS, 2006). For more details about these algorithms, see (GONZALEZ; WOODS, 2006).

Image segmentation can be appropriate for crack extraction; however, depending on the chosen algorithm and on the images under consideration, limitations of the standalone approach typically require combination with other techniques to improve performance (ZHANG, Dejin *et al.*, 2017a). For instance, the feature extraction step consists of a particular form of extracting information from the image for comparison and analysis purposes (DAVOUDI; MILLER; KUTZ, 2018b). Features are quantitative attributes or properties that, aside from describing characteristics of the image (or image regions), also help to distinguish different patterns (JAHANSHAHI *et al.*, 2013). In this context, features must be invariant to changes in condition, such as illumination and rotation, for example (NIXON; AGUADO, 2008).

The object recognition is the foremost step of this process as it entails the actual detection of the crack, which according to Ref. (MEI; GÜL, 2019) can be conducted by Rule Based (RL) or ML-based techniques. At the same time, RL-based methods are simpler to implement because they do not require annotation and training, these methods are not flexible enough to adapt to different situations (MEI; GÜL, 2019).

The structural analysis provides a deep understanding of the crack characteristics, and the most common objectives are the computation of crack's length, width, depth, position, surface, or direction of propagation (MOHAN; POOBAL, 2018). Among these properties, width is the most important and widely used one for concrete health assessment. In fact, the American Concrete Institute (ACI) prescribes width minimum allowable values for infrastructure condition assessment (QINGGUO *et al.*, 2019b).

The significant advantages of the crack detection method presented in Figure 1 are two-fold: (i) it provides automatic, fast, and accurate results, and (ii) through establishing database from these results, it is possible to evaluate and compare crack expansion (QINGGUO *et al.*, 2019b).

Various papers among the literature follow this methodology, proving its worth. They are described in Appendix A and further analyzed below.

The architecture of the crack detection method presented in Figure 1 entails that the goal is necessarily to localize the crack within a given image as it segments crack-like regions. Nevertheless, aside from classifying each segmented region as a crack or not, other classification tasks can be combined with this methodology in two ways: (i) as a step previous to segmentation, in the sense that only positively classified images follow to the next step, or (ii) as a way of differentiating segmented cracks.

An example of (i) is found in (FAN, Rui *et al.*, 2019), where a Convolutional Neural Network (CNN) is used to classify images as having cracks or not, and if so, the image is segmented through histogram thresholding, which is optimized via a quantization problem. As for (ii), the classification of different crack patterns is mostly considered for pavement applications, where cracks can be distinguished between an alligator, diagonal, longitudinal, or transverse. Examples of this classification can be found in (HOANG, 2018a; HOANG; NGUYEN, 2019, 2018; OLIVEIRA; CORREIA, 2014; QINGGUO *et al.*, 2019b). The disadvantage of this approach is that it generally achieves low accuracy due to the lack of large and more comprehensive annotated datasets, but an alternative is to apply data augmentation.

A wide literature review considering papers published in the past ten years, with a focus on new pavement and concrete applications, is summarized in Appendix A, where the techniques adopted in each step are highlighted. From these applications, 55% are tailored for concrete, 34% for pavement and 11% for other materials.

In Appendix A, the “Goal” column specifies if the paper addresses a problem of localization (L), classification (C), or both. In the latter, the problems are specified in the sequence they are performed. Moreover, the classification problem is followed by the number that indicates how many classes are taken into consideration, including the “no crack” class.

Although most applications use images acquired from digital cameras without details about the image acquisition process, some papers use vehicles mounted cameras (AMHAZ *et al.*, 2016; PRASANNA *et al.*, 2016; SUTTER *et al.*, 2018) or UAVs (DORAFSHAN; THOMAS; MAGUIRE, 2019; PAGLINAWAN *et al.*, 2018; ZHU, Q *et al.*, 2018). In fact, the use of this kind of technology makes the process of crack detection even more autonomous and efficient.

There are even some papers in which the image is acquired in a test environment, and they are marked with “*”.

Slightly more than half of the papers rely on pre-processing methods either to denoise and blur background characteristics, with the most common method being the median filter, or to enhance abrupt color change, as one of the assumptions of crack detection methods presented earlier is that cracks are darker than the background. Only a few papers focus on eliminating extraneous objects from the images, for instance, white lanes (OLIVEIRA; CORREIA, 2014).

As mentioned above, segmentation is one of the most used methods across the literature, and this can be verified in Appendix A, specifically operations such as hole filling and closing. For edge detection based segmentation, Canny and Sobel operators are generally adopted, and some papers propose improvements taking into consideration the specificities of the application such as (SHAN; ZHENG; OU, 2016; WANG, Gaochao; TSE; YUAN, 2018). Another method for segmentation found in the literature is based on clustering, more explicitly using K-means, which is an unsupervised method. The idea is to distinguish crack-like regions from background and noise by grouping them based on similarity. The criterion for grouping varies though. For example, it can be based on the color similarity (CHO *et al.*, 2018b), the extracted features (WANG, Gaochao; TSE; YUAN, 2018), or the Euclidean distance between regions (SANTOS; VALENÇA; JÚLIO, 2019).

The feature extraction step is typically performed in combination with ML methods. For crack detection, the most common features are related to gray level distribution, texture, crack characteristics (shape or orientation), and projective integrals. It is worth noting that, at the best of author’s knowledge, Ref. (OLIVEIRA; CORREIA, 2014) was the only paper studied that adopts feature normalization, what is a type of data pre-processing that corresponds to scaling each observation to have unit norm across all features (PEDREGOSA *et al.*, 2011). Another type of data preprocessing is standardization, which scales each feature, as opposed to each observation. Then, the use of data preprocessing needs to be further investigated during the development of the methodology that will be proposed in this dissertation.

Although RL techniques were widely adopted in old works due to its simplicity (SHI *et al.*, 2016), Appendix A shows that it is still well accepted in the literature. In fact, many papers rely solely on features such as the area or length of segmented regions to determine whether a region is a crack or not. The drawback of this approach is the need for fine-tuning parameters and, hence, the loss of generality since it is deterministic. Other issues that need to be addressed are: detecting fine cracks (HOANG, 2018b), robustness to various positions (HEO; JEON; SON, 2019; LEI *et al.*, 2018) and the number of cracks (LEI *et al.*, 2018; LIANG, Sun; JIANCHUN; XUN, 2018), and considering ramifications (LIANG, Sun; JIANCHUN; XUN, 2018).

Conversely to the papers mentioned above, which deal with the classification of individual segmented objects from the original image, Ref. (DORAFSHAN; THOMAS; MAGUIRE, 2018b) used CNN for inspection of concrete constructions by detecting whether the image contains or not crack(s). They verified that image quality is an essential factor in detection accuracy and emphasized the need to diversify defects in the training dataset. Ref. (CHA; CHOI; SUH; MAHMOUDKHANI; *et al.*, 2018) proposed an approach for quasi-real-time simultaneous detection of multiple types of concrete damage, using a Faster Region-based CNN (Faster R-CNN). In fact, Ref. (ZHANG, L. *et al.*, 2016) compares the use of CNN against a methodology based on Figure 1 that uses Support Vector Machine (SVM) and Boosting for classification of individual pixels, showing that CNN is superior.

Although Deep Learning (DL) techniques, such as CNN, have increasingly been applied in a wide range of areas (WANG, Huai-zhi *et al.*, 2017), its performance heavily depends on the architecture applied and on the size and variety of the training dataset (DORAFSHAN *et al.*, 2018). Because of that, training a DL model computationally intensive (ZHAO *et al.*, 2016), requiring the use of GPUs and limiting its application in standard computers. Moreover, aside from the fact that a large dataset can be proven difficult to obtain in some cases, it requires much manual effort for labeling (MNEYMNEH; ABBAS; KHOURY, 2018).

On the other hand, with approaches based on image pre-processing, segmentation, and classification, it is possible to achieve desirable performance results even for small data sets, especially given that the publicly available crack datasets are limited (MEI; GÜL, 2019). Then, this work aims at developing a robust methodology for crack detection that addresses the drawbacks of previous works based on the methodology shown in Figure 1 as well as the limitations of (i) not having big datasets, (ii) standard computational resources, and (iii) poor crack image quality, i.e., noise on the image and from the background.

## 2.2 TEXTURE ANALYSIS

Another classical technique used in CV is texture analysis. According to (LIU, Li *et al.*, 2012a), texture analysis has been an area of intense research, applied in fields such as medical image analysis, remote sensing, object recognition, document analysis, and environmental modeling. Nowadays, texture analysis is also commonly found in areas such as system reliability and fault detection. For instance, Refs. (FARDO; DONATO; RODRIGUES, 2018; RUIZ *et al.*, 2018) proposed a method to retrieve the crack contour through texture analysis. Ref. (QUINTANA; TORRES; MENÉNDEZ, 2016) combined the use of Hough transform features and Local Binary Pattern (LBP) to extract edge orientation and texture features, building crack seeds, which are further processed to obtain a clearer image of the crack. Although texture analysis is widely applied in different contexts, there is still a need to enhance its effectiveness for crack detection in images with high levels of surface texture (MOHAN; POOBAL, 2018), such as in concrete surfaces compared to a plastic one.

In this context, (TEDESCHI; BENEDETTO, 2017) designed cascade classifiers based on LBP features to detect three types of pavement distress, i.e. fatigue cracks, longitudinal-transversal cracks, and potholes, reaching an accuracy of 70% approximately. Ref. (HU; ZHAO; WANG, 2010) used texture analysis and shape descriptors to extract features from surface images, treating cracks as inhomogeneous to the pavement texture. In this case, the texture analysis included six commonly used Haralick features from the Gray-Level Co-occurrence Matrix (GLCM). As for shape descriptors, principal axes and compactness were implemented because they are translation invariant, and this is important in the context of uneven illumination. Following the approach presented in (POREBSKI; VANDENBROUCKE; MACAIRE, 2008; WANG, Guo-De *et al.*, 2012), this dissertation explores the application of the LBP operator to images and the calculation of its GLCM. Hence, we extract Haralick features (HARALICK RM, SHANMUGAM K, 1973) to use as texture descriptors. Moreover, other shape descriptors will be explored as well.

LBP is a classical and widely used method to describe textures, which was initially proposed in (OJALA; PIETIKÄINEN; HARWOOD, 1996), and is formally denoted by Equation (1), where *s(x)* is a "local pattern" operator, $p$ is the number of points, $x_{0,0}$ denotes the central pixel, and $x_{r,n}$ is the gray values of pairs of $n$ equally spaced pixels on a circular radius $r$. In this case, the operator determines the relationship between $x_{0,0}$ and its neighborhood ($x_{r,n}$), in terms of its pixel differences. Although LBP is immune to changes in lighting conditions (TEDESCHI; BENEDETTO, 2017), some limitations can be listed, such as sensibility to image rotation, small spatial support, loss of local textural information; and high sensitivity to noise (LIU, Li *et al.*, 2012a).

$$LBP_{p,r} = \sum_{n=0}^{p-1} s(x_{r,n} - x_{0,0})2^n \tag{1}$$

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

To enhance the conventional LBP power of classifying textures, Ref. (LIU, Li *et al.*, 2012a) proposed an improvement considering two different types of features in a local patch: the pixel intensities and the pixel differences. The use of LBP, conventional or not, implies in a descriptor, which is a histogram of textural information. The advantage of these descriptors is the extraction/selection of useful features to classify textures. Ref. (LIU, Li *et al.*, 2012a) tested each one of the proposed LBP separately and also together, which was called by the authors as a joint histogram and had a better performance.

The pixel intensities are divided into two components: the intensity of the central pixel and the intensities of its neighboring pixels. For the pixel difference, the radial difference is analyzed. Then, the three descriptors proposed by (LIU, Li *et al.*, 2012a) are described below: central intensity (CI) – Equation (2), neighboring intensity (NI) – Equation (3), and radial difference (RD) – Equation (4).

$$CI - LBP = s(x_{0,0} - \mu_r) \tag{2}$$

$$NI - LBP_{p,r} = \sum_{n=0}^{p-1} s(x_{r,n} - \mu)2^n \tag{3}$$

$$RD - LBP_{p,r,\delta} = \sum_{n=0}^{p-1} s(\Delta_{\delta,n}^{Rad})\, 2^n \tag{4}$$

where

- $\mu_r$ is the mean of the whole image, i.e., $\mu_r = \frac{1}{p}\sum_{n=0}^{p-1} x_{r,n}$;
- $\delta$ is an integer radial displacement;
- $\Delta_{\delta,n}^{Rad} = x_{r,n} - x_{r-\delta,n}$ is the radial difference computed with a given integer radial displacement $\delta$, where $1 \leq \delta \leq p/2$, and $x_{r-\delta,n}$ corresponds to the gray values of pairs of $\delta$ equally spaced pixels of the same radial direction;
- $\varepsilon$ is a threshold value. (LIU, Li *et al.*, 2012a) sets $\varepsilon = 0.01$;
- $\Delta_{\delta,n}^{Ang} = x_{r,n} - x_{r,mod(n+\delta,p)}$ is the angular difference computed with a given angular displacement $\delta(2\pi/p)$, and $x_{r,mod(n+\delta,p)}$ corresponds to the gray values of pairs of $\delta$ equally spaced pixels on a circular radius $r$, and function $mod(x,y)$ is the modulus $x$ of $y$.

This dissertation compares different approaches to extract features and identify cracks in concrete images. The first approach, here called texture-based, uses all the LBPs proposed in (LIU, Li *et al.*, 2012a) and calculate its GLCM to extract four Haralick features, i.e., contrast, correlation, energy, and homogeneity. Moreover, some shape descriptors will also be used, and they are described in Section 3. The second approach is the region-based, which relies on image segmentation, following the methodology shown in Figure 1. The features extracted from the images in each approach will feed ML models to detect crack existence. The features implemented for each approach are based on the existing literature, though some are new to the context of crack detection. Hence, the idea is to determine which features are relevant for crack detection to improve computational speed.

Furthermore, the proposed methodology aims at overcoming the problem caused by the variety of both crack and background characteristics by determining proper and discriminative features. Therefore, the proposed methodology will be tested in a recently available concrete crack database with diverse image characteristics. The idea is that this methodology can be coupled in autonomous vehicles such as UAVs or solutions like the one presented in (SUTTER *et al.*, 2018), which is a vehicle specifically designed for bridge inspection, for example.

## 2.3 MACHINE LEARNING (ML)

ML algorithms' goal is to learn patterns as closely as possible, which can be a regression or a classification problem, making use of labeled training data (supervised learning) or not (unsupervised learning) (HAMEL, 2009). In the former case, the algorithm learns a function that maps the inputs into the outputs, minimizing the difference between the predicted and the real (ground truth) output (BAYAR; BILIR, 2019; MODARRES *et al.*, 2018). Because it is a vast field, there is not a universal best model (or model configuration) for all classification tasks (PRASANNA *et al.*, 2016). Hence, it is essential to investigate different models and configurations (based on the definition of hyper-parameters) to determine the one that best fits each problem.

The hyper-parameters need to be set before training the model, and because they determine the configuration of the learning algorithm, they affect its performance (CLAESEN; DE MOOR, 2015). This dissertation aims at classifying images as crack or non-crack, based on the features extracted from those images, which will feed ML models. To that end, four different models were compared: SVM, Multilayer Perceptron (MLP), AdaBoost (AB) and Random Forest (RF). These models were chosen because they have been successfully applied in crack detection (LIU, Wenjun *et al.*, 2019). The following subsections will provide a brief overview of these models and their hyper-parameters.

### 2.3.1 Support Vector Machines (SVM)

SVM is a supervised ML technique, based on statistical learning theory (VAPNIK, 2000) that can be applied both to regression and classification problems. The standard version of the latter relies on the concept of a margin by the development of a decision boundary (hyperplane) that maximizes the separation between two classes (margin) while minimizing the classification error and improving generalization (HOANG; NGUYEN, 2019; KOTSIANTIS, 2007). The hyperplane is mapped by a kernel function, which can be of different types, such as linear, polynomial, sigmoid, or Radial Basis Function (RBF) (VAPNIK, 2000), making the model versatile to both linearly and non-linearly separable problems (PIRI; DELEN; LIU, 2018).

Equations (5) to (7) defines the margin optimization problem, where $\phi$ is the dual Lagrangian, $\alpha$ is the dual variable, $k(x_i, x_j)$ is the kernel function, $l$ is the number of training instances, and $C$ is the soft margin parameter. SVM involves the optimization of a convex quadratic programming problem (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2006), which ensures a single global solution as opposed to other methods (LINS *et al.*, 2013).

$$\max \phi(\bar{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\, \alpha_j y_i y_j k(x_i, x_j)\,, \tag{5}$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6}$$

$$0 \leq \alpha_i \leq C, \tag{7}$$

### 2.3.2 Multilayer perceptron (MLP)

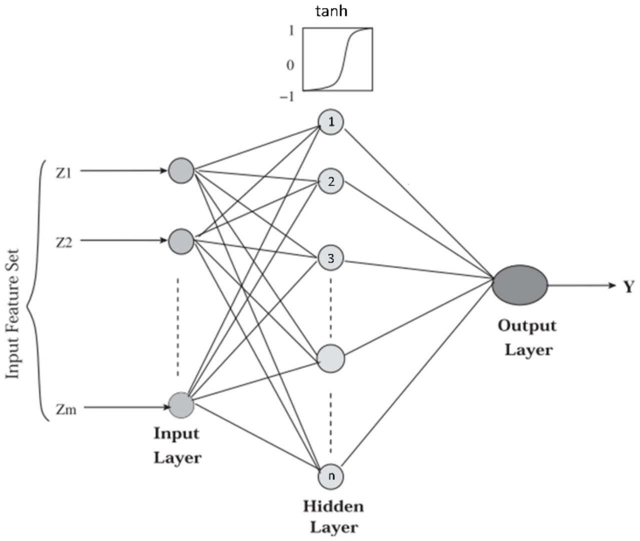
MLP is a popular feedforward Artificial Neural Network (ANN) architecture, which is arranged in layers, and it can deal with both regression and classification problems (LIU, Ruonan *et al.*, 2018). A perceptron is an operator that computes the weighted sum of the input features and adjust the result to 1 if it is above a threshold, or to 0 otherwise (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2006). The problem is that a perceptron can only classify linearly separable data, then MLP was created to solve this problem (KOTSIANTIS, 2007).

In MLP, the first layer is called the input layer, comprising a set of features that will be used to map the response variable, represented by the output layer. Between these two layers, there are hidden layers composed of a set of neurons that receives information from the preceding layer, transforms this information with a weighted linear summation, and, if activated, passes it to the next layer (PANDEY; BARAI, 1995).

The activation function, which is one hyper-parameter, introduces non-linearity to the model and can be of various types, such as the hyperbolic tangent (tanh) and the Rectified Linear Unit (RELU). An example of a generic one-hidden layer MLP with *n* neurons and activation function tanh is shown in Figure 2. Consequently, the learning process is based on a feature hierarchy, where high-level features are learned from lower level features (GLOROT; BENGIO, 2010).

Here, the weights will be learned and updated by the backpropagation method, in which the prediction error (loss function) is propagated through the network. Also, it is possible to add a regularization term (α) to the loss function, to prevent overfitting (EISENBACH *et al.*, 2017). The way the weights of the network are updated given the error depends on the solver for weight optimization, which is another hyper-parameter.

Figure 2 - Generic one hidden layer MLP



Source: adapted from (KALYANI; SWARUP, 2013)

### 2.3.3 AdaBoost (AB)

AB is an iterative procedure, proposed by (FREUND; SCHAPIRE, 1997), which addresses classification problems. It is iterative because it starts by fitting a classifier (usually decision trees) on the original data and, if a misclassification occurs, the weight of that datapoint is increased (boosted), while the weights of correctly classified data are decreased. Initially, the weights are set to $1/N$, where $N$ is the number of training samples, and they are updated over the iterations.

Finally, a new classifier is applied to the modified data at each iteration (SAGI; ROKACH, 2018), and the final score is given as a linear combination of the predictions at each stage (ZHU, J. *et al.*, 2009). Hence, the method adaptively adjusts to the errors, as samples that are difficult to predict receive ever-increasing weights, forcing the learning algorithm to focus on them (FREUND; SCHAPIRE, 1997). In this model, the main hyper-parameter that needs tuning is the number of estimators, i.e. the number of boosting iterations.

### 2.3.4 Random forests (RF)

RF is a classifier consisting of an ensemble of decision trees (HOANG; NGUYEN, 2019), which are generated by independent random vectors, but identically distributed (BREIMAN, 2001). Each tree classifies sub-samples of the input data, and the results are averaged to cast the vote of the tree, i.e., which class does the data belong. Hence, the final prediction is based on the class with the highest number of votes, considering the whole forest. Compared to other methods, aside from being easier to tune its parameters, RF produces good results and the possibility of analyzing features’ importance (SAGI; ROKACH, 2018). Also, classification error tends to converge to a limit as the number of trees, i.e. the hyper-parameter called the number of estimators, becomes larger (BREIMAN, 2001).

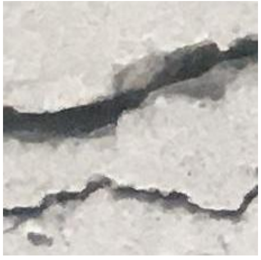
# 3 METHODOLOGY

The proposed methodology relies on the general architecture of crack detection methods based on image processing presented in Figure 1, but here another approach based on texture analysis will also be assessed. Hence, it entails the extraction of features from images to be used as input to ML techniques for crack detection. In our case, the image is the grayscale or the segmented version of the original one. Particularly, the segmentation aims to create a new image containing only white parts that resemble a crack (the procedure will be described in detail in Section 3.1). To improve segmentation precision, both pre- and post-processing techniques were implemented, and their effectiveness will be discussed.

According to (MEI; GÜL, 2019), crack detection methods can be categorized into two types: one focused on the classification of an entire region of the image and the other focused on retrieving pixel-level information and localizing the crack within the image. In this work, we separately analyzed these two approaches for automated crack detection, extracting different features in each case, and comparing their performance in labeling an image as cracked or non-cracked. The features are based on the literature and will be presented in Section 3.2. The difference between them relies on the focus of analysis, as one can only provide information about the existence of cracks, and the other analyzes each region of the segmented image (i.e., connected white pixels). The former is herein called texture-based analysis and the latter as region-based analysis, which can also localize cracks in the image.

Note the problem complexity increases from the texture-based to the region-based analysis, as the result gets more detailed by changing from one to another. For example, in Figure 3, the texture-based approach would indicate that this image contains crack(s), i.e., label this image as cracked. With this result, it is not possible to know how many cracks there are (in this case, two). On the other hand, the region-based approach attempts to indicate the exact pixels that define each crack, and hence label this image as cracked. So, the region-based approach heavily depends on the effectiveness of the segmentation step, but its result is more informative than the one provided by the texture-based approach.

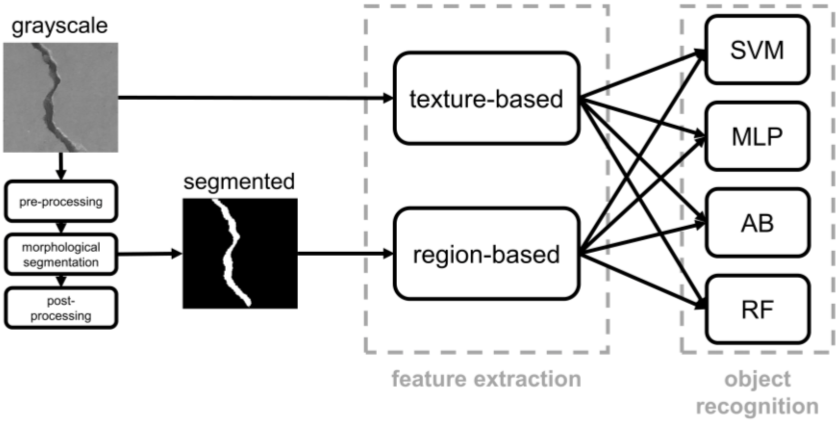
Figure 3 - Example of a concrete crack image containing two cracks.



Source: (ÖZGENEL, 2018).

As the feature extraction step comprises the concatenation of several features with different magnitudes, each providing a weak classification cue (PRASANNA *et al.*, 2016), the application of standardization will also be considered. This step aims at centering the features, by removing the mean value of each feature, and then rescaling them, by dividing by their standard deviation. The learning process could benefit from data standardization; otherwise, the objective function could be more influenced by higher values. Then, the proposed methodology is summarized in Figure 4 and detailed in the next subsections.

Figure 4 – The crack detection methodology.



Source: The Author (2020).

## 3.1 SEGMENTATION

Segmentation comprises the identification of objects that can potentially be classified as cracks in the image, removing irrelevant or unrelated information that belongs to concrete pattern characteristics (i.e., noise, background). Before the segmentation itself, bilateral filtering, which consists of a nonlinear combination between image values (TOMASI; MANDUCHI, [*s. d.*]), was applied to smooth the input images as both concrete and asphalt images are characterized by a noisy background. Not only did this specific pre-processing technique outperforms others, such as the median filter, but was also chosen because of its potential for edge preservation and Ref (FAN, Rui *et al.*, 2019) obtained good results in the same context of crack detection. The segmentation technique was implemented in MATLAB and is based on the morphological operation proposed in (JAHANSHAHI *et al.*, 2013), shown in Equation (8), where I is the grayscale image, ○ is the opening morphological operation, ● is the closing morphological operation, and SE is the linear structuring element, defined for orientations of 0º, 45º, 90º and 135º.

$$T = max\left[\left(I \circ SE_{[0°,45°,90°,135°]}\right) \bullet SE_{[0°,45°,90°,135°]}, I\right] - I \tag{8}$$

It is essential to understand that the SE is determined by two parameters: format and size. The linear format chosen for SE enables the segmentation of objects perpendicular to each orientation, while the size narrows the segmentation by imposing a constraint for object length. Therefore, the morphological operation described in Equation (8) is repeated for each image and several SE of different sizes to map the most information about objects. The resulting image of each iteration is binarized by the Otsu method (OTSU, 1979), which is an algorithm that searches for the threshold for segmentation that minimizes the between-class variance (TALAB *et al.*, 2016), and joined together to compose one single image, finally separating the cracks from the complex background and keeping as many characteristics of the original crack as possible (XING *et al.*, 2018). The reason for choosing the Otsu method relies on its high applicability in the context of crack detection, as can be seen in Appendix A.

Figure 5 presents the application of the proposed segmentation methodology, including examples of the segmentation result yet containing noise inside the crack and related to the background (i.e., black dots and discontinuity). In order to address this issue, post-processing techniques were deployed. The following steps, based on the method proposed by (LIANG, Sun; JIANCHUN; XUN, 2018), were used to fill internal discontinuity and eliminate broken parts of the crack.

(i) Fill holes in the segmented crack image;

(ii) Perform morphological closing (dilation followed by erosion);

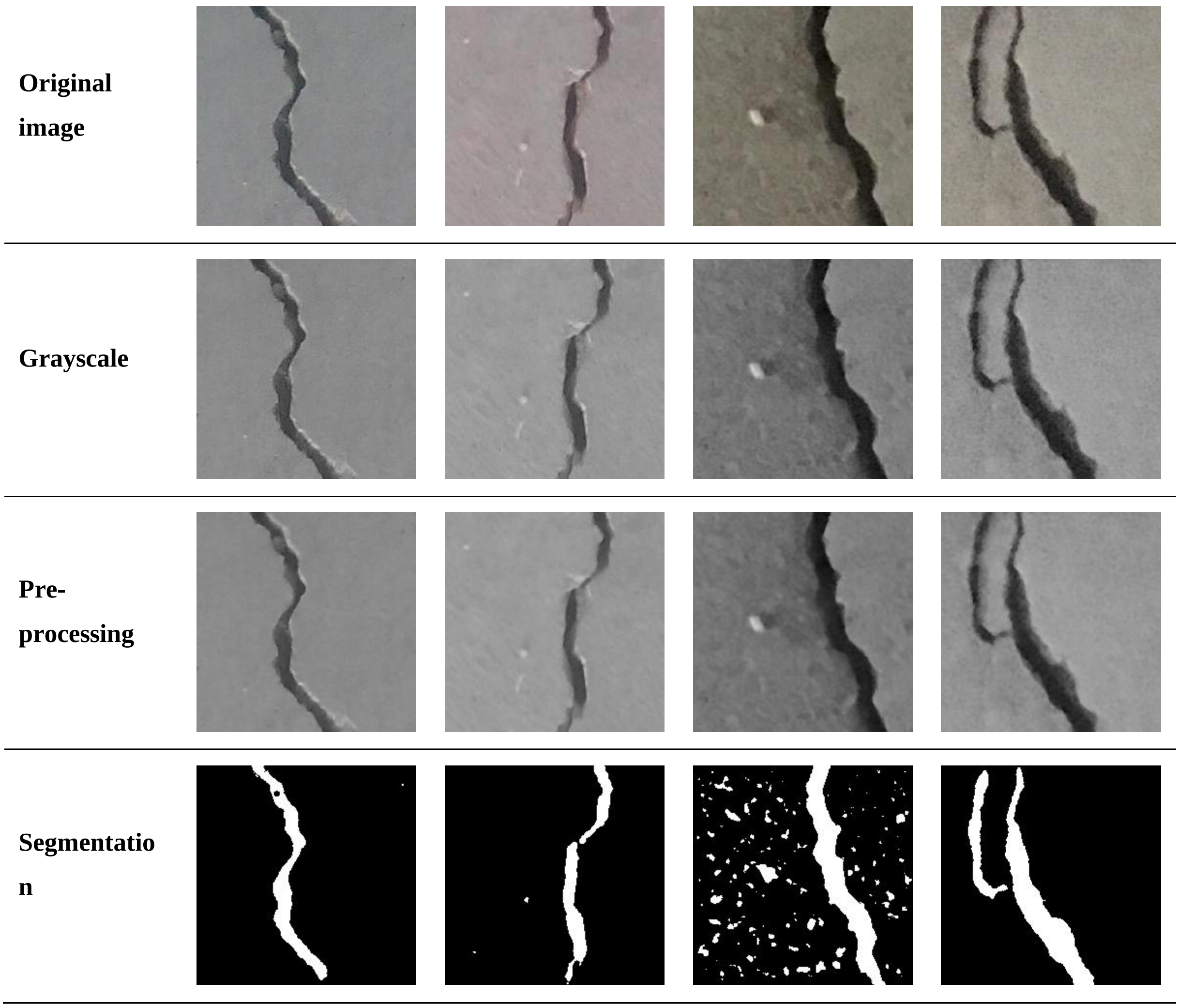
(iii) Obtain the skeleton line of the crack to identify crack direction;

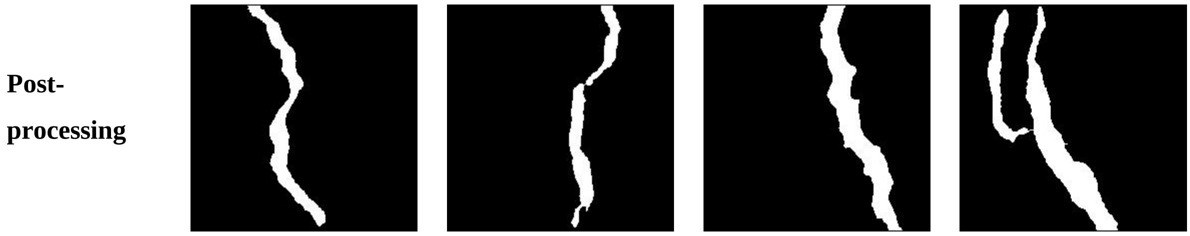
(iv) Search for breaking points of the crack and connect points that are at a given distance away from each other in the direction of the skeleton line;

(v) Remove objects, whose area falls below a threshold.

Steps (iii) and (iv) were implemented using methods of “Edge Linking and Line Segment Fitting” developed in (KOVESI, 2000). In step (v), an irrelevant object can be removed by comparing its area with the whole image (NGUYEN; KAM; CHENG, 2014). Results of the application of the whole segmentation methodology in a concrete crack dataset (ÖZGENEL, 2018), including both pre- and post-processing techniques, are shown in Figure 5.

Figure 5 - Results of the segmentation methodology applied to the concrete crack dataset.





Source: The Author (2020).

## 3.2 FEATURE EXTRACTION

Features are quantitative attributes or properties that, aside from describing characteristics of the image (or image regions), also help to distinguish different patterns (JAHANSHAHI *et al.*, 2013). However, to determine features that are discriminative and have the potential to be generalized is a challenging task (HAN *et al.*, 2019). In this context, features must be invariant to changes in condition, such as illumination, for example (NIXON; AGUADO, 2008). Then, a feature extraction process is a particular form of extracting information from the image for comparison and analysis purposes (DAVOUDI; MILLER; KUTZ, 2018b), to determine the class of the objects in the image (RUCK; ROGERS; KABRISKY, 1990).

Texture- and region-based approaches for feature extraction were analyzed and implemented in MATLAB and are described in the following subsections. Each approach comprises the extraction of multiple features. The idea is that each feature provides a weak classification cue, but their combination enables a more accurate classification (PRASANNA *et al.*, 2016). Thus, the feature vector in each case, whose classification is already known, will feed ML models to classify images as crack or non-crack.

### 3.2.1 Texture-based analysis

The first approach considered extracting features from the entire image (i.e., textural properties). According to (NIXON; AGUADO, 2008), there is not a precise definition for texture, as it is usually attributed to human perception, and then the concepts for its analysis are flexible enough to handle different definitions. We are here interested in distinct textures presented in an image with a crack, once it is considerably different from the concrete image without any crack.

A classical and widely used method to describe textures is the LBP, which was initially proposed in (OJALA; PIETIKÄINEN; HARWOOD, 1996), and consists of creating a histogram of textural information by determining the relationship between each pixel and its neighborhood. LBP presents advantages such as a notable computational efficiency and a good texture discriminative property. However, limitations such as sensibility to noise and image rotation, small spatial support, and loss of local textural information motivate improvements in the method.

The LBP and its three above presented variants proposed in (LIU, Li *et al.*, 2012b) were applied in the grayscale image, generating four different histograms for each image. For each histogram and the grayscale image, the GLCM was computed, and four Haralick features (i.e., contrast, correlation, energy, and homogeneity in Table 1) were extracted, generating 20 features. Moreover, the other four features were extracted from the segmented image. The outcome of this step is a feature vector composed of twenty-four features, which are described in Table 1, and these features will be used to train the ML models that will be discussed in the next section.

Table 1 - Description of textural properties.

| Feature | Description | Examples |
|---|---|---|
| Contrast | Intensity contrast between a pixel and its neighbor over the whole image. | $\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$ |
| Correlation | Correlation between a pixel and its neighbor over the whole image. | $\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$ |
| Energy | Sum of squared elements in the Gray Level Co-occurrence Matrix (GLCM) | $\sum_{i,j} p(i,j)^2$ |
| Homogeneity | The closeness between the distribution of elements in GLCM and its diagonal. | $\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$ |
| Thresh Out | The relative proportion of the cracks to non-crack objects. | The threshold value for the Sobel edge detection method |
| Entropy | A statistical measure of randomness. | $-\sum_{p} p \log_2 p$ |
| Local variance | The average local standard deviation of 3 × 3 neighborhood around each pixel in the image. | $stdfilt$ function in MATLAB divided by the number of pixels in the image |
| Standard Deviation | The standard deviation of all values. | $std2$ function in MATLAB |

Source: The Author (2020).

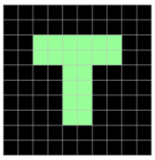
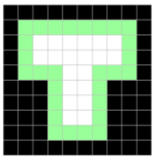
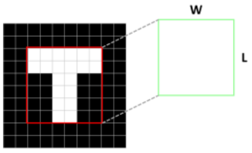
### 3.2.2 Region-based analysis

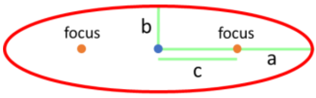
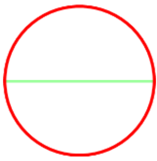
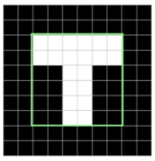
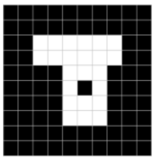
Unlike the texture-based approach, the region-based analysis is concerned with parts of the image. In this way, it focuses on the extraction of features for each segmented region of the image. Here, a region is defined in white pixels as an 8-connected component that represents a crack.

We considered that the feature vector for each region includes features provided in (JAHANSHAHI *et al.*, 2013; LIANG, Sun; JIANCHUN; XUN, 2018) as well as some implemented in MATLAB. All thirteen features are described in Table 2, and, together, they should be able to capture the characteristics of the crack that distinguishes it from the background and other objects. To make descriptions clearer, Table 2 provides examples, where components of interest are identified in green.

These features will be used to train the ML models that will be discussed in the next section. By classifying segmented regions, it is possible to determine the number of cracks in the image and the precise location of each crack. Note that one feature vector is generated for each region of the image. As the same image may contain more than one crack and other misleading objects, it may generate more than one feature vector.

Table 2 - Description of region properties.

| Feature | Description | Example |
|---|---|---|
| Filled Area (A) | The number of pixels in the region with filled holes |  |
| Perimeter (P) | Distance around the boundary of the region. |  |
| Proportion of crack pixels | The ratio of the area of the region (A) and the area of the smallest circumscribed rectangle of the crack. |  $\frac{A}{WL}$ |
| Length-width ratio | The length-width ratio of the smallest circumscribed rectangle of the crack. | $\frac{L}{W}$ |

| Feature | Description | Example |
|---|---|---|
| Image Concentration (K) | The ratio between the area of the region (A) and its perimeter (P). | $K = \frac{2 \times \sqrt{\pi} \times \mathrm{A}}{\mathrm{P}}$ |
| Compactness | The ratio between the square root area of the region (A) and its perimeter (P). | $\frac{A}{P}$ |
| Eccentricity | The ratio between the distance of the ellipse foci (c) that has the same second-moments as the region and its major axis length (a). |  $\frac{c}{a}$ |
| Area ratio | The ratio between the area of the region (A) and the area of the ellipse ($ellipse\ area = \pi ab$) that has the same second-moments as the region. | $\frac{A}{\pi ab}$ |
| Major Axis Length | Length of the major axis of the ellipse that has the same normalized second central moments as the region. | $2a$ |
| Equivalent Diameter | The diameter of a circle with the same area as the region. |  $2\sqrt{\frac{A}{\pi}}$ |
| Convex Area | The number of pixels in the image that specifies the convex hull, which is the smallest convex polygon that can contain the region. |  $WL$ |
| Solidity | The proportion of the pixels in the convex hull that is also in the region. | $\frac{A}{CA}$ |
| Euler Number | The number of objects in the region minus the number of holes in those objects. |  $=$ 0, in this case |

Source: The Author (2020).

## 3.3 MODEL SELECTION

Although the segmentation method proposed is time effective, misleading objects could be segmented in the process, raising the need for an image classifier (PANELLA *et al.*, 2018). Hence, the next step is the classification (see Figure 4). To that end, we chose some popular classification models: SVM, MLP, AB, and RF, comparing results provided by them. Given that data is annotated, these models will be trained using the feature vector of each approach.

All models rely on hyper-parameters, which are inputs of the models instead of being learned. Because there are several combinations between their values, an exhaustive search of all hyper-parameter combinations (grid search) was conducted to select the ones with the best cross-validation score. In *k*-fold cross-validation, the data is (approximately) equally split into *k* subsets, and the model is trained $k$ times, each time leaving one of the subsets for validation (KOHAVI, 1995). The final score of the model is the average over $k$ training sets, providing a pessimistic estimate (since the training data is smaller) of the generalization performance of the model (CAWLEY; TALBOT, 2010).

For each model, the hyper-parameter space is shown in Table 3, along with a brief description (PEDREGOSA *et al.*, 2011). Note that, in SVM, as gamma is a hyper-parameter related to the kernel RBF, it will only be set if the kernel selected is the RBF. As for MLP, the parameter "nº of neurons per layer" indicates both the number of hidden layers and the number of neurons in each layer. Hence, the number of hidden layers is determined by the number of elements between parentheses, where each element represents the number of neurons in each layer. For the grid search, 4 different combinations of the number of hidden layers and the number of neurons in each layer were determined, as shown in Table 3.

Table 3 - Hyper-parameter space

| Model | Hyper-parameters | Description | Values |
|---|---|---|---|
| SVM | kernel | Decision function | linear and RBF |
|  | C | Soft margin parameter | [1, 10, 100, 1000] |
|  | $\gamma$ | RBF parameter | [1, 0.1, 0.001, 0.0001] |
| MLP | nº of layers | Number of layers | [1, 2, 3] |
|  | nº of neurons per layer | Number of neurons in each hidden layer | [(100), (100,100), (100,50), (100,100,100)] |
|  | solver | Weight optimization | SGD and adam |
|  | α | Parameters of the regularization term | [0.1, 0.01, 0.001, 0.0001, 0.00001] |
|  | Activation | Neurons' activation function | ReLu |
| AB | Number of estimators | Maximum number of decision trees at which boosting is terminated | [50, 100, 200] |
| RF | Number of estimators | Number of trees in the forest | [50, 100, 200] |

Source: The Author (2020).

The selection of hyper-parameters will be based on Balanced Accuracy (BA) score using 4-fold cross-validation. BA is a useful metric when the classes in the training set are imbalanced, and is defined in Equation (9) as the average of recall obtained on each class, where $P$ is the number of positive predictions, $TP$ is the number of true positive predictions, $N$ is the number of negative predictions, and $TN$ is the number of true negative predictions.

$$BA = \frac{\left(\frac{TP}{P} + \frac{TN}{N}\right)}{2} \tag{9}$$

Once the hyper-parameters are selected, all classification models were trained for the texture-based and region-based features along with the ground truth, resulting in eight different models. Results are presented in the next chapter, comparing the performance of the models. For this step, the Python computational language and the Google Collaboratory, which is a free cloud platform for Python, were used.

The performance is evaluated in terms of three metrics: BA, Specificity (S) and Recall (R), which are described in Equations (9) to (11) respectively (SENSITIVITY AND SPECIFICITY, [*s. d.*]). Accuracy is the proportion of the correctly classified samples to the total number of samples. Specificity is the ability of the classifier not to label as positive a sample that is negative, representing a useful metric, when the cost of false positive is high. Recall (or sensitivity) is the ability of the classifier to find all the positive samples and, hence, is an informative measure when there is a high cost associated with false negatives. In reliability analysis, in general, recall is the most critical metric once the cost of a false negative can be catastrophic if, for instance, human safety is involved.

$$S = \frac{TN}{N} \tag{10}$$

$$R = \frac{TP}{P} \tag{11}$$

# 4 RESULTS AND DISCUSSION

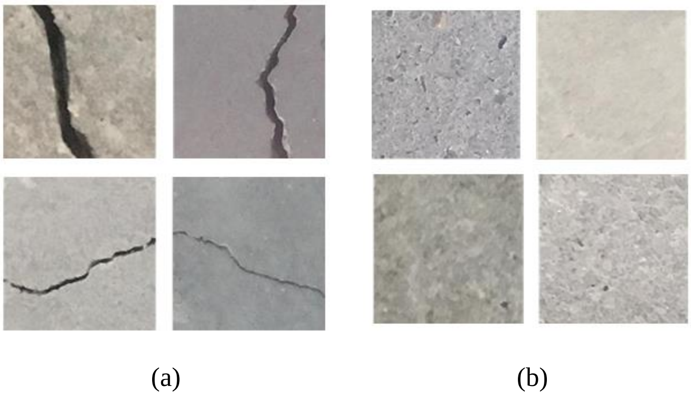
According to (MEI; GÜL, 2019), there is a need for the development of challenging and pixel-level annotated crack datasets, as it enables the deployment of novel algorithms and the comparison of results. In fact, this is especially true for specific materials, such as concrete and wind turbines. One of the largest concrete crack datasets available (ÖZGENEL, 2018) contains images with 96 dpi and dimension 227 x 227, taken from the Technical University of the Middle East (METU) buildings, representing concrete parts of these buildings containing or not cracks.

Because we are dealing with supervised learning, the training of the ML models is based on positive and negative samples (TEDESCHI; BENEDETTO, 2017), which are crack and no crack images respectively. Moreover, the latter consists of background texture, which could be incorrectly identified as the object of interest. However, the dataset does not contain pixel-level annotation, and, according to (CHAMBON; MOLIARD, 2011), accurate segmentation must be provided as a reference. In order to overcome these aspects, a semi-automatic annotation was deployed, where the images were segmented with the aid of the methodology presented in Section 3.1, combined with manual interference to establish the ground-truth.

As mentioned in Section 3.1, the segmentation step involves the specification of some parameters. For example, as the idea is to make the model flexible enough to segment both thin and thick cracks with different lengths, the size of SE varied from 10 to 20 (by a step of 2) and from 20 to 104 (by a step of 6). These values were empirically determined by the observation that smaller SEs produce noisy results and that the information retrieved by each SE at smaller sizes is more distinctive and essential to catch fine details than at larger sizes (above 20, in this case). At the post-processing step, it was established that breaking points would be connected if they were at most 8 pixels away from each other. This value was sufficient to reduce discontinuities on the segmentation due to the effect of shadows or changes in illumination. Moreover, objects with areas smaller than 100 pixels were removed. The definition of this threshold was based on the results obtained in (CHO *et al.*, 2018b) and empirical observation.

Then, to evaluate the overall performance of the proposed crack detection algorithm, a total of 1,768 crack images and 1,166 non-crack images were selected and annotated. Examples of the dataset are shown in Figure 6, demonstrating the variety in both the crack and the concrete characteristics. With these images, a comparison between texture- and region-based analysis was performed using SVM, MLP, AB, and RF.

Figure 6 - Examples of (a) cracked and (b) non-cracked images.



Source: (ÖZGENEL, 2018).

As mentioned before, in Section 3.2.2, the same image may generate more than one feature vector for the region-based approach because images may contain more than one region. Indeed, from the ground truth, the region-based approach extracted 2,082 positive and 1,206 negative feature vectors. On the other hand, the number of feature vectors generated by the texture-based approach was the exact number of images.

For each case, 10 random splits of the feature vector were generated, considering 80% of the data for training purposes and the remaining 20% for testing. Hence, for each ML model, grid search, train and test were conducted 10 times, one for each of the different training and testing sets randomly generated. The performance of each model was averaged across the 10 runs and the results are presented in Table 4, in terms of the mean and the standard deviation. Table 4 also presents the hyper-parameters selected in each case, based on the best grid search score across the 10 runs.

The proposed methodology will be also compared to existing methodologies in the literature. For the texture-based, the benchmark model is composed only by the features of the traditional LBP (instead of all the other features proposed in the texture-based approach), whereas for the region-based, the benchmark model is the one proposed by Ref. (JAHANSHAHI *et al.*, 2013). For the benchmark models, the results are obtained following the same steps as the proposed methodology, i.e. average results are obtained with grid search, train, and test performed in 10 random splits of the dataset. However, for convenience, the hyper-parameters selected for the benchmark models will not be shown.

From Table 4 it is possible to note that the proposed methodology outweighs the benchmark methodologies, especially when dealing with SVM for the texture-based approach, since no results were obtained in a timely manner for the benchmark case. Also, note that for MLP the regularization term α was higher for the region-based approach, compared to the texture-based. This indicates that the value of the weights in the former is smaller than the latter (RASCHKA; JULIAN; HEARTY, 2016). Moreover, not only did MLP presented the highest standard deviations in the performance metrics, but also the worst performance, especially for the texture-based approach.

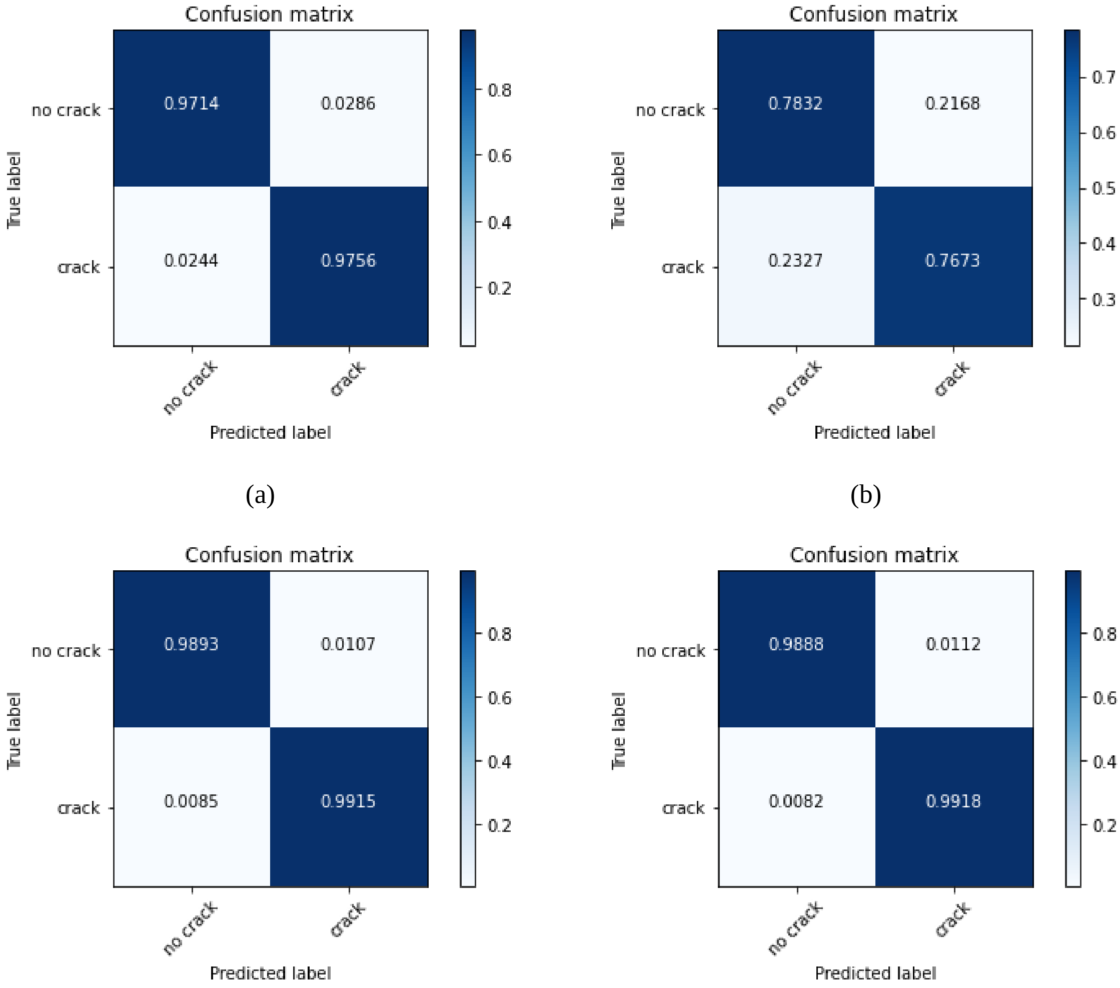
Table 4 - Classification performance of the ML models, in terms of mean and standard deviation.

| Method | Classification | Hyper-parameters | BA benchmark | BA | R | S |
|---|---|---|---|---|---|---|
| Texture-based | SVM | C: 10 or 1000, kernel: linear | - | 0.9735 ± 0.0060 | 0.9755 ± 0.0295 | 0.9714 ± 0.0550 |
| | MLP | α: 0.0001, nº of layers: 1, nº of neurons per layer: (100), solver: adam | 0.5004 ± 0.0008 | 0.7786 ± 0.1624 | 0.7673 ± 0.2960 | 0.7832 ± 0.3202 |
| | AB | Number of estimators: 100 or 200 | 0.7149 ± 0.0177 | 0.9904 ± 0.0031 | 0.9915 ± 0.0332 | 0.9893 ± 0.0026 |
| | RF | Number of estimators: 100 | 0.7356 ± 0.0203 | 0.9902 ± 0.0047 | 0.9918 ± 0.0305 | 0.9888 ± 0.0526 |
| Region-based | SVM | C: 1, kernel: linear | 0.9575 ± 0.0095 | 0.9751 ± 0.0046 | 0.9710 ± 0.0156 | 0.9792 ± 0.0189 |
| | MLP | α: 0.001, nº of layers: 1, nº of neurons per layer: (100), solver: adam | 0.9566 ± 0.0079 | 0.9684 ± 0.0089 | 0.9531 ± 0.0265 | 0.9841 ± 0.0408 |
| | AB | Number of estimators: 100 | 0.9545 ± 0.0087 | 0.9687 ± 0.0038 | 0.9758 ± 0.0136 | 0.9615 ± 0.0218 |
| | RF | Number of estimators: 50 | 0.9616 ± 0.0083 | 0.9753 ± 0.0073 | 0.9763 ± 0.0307 | 0.9742 ± 0.0528 |

Source: The Author (2020).

The result of the texture-based analysis is the classification of the image as cracked or non-cracked, while the outcome of the region-based approach is the classification of each segmented region as a crack or not. Three direct conclusions are immediately derived from the results in Table 4. Firstly, the performance of all ML models was excellent, especially the SVM, AB and RF. Figure 7 presents a visual representation of the information in Table 4, in terms of the mean value, showing that both AB and RF outperformed the other methods in terms of correctly classifying cracks for the texture-based approach, both presenting the best performances.

Figure 7 - Confusion matrix for the texture-based approach for (a) SVM, (b) MLP, (c) AB, and (d) RF.



(a)

(b)

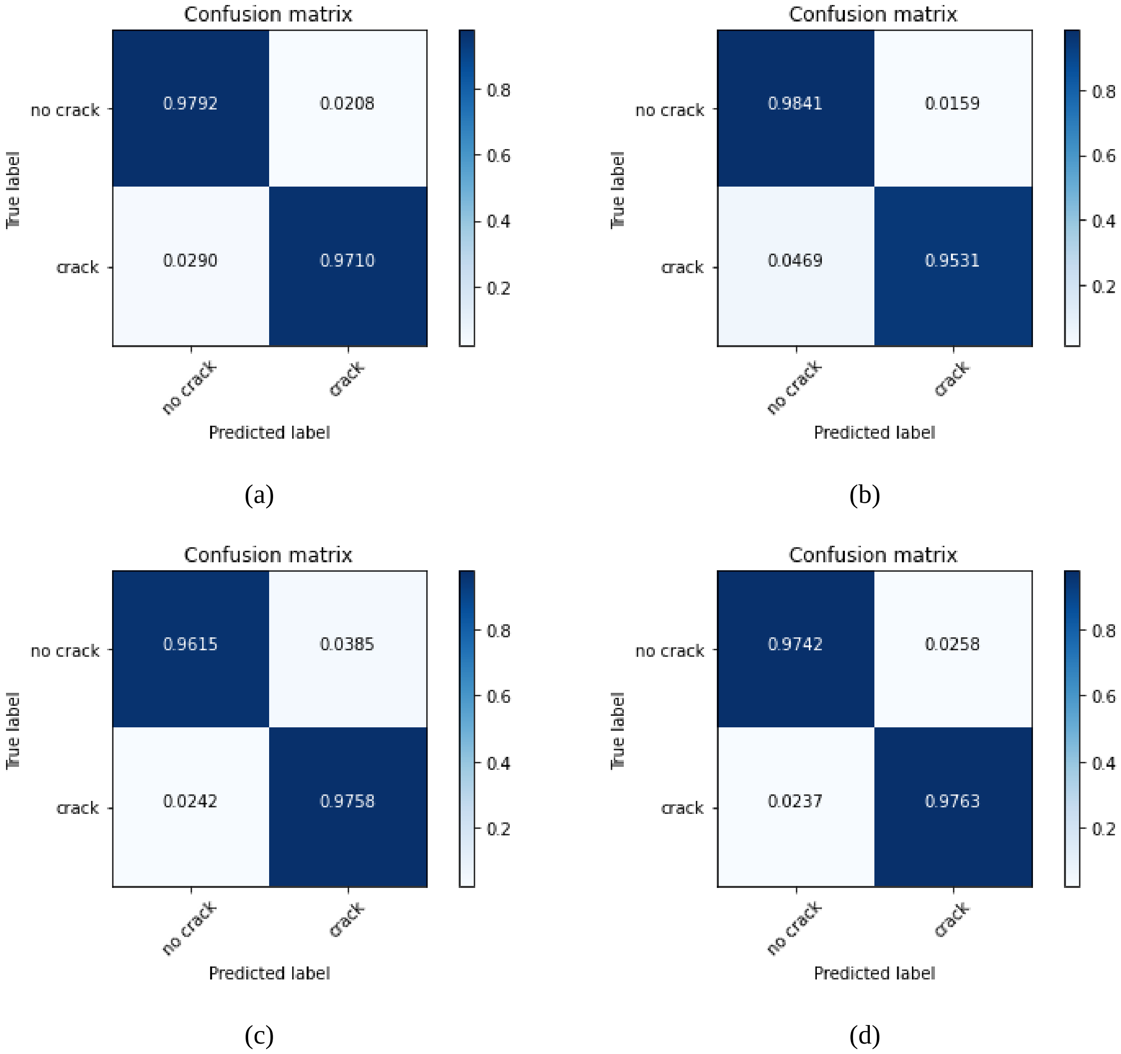
(c)

(d)

Source: The Author (2020).

Generally, the results of the region-based approach were slightly worse than the texture-based, except for the MLP, in which case not only did the region-based performed better but was also more consistent across the different data splits, i.e. small standard deviation. Figure 8 shows that SVM and RF results were better in terms of correctly classifying cracks. In terms of correctly classifying "no crack" objects, MLP achieved the best result by compromising the recall rate. As mentioned before, in the context of reliability, recall is the most critical metric since the cost of false negative classifications can be catastrophic.

Figure 8 - Confusion matrix for the region-based approach for (a) SVM, (b) MLP, (c) AB, and (d) RF .



(a)

(b)

(c)

(d)

Source: The Author (2020).

Secondly, the texture-based approach presents a slightly better accuracy performance than the region-based one, i.e., the best result for the texture-based in terms of averaged BA was 99% compared to 97% for the region-based. In fact, as mentioned before, the problem complexity increases from the former to the latter, since the former is concerned with labeling the whole image as cracked or not, while the latter aims at locating the crack on the image. The best performance was obtained with AB and RF for the texture-based approach, reaching 99% BA and R, and this result surpassed all the others, while the best performance for the region-based approach was obtained with SVM and RF, reaching 97% BA, R and S.

Lastly, note that MLP presented the worst performance for both approaches. The reason for that was the difficulty of convergence. In order to overcome this problem, data standardization was applied as it has the potential to prevent the domination of any feature on the objective function (KALYANI; SWARUP, 2013). So, with the application of data standardization, the performance of MLP considerably improved, as well as the performance of the other ML models (see Table 5). This improvement is more considerable for the texture-based approach since the chance of correctly classifying the observation jumped from 77% to 99% with data standardization.

Table 5 – Classification performance of the ML models with data standardization, in terms of mean and standard deviation.

| Method | Classification | Hyper-parameters | BA benchmark | BA | R | S |
|---|---|---|---|---|---|---|
| Texture-based | SVM | C: 1 or 10, kernel: linear | 0.7649 ± 0.0140 | 0.9975 ± 0.0013 | 0.9972 ± 0.0284 | 0.9979 ± 0.0424 |
| | MLP | α: 0.1, nº of layers: 3, nº of neurons per layer: (100, 100, 100), solver: adam | 0.7563 ± 0.0177 | 0.9956 ± 0.0025 | 0.9980 ± 0.0304 | 0.9931 ± 0.0449 |
| | AB | Number of estimators: 200 | 0.7152 ± 0.0155 | 0.9907 ± 0.0039 | 0.9924 ± 0.0229 | 0.9891 ± 0.0396 |
| | RF | Number of estimators: 200 | 0.7414 ± 0.0123 | 0.9863 ± 0.0035 | 0.9901 ± 0.0173 | 0.9824 ± 0.0293 |
| Region-based | SVM | C: 10, kernel: linear | 0.9589 ± 0.0095 | 0.9774 ± 0.0048 | 0.9704 ± 0.0267 | 0.9845 ± 0.0405 |
| | MLP | α: 0.01, nº of layers: 1, nº of neurons per layer: (100), solver: adam | 0.9652 ± 0.0056 | 0.9763 ± 0.0036 | 0.9745 ± 0.0242 | 0.9779 ± 0.0369 |
| | AB | Number of estimators: 50 or 200 | 0.9519 ± 0.0073 | 0.9719 ± 0.0048 | 0.9750 ± 0.0266 | 0.9689 ± 0.0546 |
| | RF | Number of estimators: 100 | 0.9598 ± 0.0073 | 0.9722 ± 0.0055 | 0.9735 ± 0.0264 | 0.9708 ± 0.0360 |

Source: The Author (2020).

Still, the performance of the proposed methodologies surpassed the benchmark models with the application of data standardization. Generally speaking, in terms of recall rate, which, as mentioned before, is very important in the context of infrastructure inspection, results were outstanding as the minimum was 96%. The same happened with the correct classification of negative observations. Because one of the aims of the proposed methodology is to reduce the workload during inspections, although false negatives classifications have the potential to impose risks not only to the system, but also to its users, false positives must be avoided as well.

## 4.1 FLEXIBILITY TEST

Although Section 4.1 presented results on a concrete crack dataset, it is essential to highlight that the proposed methodology can be applied to different materials, such as pavement, for instance, with varied crack sizes and characteristics. Note that if the image characteristics are too different from the ones in (ÖZGENEL, 2018), it will be necessary to retrain the ML model. Here, the models trained with the images from Ref. (ÖZGENEL, 2018) will be tested in another dataset, the SDNET (MAGUIRE; DORAFSHAN; THOMAS, 2018). The SDNET dataset contains about 56,000 images of concrete bridge decks, walls, and pavements from the Utah State University with cracks (from 0.06mm to 25mm) or without. Only a small portion of this dataset was used because it does not contain pixel-level annotation and generating the ground-truth is extremely time-consuming.

Here, both the texture-based and the region-based approaches were applied to 39 crack images and 19 without crack from the SDNET dataset. Conversely to what was presented before, the test results on the SDNET dataset were superior without data standardization and they are shown in Table 6. In general, the performance of all ML models in the texture-based approach did not meet expectations. In fact, it is possible to conclude from Table 6 that the SVM and MLP are biased to the “no crack” class in this case, whereas the AB and RF are biased to the “crack” class.

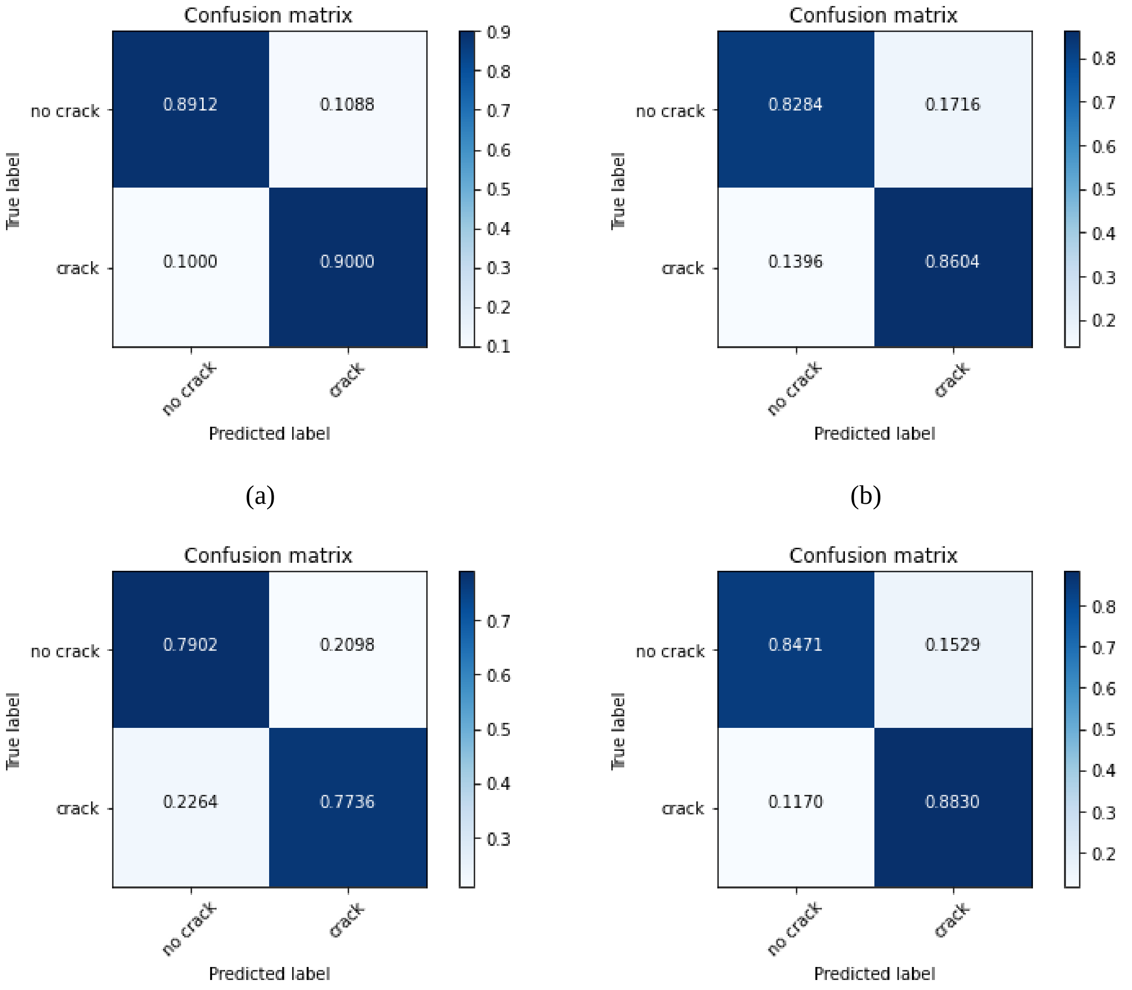
Table 6 – Classification performance of ML models on SDNET dataset

| Method | Classification | BA | R | S |
|---|---|---|---|---|
| Texture-based | SVM | 0.6403 ± 0.0106 | 0.4384 ± 0.0213 | 0.8421 ± 0.0000 |
| | MLP | 0.5528 ± 0.1176 | 0.3897 ± 0.2148 | 0.7158 ± 0.2504 |
| | AB | 0.5128 ± 0.0389 | 0.7307 ± 0.3344 | 0.2947 ± 0.3363 |
| | RF | 0.4845 ± 0.0490 | 0.8743 ± 0.2545 | 0.0947 ± 0.2022 |
| Region-based | SVM | 0.8956 ± 0.0068 | 0.9000 ± 0.0170 | 0.8912 ± 0.0155 |
| | MLP | 0.8444 ± 0.0404 | 0.8604 ± 0.0722 | 0.8284 ± 0.0517 |
| | AB | 0.7819 ± 0.0197 | 0.7736 ± 0.0462 | 0.7902 ± 0.0626 |
| | RF | 0.8650 ± 0.0111 | 0.8830 ± 0.0164 | 0.8471 ± 0.0220 |

Source: The Author (2020).

On the other hand, the results for the region-based approach were valid, with the SVM presenting the best performance. As mentioned before, the same image may generate more than one feature vector for the region-based approach because images may contain more than one region. Indeed, from the images gathered, the region-based approach extracted 53 positive and 102 negative feature vectors. A comparison between the performance of all the models is shown in Figure 9, proving SVM superiority in terms of R and S. This result also implies that the fact that the dataset selected for training presents a variety of crack characteristics made it possible to obtain good results in another dataset.

Figure 9 - Confusion matrix of SDNET classification with (a) SVM, (b) MLP, (c) AB, and (d) RF, for the region-based approach.



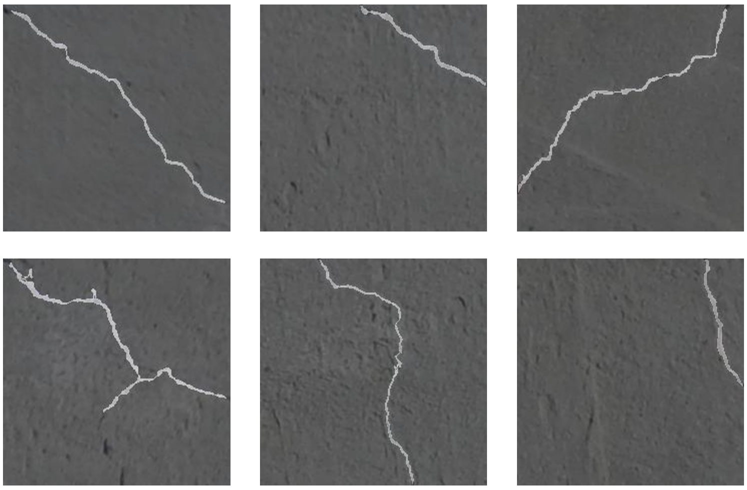
(a) (b)

(c) (d)

Source: The Author (2020).

Although the texture-based approach generally presented better results, this behavior was not repeated with the test in a different dataset. In this Section, the idea is to evaluate the flexibility of the model for localizing cracks, especially in this dataset that mostly contains thin cracks. Figure 10 shows the result of the proposed methodology correctly classifying images from SDNET, where the lighter parts correspond to the overlay of the real crack and the segmented one.

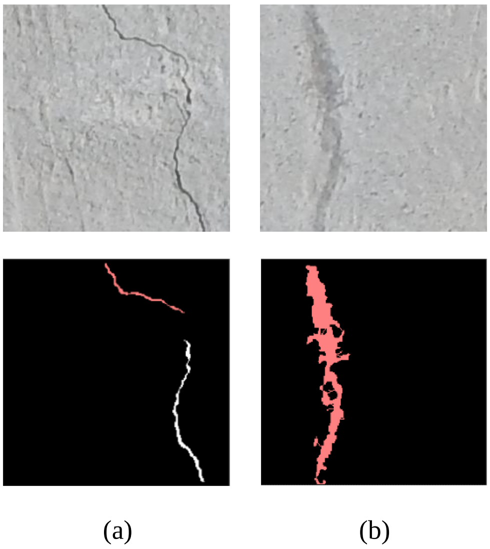
Figure 10 - Test on the SDNET dataset.



Source: The Author (2020).

Examples of misclassification are displayed in red for a positive and a negative case in Figure 11(a) and Figure 11(b), respectively. Note that in the first case, the segmentation generated a discontinuity in the crack, and then one part of the crack was not correctly classified (false negative). The second case exemplifies the same problem with the illumination and shadows discussed in Section 3.1, where a region was segmented because it resembles a crack, and it was indeed classified as a crack by the model, but in fact, it is not (false positive).

Figure 11 – Examples of misclassification (in red) from SDNET (a) positive sample and (b) negative sample.



(a) (b)

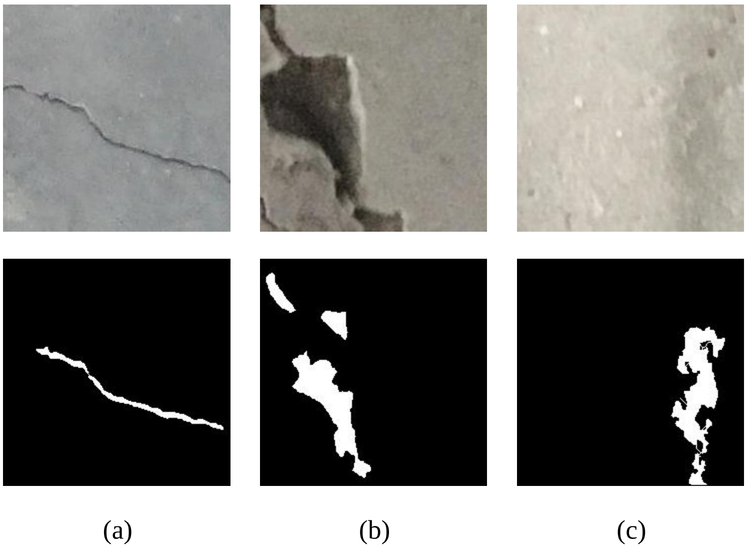
Source: The Author (2020).

## 4.2 LIMITATIONS

As mentioned in Section 2, one of the drawbacks of the general crack detection model based on image processing is the need for fine-tuning parameters according to the dataset characteristics and, hence, the loss of generality. In the proposed methodology, three parameters need some attention: they are the size of the SE, the minimum area (for removal of irrelevant objects), and the maximum distance between the regions for making a connection. While the first is related to the segmentation itself, both second and third ones are related to the post-processing stage, which means that they aim to improve the performance of the segmentation (as shown in Figure 5) and, as a consequence, the speed of classification.

Note that the selection of SE's size is based on the size of the cracks of interest. As the dataset comprises a variety of cracks, we addressed this issue by selecting a range of SE and combining their results. In this way, it is possible not only to detect various crack sizes but also joining information from different scales. The significant difficulty of the proposed methodology was related to the quality of the image, i.e., shadows, especially for cracks too thin (Figure 12.a) or too large (Figure 12.b), and changes in illumination (Figure 12.c), as reported by other authors (LIANG, Dong *et al.*, 2019; LIANG, Sun; JIANCHUN; XUN, 2018; MOHAN; POOBAL, 2018; ZHANG, Dejin *et al.*, 2017b). Other drawbacks highlighted by some papers were successfully addressed by the region-based approach, such as detecting fine cracks (HOANG, 2018b), robustness to various positions (HEO; JEON; SON, 2019; LEI *et al.*, 2018) and the number of cracks (LEI *et al.*, 2018; LIANG, Sun; JIANCHUN; XUN, 2018), since the segmentation step considers different orientations, locations, and crack ramifications (LIANG, Sun; JIANCHUN; XUN, 2018). The reason behind it is that the segmentation step provides flexibility by using several SE sizes and orientations.

Figure 12 - Original images and the results of the segmentation methodology on challenging contexts, such as (a) shadow within thin cracks, (b) shadow caused by the crack depth, and (c) changes in illumination.



(a) (b) (c)

Source: The Author (2020).

# 5 CONCLUDING REMARKS

In order to create an automatic crack detection methodology, two different approaches were analyzed in this dissertation. The first one was based on segmentation techniques (region-based), while the second one relies on texture analysis (texture-based) for the extraction of features from images. Given that images are annotated, ML models (i.e., SVM, MLP, AB, and RF) were trained using the feature vector of each approach to assess the existence of cracks and provide an automatic classification method. The hyper-parameters of the models were tuned, and a real concrete crack image dataset was used to evaluate performances.

As a result, although all models reached a good performance in terms of balanced accuracy, recall, and specificity, models with data standardization and the texture-based feature extraction approach presented better results, reaching a balanced accuracy of 99%. In fact, the implementation of data standardization was essential to improve MLP results, as its accuracy went from 77% to 99% in the texture-based approach and from 96% to 97% in the region-based.

Furthermore, a test was also conducted on a small set of images from another dataset (SDNET), which was different from the one used to train the models. In this case, the best result was obtained with SVM in the region-based approach with a balanced accuracy of 89%, showing the ability of generalization of the model. However, it is important to mention the limitation of the segmentation step concerning changes in lighting conditions and to the presence of shadows in the images, which may lead to misclassifications by the ML models.

Note that, although we are analyzing the approaches mentioned above separately, for comparison purposes, our ongoing research focuses on combining them. Combining both approaches could be interesting when dealing with a large area to be covered by the inspection, in which case the area could be divided into several parts. For each part, one could use the texture-based approach to detect whether there is a crack or not and, then, for each positive result, use the region-based analysis to narrow down the precise location of the crack(s).

As future work, it is expected we improve the image detection methodology to classify not only cracks but also other types of fault (e.g., corrosion, deformation, wear). Some works have been dealing with this problem, such as (HOANG; NGUYEN, 2019; OLIVEIRA; CORREIA, 2014; QINGGUO *et al.*, 2019b), but the biggest issue is the lack of a comprehensive and fully annotated dataset, which can be overcome with data augmentation. Also, it is important to address the issues with shadows and changes in illumination within images. Recently, researchers have been focusing on DL methods to deal with these problems, especially CNN such as (DUNG; ANH, 2019; LIANG, Dong *et al.*, 2019; MAEDA *et al.*, 2018; PANELLA *et al.*, 2018; STANIEK; CZECH, 2018), and they are obtaining good results, which proves its worth. In fact, Ref. (DORAFSHAN; THOMAS; MAGUIRE, 2018b) compares the adoption of CNN methods against an approach based on edge detection, showing that not only did the former outperformed the latter, but also the combination of both approaches showed significant promise. Hence, in possession of a considerable number of images, DL approaches, such as CNN, can be applied to this context and compared with the results presented here in terms of performance metrics and required computational power.

# REFERENCES

ABUOBAID, A.; HEIDER, D.; YARLAGADDA, S. A time-domain reflectometry method for automated measurement of crack propagation in composites during mode I DCB testing under cold, hot, and hot/wet conditions. **Journal of Thermoplastic Composite Materials**, vol. 32, no. 4, p. 558–573, 10 Apr. 2019. DOI 10.1177/0892705718772873. Available at: http://journals.sagepub.com/doi/10.1177/0892705718772873. Accessed on: 1 Aug. 2019.

ADHIKARI, R. S.; MOSELHI, O.; BAGCHI, A. Automation in Construction Image-based retrieval of concrete crack properties for bridge inspection. **Automation in Construction**, vol. 39, p. 180–194, 2014a. DOI 10.1016/j.autcon.2013.06.011. Available at: http://dx.doi.org/10.1016/j.autcon.2013.06.011. Accessed on: 1 Aug. 2019.

ADHIKARI, R. S.; MOSELHI, O.; BAGCHI, A. Image-based retrieval of concrete crack properties for bridge inspection. **Automation in Construction**, vol. 39, p. 180–194, Apr. 2014b. DOI 10.1016/j.autcon.2013.06.011. Available at: http://dx.doi.org/10.1016/j.autcon.2013.06.011. Accessed on: 1 Aug. 2019.

ALAM, S. Y.; LOUKILI, A.; GRONDIN, F.; ROZIÈRE, E. Use of the digital image correlation and acoustic emission technique to study the effect of structural size on cracking of reinforced concrete. **Engineering Fracture Mechanics**, vol. 143, p. 17–31, Jul. 2015. DOI 10.1016/j.engfracmech.2015.06.038. Available at: http://dx.doi.org/10.1016/j.engfracmech.2015.06.038. Accessed on: 1 Aug. 2019.

ALMEIDA, V. Novo vídeo mostra prédio desabar sobre pessoas em Fortaleza. 2019. **G1**. Available at: https://g1.globo.com/ce/ceara/noticia/2019/10/16/video-mostra-pessoas-atingidas-por-desabamento-de-predio-em-fortaleza.ghtml. Accessed on: 31 Oct. 2019.

AMHAZ, R.; CHAMBON, S.; IDIER, J.; BALTAZART, V. Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection. **IEEE Transactions on Intelligent Transportation Systems**, vol. 17, no. 10, p. 2718–2729, Oct. 2016. DOI 10.1109/TITS.2015.2477675. Available at: http://ieeexplore.ieee.org/document/7572082/. Accessed on: 1 Aug. 2019.

ASTORGA, N.; DROGUETT, E. L.; MERUANE, V. Computer vision for structural damage quantification: A novel residual deep learning based approach. **Safety and Reliability – Safe Societies in a Changing World**. Trondheim, Norway: CRC Press, 2018. p. 1057–1063. DOI 10.1201/9781351174664-133. Available at: https://www.taylorfrancis.com/books/9781351174657/chapters/10.1201/9781351174664-133. Accessed on: 1 Aug. 2019.

ATTARD, L.; DEBONO, C. J.; VALENTINO, G.; DI CASTRO, M. Vision-based change detection for inspection of tunnel liners. **Automation in Construction**, vol. 91, no. February, p. 142–154, Jul. 2018. DOI 10.1016/j.autcon.2018.03.020. Available at: https://doi.org/10.1016/j.autcon.2018.03.020. Accessed on: 1 Aug. 2019.

BAYAR, G.; BILIR, T. A novel study for the estimation of crack propagation in concrete using machine learning algorithms. **Construction and Building Materials**, vol. 215, p. 670–685, Aug. 2019. DOI 10.1016/j.conbuildmat.2019.04.227. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950061819311110. Accessed on: 1 Aug. 2019.

BREIMAN, L. Random Forests. **Machine Learning**, vol. 45, p. 5–32, 8 Jun. 2001.

CAWLEY, G. C.; TALBOT, N. L. C. On over-fitting in model selection and subsequent selection bias in performance evaluation. **Journal of Machine Learning Research**, vol. 11, p. 2079–2107, 2010.

CHA, Y.; CHOI, W.; SUH, G.; MAHMOUDKHANI, S.; BÜYÜKÖZTÜRK, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. **Computer-Aided Civil and Infrastructure Engineering**, vol. 33, no. 9, p. 731–747, Sep. 2018. DOI 10.1111/mice.12334. Available at: http://doi.wiley.com/10.1111/mice.12334. Accessed on: 1 Aug. 2019.

CHAMBON, S.; MOLIARD, J. Automatic Road Pavement Assessment with Image Processing: Review and Comparison. **International Journal of Geophysics**, vol. 2011, p. 1–20, 2011. DOI 10.1155/2011/989354. Available at: http://www.hindawi.com/journals/ijge/2011/989354/. Accessed on: 1 Aug. 2019.

CHAWLA, R.; SINGAL, P.; GARG, A. K. A Mamdani Fuzzy Logic System to Enhance Solar Cell Micro-Cracks Image Processing. **3D Research**, vol. 9, no. 3, p. 34, 27 Sep. 2018. DOI 10.1007/s13319-018-0186-7. Available at: https://doi.org/10.1007/s13319-018-0186-7. Accessed on: 1 Aug. 2019.

CHEN, H.; PANG, Y.; HU, Q.; LIU, K. Solar cell surface defect inspection based on multispectral convolutional neural network. **Journal of Intelligent Manufacturing**, 14 Dec. 2018. DOI 10.1007/s10845-018-1458-z. Available at: https://doi.org/10.1007/s10845-018-1458-z. Accessed on: 1 Aug. 2019.

CHO, S.; PARK, S.; CHA, G.; OH, T. Development of Image Processing for Crack Detection on Concrete Structures through Terrestrial Laser Scanning Associated with the Octree Structure. **Applied Sciences**, vol. 8, no. 12, p. 2373, 23 Nov. 2018b. DOI 10.3390/app8122373. Available at: http://www.mdpi.com/2076-3417/8/12/2373. Accessed on: 1 Aug. 2019.

CHOTZOGLOU, A.; PISSAS, M.; ZERVAKI, A. D.; HAIDEMENOPOULOS, G. N.; PISSAS, T. Visualization of the Rolling Contact Fatigue Cracks in Rail Tracks with a Magnetooptical Sensor. **Journal of Nondestructive Evaluation**, vol. 38, no. 3, p. 68, 16 Sep. 2019. DOI 10.1007/s10921-019-0606-5. Available at: https://doi.org/10.1007/s10921-019-0606-5. Accessed on: 1 Aug. 2019.

CHOUDHURY, S.; THATOI, D. N.; MAITY, K.; SAU, S.; RAO, M. D. A Modified Support Vector Regression Approach for Failure Analysis in Beam-Like Structures. **Journal of Failure Analysis and Prevention**, vol. 18, no. 4, p. 998–1009, 12 Aug. 2018. DOI 10.1007/s11668-018-0494-5. Available at: https://doi.org/10.1007/s11668-018-0494-5. Accessed on: 1 Aug. 2019.

CLAESEN, M.; DE MOOR, B. Hyperparameter Search in Machine Learning. 2015. *In*: **MIC 2015: The XI Metaheuristics International Conference** [...]. Agadir: [*s. n.*], 2015. p. 10–14. Available at: http://arxiv.org/abs/1502.02127. Accessed on: 1 Aug. 2019.

DAVOUDI, R.; MILLER, G. R.; KUTZ, J. N. Data-driven vision-based inspection for reinforced concrete beams and slabs: Quantitative damage and load estimation. **Automation in Construction**, vol. 96, no. October, p. 292–309, Dec. 2018a. DOI 10.1016/j.autcon.2018.09.024. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0926580518303686. Accessed on: 1 Aug. 2019.

DAVOUDI, R.; MILLER, G. R.; KUTZ, J. N. Structural Load Estimation Using Machine Vision and Surface Crack Patterns for Shear-Critical RC Beams and Slabs. **Journal of Computing in Civil Engineering**, vol. 32, no. 4, p. 04018024, Jul. 2018b. DOI 10.1061/(ASCE)CP.1943-5487.0000766. Available at: http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000766. Accessed on: 1 Aug. 2019.

DORAFSHAN, S.; MAGUIRE, M.; HOFFER, N. V.; COOPMANS, C. Challenges in Bridge Inspection Using Small Unmanned Aerial Systems : Results and Lessons Learned. 2017. *In*: **2017 International Conference on Unmanned Aircraft Systems (ICUAS)** [...]. Miami: [*s. n.*], 2017. p. 1722–1730.

DORAFSHAN, S.; THOMAS, R. J.; COOPMANS, C.; MAGUIRE, M. Deep Learning Neural Networks for sUAS-Assisted Structural Inspections: Feasibility and Application. Jun. 2018. *In*: **2018 International Conference on Unmanned Aircraft Systems (ICUAS)** [...]. [*S. l.*]: IEEE, Jun. 2018. p. 874–882. DOI 10.1109/ICUAS.2018.8453409. Available at: https://ieeexplore.ieee.org/document/8453409/. Accessed on: 1 Aug. 2019.

DORAFSHAN, S.; THOMAS, R. J.; MAGUIRE, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. **Construction and Building Materials**, vol. 186, p. 1031–1045, Oct. 2018a. DOI 10.1016/j.conbuildmat.2018.08.011. Available at: https://doi.org/10.1016/j.conbuildmat.2018.08.011. Accessed on: 1 Aug. 2019.

DORAFSHAN; THOMAS; MAGUIRE. Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures. **Infrastructures**, vol. 4, no. 2, p. 19, 30 Apr. 2019. DOI 10.3390/infrastructures4020019. Available at: https://www.mdpi.com/2412-3811/4/2/19. Accessed on: 1 Aug. 2019.

DUNG, C. V.; ANH, L. D. Autonomous concrete crack detection using deep fully convolutional neural network. **Automation in Construction**, vol. 99, no. December 2018, p. 52–58, Mar. 2019. DOI 10.1016/j.autcon.2018.11.028. Available at: https://doi.org/10.1016/j.autcon.2018.11.028. Accessed on: 1 Aug. 2019.

DUNG, C. V.; SEKIYA, H.; HIRANO, S.; OKATANI, T.; MIKI, C. A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. **Automation in Construction**, vol. 102, no. March, p. 217–229, Jun. 2019. DOI 10.1016/j.autcon.2019.02.013. Available at: https://doi.org/10.1016/j.autcon.2019.02.013. Accessed on: 1 Aug. 2019.

EISENBACH, M.; STRICKER, R.; SEICHTER, D.; AMENDE, K.; DEBES, K.; SESSELMANN, M.; EBERSBACH, D.; STOECKERT, U.; GROSS, H. How to get pavement distress detection ready for deep learning? A systematic approach. May 2017. *In*: **2017 International Joint Conference on Neural Networks (IJCNN)** [...]. [*S. l.*]: IEEE, May 2017. p. 2039–2047. DOI 10.1109/IJCNN.2017.7966101. Available at: https://www.tuilmenau.de/fileadmin/media/neurob/publications/conferences_int/2017/Eisenbach-IJCNN-2017.pdf. Accessed on: 1 Aug. 2019.

FAN, R.; BOCUS, M. J.; ZHU, Y.; JIAO, J.; WANG, L.; MA, F.; CHENG, S.; LIU, M. Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding. 17 Apr. 2019. Available at: http://arxiv.org/abs/1904.08582. Accessed on: 1 Aug. 2019.

FAN, Z.; WU, Y.; LU, J.; LI, W. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network. , p. 1–9, 1 Feb. 2018. Available at: http://arxiv.org/abs/1802.02208. Accessed on: 1 Aug. 2019.

FARDO, F. A.; DONATO, G. H. B.; RODRIGUES, P. S. Texture Analysis for Crack Detection in Fracture Mechanics. **Journal of Failure Analysis and Prevention**, vol. 18, no. 3, p. 526–537, 28 Jun. 2018. DOI 10.1007/s11668-018-0432-6. Available at: https://doi.org/10.1007/s11668-018-0432-6. Accessed on: 1 Aug. 2019.

FENG, D.; FENG, M. Q. Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection – A review. **Engineering Structures**, vol. 156, no. November 2017, p. 105–117, Feb. 2018. DOI 10.1016/j.engstruct.2017.11.018. Available at: https://doi.org/10.1016/j.engstruct.2017.11.018. Accessed on: 1 Aug. 2019.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, vol. 55, p. 119–139, 1997.

FUJITA, Y.; HAMAMOTO, Y. A robust automatic crack detection method from noisy concrete surfaces. **Machine Vision and Applications**, vol. 22, no. 2, p. 245–254, 2 Mar. 2011. DOI 10.1007/s00138-009-0244-5. Available at: http://link.springer.com/10.1007/s00138-009-0244-5. Accessed on: 1 Aug. 2019.

GLOROT, X.; BENGIO, Y. Initializing weights to a hidden layer of a multilayer neural network by linear programming. *In*: **the Thirteenth International Conference on Artificial Intelligence and Statistics** [...]. [*S. l.*: *s. n.*], 2010. vol. 9, p. 249–256.

GLUD, J. A.; DULIEU-BARTON, J. M.; THOMSEN, O. T.; OVERGAARD, L. C. T. Automated counting of off-axis tunnelling cracks using digital image processing. **Composites Science and Technology**, vol. 125, p. 80–89, Mar. 2016. DOI 10.1016/j.compscitech.2016.01.019. Available at: http://dx.doi.org/10.1016/j.compscitech.2016.01.019. Accessed on: 1 Aug. 2019.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. Third. Upper Saddle River, NJ: Prentice-Hall, Inc., 2006.

HAMEL, L. **Knowledge Discovery with Support Vector Machines**. Hoboken, New Jersey: [*s. n.*], 2009.

HAN, Y.; LIU, Z.; LYU, Y.; LIU, K.; LI, C.; ZHANG, W. Deep learning-based visual ensemble method for high-speed railway catenary clevis fracture detection. **Neurocomputing**, Apr. 2019. DOI 10.1016/j.neucom.2018.10.107. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0925231219304916. Accessed on: 1 Aug. 2019.

HARALICK, R. M; SHANMUGAM, K.; et al. Textural features for image classification. **IEEE Transactions on systems, man, and cybernetics**, vol. 3, no. 6, 1973.

HEO, G.; JEON, J.; SON, B. Crack Automatic Detection of CCTV Video of Sewer Inspection with Low Resolution. **KSCE Journal of Civil Engineering**, vol. 23, no. 3, p. 1219–1227, 14 Mar. 2019. DOI 10.1007/s12205-019-0980-7. Available at: http://link.springer.com/10.1007/s12205-019-0980-7. Accessed on: 1 Aug. 2019.

HOANG, N.-D. Classification of Asphalt Pavement Cracks Using Laplacian Pyramid-Based Image Processing and a Hybrid Computational Approach. **Computational Intelligence and Neuroscience**, vol. 2018, p. 1–16, 1 Oct. 2018a. DOI 10.1155/2018/1312787. Available at: https://www.hindawi.com/journals/cin/2018/1312787/. Accessed on: 1 Aug. 2019.

HOANG, N.-D. Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding. **Advances in Civil Engineering**, vol. 2018, p. 1–10, 2018b. DOI 10.1155/2018/3924120. Available at: https://www.hindawi.com/journals/ace/2018/3924120/. Accessed on: 1 Aug. 2019.

HOANG, N.-D.; NGUYEN, Q.-L. A novel method for asphalt pavement crack classification based on image processing and machine learning. **Engineering with Computers**, vol. 35, no. 2, p. 487–498, 18 Apr. 2019. DOI 10.1007/s00366-018-0611-9. Available at: http://dx.doi.org/10.1007/s00366-018-0611-9. Accessed on: 1 Aug. 2019.

HOANG, N.-D.; NGUYEN, Q.-L. Fast Local Laplacian-Based Steerable and Sobel Filters Integrated with Adaptive Boosting Classification Tree for Automatic Recognition of Asphalt Pavement Cracks. **Advances in Civil Engineering**, vol. 2018, p. 1–17, 24 Sep. 2018. DOI 10.1155/2018/5989246. Available at: https://www.hindawi.com/journals/ace/2018/5989246/. Accessed on: 1 Aug. 2019.

HU, Y.; ZHAO, C.; WANG, H. Automatic Pavement Crack Detection Using Texture and Shape Descriptors. **IETE Technical Review**, vol. 27, no. 5, p. 398, 2010. DOI 10.4103/0256-4602.62225. Available at: http://tr.ietejournals.org/text.asp?2010/27/5/398/62225. Accessed on: 1 Aug. 2019.

HWANG, H.; OH, J.; LEE, K.-H.; CHA, J.-H.; CHOI, E.; YOON, Y.; HWANG, J.-H. Synergistic approach to quantifying information on a crack-based network in loess/water material composites using deep learning and network science. **Computational Materials Science**, vol. 166, no. May, p. 240–250, Aug. 2019. DOI 10.1016/j.commatsci.2019.04.014. Available at: https://doi.org/10.1016/j.commatsci.2019.04.014. Accessed on: 1 Aug. 2019.

INZERILLO, L.; DI MINO, G.; ROBERTS, R. Image-based 3D reconstruction using traditional and UAV datasets for analysis of road pavement distress. **Automation in Construction**, vol. 96, no. October, p. 457–469, Dec. 2018. DOI 10.1016/j.autcon.2018.10.010. Available at: https://doi.org/10.1016/j.autcon.2018.10.010. Accessed on: 1 Aug. 2019.

IYER, S.; SINHA, S. K. A robust approach for automatic detection and segmentation of cracks in underground pipeline images. **Image and Vision Computing**, v. 23, n. 10, p. 921–933, Sep. 2005. DOI 10.1016/j.imavis.2005.05.017. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0262885605000764. Accessed on: 1 Aug. 2019.

JAHANSHAHI, M. R.; MASRI, S. F.; PADGETT, C. W.; SUKHATME, G. S. An innovative methodology for detection and quantification of cracks through incorporation of depth perception. **Machine Vision and Applications**, v. 24, n. 2, p. 227–241, 27 Feb. 2013. DOI 10.1007/s00138-011-0394-0. Available at: http://link.springer.com/10.1007/s00138-011-0394-0. Accessed on: 1 Aug. 2019.

KALYANI, S.; SWARUP, K. S. Pattern analysis and classification for security evaluation in power networks. **International Journal of Electrical Power & Energy Systems**, v. 44, n. 1, p. 547–560, Jan. 2013. DOI 10.1016/j.ijepes.2012.07.065. Available at: http://dx.doi.org/10.1016/j.ijepes.2012.07.065. Accessed on: 1 Aug. 2019.

KIM, B.; CHO, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. **Sensors**, v. 18, n. 10, p. 3452, 14 Oct. 2018. DOI 10.3390/s18103452. Available at: http://www.mdpi.com/1424-8220/18/10/3452. Accessed on: 1 Aug. 2019.

KOCH, C.; GEORGIEVA, K.; KASIREDDY, V.; AKINCI, B.; FIEGUTH, P. Advanced Engineering Informatics A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure q. **Advanced Engineering Informatics**, v. 29, p. 196–210, 2015. DOI 10.1016/j.aei.2015.01.008. Available at: http://dx.doi.org/10.1016/j.aei.2015.01.008. Accessed on: 1 Aug. 2019.

KOHAVI, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. 1995. *In*: **IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence** [...]. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995. p. 1137–1143.

KONG, X.; LI, J. Vision-Based Fatigue Crack Detection of Steel Structures Using Video Feature Tracking. **Computer-Aided Civil and Infrastructure Engineering**, v. 33, n. 9, p. 783–799, Sep. 2018. DOI 10.1111/mice.12353. Available at: http://doi.wiley.com/10.1111/mice.12353. Accessed on: 1 Aug. 2019.

KOTSIANTIS, S. B. Supervised Machine Learning: A Review of Classification Techniques. **Informatica**, v. 31, p. 249–268, 2007.

KOTSIANTIS, S. B.; ZAHARAKIS, I. D.; PINTELAS, P. E. Machine learning: a review of classification and combining techniques. **Artificial Intelligence Review**, v. 26, n. 3, p. 159–190, 10 Nov. 2006. DOI 10.1007/s10462-007-9052-3. Available at: http://link.springer.com/10.1007/s10462-007-9052-3. Accessed on: 1 Aug. 2019.

KOVESI, P. MATLAB and Octave Functions for Computer Vision and Image Processing. 2000.

LEE, F. W.; CHAI, H. K.; LIM, K. S.; LAU, S. H. Concrete Sub-Surface Crack Characterization by Means of Surface Rayleigh Wave Method. **ACI Materials Journal**, v. 116, n. 1, p. 113–123, Jan. 2019. DOI 10.14359/51710967. Available at: http://www.concrete.org/Publications/InternationalConcreteAbstractsPortal.aspx?m=details&i=51710967. Accessed on: 1 Aug. 2019.

LEI, B.; WANG, N.; XU, P.; SONG, G. New Crack Detection Method for Bridge Inspection

Using UAV Incorporating Image Processing. **Journal of Aerospace Engineering**, v. 31, n. 5, p. 04018058, Sep. 2018. DOI 10.1061/(ASCE)AS.1943-5525.0000879. Available at: http://ascelibrary.org/doi/10.1061/%28ASCE%29AS.1943-5525.0000879. Accessed on: 1 Aug. 2019.

LI, X.; JIANG, H.; YIN, G. Detection of surface crack defects on ferrite magnetic tile. **NDT & E International**, v. 62, p. 6–13, Mar. 2014. DOI 10.1016/j.ndteint.2013.10.006. Available at: http://dx.doi.org/10.1016/j.ndteint.2013.10.006. Accessed on: 1 Aug. 2019.

LI, Y.; ZHAO, W.; ZHANG, X.; ZHOU, Q. A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks. **IEICE Transactions on Information and Systems**, vol. E101.D, n. 12, p. 3249–3252, 1 Dec. 2018. DOI 10.1587/transinf.2018EDL8150. Available at: https://www.jstage.jst.go.jp/article/transinf/E101.D/12/E101.D_2018EDL8150/_article. Accessed on: 1 Aug. 2019.

LIANG, D.; ZHOU, X.-F.; WANG, S.; LIU, C.-J. Research on Concrete Cracks Recognition based on Dual Convolutional Neural Network. **KSCE Journal of Civil Engineering**, vol. 23, no. 7, p. 3066–3074, 10 Jul. 2019. DOI 10.1007/s12205-019-2030-x. Available at: http://link.springer.com/10.1007/s12205-019-2030-x. Accessed on: 1 Aug. 2019.

LIANG, S.; JIANCHUN, X.; XUN, Z. An Algorithm for Concrete Crack Extraction and Identification Based on Machine Vision. **IEEE Access**, v. 6, p. 28993–29002, 2018. DOI 10.1109/ACCESS.2018.2844100. Available at: https://ieeexplore.ieee.org/document/8373690/. Accessed on: 1 Aug. 2019.

LINS, I. D.; ARAUJO, M.; MOURA, M. das C.; SILVA, M. A.; DROGUETT, E. L. Prediction of sea surface temperature in the tropical Atlantic by support vector machines. **Computational Statistics & Data Analysis**, v. 61, p. 187–198, May 2013. DOI 10.1016/j.csda.2012.12.003. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0167947312004240. Accessed on: 1 Aug. 2019.

LIU, L.; ZHAO, L.; LONG, Y.; KUANG, G.; FIEGUTH, P. Extended local binary patterns for texture classification. **Image and Vision Computing**, v. 30, n. 2, p. 86–99, Feb. 2012b. DOI 10.1016/j.imavis.2012.01.001. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0262885612000066. Accessed on: 1 Aug. 2019.

LIU, R.; YANG, B.; ZIO, E.; CHEN, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. **Mechanical Systems and Signal Processing**, v. 108, p. 33–47, 2018. DOI 10.1016/j.ymssp.2018.02.016. Available at: https://doi.org/10.1016/j.ymssp.2018.02.016. Accessed on: 1 Aug. 2019.

LIU, W.; HUANG, Y.; LI, Y.; CHEN, Q. FPCNet: Fast Pavement Crack Detection Network Based on Encoder-Decoder Architecture. , p. 1–11, 4 Jul. 2019. Available at: http://arxiv.org/abs/1907.02248. Accessed on: 1 Aug. 2019.

LIU, Z.; CAO, Y.; WANG, Y.; WANG, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. **Automation in Construction**, v. 104, n. April, p. 129–139, Aug. 2019. DOI 10.1016/j.autcon.2019.04.005. Available at: https://doi.org/10.1016/j.autcon.2019.04.005. Accessed on: 1 Aug. 2019.

LUO, Q.; GE, B.; TIAN, Q. A fast adaptive crack detection algorithm based on a double-edge extraction operator of FSM. **Construction and Building Materials**, v. 204, p. 244–254, Apr. 2019. DOI 10.1016/j.conbuildmat.2019.01.150. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950061819301849. Accessed on: 1 Aug. 2019.

MAEDA, H.; SEKIMOTO, Y.; SETO, T.; KASHIYAMA, T.; OMATA, H. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. **Computer-Aided Civil and Infrastructure Engineering**, v. 33, n. 12, p. 1127–1141, 29 Dec. 2018. DOI 10.1111/mice.12387. Available at: http://arxiv.org/abs/1801.09454. Accessed on: 1 Aug. 2019.

MAGUIRE, M.; DORAFSHAN, S.; THOMAS, R. J. SDNET2018: A concrete crack image dataset for machine learning applications. 2018. DOI https://doi.org/10.15142/T3TD19. Available at: https://digitalcommons.usu.edu/all_datasets/48. Accessed on: 1 Aug. 2019.

MEI, Q.; GÜL, M. A Conditional Wasserstein Generative Adversarial Network for Pixel-level Crack Detection using Video Extracted Images. 13 Jul. 2019. Available at: http://arxiv.org/abs/1907.06014. Accessed on: 1 Aug. 2019.

MNEYMNEH, B. E.; ABBAS, M.; KHOURY, H. Evaluation of computer vision techniques for automated hardhat detection in indoor construction safety applications. **Frontiers of Engineering Management**, v. 5, n. 2, p. 227–239, 2018. DOI 10.15302/J-FEM-2018071. Available at: https://doi.org/https://doi.org/10.15302/J-FEM-2018071. Accessed on: 1 Aug. 2019.

MODARRES, C.; ASTORGA, N.; DROGUETT, E. L.; MERUANE, V. Convolutional neural networks for automated damage recognition and damage type identification. **Structural Control and Health Monitoring**, v. 25, n. 10, p. e2230, Oct. 2018. DOI 10.1002/stc.2230. Available at: http://doi.wiley.com/10.1002/stc.2230. Accessed on: 1 Aug. 2019.

MOHAN, A.; POOBAL, S. Crack detection using image processing: A critical review and analysis. **Alexandria Engineering Journal**, v. 57, n. 2, p. 787–798, Jun. 2018. DOI 10.1016/j.aej.2017.01.020. Available at: https://doi.org/10.1016/j.aej.2017.01.020. Accessed on: 1 Aug. 2019.

NGUYEN, H.-N.; KAM, T.-Y.; CHENG, P.-Y. An Automatic Approach for Accurate Edge Detection of Concrete Crack Utilizing 2D Geometric Features of Crack. **Journal of Signal Processing Systems**, v. 77, n. 3, p. 221–240, 14 Dec. 2014. DOI 10.1007/s11265-013-0813-8. Available at: http://link.springer.com/10.1007/s11265-013-0813-8. Accessed on: 1 Aug. 2019.

NHAT-DUC, H.; NGUYEN, Q.-L.; TRAN, V.-D. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. **Automation in Construction**, v. 94, no. January, p. 203–213, Oct. 2018. DOI 10.1016/j.autcon.2018.07.008. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0926580518300499. Accessed on: 1 Aug. 2019.

NIXON, M.; AGUADO, A. **Feature Extraction and Image Processing**. Second. Oxford: Academic Press, 2008.

NTSB BLAMES FATAL MIAMI BRIDGE FALL ON DESIGN, LACK OF OVERSIGHT. 2019. **NBC News**. Available at: https://www.nbcnews.com/news/us-news/ntsb-blames-fatal-miami-bridge-fall-design-lack-oversight-n1070226. Accessed on: 27 Nov. 2019.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on feature distributions. **Pattern Recognition**, v. 29, n. 1, p. 51–59, 1996. .

OLIVEIRA, H.; CORREIA, P. L. CrackIT - An image processing toolbox for crack detection and characterization. Oct. 2014. *In*: **2014 IEEE International Conference on Image Processing (ICIP)** [...]. [*S. l.*]: IEEE, Oct. 2014. p. 798–802. DOI 10.1109/ICIP.2014.7025160. Available at: http://ieeexplore.ieee.org/document/7025160/. Accessed on: 1 Aug. 2019.

OLIVEIRA SANTOS, B.; VALENÇA, J.; JÚLIO, E. Automatic mapping of cracking patterns on concrete surfaces with biological stains using hyper-spectral images processing. **Structural Control and Health Monitoring**, v. 26, n. 3, p. e2320, Mar. 2019. DOI 10.1002/stc.2320. Available at: http://doi.wiley.com/10.1002/stc.2320. Accessed on: 1 Aug. 2019.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Trans. Syst. Man. Cybern**, v. 9, n. 1, p. 62–66, 1979. .

ÖZGENEL, Ç. F. Concrete Crack Images for Classification. 2018. Available at: https://data.mendeley.com/datasets/5y9wdsg2zt/1. Accessed on: 1 Nov. 2018.

PAGLINAWAN, A. C.; CRUZ, F. R. G.; CASI, N. D.; INGATAN, P. A. B.; KARGANILLA, A. B. C.; MOSTER, G. V. G. Crack Detection Using Multiple Image Processing for Unmanned Aerial Monitoring of Concrete Structure. 2018-Octob., Oct. 2018. *In*: **TENCON 2018 - 2018 IEEE Region 10 Conference** [...]. [*S. l.*]: IEEE, Oct. 2018. vol. 2018-Octob, p. 2534–2538. DOI 10.1109/TENCON.2018.8650313. Available at: https://ieeexplore.ieee.org/document/8650313/. Accessed on: 1 Aug. 2019.

PANDEY, P. C.; BARAI, S. V. Multilayer perceptron in damage detection of bridge structures. **Computers & Structures**, v. 54, n. 4, p. 597–608, Feb. 1995. DOI 10.1016/0045-7949(94)00377-F. Available at: https://linkinghub.elsevier.com/retrieve/pii/004579499400377F. Accessed on: 1 Aug. 2019.

PANELLA, F.; BOEHM, J.; LOO, Y.; KAUSHIK, A.; GONZALEZ, D. Deep Learning and Image Processing for Automated Crack Detection and Defect Measurement in Underground Structures. XLII–2., 30 May 2018. *In*: **ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences** [...]. [*S. l.*: *s. n.*], 30 May 2018. vol. XLII–2, p. 829–835. DOI 10.5194/isprs-archives-XLII-2-829-2018. Available at: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/829/2018/. Accessed on: 1 Aug. 2019.

PARKER, J. R. Algorithms for image prcessing and computer vision. Second. Indianapolis, Indiana: Wiley Publishing, Inc., 2011.

PAYAB, M.; ABBASINA, R.; KHANZADI, M. A Brief Review and a New Graph-Based

Image Analysis for Concrete Crack Quantification. **Archives of Computational Methods in Engineering**, v. 26, n. 2, p. 347–365, 20 Apr. 2019. DOI 10.1007/s11831-018-9263-6. Available at: https://doi.org/10.1007/s11831-018-9263-6. Accessed on: 1 Aug. 2019.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIPINATO, A. **Innovative Bridge Design Handbook: Construction, Rehabilitation and Maintenance**. Oxford: Butterworth Heinemann, 2016.

PIRI, S.; DELEN, D.; LIU, T. A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. **Decision Support Systems**, v. 106, p. 15–29, Feb. 2018. DOI 10.1016/j.dss.2017.11.006. Available at: https://doi.org/10.1016/j.dss.2017.11.006. Accessed on: 1 Aug. 2019.

POREBSKI, A.; VANDENBROUCKE, N.; MACAIRE, L. Haralick feature extraction from LBP images for color texture classification. Nov. 2008. *In:* **2008 First Workshops on Image Processing Theory, Tools and Applications** [...]. [*S. l.*]: IEEE, Nov. 2008. p. 1–8. DOI 10.1109/IPTA.2008.4743780. Available at: http://ieeexplore.ieee.org/document/4743780/. Accessed on: 1 Aug. 2019.

PRASANNA, P.; DANA, K. J.; GUCUNSKI, N.; BASILY, B. B.; LA, H. M.; LIM, R. S.; PARVARDEH, H. Automated Crack Detection on Concrete Bridges. **IEEE Transactions on Automation Science and Engineering**, v. 13, n. 2, p. 591–599, Apr. 2016. DOI 10.1109/TASE.2014.2354314. Available at: http://ieeexplore.ieee.org/document/6917066/. Accessed on: 1 Aug. 2019.

PROTOPAPADAKIS, E.; VOULODIMOS, A.; DOULAMIS, A.; DOULAMIS, N.; STATHAKI, T. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. **Applied Intelligence**, v. 49, n. 7, p. 2793–2806, 7 Jul. 2019. DOI 10.1007/s10489-018-01396-y. Available at: http://link.springer.com/10.1007/s10489-018-01396-y. Accessed on: 1 Aug. 2019.

QINGGUO, T.; QIJUN, L.; GE, B.; LI, Y. A methodology framework for retrieval of concrete surface crack′s image properties based on hybrid model. **Optik**, v. 180, n. October 2018, p. 199–214, Feb. 2019b. DOI 10.1016/j.ijleo.2018.11.013. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0030402618317571. Accessed on: 1 Aug. 2019.

QU, Z.; CHEN, Y.-X.; LIU, L.; XIE, Y.; ZHOU, Q. The Algorithm of Concrete Surface Crack Detection Based on the Genetic Programming and Percolation Model. **IEEE Access**, v. 7, p. 57592–57603, 2019. DOI 10.1109/ACCESS.2019.2914259. Available at: https://ieeexplore.ieee.org/document/8704292/. Accessed on: 1 Aug. 2019.

QUINTANA, M.; TORRES, J.; MENÉNDEZ, J. M. A Simplified Computer Vision System for Road Surface Inspection and Maintenance. v. 17, n. 3, p. 608–619, 2016. .

RASCHKA, S.; JULIAN, D.; HEARTY, J. **Python: Deeper Insights into Machine**

**Learning**. Birmingham, UK: Packt Publishing Ltd, 2016.

REAGAN, D.; SABATO, A.; NIEZRECKI, C. Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges. **Structural Health Monitoring**, v. 17, n. 5, p. 1056–1072, 10 Sep. 2018. DOI 10.1177/1475921717735326. Available at: http://journals.sagepub.com/doi/10.1177/1475921717735326. Accessed on: 1 Aug. 2019.

RUCK, D. W.; ROGERS, S. K.; KABRISKY, M. Feature Selection Using a Multilayer Perceptron. **Journal of Neural Network Computing**, v. 2, n. 2, p. 40–48, 1990. .

RUIZ, M.; MUJICA, L. E.; ALFÉREZ, S.; ACHO, L.; TUTIVÉN, C.; VIDAL, Y.; RODELLAR, J.; POZO, F. Wind turbine fault detection and classification by means of image texture analysis. **Mechanical Systems and Signal Processing**, v. 107, p. 149–167, Jul. 2018. DOI 10.1016/j.ymssp.2017.12.035. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0888327017306738. Accessed on: 1 Aug. 2019.

SAGI, O.; ROKACH, L. Ensemble learning: A survey. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 8, n. 4, p. 1–18, 27 Jul. 2018. DOI 10.1002/widm.1249. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249. Accessed on: 1 Aug. 2019.

SALMAN, M.; MATHAVAN, S.; KAMAL, K.; RAHMAN, M. Pavement crack detection using the Gabor filter. 2013. *In:* **16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)** [...]. The Hague, The Netherlands: IEEE, 2013. p. 2039–2044. Available at: https://doi.org/10.1109/ITSC.2013.6728529. Accessed on: 1 Aug. 2019.

SANKARASRINIVASAN, S.; BALASUBRAMANIAN, E.; KARTHIK, K.; CHANDRASEKAR, U. Health Monitoring of Civil Structures with Integrated UAV and Image Processing System. **Procedia Computer Science**, v. 54, p. 508–515, 2015. DOI 10.1016/j.procs.2015.06.058. Available at: https://doi.org/10.1016/j.procs.2015.06.058. Accessed on: 1 Aug. 2019.

SANTOS, B. O.; VALENÇA, J.; JÚLIO, E. Automatic mapping of cracking patterns on concrete surfaces with biological stains using hyper-spectral images processing. **Structural Control and Health Monitoring**, v. 26, n. 3, p. 1–2, 2019. DOI 10.1002/stc.2320. Available at: https://doi.org/10.1002/stc.2320. Accessed on: 1 Aug. 2019.

SENSITIVITY AND SPECIFICITY. [*s. d.*]. **Wikipedia**. Available at: https://en.wikipedia.org/wiki/Sensitivity_and_specificity. Accessed on: 9 Sep. 2019.

SHAN, B.; ZHENG, S.; OU, J. A stereovision-based crack width detection approach for concrete surface assessment. **KSCE Journal of Civil Engineering**, v. 20, n. 2, p. 803–812, 20 Mar. 2016. DOI 10.1007/s12205-015-0461-6. Available at: http://link.springer.com/10.1007/s12205-015-0461-6. Accessed on: 1 Aug. 2019.

SHI, Y.; CUI, L.; QI, Z.; MENG, F.; CHEN, Z. Automatic Road Crack Detection Using Random Structured Forests. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, n. 12, p. 3434–3445, Dec. 2016. DOI 10.1109/TITS.2016.2552248. Available at:

http://ieeexplore.ieee.org/document/7471507/. Accessed on: 1 Aug. 2019.

SINHA, S. K.; FIEGUTH, P. W. Automated detection of cracks in buried concrete pipe images. **Automation in Construction**, v. 15, n. 1, p. 58–72, Jan. 2006. DOI 10.1016/j.autcon.2005.02.006. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0926580505000452. Accessed on: 1 Aug. 2019.

STANIEK, M.; CZECH, P. Self-correcting neural network in road pavement diagnostics. **Automation in Construction**, v. 96, n. February, p. 75–87, Dec. 2018. DOI 10.1016/j.autcon.2018.09.001. Available at: https://doi.org/10.1016/j.autcon.2018.09.001. Accessed on: 1 Aug. 2019.

SULEIMAN, A. R.; NELSON, A. J.; NEHDI, M. L. Visualization and quantification of crack self-healing in cement-based materials incorporating different minerals. **Cement and Concrete Composites**, v. 103, n. April, p. 49–58, Oct. 2019. DOI 10.1016/j.cemconcomp.2019.04.026. Available at: https://doi.org/10.1016/j.cemconcomp.2019.04.026. Accessed on: 1 Aug. 2019.

SUTTER, B.; LELEVÉ, A.; PHAM, M. T.; GOUIN, O.; JUPILLE, N.; KUHN, M.; LULÉ, P.; MICHAUD, P.; RÉMY, P. A semi-autonomous mobile robot for bridge inspection. **Automation in Construction**, v. 91, n. January, p. 111–119, Jul. 2018. DOI 10.1016/j.autcon.2018.02.013. Available at: https://doi.org/10.1016/j.autcon.2018.02.013. Accessed on: 1 Aug. 2019.

TALAB, A. M. A.; HUANG, Z.; XI, F.; HAIMING, L. Detection crack in image using Otsu method and multiple filtering in image processing techniques. **Optik**, v. 127, n. 3, p. 1030–1033, Feb. 2016. DOI 10.1016/j.ijleo.2015.09.147. Available at: http://dx.doi.org/10.1016/j.ijleo.2015.09.147. Accessed on: 1 Aug. 2019.

TEDESCHI, A.; BENEDETTO, F. A real-time automatic pavement crack and pothole recognition system for mobile Android-based devices. **Advanced Engineering Informatics**, v. 32, p. 11–25, Apr. 2017. DOI 10.1016/j.aei.2016.12.004. Available at: http://dx.doi.org/10.1016/j.aei.2016.12.004. Accessed on: 1 Aug. 2019.

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. [*s. d.*]. *In:* **Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)** [...]. [*S. l.*]: Narosa Publishing House, [*s. d.*]. p. 839–846. DOI 10.1109/ICCV.1998.710815. Available at: http://ieeexplore.ieee.org/document/710815/. Accessed on: 1 Aug. 2019.

VALENÇA, J.; JÚLIO, E. MCrack-Dam: the scale-up of a method to assess cracks on concrete dams by image processing. The case study of Itaipu Dam, at the Brazil–Paraguay border. **Journal of Civil Structural Health Monitoring**, v. 8, n. 5, p. 857–866, 15 Nov. 2018. DOI 10.1007/s13349-018-0309-0. Available at: http://link.springer.com/10.1007/s13349-018-0309-0. Accessed on: 1 Aug. 2019.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. New York, NY: Springer New York, 2000. v. 2, . DOI 10.1007/978-1-4757-3264-1. Available at: https://books.google.ie/books?id=jzbY4NisxckC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. Accessed on: 1 Aug. 2019.

VARONA, B.; MONTESERIN, A.; TEYSEYRE, A. A deep learning approach to automatic road surface monitoring and pothole detection. **Personal and Ubiquitous Computing**, 27 May 2019. DOI 10.1007/s00779-019-01234-z. Available at: http://link.springer.com/10.1007/s00779-019-01234-z. Accessed on: 1 Aug. 2019.

WANG, G.-D.; ZHANG, P.-L.; LIN, G.-Q.; KOU, X. Texture Feature Extraction Method Fused with LBP and GLCM. **Computer Engineering**, v. 38, n. 11, p. 199–201, 2012. DOI 10.3969/j.issn.1000-3428.2012.11.061. Available at: https://doi.org/10.3969/j.issn.1000-3428.2012.11.061. Accessed on: 1 Aug. 2019.

WANG, G.; TSE, P. W.; YUAN, M. Automatic internal crack detection from a sequence of infrared images with a triple-threshold Canny edge detector. **Measurement Science and Technology**, v. 29, n. 2, p. 025403, 1 Feb. 2018. DOI 10.1088/1361-6501/aa9857. Available at: http://stacks.iop.org/0957-0233/29/i=2/a=025403?key=crossref.425508d0302e925b8616761f36c412b8. Accessed on: 1 Aug. 2019.

WANG, H.; LI, G.; WANG, G.; PENG, J.; JIANG, H.; LIU, Y. Deep learning based ensemble approach for probabilistic wind power forecasting. **Applied Energy**, v. 188, p. 56–70, 2017. DOI 10.1016/j.apenergy.2016.11.111. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0306261916317421. Accessed on: 1 Aug. 2019.

WANG, N.; ZHAO, X.; ZHAO, P.; ZHANG, Y.; ZOU, Z.; OU, J. Automatic damage detection of historic masonry buildings based on mobile deep learning. **Automation in Construction**, v. 103, n. March, p. 53–66, Jul. 2019. DOI 10.1016/j.autcon.2019.03.003. Available at: https://doi.org/10.1016/j.autcon.2019.03.003. Accessed on: 1 Aug. 2019.

WANG, P.; HUANG, H. Comparison analysis on present image-based crack detection methods in concrete structures. Oct. 2010. *In:* **2010 3rd International Congress on Image and Signal Processing** [...]. [*S. l.*]: IEEE, Oct. 2010. p. 2530–2533. DOI 10.1109/CISP.2010.5647496. Available at: http://ieeexplore.ieee.org/document/5647496/. Accessed on: 1 Aug. 2019.

WANG, X.; WU, N. Crack identification at welding joint with a new smart coating sensor and entropy. **Mechanical Systems and Signal Processing**, v. 124, p. 65–82, Jun. 2019. DOI 10.1016/j.ymssp.2019.01.044. Available at: https://doi.org/10.1016/j.ymssp.2019.01.044. Accessed on: 1 Aug. 2019.

WENG, X.; HUANG, Y.; WANG, W. Segment-based pavement crack quantification. **Automation in Construction**, v. 105, n. May, p. 102819, Sep. 2019. DOI 10.1016/j.autcon.2019.04.014. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0926580518312196. Accessed on: 1 Aug. 2019.

WU, R.; ZHANG, D.; YU, Q.; JIANG, Y.; AROLA, D. Health monitoring of wind turbine blades in operation using three-dimensional digital image correlation. **Mechanical Systems and Signal Processing**, v. 130, p. 470–483, Sep. 2019. DOI 10.1016/j.ymssp.2019.05.031. Available at: https://doi.org/10.1016/j.ymssp.2019.05.031. Accessed on: 1 Aug. 2019.

XING, C.; HUANG, J.; XU, Y.; SHU, J.; ZHAO, C. Research on crack extraction based on the improved tensor voting algorithm. **Arabian Journal of Geosciences**, v. 11, n. 13, p. 342, 27 Jul. 2018. DOI 10.1007/s12517-018-3676-2. Available at: http://link.springer.com/10.1007/s12517-018-3676-2. Accessed on: 1 Aug. 2019.

YAMAGUCHI, T.; HASHIMOTO, S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. **Machine Vision and Applications**, v. 21, n. 5, p. 797–809, 11 Aug. 2010. DOI 10.1007/s00138-009-0189-8. Available at: http://link.springer.com/10.1007/s00138-009-0189-8. Accessed on: 1 Aug. 2019.

YANG, F.; ZHANG, L.; YU, S.; PROKHOROV, D.; MEI, X.; LING, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. **IEEE Transactions on Intelligent Transportation Systems**, , p. 1–11, 2019. DOI 10.1109/tits.2019.2910595. Available at: https://doi.org/10.1109/tits.2019.2910595. Accessed on: 1 Aug. 2019.

YANG, Y.-S.; YANG, C.-M.; HUANG, C.-W. Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis. **Advances in Engineering Software**, v. 83, p. 99–108, May 2015. DOI 10.1016/j.advengsoft.2015.02.005. Available at: http://dx.doi.org/10.1016/j.advengsoft.2015.02.005. Accessed on: 1 Aug. 2019.

YANG, Y.; WU, C.; HSU, T. T. C.; YANG, H.; LU, H.; CHANG, C. Image analysis method for crack distribution and width estimation for reinforced concrete structures. **Automation in Construction**, v. 91, n. January, p. 120–132, Jul. 2018. DOI 10.1016/j.autcon.2018.03.012. Available at: https://doi.org/10.1016/j.autcon.2018.03.012. Accessed on: 1 Aug. 2019.

YIYANG, Z. The design of glass crack detection system based on image preprocessing technology. Dec. 2014. *In:* **2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference** [...]. [*S. l.*]: IEEE, Dec. 2014. p. 39–42. DOI 10.1109/ITAIC.2014.7065001. Available at: http://ieeexplore.ieee.org/document/7065001/. Accessed on: 1 Aug. 2019.

ZHANG, D.; LI, Q.; CHEN, Y.; CAO, M.; HE, L.; ZHANG, B. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. **Image and Vision Computing**, v. 57, p. 130–146, Jan. 2017a. DOI 10.1016/j.imavis.2016.11.018. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0262885616302153. Accessed on: 1 Aug. 2019.

ZHANG, H.; JACKMAN, J. Feasibility of Automatic Detection of Surface Cracks in Wind Turbine Blades. **Wind Engineering**, v. 38, n. 6, p. 575–586, 2014. DOI 10.1260/0309-524X.38.6.575. Available at: http://journals.sagepub.com/doi/10.1260/0309-524X.38.6.575. Accessed on: 1 Aug. 2019.

ZHANG, L.; YANG, F.; DANIEL ZHANG, Y.; ZHU, Y. J. Road crack detection using deep convolutional neural network. Sep. 2016. *In:* **2016 IEEE International Conference on Image Processing (ICIP)** [...]. [*S. l.*]: IEEE, Sep. 2016. p. 3708–3712. DOI 10.1109/ICIP.2016.7533052. Available at: http://ieeexplore.ieee.org/document/7533052/. Accessed on: 1 Aug. 2019.

ZHANG, Y.; CHEN, C.; WU, Q.; LU, Q.; ZHANG, S.; ZHANG, G.; YANG, Y. A Kinect-Based Approach for 3D Pavement Surface Reconstruction and Cracking Recognition. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 12, p. 3935–3946, Dec. 2018. DOI 10.1109/TITS.2018.2791476. Available at:

https://ieeexplore.ieee.org/document/8357931/. Accessed on: 1 Aug. 2019.

ZHAO, R.; YAN, R.; CHEN, Z.; MAO, K.; WANG, P.; GAO, R. X. Deep Learning and Its Applications to Machine Health Monitoring: A Survey. v. 115, p. 213–237, 2016. DOI 10.1016/j.jocs.2017.06.006. Available at: http://arxiv.org/abs/1612.07640. Accessed on: 1 Aug. 2019.

ZHU, J.; ROSSET, S.; ZOU, H.; HASTIE, T. Multi-class AdaBoost. **Statistics and Its Interface**, v. 2, n. 3, p. 349–360, 2009. DOI 10.4310/SII.2009.v2.n3.a8. Available at: http://www.intlpress.com/site/pub/pages/journals/items/sii/content/vols/0002/0003/a008/. Accessed on: 1 Aug. 2019.

ZHU, Q.; DINH, T. H.; HOANG, V. T.; PHUNG, M. D.; HA, Q. P. Crack Detection Using Enhanced Thresholding on UAV based Collected Images. 19 Dec. 2018. Available at: https://ssl.linklings.net/conferences/acra/acra2018_proceedings/views/includes/files/pap105s1-file1.pdf. Accessed on: 1 Aug. 2019.

ZHUANG, L.; WANG, L.; ZHANG, Z.; TSUI, K. L. Automated vision inspection of rail surface cracks: A double-layer data-driven framework. **Transportation Research Part C: Emerging Technologies**, v. 92, n. September 2017, p. 258–277, Jul. 2018. DOI 10.1016/j.trc.2018.05.007. Available at: https://doi.org/10.1016/j.trc.2018.05.007. Accessed on: 1 Aug. 2019.

ZOU, Q.; CAO, Y.; LI, Q.; MAO, Q.; WANG, S. CrackTree: Automatic crack detection from pavement images. **Pattern Recognition Letters**, v. 33, n. 3, p. 227–238, Feb. 2012. DOI 10.1016/j.patrec.2011.11.004. Available at: http://dx.doi.org/10.1016/j.patrec.2011.11.004. Accessed on: 1 Aug. 2019.

ZOU, Q.; ZHANG, Z.; LI, Q.; QI, X.; WANG, Q.; WANG, S. DeepCrack: Learning hierarchical convolutional features for crack detection. **IEEE Transactions on Image Processing**, v. 28, n. 3, p. 1498–1512, 2019. DOI 10.1109/TIP.2018.2878966. Available at: https://doi.org/10.1109/TIP.2018.2878966. Accessed on: 1 Aug. 2019.

# APPENDIX A – LITERATURE REVIEW OF CRACK DETECTION METHODS BASED ON IMAGE PROCESSING

| Paper | Year | Goal | Image acquisition | Pre-processing | Segmentation | Feature extraction | Classification | Structural analysis |
|---|---|---|---|---|---|---|---|---|
| (HU; ZHAO; WANG, 2010) | 2010 | C2/L | Camera | x | Binarization: threshold<br>Post-processing: area filtering | Texture (Haralick features) and crack shape | ML | Area, length, orientation, and width |
| (YAMAGUCHI; HASHIMOTO, 2010) | 2010 | L | Camera | x | Method: percolation model | x | RL | x |
| (CHAMBON; MOLIARD, 2011) | 2011 | L | Camera | Denoising: erosion in gray levels and median filtering,<br>Enhancement: histogram equalization and mean filtering | Binarization: local threshold<br>Post-processing: morphological closing and shape filtering | x | RL | Height and width |
| (FUJITA, Yusuke; HAMAMOTO, 2011) | 2011 | L | Camera | Denoising: median filter<br>Enhancement: a multi-scale line filter with the Hessian matrix | Method: probabilistic relaxation<br>Binarization: locally adaptive thresholding | x | RL | x |
| (JAHANSHAHI *et al.*, 2013) | 2013 | L | Camera | x | Method: morphological operation<br>Binarization: Otsu method<br>Post-processing: length filtering | Crack shape | ML | Width |
| (OLIVEIRA; CORREIA, 2014) | 2014 | L/C4 | Camera | Denoising: anisotropic diffusion, morphological smoothing, erosion, and dilation, wavelet transform, an adaptive filter<br>Enhancement: pixel intensity normalization and saturation<br>Object removal: lane | Binarization: the dual intensity threshold<br>Post-processing: connect regions, width and shape filtering | Intensity-based<br>Normalization | ML/RL | Severity level |
| (AMHAZ *et al.*, 2016) | 2016 | L | Vehicle-mounted camera | x | Method: minimal path with Dijkstra algorithm<br>Post-processing: length filtering | x | RL | Width |
| (PRASANNA *et al.*, 2016) | 2016 | L | Vehicle-mounted camera | x | Method: line detector based on curve fitting<br>Post-processing: morphological closing and hole filling | Intensity-based, gradient-based and scale-space | ML | x |
| (SHAN; ZHENG; OU, 2016) | 2016 | L | Camera | Denoising: median filter<br>Enhancement: self-adaptive histogram | Binarization: Otsu method<br>Edge detection: Canny operator | x | RL | Width |

| Paper | Year | Goal | Image acquisition | Pre-processing | Segmentation | Feature extraction | Classification | Structural analysis |
|---|---|---|---|---|---|---|---|---|
| (TALAB *et al.*, 2016) | 2016 | L | Camera* | x | Edge detection: Sobel filter<br>Binarization: Otsu method<br>Post-processing: area filtering | x | RL | x |
| (ZHANG, Dejin *et al.*, 2017a) | 2017 | L | Camera | Enhancement: intensity correction and Grid Cell Analysis<br>Object removal: lane and sign | Method: propose Maximally Stable Extremal Regions<br>Binarization: adaptive thresholding<br>Post-processing: connect regions and region growing | x | RL | x |
| (CHO *et al.*, 2018b) | 2018 | L | Camera | Denoising: median filter<br>Object removal: background | Method: K-means<br>Binarization: Otsu method<br>Post-processing: morphological closing and area filtering | x | RL | x |
| (DAVOUDI; MILLER; KUTZ, 2018b) | 2018 | L | Camera* | x | Edge detection: Canny operator | Overall image and crack shape averaged | ML | x |
| (HOANG; NGUYEN, 2018) | 2018 | L/C5 | Camera | Denoising: median filter<br>Enhancement: Fast Local Laplacian Filter | Edge detection: steerable Gaussian filters and Sobel filter | Projective Integrals | ML | x |
| (HOANG, 2018b) | 2018 | L | Camera | Enhancement: Min-Max Gray Level Discrimination | Binarization: Otsu method<br>Post-processing: area and shape filtering | x | RL | Area, length, orientation, perimeter, and width |
| (HOANG, 2018a) | 2018 | L/C5 | Camera | Denoising: Median filter (window size 5x5) | Method: Laplacian pyramid | Projective integrals | ML | x |
| (LEI *et al.*, 2018) | 2018 | L | Camera | Denoising: Gaussian filter<br>Enhancement: histogram equalization | Method: find crack center points based on the gray level distribution | x | RL | x |
| (LIANG, Sun; JIANCHUN; XUN, 2018) | 2018 | L | Camera | Enhancement: Linear Grayscale Transformation | Binarization: Otsu method<br>Post-processing: length and area filtering, morphology fill, expand and corrode, and connect regions | Crack shape | ML | x |
| (WANG, Gaochao; TSE; YUAN, 2018) | 2018 | L | Infrared* | Denoising: Gaussian filter | Edge detection: Canny operator<br>Post-processing: K-means | Frequency domain | RL | x |

| Paper | Year | Goal | Image acquisition | Pre-processing | Segmentation | Feature extraction | Classification | Structural analysis |
|---|---|---|---|---|---|---|---|---|
| (XING *et al.*, 2018) | 2018 | L | Camera | Denoising: black hat algorithm | Binarization: adaptive width template method<br>Post-processing: Tensor voting algorithm, and length, area and shape filtering | x | RL | x |
| (ZHU, Q *et al.*, 2018) | 2018 | L | UAV mounted camera | x | Method: recursively search for the darker region until a stop condition is met, using the Otsu method | x | RL | x |
| (ZHUANG *et al.*, 2018) | 2018 | L | Camera | x | Method: proposed feature-based linear iterative crack aggregation method | Haar-like features | ML | x |
| (DORAFSHAN; THOMAS; MAGUIRE, 2019) | 2019 | L | UAV mounted camera | x | Edge detection: filtering in both spatial and frequency domain<br>Binarization: local threshold and area threshold<br>Post-processing: area filtering | x | RL | x |
| (FAN, Rui *et al.*, 2019) | 2019 | C2/L | Camera | Denoising: bilateral filter | Binarization: only for positively classified images, with threshold optimization via quantization problem | x | ML/RL | x |
| (HEO; JEON; SON, 2019) | 2019 | L | Video | Enhancement: top-hat filtering and histogram contrast stretch | Binarization: proposed histogram optimum thresholding<br>Post-processing: hole filling, morphological erode and close, and length and width filtering | x | RL | x |
| (HOANG; NGUYEN, 2019) | 2019 | L/C4 | Camera | Enhancement: Gaussian steerable filters | Binarization: Otsu method | Projective integrals and crack characteristics | ML | x |
| (PAGLINAWAN *et al.*, 2018) | 2019 | L | UAV mounted camera | x | Binarization: adaptive threshold<br>Post-processing: image inversion, median filter and area filtering | x | RL | Condition, distance, length, and width |
| (QINGGUO *et al.*, 2019b) | 2019 | L/C4 | Camera | x | Edge detection: Canny operator<br>Post-processing: region growing algorithm, hole filling, and shape filtering | x | RL | Length, intersection properties, and width |
| (SANTOS; VALENÇA; JÚLIO, 2019) | 2019 | L | Hyper-spectral* | Enhancement: spectral alignment and brightness normalization | Method: K-means<br>Post-processing: connect regions | x | RL | x |