Deep Neural Networks Airport Runway Crack Detection and Densification

Thesis Submitted
In Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

Software Engineering

by

Aryan Bansal
(2K22/SWE/04)

Under the Supervision of
Dr. Abhilasha Sharma
(Assistant Professor, SE, DTU)

To the
Department of Software Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May, 2024

ACKNOWLEDGEMENT

I would like to express my deep appreciation to **Dr. Abhilasha Sharma**, Assistant Professor at the Department of Software Engineering, Delhi Technological University, for her invaluable guidance and unwavering encouragement throughout this research. Her vast knowledge, motivation, expertise, and insightful feedback have been instrumental in every aspect of preparing this research plan.

I am also grateful to **Prof. Ruchika Malhotra**, Head of the Department, for her valuable insights, suggestions, and meticulous evaluation of my research work. Her expertise and scholarly guidance have significantly enhanced the quality of this thesis.

My heartfelt thanks go out to the esteemed faculty members of the Department of Software Engineering at Delhi Technological University. I extend my gratitude to my colleagues and friends for their unwavering support and encouragement during this challenging journey. I have had some friends that I am thankful to be around. They made me feel truly at home. In particular, I would like to thank Roshni Singh whom I had such a great time. Last two years were great with such a lovely bunch of people around me. Their intellectual exchanges, constructive critiques, and camaraderie have enriched my research experience and made it truly fulfilling.

While it is impossible to name everyone individually, I want to acknowledge the collective efforts and contributions of all those who have been part of this journey. Their constant love, encouragement, and support have been indispensable in completing this MTech thesis.

Aryan Bansal

(2K22/SWE/04)

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE DECLARATION

I ARYAN BANSAL (2K22/SWE/04) hereby certify that the work which is being presented in the thesis entitled "**Deep Neural Networks Airport Runway Crack Detection And Densification"** in partial fulfillment of the requirements for the award of the Degree of Master of Technology submitted in the Department of Software Engineering, Delhi Technological University in an authentic record of my work carried out during the period from August 2022 to May 2024 under the supervision of Dr. Abhilasha Sharma.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Aryan Bansal

This is to certify that the student has incorporated all the corrections suggested by the examiner in the thesis and that the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor(s)

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that Aryan Bansal (2K22/SWE/04) has carried out their project work presented in this thesis entitled "**Deep Neural Networks Airport Runway Crack Detection And Densification"** for the award of **Master of Technology** from the Department of Software Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Abhilasha Sharma

Date:

Assistant Professor

Department of Software Engineering,

DTU-Delhi, India

Deep Neural Networks Airport Runway Crack Detection and Densification

Aryan Bansal

ABSTRACT

As a greater number of people travel due to the built many airports by improved Infrastructure, a lot of runways are used constantly by planes and worries about safeties are increasing since new ways of using them keep on mushrooming because of this line of action. Runway maintenance has become an essential duty due to check of cracks. However, traditional methods such as manual inspection because cracks always have differing degree on division have never had good performance in time and needed more time to proof the exact problem.

To solve these problems, it is possible to present an dataset that goes by the name of ARID. It splits into eight divisions of runway cracks. It contains 8228 anotated illustrations hence serving as a good source for acquiring the models to work with while training or evaluating them. By using modern methods of deep learning like YOLO v5 and Faster R-CNN, this idea creates a good working algorithm for detecting runway cracks.

To get optimal model performance tuning and changing different parameters being fine-tuned are involved. We have noticed a significant improvement in the performance metrics of our model as a result of intense experimentation. The precision of crack detection has increased from 83% up to 92%, while the recall rate has escalated steadily from an initial 62.8% to its current level which stands at 76%. The proposed model has been demonstrated by these results to be effective in precisely identifying and categorizing cracks on runways, so that runway maintenance and safety can be upgraded.

Keyword: YOLO, Faster RCNN, Airport Runway, AIRD, Crack Detection.

TABLE OF CONTENT

LIST OF TABLE(S)

LIST OF FIGURE(S)

LIST OF ABBREVIATION(S)

YOLO	You Only Look Once
FRCNN	Faster Recurrent Neural Network
RNN	Recurrent Neural Network
PCA	Principal Component Analysis
CNN	Convolutional Neural Network
ARID	Airport Runway Image Dataset
AI	Artificial Intelligence
ML	Machine Learning
DNN	Deep Neural Network
ICAO	Internation Civil Aviation Organisation
ReLU	Rectified Linear Unit
GPU	Graphics Processing Unit
IP	Image Processing
IRI	International Roughness Index
NDHM	Neighbouring Difference Histogram Model
SVM	Support Vector Machine
DL	Deep Learning
CRNN	Convolutional Recurrent Neural Network
LSTM	Long Short-Term Memory
LR	Location Refinement
FEC	Future Extraction Classification
RON	Reason Proposal Network

CHAPTER 1

INTRODUCTION

1.1 Overview

In the past few years, extreme travel and travel changing have greatly increased all over the world. It is the aviation industry that has experienced major technology advancements leading to safe and comfortable flights. Current airplanes have cutting-edge navigation systems, highly developed safety features together with better cabin amenities that provide an enhanced air travel experience for travellers. Various factors influence the development of surface cracks on runways despite the way in which heightened transportation activity impacts infrastructure service performance and life. This is why routine maintenance has become increasingly important especially in detecting and classifying runway cracks. Consequently, safety and service life may be threatened through their structural degradation. Cracks in the pavement can cause damage on those who use it as well as risk their safety when they are at risk. Nonetheless, these are most typical problems that occur in airport runways leading to reduced tension and even accidents in some cases. Should these broken down cracks not be fixed, the instance will be exacerbated as a result of persistent establishment or human actions. Fluctuations in temperature, precipitation, freeze-thaw cycles, along with excessive loads and improper maintenance by humans can cause damage to runways. It is important to inspect and repair runways regularly to prevent damage. Ground penetrating radar as well as automatic crack detection are employed to find deep-set extreme cracks. They assist in maintenance planning thus promoting speedy, efficient and effective repairs.

Apart from technology-oriented strategies, the use of good designs for airport runways, as well as their proper construction, should be given serious thought as they can reduce incidences of cracks occurring. Good quality materials, proper drainage system and coming up with efficient methods of managing different heavy loads. All these measures are important in extending the life span of pavement runways. The ongoing oversight and maintenance of runway infrastructure are imperative regarding the safety and efficiency of air travel. If the aviation industry takes this step towards cracks ' damage first, airport operations will be more reliable. This will not only help increase integration but enhance growth and sustainability globally as well in terms of transport. Maintain pavement and reduce green house emissions by sealing cracks in roads before they expand into potholes. By removing cracks at an early stage rehabilitation costs might be minimized up to 80% compared with rehabilitating deteriorated pavement. In the last few years there has been explosive growth in the Indian economy quickening the pace of airport development which has in turn allowed aviation industry to get over pre-pandemic levels. New routes and startup carriers are appearing on the horizon. By the year 2025, the Indian government plans to construct 220 more airports while in 2027, India should have 1,200 aircrafts and 400 million passengers according to Jyotiraditya Scindia who is the Minister for Civil Aviation. In a market expected to see great growth, the country is constructing new greenfield airports funded using the public money and public-private partnerships. So far, eight out of the 21 greenfield airports that have been planned are already in operation. The increase in the number of people deciding to fly would mean that there would be different kinds of damages on the runways. Fuel marks and wheel marks observed on the runway are usually numerous. Furthermore, there could be very fine cracks that may indicate complete failure. The pictures of the runways have a lot of noise and contain features such as fine cracks, fuel spills, and rough textures.

The analysis process within intelligent transportation system has benefited significantly from the advent of automated crack detection technologies characterized by quick and accurate outcomes that have replaced slow and subjective conventional methods. An airport authority, say International Civil Aviation Organization(ICAO) could utilize an automated crack detection system to examine runways with great efficiency ranking them according to urgency before undertaking repairs as way of extending their lifespan. Computer vision aims at enabling computers to learn from the visual representations of digital images and videos. This helps in improving understanding of features as well as patterns using visual data. In these research areas, there is an abundance of pictorial data which can be accessed through cell phones including digital cameras. Several scholars have been researching deep architectures that detect cracks. For example, Gopalakrishna et al.[1] reviewed a series of deep learning approaches which are applicable to crack detection. By using Otsu's improved threshold segmentation algorithm, road signs can be removed from an image of a road surface. Once the marks are removed and the crack is traced, we use the enhanced adaptive threshold segmentation for segmenting the image. Oliveira et al.[2] used various image analysis techniques to detect and describe cracks in road surface.

1.2 Problem Statement

The development of AI and deep learning technology helps improve plane operation safety and efficacy since it promotes better runway conditions. Using these technologies guarantees that airplane runway upkeep approaches are always forward-looking enough to prevent any possible issues before they pose a threat to civil aviation growth as well as sustainability. Although these techniques have been shown to work well in identifying cracks found in high quality image databases[3], it should be noted that they may not provide enough precision to distinguish between crack and complex background in images of poor quality. Effective modeling of pavement distress would require identification and further analysis of critical surface cracks. The detection rate for surface cracks can be influenced by a number of factors such as age of layers, volume of traffic, climatic conditions, stratification and layer quality. Road administrators can rely on these specifications for developing road maintenance strategies focusing on type, extent and stage various faults are in when detected. Previously, numerous studies have tried to tackle these problems by following different methods to find the solutions but were not so fruitful. For instance, in trying to determine any form of failure points, CrackNet[4] sometimes managed to succeed though it failed at establishing different types of faults along the same failure surfaces.. In contrast, Zalama et al.[5] examined horizontal, vertical varieties of distress as proposed by Akarsu et al.[6] who further proposed that three types of distresses exist; horizontal, vertical and alligator cracks. Other studies have looked into identifying blurred road signs or distinguishing sealed cracks from other categories of pavement imperfections.

To achieve improved effectiveness in deep learning techniques, good quality of data used during training and testing of models are essential. For the airport runway distress surface database, crucial is availability of labeled datasets. Accurate learning of different types of cracks and distress pattern's distinguishing features is facilitated by high-quality labeled datasets so that they should encompass diverse images corresponding to multiple environments, various lighting conditions as well as noise levels in order to guarantee model's robustness. Researchers are looking for new ways for improving the models of crack detecting that include modern deep learning methods like convolutional neural networks, recurrent neural networks and hybrid models up which are mixed several ways of coping with difficult data. One way of increasing the robustness of such models to diverse environments is by using data augmentation methods such as rotating or scaling the image during training together with flipping it horizontally.

One additional benefit, in addition to assisting with including sensor data like temperature and humidity senses with the visual data for the enhancement of model predictability, is integrating drones or mobile imaging systems for obtaining updated information sets to train crack detection models efficiently. The objective is eventually to invent a type of system which can automatically detecting cracks and identifying them so as to operate almost anywhere and give precise, useful feedback concerning repairs of roads. With such a system airports will check out the decay ahead of time, reducing costs and pollution, as well as raising safety levels plus saving on time by handling the pavements properly. I introduce a new dataset called the "Airport Runway Image Dataset" (ARID). Initially, 8,228 images were extracted from 10 different surface sections. Those photos were collected through Google Applications Interface (API) using street view. The primary operation to be done using this dataset is to identify all the segments that show any forms of distress by placing bounding boxes around them to form a segmentation mask.

Two models that use DL (deep learning) have been improved based on the concept of deep learning and have been put into a single frame from the figure one. "This thesis makes the following important contributions:

- They introduced a new dataset that simultaneously depicts surface cracks and counts them with various cameras. Such cameras encompass overhead views as well as common angles. The information on break points came from overhead photos, whereas categorizations were based on wide angle images.
- The wide-field pictures have been tagged to show all nine identified types and their identifications i.e., D0-D8. As a result transverse, block, longitudinal, alligator, sealed transverse, sealed longitudinal and lane longitudinal cracking is observed, together with critical potholes that are used to assess the quality of crack surface.

The proposed model, implemented on two deep learning approaches, namely YOLO v5 and Faster R-CNN, is trained on the dataset mentioned above.

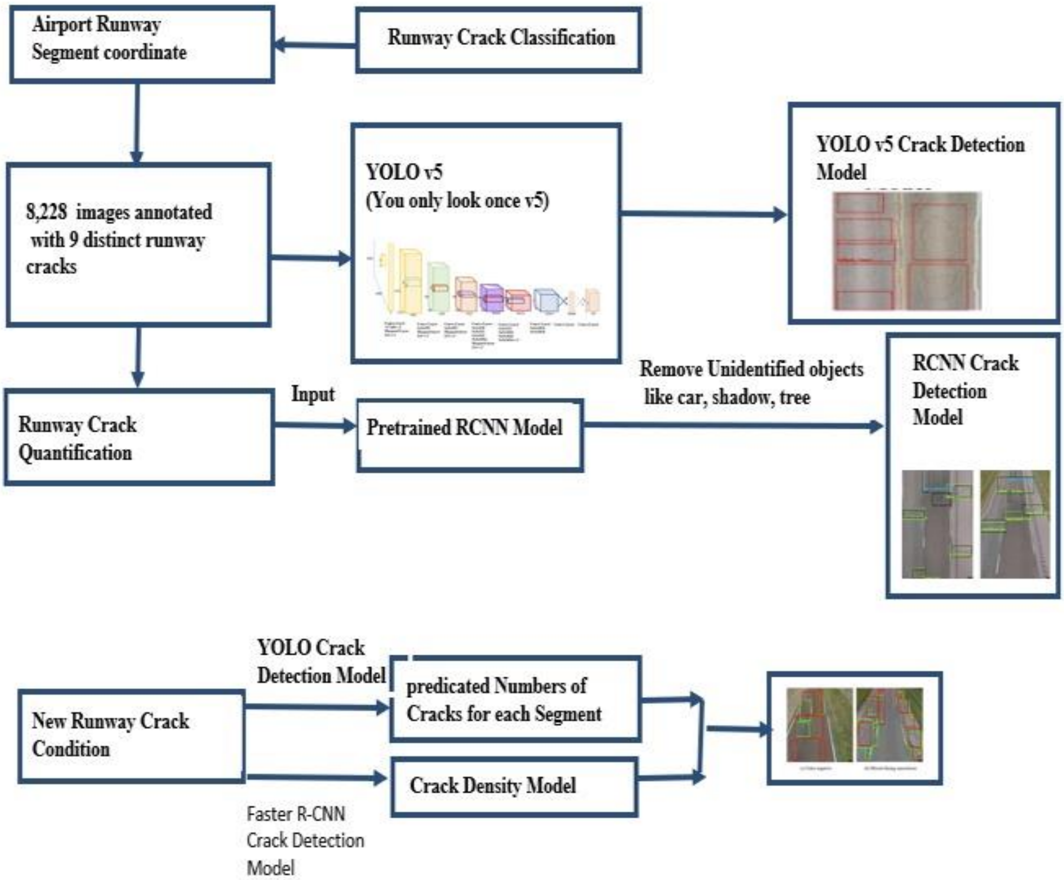


Figure 1.1. Outline for Airport Runway Crack Detection and its Classification

The following are chapters of these dissertations that are structured in order to offer an extensive review pertaining to primary ideas, methodologies, and discoveries in relation to the exploration of crack detection as well as segmentation using deep learning methods. The next section presents a comprehensive summary about the same topic areas. Delving into the basics of deep learning, Chapter 2 traces its historical development. An emphasis is placed on Convolutional Neural Networks (CNNs) as they play a crucial role in image processing applications. This section delves on the organization of CNNs which include convolutional layer, pooling layer and fully connected layer among others in addition to explaining how networks of such nature are taught to detect patterns as well as characteristics within pictures.This chapter explores different algorithmic approaches historically used for image analysis and processing e.g., edge detection, thresholding and morphological operations. In Chapter 3, we present an exhaustive literature review on conventional image processing techniques particularly applied to crack detection. Besides, this chapter offers an in-depth review of previous studies about crack detection which shows us where it has come from, in terms of efficiency, practicality, and feasibility. These improvements from traditional methods are what we are going to look at more deeply later on, in relation to this research work done using newer forms of machine learning called deep neural networks. In Chapter 4, the architecture for Crack Detection and Segmentation is proposed in detail. State-of-the-art deep learning models are used in this architecture. Thus, it explains how the architecture was designed and implemented by integrating You Only Look Once (YOLO) and Faster R-CNN models. For those who may not be familiar with these types of models, we will describe them later on in this chapter along with an outline of how they were configured and trained in order to perform tasks related to crack detection and segmentation. Chapter 5 explains the experimental design that was used to assess the suggested crack identification and division method. This section describes the methodology applied during the data collection step, data preprocessing task and particular parameters for model training purposes. The subsequent section contains different types of tests in which the respective configurations are evaluated. We are displaying and looking at the outcomes while illustrating how effectively the recommended models work on precision, recall and overall accuracy. In essence, the preceding paragraphs have presented a comprehensive view on what has been found out as well as the degree to which they added value through these chapters of the dissertation paper by stating Chapter 6 as its last section while giving an abridged account of different topics like thesis aims and methods used among others culminating into final chapter.

.

CHAPTER 2

DEEP LEARNING

Today, ML techniques have facilitated numerous parts of the contemporary civilization. More and more data is constantly produced, and it will further grow in the future. 80%-90% of the total data cannot perform most of the tasks as structured data (unstructured data). However, traditional ML techniques like logistic regression, support vector machine, decision tree and k-nearest neighbours' were constrained by the fact that they could not handle unstructured data. It has only been within the past few decades that machine learning has evolved from a methodology that required significant domain expertise and careful engineering to one where an algorithm might transform unstructured data points, such as pixel values for images, into appropriate representations that could then be used by other machine learning algorithms [7]. Representation learning is a suite of techniques that enable computers to analyse disorganised information and identify how it can be used for a given purpose without any specific instructions. Machine learning algorithms in deep learning have several layers of representation. The deep representation learning is achieved by using multiple layers of simple complex nodes, which can change the input from one form to another at a slightly higher level of abstraction. When enough of these are put together, it becomes possible to discover very sophisticated functions thereby making it easy for professionals with diverse research topics different fields take much attention. These novel technologies have been applied to tackle a difficult issue in civil engineering. Following section covers the basic idea of DL alongside the distinct constituents that must be put together without fail to develop an efficient DL model. Based on these insights and methodologies, an asphalt specific pavement-crack identifying framework will be brought forth.

2.1 History

In the early days of the AI construction, very high intelligence computing power tried very hard to resolve problems within the range of possibilities for human intellect; problems were thought of in a row of formal-mathematical rules, hence, they were simple enough for machine value. Therefore, the real aim of AI development is to handle tasks that are simple for people in such a way as they understand them "intuitively", but impossible to describe on the basis of any formal language for programming computers[8]. Solving these challenges is possible by using DL. DL aims not only to learn the mapping but as well as acquiring the most favourable data representation [8]. People have been using the terms AI and DL simultaneously ever since the first learning algorithms designed were imitative of brain functions. Essentially, the idea of artificial neural networks (ANNs) being the same as deep learning is now commonplace among practitioners in this field. About fifty years ago, Rosenblatt[9] popularized neural networks (NNs) through various types of perceptron networks. However, in 1969 Minsky and Papert considered them very limited in their function [10]. A lot of people generalized these restrictions improperly, which in turn caused a significant decrease in the popularity of neural networks. A number of deep learning techniques were developed in the 1980s and 1990s like long short term memory (LSTM)[11-12] as well as back propagation algorithm. The 1990s saw unrealistic claims made by the artificial intelligence community that failed to meet these expectations when artificial intelligence research could not live up to it. Kernel machines and graphical models also found success in their own right; this, coupled with a drop in interest for neural networks as of 2007. This led to NNs losing their enthusiasm between 2001-04 or so [8]. In 2006 Hochreiter et al.[10], demonstrated how one might construct deep-belief-network that could be trained effectively through unsupervised layer-wise learning: while still others adopted similar techniques when dealing with different types of hierarchical architectures [13,14]. These studies have waken up AI from coma. With performance better than other techniques in multiple artificial intelligence challenges, DL is now one of effective methods among supervised, unsupervised, and reinforcement learning.

2.2 Machine Learning

Since DL falls under a wider range of other ML methods, some basic concepts in ML have to be talked about. In different fields, ML algorithms and models have been utilised hence the multiple definitions of ML. The name "machine learning" was given in the year 1959 [15] hence this relates to how mathematical models and algorithms are employed for performing specific functions using data generated by computer systems together with experience [16]. Learning from data is the process of analysing situations endowed with certain patterns that do not have a known theoretical solution. In such situations, it means that Machine Learning will always provide ways through which such patterns can be identified through which patterns can be determined. The machine learning problems generally fall under three categories: supervised, unsupervised and reinforcement learning as shown in Figure 2.1. In supervised learning a naive model can only learn a regulated data with beginners guide (The learning set). From where it gets ins and outs together Proactive Maintenance; we can travel through multiple articles including step in step Self-Instructional. For example, when it comes to detecting whether an image has a particular object, training data will involve images containing the object or images that do not have it (the in-put), with each image receiving a label depending on whether or not it contains the object [16]. Contrastingly, outputs being non-existent serves as a basis for application of unsupervised learning models. Unsupervised study focuses on how systems can find a function to reflect a latent structure from data without labels. Refereeing to reinforcement learning is a method used by machines to learn through experimentation with reward from themselves experiences and actions in an interactive setting. The agent increases its performance by automatically discovering the best way of behaving in a given situation.

2.3 What is learning?

The traditional frameworks are used to explain the aspects of learning algorithms and for learning to be considered as feasible, provide mathematical proof of this fact—Shai Shalev-Shwartz, Shai Ben-David [18] presented examples that could help in understanding how basic learning process work alongside what have been identified as principal challenges within machine learning (ML). Rats learn how

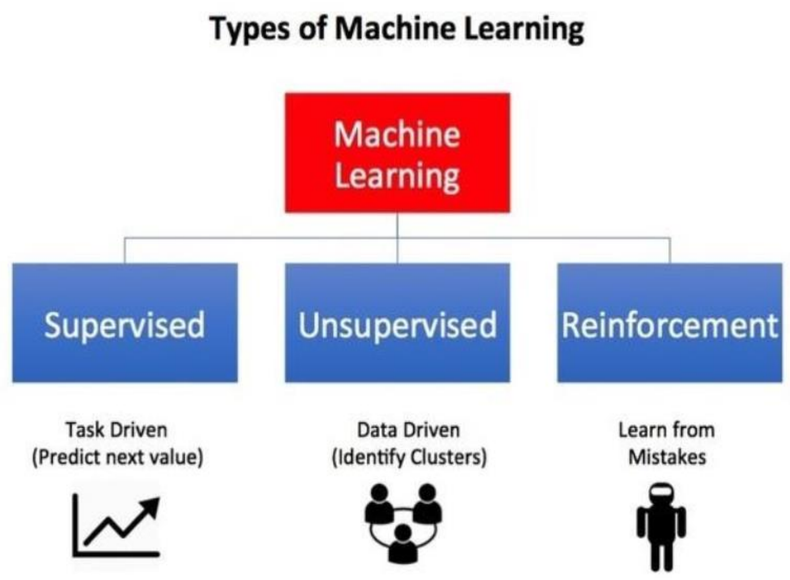


Figure 2.1 . Different ML problem categories [17]

to avoid poisoned food starting from their childhood. Rats usually take a small amount of new food first and are careful to investigate the physical consequences. If the food causes sickness, they never eat it forever. The experiment involved an animal in search of a harmless meal. In this case, the animal would expect that if it experienced a negative label then it would also develop negatively. Assume we are attempting to write a spam detector program. For instance, one straightforward way is to remember every email determined to be spam by a user. When an incoming email is received, it is verified against the spam set. If it is found in the spam set, then it is marked as a spam message; else, it is saed in the inbox folder. Memorization is occasionally helpful, but it does not have much in common with learning because it cannot be generalized. An intelligent learner who truly understood should be able to extract wider generalizations from diverse instances. It therefore means that generalizing constitutes the ultimate definition of intelligence. When compared with other creatures, man’s special gift is his ability to think and understand concepts widely, putting us one step ahead. For instance, given a realistic picture of an elephant, a child might be able to recognize a drawn elephant that looks very different (Figure 2.2). Another problem is when the learner comes to a wrong conclusion. In explaining this notion, Skinner’s superstition experiments are the most useful example. To be precise, Skinner put some hungry pigeons in a box that came with an automatic device meant to supply food for the hen occasionally with no consideration given to its actions. He found that pigeons would exhibit behaviours signalling expectancy only during feeding time and for more or less two minutes after that. While waiting for food, a particular bird spun round and round in a counter clockwise direction before making one or two turns in the opposite direction before it was rewarded. But there were sometimes when it was fed by Andy and would peck continuously at the upper edge of its basin."’A bird thrust its head out and swung it sharply rightwards from leftwards then back again with some slowness so as to make it like a pendulum while another bird began shaping up like it was making quotations (this means they stuck their heads beneath an unseen pole raised them up multiple times’[19].

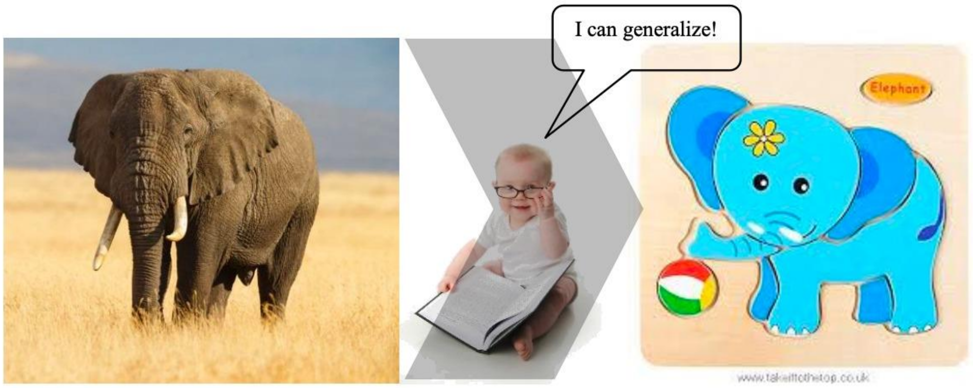


Figure 2.2. Concept of generalisation and intelligence

When humans learn, they use their common sense and ignore random patterns or conclusions from learning that are meaningless, but machines do not. A machine requires well defined principles to steer it out of arriving at irrelevant conclusions. In simpler terms, the algorithm should be able to discern a pattern in the data but not in the noise.

2.4 Convolutional Neural Network (CNN)

In this section, we will introduce the basic notions of NNs and discuss various parts of CNNs before explaining why each architecture is worth considering. There is a standard NN architecture shown in Figure 2.3 with input i given as a single feature vector, x^k. The input is passed through successive hidden layers, to estimate an output ŷ. All the layer consists of neurons (nodes), each of which is completely linked to all nodes in the previous layer and the following layer. You can do this at arbitrary patches because each layer has no connections to the others. With respect to this particular patch, the output of the one that came before it $a_k^{[l-1]}$ is modified by the weight $\omega_{jk}^{[l]}$ and added to a bias term $b_j^{[l]}$.

After this happens, it passes through an activation function $g^{[l]}$ which decides what will be outputted from the node $a_j^{[l]}$.

The result of each node is generally formulated as

$$a_j^{[1]} = g^{[1]}(\textstyle\sum_k \omega_{jk}^{[1]} a_k^{[1-1]} + b_j^{[1]}) \qquad (1)$$

The input vector is denoted by $a^{[0]}$. The final fully-connected layer $a^{[3]}$ is given the name “output layer” in this example, while in classification tasks it shows the likelihoods of classes. It should be noted that the weights $\omega_{jk}^{[l]}$ as well as the biases $b_j^{[l]}$ are actually calculated while training the model.

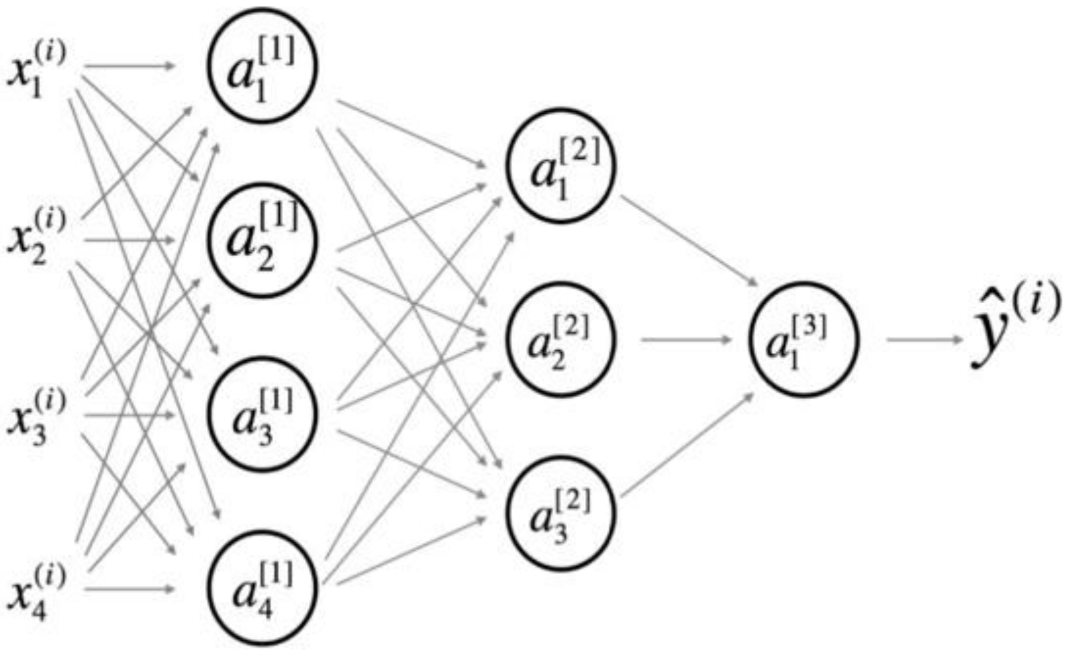


Figure 2.3. A typical NN containing two hidden layers.

2.4.1 Implementing Digital Images

We first convert a tensor with three channels (an order of 3) into one with a smaller order, meaning one (a vector) when you intend on using an ordinary network to handle a digital image. For example, consider an image of 100×100 pixels resolution stored in RGB format with 3 channels hence appearing as a vector of 30,000 elements while each element represents a single input feature. Building an NN model requires thirty thousand weight parameters for one node located at layer one. Therefore, it implies that if you want to employ larger images or insert additional nodes into the first layer then you will have increased number of parameters. This approach does not really work for image NN development and it is cumbersome. Convolutional neural networks take better advantage of the forms of input data to set an architecture using weights more effectively. CNNs capitalizes on two vital ideas to enhance network performance: handy interactions and shared parameters. In an ordinary nerual system, each output node aj[l] interacts with every input neuron $a_k^{[l-1]}$ whereas CNNs are sparse in terms of connections usually. This can be achieved by making use of filters having less size compared to the initial data. For example an input image may contain many pixels while filters consisting merely tens or hundreds of pixels can identify minor yet important characteristics like contours. Other techniques like a dropout layer can be used to improve performance and avoid over-fitting of data. This paper describes how each of these layers works as well as their configurations within the CNN system.

2.4.2 Convolution Layer

Convolution layers are the main computational elements of CNNs. A series of filters with learnable weights is included in each block. These filters are convolved with input from the previous layer to look for important patterns across the whole picture. An error function is minimized by designing filters in a certain manner for each network. In a CNN, a convolution operation is identical to a cross-correlation operation in two-dimensional signal processing. In Figure 39, we see the 2D image I of size 5 x 5 pixels and filter K of size 3 x 3 pixels being subjected to convolution operation.Applying the filter to one pixel at a time means moving one pixel at a time on the input image; this stride (i.e., one) makes the output image smaller than the input image. This is solved by placing zero pixels on the edge of the input image during convolution while using a filter (see Figure 2.4). The convolutional layer's output is calculated using the addition operation on the result of the convolution operation and a bias "b," which is then passed through an activation function “a". A formula; Conv(I, K)xy of a pixel’s convolutional layer in (x, y) coordinate is:

$$Conv(I,K)_{xy} = a\left(b + \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{d} K_{ijk} * I_{x+i-1,y+j-1,k}\right) \quad (2)$$

where h and w stand for the dimensions of the filter, whereas d corresponds to the number of input channels.

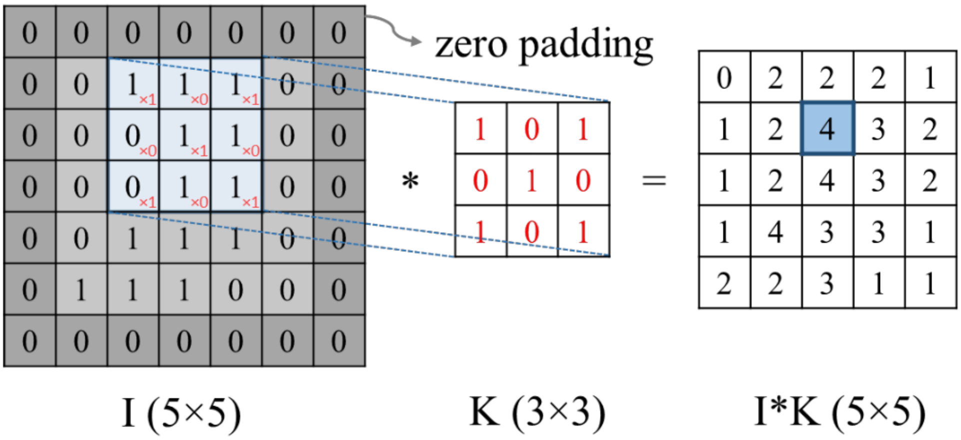


Figure 2.4. Convolution operation in CNN.

2.4.3 Activation Function

To introduce nonlinearity, it is important for you to include a nonlinear activation function in the network. Check the following diagram where the three activation functions are commonly seen in DL. In the early days of DL, many people loved the sigmoid function. Nevertheless, nowadays it is well known that tanh function outperforms it [20]. One issue with these functions is that their gradients vanish at the end points, making them stagnate. As a result, learning becomes drastically slow when a gradient-based optimizer is employed. In recent times, the Rectified Linear Unit (ReLU), which is non-saturating, has gained popularity as an activation function [21,22]. The use of this activation function has been found to increase network performance. In this study, we are using ReLU activation functions for all activation functions except the final layer of the network; it will consist of a softmax activation function to help classify input data. Softmax function si($\vec{x}$) of class i defines probabilities of input points belonging to each class, defined as:

$$S_i(\vec{X}) = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}} \quad (3)$$

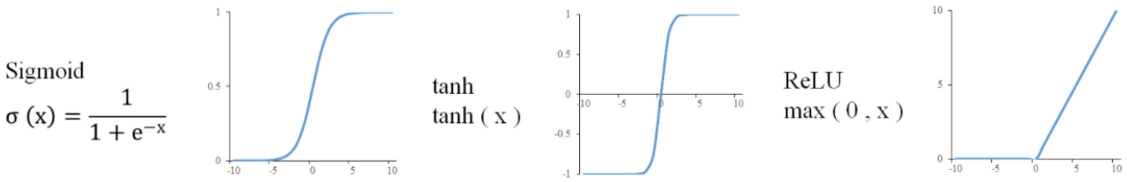


Figure 2.5. Activation functions in DL.

2.4.4 Pooling Layer

Pooling layers are mostly used by CNNs for reducing the size of the input layers so that computation is accelerated while at the same time increasing detection robustness. The most commonly used types of pooling are max-pooling and average pooling in DL. For image-like data, max-pooling has been shown to be far much better [23]. Every pooling layer in this study is a max-pooling layer unless otherwise indicated. In Figure 2.6, notice that with a 2×2 window and a stride of 2 the max-pooling mechanism is illustrated. As it goes through the input data, the highest value in the 2×2 window is selected. With each two-pixel shift of the 2×2 window, the whole input will be operated upon in this way. In this way, size of input data is reduced (in this example, the output data is half the size of the input data).

2.4.5 Dropout

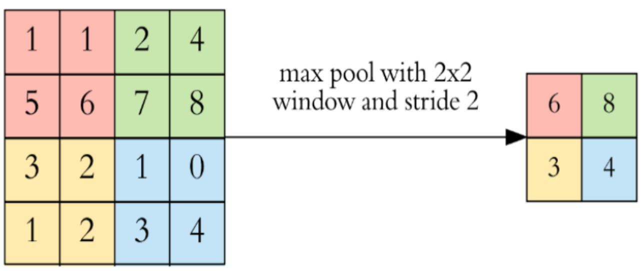


Figure 2.6. Max pooling mechanism[24].

Dropout could make neural net more flexible by applying diverse architectures or avoiding from overfitting-many various nets can combine into one net [25]. Actually, it means to randomly remove neurons in a NN("dropout"). Figure 2.7 illustrates the disappearing of temporarily their input-output links together with output layers themselves during dropout applied on neural networks (see fig. forty-two). As part of this analysis, the dropout method will be used on all the layers with 0.5 as the threshold probability rate [25].

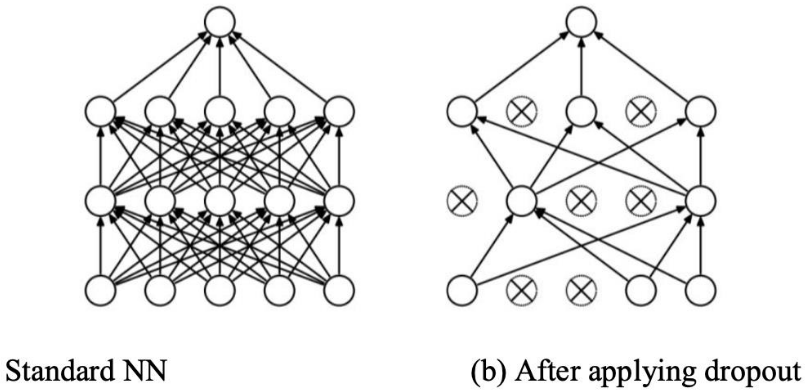


Figure 2.7. Net Modal Dropout Nerual

2.4.6 Cost Function

Training a CNN is essential for finding a group of weights and bias that minimizes mistake in prediction and actual. To numerically measure error, we would have to define loss functions. Categorical cross entropy (Equation 4) is the designated loss function Li that is used for estimating the difference between the true class y from the probability distribution over the predicted class ŷ of a single image. Through the use of the softmax function, it becomes easy to calculate the probability distribution of the anticipated class.

$$L_i(\widehat{y_i}, y_i) = \sum_{i=1}^{k} -y_i \ln \widehat{y_i} \qquad (4)$$

When it comes to incorporating image labels into neural networks, modelers use a one-hot encoding scheme. Two classes- one and two are represented in binary classification by (0, 1) as well as (1, 0). Consequently, the network model output is represented in the form probabilities for every class which are denoted by (ŷ1, ŷ2). This means that in our example case, if we have output vector (0.3, 0.7) then it implies that there is a 30% possibility that it belongs to class one while there is a 70% chance that this same vector belongs to class two.

In equation 4, the lost value is 0.36 given that the true class is one. To illustrate, (-0* ln(0.3)-1*ln(0.7)). is what the context is? For instance, poor prediction for the same example shall attract loss of 0.92 (0.6, 0.4) while good predictions like (0.05, 0.95) have such small losses as 0.05 The cost function, C, is merely a summation over the loss function L that has been applied to all images divided by their number, N.

$$Cost = \frac{1}{N}\sum_{i}^{N} L_i(\hat{y_i}, y_i) \qquad (5)$$

In order to add regularization into the model, it is necessary to insert the L2 regularization formula. This formula is defined as the sum of the squares of the weights on features and then is put into cost function together with its parameter.

2.4.7 Optimization

Broadly speaking, the question of learning is about optimisation. It follows therefore, that this optimisation aims at determining the most appropriate parameters that help to reduce the cost function, that is, weights and biases. In cases where neural networks are large, closed form solutions to their optimisation are not available so these are determined via gradient descent among other methods while using iterative algorithms. Common neural networks have non-convex search spaces, therefore it is logical to consider using a modified stochastic gradient descent algorithm. The cost function has millions of parameters in the proposed CNNs which should be fine-tuned. In this research work, we use Adam (adaptive moment estimation) to minimise the cost function. Adam is a first-order gradient-based optimisation algorithm for stochastic objective functions.

In the training phase, this optimisation algorithm will be administered as the optimisation algorithm [26, 28] since Adam optimiser is computationally efficient, with small amounts of memory required, invariants to gradient rescaling along diagonals, and suitable for huge data and/or parameter non-convex optimisation problems in ML [26,27]. An effective method of finding gradients of parameters through backwards and forwards application of chain rule on a computational graph is back propagation. When every forward pass is done, the expense function calculates; thus, depending on the output from such activity, besides inputs used as ground truths we can compute value derivative with respect to learning parameters by performing back propagation. Furthermore, this information feeds into Adam optimiser which modifies learning rates according to them. The computational parallelism gets quicker-due to vectorisation that is increased on Graphics Processing Unit (GPU) processors. Nonetheless, the computation will move slower since with a larger data set there is need for large memory to implement vectorisation. As a remedy to this problem, the training data is broken into smaller mini-batches [29]. Despite enlarger mini-batches offers more computational parallelism, smaller mini-batch training, nevertheless, tends to give better generalisation performance, as well as having a much smaller memory footprint that can be leveraged to increase the speed of machines used for this purpose [30]. According to Masters & Luschi, mini-batches with fewer samples lead to gradients being calculated closer to their current value thus giving rise for both stable learning procedures with less noise in them or simply put improving reliability of such systems [30]. Thus this survey is going to employ mini-batches of 32 images (N in Equation (5) will be 32 instead of the total number of images).

CHAPTER 3

LITERATURE REVIEW

3.1 Image Processing

A 2D function f(X, Y) is used to explain an image, where X and Y serve as spatial coordinates pinpointing a point's location within the said image and with the "f" value showing how intense a pixel is at this very point. All pixels as well as their intensity levels remain discrete and limited in number. Digital Image Processing, as described by Gonzalez[31], refers to the application of computer to manipulate digital images which is referred to as "The field of Digital Image Processing".
One controversial issue that is facing researchers concerning the boundary within image processing and related fields such as computer vision and image analysis. It is tough to tell the difference between computer vision as well as image processing. However, it should be understood that these computerized procedures can be divided into low-level, mid-level, and high-level processes.

1. Low-level processing refers to a having such as receiving or giving of images which consist of actions including noise suppression, difference amplification as well as image acuity.
2. Images are consumed as inputs by intermediate-level processes and then they generate image features, for example contours and edges.
3. High-level processing, akin to image analysis, encompasses advanced tasks such as recognition and object detection.

3.2 Traditional Methods for Crack Detection using Image Processing

In traditional maintenance systems, experts or technicians typically identify and evaluate cracks under expert supervision, a process that demands considerable time and labor. Consequently, there is an expectation that automated or semi-automated methods employing image analysis will streamline the process, leading to time savings and improved performance in assessing crack indexes and conditions. Numerous studies explore the automated identification of cracks on road surfaces [32, 33, 34, 35–38]. These investigations highlight the inherent challenge in crack detection, primarily stemming from images being riddled with noise and undesired elements such as shadows, dust, and paint lines. Furthermore, in practical survey systems, images are captured by a affixed camera to approprite vehicle traveling at particular speeds and encountering various environmental factors [33]. As a result, the quality of the images fluctuates, posing challenges for crack detection.

There exist two broad categories that encompass the techniques used in the detection, evaluation and differentiation of cracks These include those borne from traditional image processing and the ones that use new technologies such as neural networks plus machine learning. The next few sub-sections would take us through earlier solutions that employ edge detection, Gabor filters as well as image feature extraction techniques.

3.2.1 Edge detection

Image Processing (IP) techniques, such as edge detection, are commonly employed in detection and classification of cracks. Edge detection determines the edges on an image by using their intensity gradients[39]. Several methods which have also been based on Canny's methodology [40] have been proposed for detecting sites of cracks [41, 42]. Prior to utilizing Canny's edge detection algorithm, contrast of images is enhanced by Huili Zhao et al. [41]. Zhao's approach successfully identifies cracks and is more effective in eliminating dot noise than the Canny method. Agaian's [42] extended Canny approach with the fusion of two edge detection test images (AC test). To modify images with Overview of "Modified Canny kernel" or "Modified Canny kernel and Modified gradient Canny" an image processing technique that focuses on branch edge.

This study looked at what distinguishes crack from other objects in terms of seven different methods in an attempt to locate cracks through edge detection [43]. From these images, the performance of these various methods was compared across 30 pictures revealing that dynamic optimization results in improved performance. Nevertheless, experimental results obtained indicate that crack segmentation remains a challenge since no distinction can be made between actual cracks and paint-like features as shown by our tests on this subject matter. Wavelet transform can serve as another basis for edge detection [44–46]. Cuhadar et al.[47] used wavelet transformation to evaluate road condition. They applied this method in analyzing data from International Roughness Index (IRI) and different types of pavement conditions using wavelet transformation as a means for segmentation. Figure 3.1 illustrates a comparison between edge detection methods with various common filters and the a` trous algorithm across scales. The outcomes point to the fact that the at trous method has better performance in terms of noise removal when compared against other common filtering techniques in this context though they are largely influenced by scale.

3.2.2 Gabor Filters

Gabor filters conduct filtering of images through Gabor functions [48]. They are also employed for extracting feature images [49, 50]. Recognized for their efficacy in texture segmentation [51], Gabor filters find application in pavement crack detection and segmentation since images can be viewed as texture images [52, 53]. The output quality is dependent on the number of filters used in a Gabor filter bank design, as seen in Salman's approach to crack detection [52]. Salman employed various orientation filters as well as Gabor filters. Increasing the number of orientations enhances accuracy but also leads to longer computational times and a higher false positive rate. Despite its capability to detect the majority of crack pixels and effectively segment the crack line, this method remains susceptible to noise.

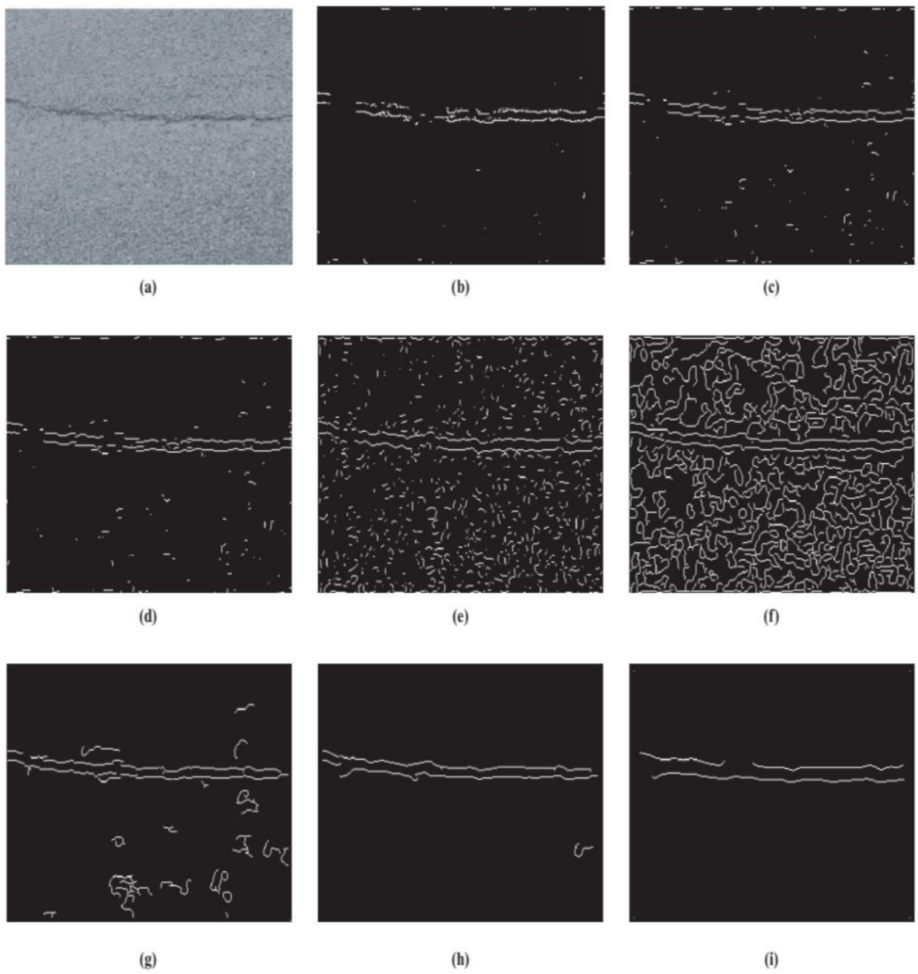


Figure 3.1. Result edge detection: (a) original, (b) Robert edges ,(c) Sobel edges, (d) Prewitt edges , (e) LOG edges, (f) Canny edges, (g,h,i) edge based a' trous algorithm with scaling 21, 22, 23 [55]

3.2.3 Adaptive thresholding

A lot of computer vision and graphics applications use Adaptive thresholding as a major technique [54]. This technique is notable for its versatility because it works by comparing a pixel with the mean of its neighboring pixels and ignoring low gradient differences. Bradley et al.[54] employed "integral image" tool, which is utilized in a face detection [56], to reduce the operations required while computing the average of a rectangular region within the image. Adaptive thresholding stands out as potent image processing technique capable of addressing variations in illumination across spatial regions and effectively eliminating noise. Fan et al.[57] employed a Deep

Convolutional Neural Network (CNN) model which identifies the regions containing cracks, subsequently applying an adaptive thresholding technique for crack segmentation from the regions identified as containing cracks within the image. This approach is characterized by its simplicity and speed. Adaptive thresholding also demonstrates high accuracy when applied to images with prominent cracks. Nonetheless, it is ineffective in eliminating dot-like noise, which is commonly prevalent in road image datasets.

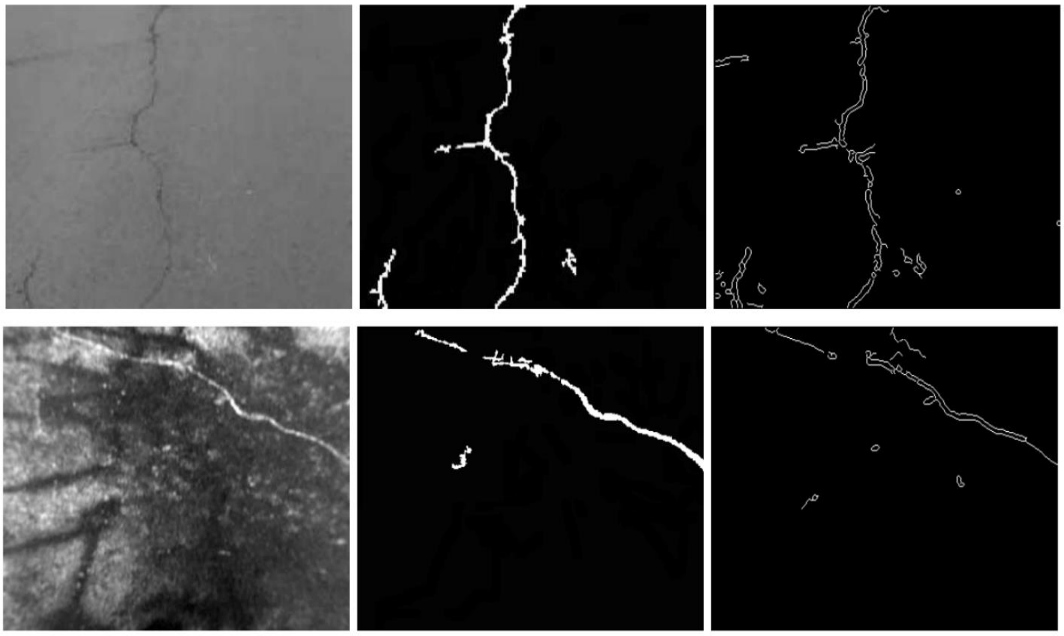


Figure 3.2. LBP feature based Crack detection[60]

3.2.4 Crack Detection Based on image Feature

For crack detection, one should consider utilizing image attributes like Neighboring Difference Histogram Model (NDHM) [61]. This formula determines inweight differences between possible dangerous pixels to adjacent ones, which are typical in most situations and locations. whenever the quantity of unsafe pixels within its surrounding rises in value, there is more chance that a given point could correspond to dangerous things. Li et al. [61] conducted a comparison between the proposed method with several thresholding techniques like Otsu [58] and Kapur [59], demonstrating the superior performance of the NDHM method. While this approach effectively highlights nearly all crack pixels, it also tends to detect noise, resulting in the presence of dots and salt-and-pepper noise in the output. Hu et al. [60] emphasises Local Binary Patterns (LBP) feature classification is utilized for pavement crack detection. The research involves extracting 35 features within the image, which are subsequently organized into six sub-classes. An LBP feature is associated with each block of the 3x3 partitioned image, which is then divided into a distinctive pattern so that in the end each pixel is either classified as cracked or not depending on the patterns that were identified. LBP feature and Canny detection method comparison appears at the top-left Figure 3.2. However, in challenging scenarios like the presence of paint in the bottom right location, both LBP features and Canny filter may incorrectly distinguish paint from crack.

3.2.5 Geographic Features

Geometric attributes have found application in crack detection across various materials [62] and specifically in road crack detection. Nguyen et al.'s proposal [63] defines a crack as a linear structure characterized by a Gaussian cross-section intensity profile. Following the enhancement of crack images using a Gaussian function-based filter, the center-line of the crack is extracted. The subsequent step involves eliminating spurious cracks, such as undesired edges. This approach requires certain pre-processing and post-processing measures to accurately discern genuine cracks.
A proposal by Median [64] employs a blend of 2D images and 3D data processing techniques to enhance crack detection, revealing detailed crack features. The Median method operates by extracting geometric information derived from road images. In this model, transverse cracks attain 96% balanced accuracy whereas longitudinal cracks attain 93%. Similarly, in previous suggestions in [38, 65], it was cracked attributes including the width of cracks, curve of cracks, or length of cracks which were used to determine whether pixels were or were not representing cracks.

3.3 Deep Learning and Crack Detection

Deep learning (DL) techniques have insignificantly demonstrated remarkable efficacy in addressing numerous practical challenges [66-69]. With a greater emphasis on automatic learning and reduced reliance on heuristics, LeCun et al.[70] illustrated the potential of constructing superior pattern recognition systems. The introduction of AlexNet in 2012 by Krizhevsky et al. yielded groundbreaking outcomes in an image classification competition (ImageNet challenge [71]), showcasing the capabilities of CNN architectures [72]. Following this, many researchers have utilized AlexNet and various other Convolutional Neural Network (CNN) architectures for damage detection in civil infrastructure. Cha et al. devised a conventional CNN specifically for detecting cracks in concrete and conducted a comparative analysis against edge detection technique by Canny and Sobel[73]. Their study utilized a training dataset consisting of 40,000 images with resolutions of 256×256 pixels, while testing was conducted on 55 images with resolutions of 5,888×3,584 pixels. They demonstrated that CNN outperforms in detecting concrete cracks under realistic conditions [73]. Cha et al.[74] employed Faster Region-based CNN to detect different types of destruction. Using FCNN, Huang et al. managed to detect crack and leak faults on the internal surface of underground tunnels made of concrete [75]. While using the Naive Bayes CNN approach, Chen and Jahanshahi identified cracks in video frames shot at separate nuclear energy installations [76]. The "Jackscrew" paper by Wang et al.[77] presented a CNN design targeting at crack detection in 3D asphalt surfaces but excluded pooling layers typical for a usual CNN that would ensure precision of pixel accurate. Meanwhile, in his research, Zhang[80] used a three-convolutional-layered CNN instead of more complex variants. An assessment of five hundred images taken by a low-cost smartphone with the resolution of 3, 264 x 2, 448 pixels was performed. The result showed that the deep learning (DL) system outperformed all other methods used to process them before. They collected 9,053 images showing various forms of damage on roads which were captured using smartphone cameras attached to vehicles; then, they used two different convolutional neural networks (CNNs) namely Single Shot Multibox Detector (SSD) Inception V2 and SSD MobileNet to identify where cracks appear in each photograph [81]. The German Asphalt Pavement Distress dataset was used by Eisenbach et al. to examine computer vision as well as DL-based crack detection methodologies. At the same time, Pauly and others further investigated the effects of additional CNN architecture layers on crack detection systems for pavement surfaces and examined how DL accuracy was affected due to varying positions of training data in comparison with the test sets [78, 79].

3.4 Crack Processing Using CNNs

3.4.1 Crack Detection

Deep learning comprises a range of machine learning techniques that rely on multiple layers of artificial neural networks. Within machine learning, neural networks find extensive application in crack detection and segmentation, boasting numerous advantages over traditional machine learning models [36, 37, 62, 82, 83]. DL models possess the capability to autonomously learn features within images, whereas traditional machine learning methods require users to manually design image features. Deep learning has demonstrated its capability to address subjective defects, such as minor product labeling errors, which are challenging to train for. In recent times, deep learning has emerged as a potent approach for tackling detection and segmentation tasks. Out of the 12 methods studies that aimed at crack detection, neural networks form the basis of six methods combining both unsupervised and supervised techniques [65]. Oliveira implemented two levels of crack detection: pixel-based crack detection and block-based crack detection. 90% approximate high precision comes with a require that is complex on pre-processing block-based before training and testing to detect cracks correctly using the technology [36, 37, 64, 84].

The research used pictures taken through the mobile phone from cracks in the pavement, which had differing colours far afield from those acquired through real systems usually for the fast moving cameras that are optimized to capture black and white images. In so doing, they presented an ICIP paper on CNN applications in ICIP proceedings [36] focusing on image processing, where deep neural networks such as convolutional neural network(s) served this purpose. According to Zhang, a good patch is suggested to be located within a distance below 5 pixels from the center of the fissure, while a bad patch does not have any fissures. The study compares the ROC curve of Support Vector Machine (SVM) [85], Boosting algorithm [86] and the one offered by the authors. The results demonstrate that the proposed method has the best AUC value (Area Under the receiver operating characteristic (ROC) Curve) which is equal to 0.845. Nevertheless, with an accuracy level of 89%, the F1-score that is being got cannot be considered sufficient enough as regards crack detection, demonstrating that detected cracks in outcomes are often wider than true ones.

In their approach [84], Maeda et al. explored various deep learning architectures and convolutional networks applied to the detection of road damage and cracks in images. They gathered a new dataset using a smartphone affixed to a moving vehicle, comprising eight types of road damage, including five types of cracks, rutting, white line blur, and crosswalk blur..

Fan et al.[37] used a model called the CNN on data from two public road cracks – the CFD and AigleRN . The CFD data set had about 65% positive pixels while the AigleRN had around 98.5% positive vs negative ratio without any modifications to the data sets, which made them skewed in terms of positive class sensitivity or specificity on the CFD dataset and sensitivity on CFD's. Crack images with corresponding labels were used during the training phase and the experiments showed that training with ground truths representing thinner cracks resulted in thinner output cracks [37]. The proposed architecture is composed of 9 layers as follows: two CNN layers each with 16 kernels followed by a max pooling layer after that there are two more CNN layers – one having 32 kernels and another having 32 kernels then we have three fully connected layers for this network. The CNN layer's kernel size should be this size: 3×3. The max-pooling layer's kernel size should be 2×2. Consequently, there exist papers that elaborate on applications of this kernel size in crack detection. Zhang et al. [36] implemented another model based on CNNs which made use of deep learning in detecting cracks. However, the resulting crack detection probability map proved that both the real crack pixels and several neighboring pixels were all given the same probability during crack detection. Hence the crack as detected seemed more enormous than it actually was; low crack coherence was observed, in a proposed system by Nguyen et al. [87] whose system also generated cracks far larger than the based truth. You Only Look Once (YOLO) is a method of object detection in real-time using minimal computational processing power [88]. This is because it divides the image into grid boxes and predicts their classes and the locations of bounding boxes as its primary function combining both localization and classification into single convolutions to increase speed. YOLO v2 was used to detect cracks and other damages on roads. Its overall F1 score was 0.87. However, detecting the area with cracks and not pixel-wise segmentation only was the focus of this analysis. RetinaNet [89] introduction aimed at detecting road damages based on deep learning. The backbone for feature maps in RetinaNet while using different neural networks during the process of learning. RetinaNet happens to possess some limitations one of which is to detect paint artifacts and shadow lines instead of cracks. To address this issue, a two-step model that can detect cracks quickly was proposed [90]. Cracks are first extracted from streets' images without changing sizes of original images in order to generate patches that have cracks. Then, these cropped patches are used to detect cracks by the second module. Indeed, this method achieved a recall rate and precision rate of 0.9521 and 0.9774. Nevertheless, the model presented serious problems with images in which cracks and road markings coexisted, or in cases of border cracks, as in Park…'s model. To sum up, for crack detection CNN models usually have input samples that are classified as either positive or negative. Crack pixels are found in the center of the positive samples while there are no cracks at all in the negative samples.

A common CNN architecture for detecting cracks incorporates 3-5 pieces of convoluted iceland, with kernels in each layer being the same or increasing from the beginning to the end. Furthermore, it uses max-pooling layers for parametric reduction. At the end, a fully-connected layer that has two neurons is used for determining whether or not input images should be grouped into categories - either as positive (that is cracked) or negative (that is not). All the above mentioned strategies in relation to detecting cracks usually work by samples. Instead of pixel classification level, evaluation accuracy is done at the image region classification level. Therefore, there have existed several segmentation methods for crack detection proposed to operate within the smallest unit of the image: a pixel. In the subsequent section we shall review a few methods for crack detection using pixels, or segmenting cracks.

3.4.2 Crack Segmentation

First, this section examines several papers employing CNNs for general object segmentation. Subsequently, it explores works specifically focused on crack segmentation. The concept of Region-Based Semantic Segmentation (R-CNN) was introduced by Girshick et al. In their study, they introduced a model capable of detecting and segmenting images based on regions with CNN features, termed as R-CNN (Region-Convolutional Neural Networks). R-CNN is created from three principal components: the first one is of a module that generates approximately 2000 region proposals without considering the category, the second one involves extracting a feature vector from each region by means of a big CNN, and finally there is a third module containing a Support Vector Machine (SVM) which classifies regions in a linear approach. Underpinned by bounding boxes for all proposed regions, R-CNN allows the network to break an image into parts corresponding to these regions; yet the results of this approach showed in Girshick's proposal point to several disadvantages in the design of R-CNN. The mAP of this model on VOC2012 (The PASCAL Visual Object Classes Challenge 2012) stands at 53.3%. While R-CNN calls for large memories and its training is slow, in contrast, during test time object detection is slow.

Fast R-CNN is a more advanced version of R-CNN that proposes sharing computation between different object proposals during the forward pass of each convolutional layer. Two major differences are apparent in the architecture of Fast R-CNN. First, regions of interest (RoIs) are used as the pooling layer input for feature extraction from the feature map and not the whole image. In conclusion, there are two sibling layers output from Fast R-CNN. The first layer is used to assign each input to an object class while the second one refines bounding boxes on object instances. For Faster R-CNN, training and testing times are reduced compared to R-CNN and SPPNet. Nonetheless, selective search computations are still necessary to accelerate the training process.

Mask R-CNN, an extension of Faster R-CNN, was introduced by He et al. [91] with the objective of achieving pixel-level segmentation [92]. In addition to the already existing bounding box recognition branch, Mask R-CNN also incorporates a mask component. Furthermore, key features of Mask R-CNN have been introduced by He et al, of which RoIAlign is one example – it is an alignment technique that works pixel-to-pixel. RoIAlign addresses the limitations of the RoI pooling layer by eliminating harsh quantization and ensuring proper alignment of extracted features with the input. Unlike previous R-CNN versions that only provide bounding boxes, Mask R-CNN precisely identifies the pixels of each object. Experimental results demonstrate that Mask R-CNN outperforms earlier R-CNN versions in terms of mAP. Despite this, the mAP of Mask R-CNN for small objects is still not relatively high, which shows that it is not appropriate for small objects segmentation.

Fully Convolutional Networks introduced by Long et al. for semantic segmentation [93], utilize knowledge transfer from VGG16 [94]. Fully connected layers in VGG16 are changed into fully convolutional layers using a 1x1 convolutional layer in the FCN architecture. By doing this, the classification network can now provide a low-resolution heatmap. The sub-sections of the upscaling part are implemented with transposed convolutions which use semantic primitives represented with low resolution. Again, it should be noted that at each phase the up-sampling is fine-tuned again. FCN has the flexibility to accept inputs of arbitrary size and produce corresponding-sized outputs. Moreover, it excels in extracting spatial information of objects for semantic segmentation, yielding a higher mean Intersection over Union (IoU) compared to other segmentation methods like R-CNN. However, FCN's complexity and long training duration are attributed to the large number of kernels in its convolutional layers.

Originally, U-Net was initially developed for biomedical image segmentation [95]. However, it is now used for numerous other image segmentation jobs. Its architecture comprises two components: a contracting segment for feature computation and an expanding segment for spatial pattern localization within the image. The contracting segment employs max-pooling layers to reduce parameters and shrink image dimensions, while the expanding segment employs up-sampling layers to enlarge the image size. As a result, the output image maintains the same size as the input image. Within the U-Net architecture, concat layers are utilized, which merge feature maps at the same level from both segments, enhancing object localization accuracy. This network incorporates a total of 23 convolutional layers. Ronneberger's approach emphasizes two key strategies: Overlap Tile and separation of touching objects. Overlap Tile allows the U-Net model to predict the entire image in parts, while the separation method encourages the network to discern small boundaries between adjacent cells. With these strategies, the U-Net model achieved a high IoU score of 0.9203 in the ISBI cell tracking challenge.

SegNet is an encoder-decoder architecture designed for image segmentation [96], utilized in various scenarios such as indoor and outdoor scene prediction. Its encoder network mirrors the initial 13 layers of the VGG16 network [94], generating a series of feature maps. Each encoder is paired with a decoder responsible for up-sampling the feature map. A notable feature of SegNet is its storage of max-pooling indices only, effectively reducing parameter count. Comparisons demonstrate SegNet's performance equivalence to FCN while consuming less memory. Notably, SegNet excels in boundary delineation, surpassing other methods in most scenarios.

XNet, a CNN tailored for medical X-ray image segmentation, adopts an encoder-decoder architecture commonly seen in segmentation tasks [97]. Despite training on a relatively small dataset comprising 108 images across 10 body parts, XNet achieves an impressive overall accuracy of 92% and an AUC of 0.98. Unlike many segmentation networks, XNet incorporates a compact serial down-sampling module within its encoder architecture. Additionally, Bullock et al. integrate two encoder-decoder modules to enhance feature extraction while preserving image resolution. Across all evaluation metrics, including F1-Score and AUC, XNet consistently surpasses the 90% mark when applied to three X-ray image categories. Comparative analysis against SegNet demonstrates XNet's superior F1 score across all categories, coupled with a reduced parameter count.

DeepCrack uses a new methodology that uses a deep hierarchical neural network for crack segmentation [98]. A hierarchical CNN based on pixels for cracks was proposed by Liu et al. Instead of applying all of the layers to VGG-16, DeepCrack neglects fully connected layer and its fifth pooling layer in order to produce auxiliary outputs at multiple scales which are significant hence cutting down on memory footprint as well as computational costs. Incorporating a guide filter inspired by guided feathering [99], DeepCrack refines predictions and eliminates noise in low-level predictions. The dataset utilized comprises over 500 images sourced from both internet downloads and authors' captures, exhibiting variations in spatial resolution and characteristics, posing challenges for real-world applications. DeepCrack achieves a mean IoU of 85.9 and an F1 score of 86.5%.

In overview, the outlined image segmentation architectures commonly exhibit several shared traits: They typically feature a backbone within the network architecture, often comprising two symmetric segments: encoding and decoding, or contracting and expanding. Certain approaches leverage pre-existing networks like VGG16 [94] for the encoding segment, while introducing novel architectures for the decoding segment. Concatenation layers are frequently employed across these methods. The techniques utilize a convolutional 1×1 layer for channel-wise pooling, often referred to as a projection layer or feature map pooling. However, while crack detection methods are criticized for detecting larger areas containing cracks than the actual crack size [36, 87], crack segmentation methods face challenges due to road images varying in resolution and containing numerous artifacts, notably noise that is difficult to eliminate. Consequently, recent approaches have adopted a two-stage architecture to address both crack detection and segmentation within a single framework. The subsequent section will delve into reviewing these novel proposals.

CHAPTER 4

PROPOSED ARCHITETURE

4.1 Introduction

The index of cracking serves as a measure of the damage resulting from the separation of segments on the runway surface, typically manifested as cracks. Crack detection involves examining and pinpointing cracks on a runway surface to assess airport runway conditions and plan maintenance activities. Manually executing crack detection relies on human-vision or cognition, semi-automatically or fully automatically through computer vision. Manual inspection demands expertise and is both labor-intensive and time-consuming. Automatic methods for detecting cracks in images had been devised to enhance operating speed and achieve performance surpassing humans [65]. This represents a formidable challenge in computer vision and image processing, attracting research attention for decades [100, 33, 34, 36, 98]. In this chapter will introduce an deep learning method for automatically detecting and segmenting road cracks to assess runway conditions. However, the subjective aspects involved in the actual evaluation of the crack detection challenges for automation, making it an ongoing area of research independent of this thesis. This chapter concentrates to achieve precise pixel-level segmentation of cracks through deep learning techniques, having aim of eventually facilitating the automated computation of the cracking index. Shifting from manually measured dimensions of crack and density to semi-automatic computation of crack positions and networking through image processing and deep learning holds promise for substantial enhancement in crack evaluation. Semi-automatic detection methods offer greater efficiency for surveys compared to manual inspection [35, 65].

Certain previous methods concentrate solely on either region-level detection or pixel-level segmentation, aiming to optimize either crack detection or crack segmentation performance independently. However, by focusing solely on one aspect, these methods fail to address the problem comprehensively, leading to an overall reduction in performance. Specifically, obtaining significant results proves challenging, especially when dealing with demanding data types like noisy images with subtle crack features and imbalanced datasets. Numerous methods have been proposed for detecting and segmenting cracks in runway images. Traditional image processing (IP) techniques like the Canny edge detector [40], as well as approaches utilizing Gabor filters [48], leverage variations in pixel intensity to delineate edges, treating cracks as features that are responsive to edge detection filters. Nevertheless, these methods are sensitive to numerous images with small details, which limits their ability to effectively reject noise. Moreover, determining the optimal parameters needed to balance noise removal with preserving weak crack features varies from one image to another.

In contrast, deep learning techniques, particularly those employing neural networks, have gained significant traction for object detection. CNNs stand out as one of the most robust recognition techniques. While various research employs CNNs for crack detection [36,37], others utilize CNNs for pixel-wise segmentation of cracks in images [98].

During the training phase, small sections of images are utilized for training, where positive inputs encompass sections with cracks and negative inputs comprise sections without cracks. The model is deployed for crack detection once it surpasses its training. The section of images used during the training phase for detection systems typically represent small segments of images, with positive inputs negative inputs featuring crack and non-crack images. The results entails a binary decision regarding the existence of a crack in the input image. Nguyen et al.[87] developed crack detection CNN model, highlighting its capability to eliminate nearly all noise and artifacts in the actual image, which is comparably large at 750 × 1900 pixels, while effectively obtaining crack in image patches. This model lacks in the localization of detected cracks which are not as precise as the ground truth. Zhang et al.[36] introduced CNN which results a high score but encounter with the issue that the cracks detected are higher with respect to ground truth cracks.

Image acquisition can be achieved through various means such as a camera affixed to a vehicle, a dedicated drone, or a smartphone. Two distinct stages, employing CNNs, are trained using these samples: initial stage is for detection following the second for crack segmentation. Existing studies have primarily concentrated on either crack detection or crack segmentation individually. However, in this chapter, both detection and segmentation are integrated into a single framework using a deep learning approach. The proposed framework demonstrates significant performance enhancement, particularly when referred to imbalanced datasets containing cracks, where the crack pixels is considerably smaller than non-crack pixels in the image.

The primary focus of this chapter include: Introducing a two-stage architecture with CNNs, tailored for handling noisy, low-resolution images, and imbalanced datasets. The performance of model surpasses that of combining previous detection and segmentation methods in the formation two-stage approach. This underscores how specially designed two-stage appraoch maximizes the benefits of a detection and segmentation paradigm.

4.2 Two Stage Approach

For the majority of the positive training examples, a crack is generally located at the centeriod. Even though the center position tends to contain cracks when testing with part of the cases, it is still worth noticing that a few of these samples will also be detected as positives. All cracks found by regression may have their boundaries extending into other areas adjacent thereto plus some percentage/portion overlap of those regions' pixel values which are next closest outside these boundaries. Hence, the size of training samples plays a crucial role. Smaller input region samples tend to result in better localization of detected cracks relative to the actual cracks. In the event of low-resolution images, distinguishing adequately between positive and negative samples in the crack region sample should require it to be quite sizeable; especially so for faint cracks as well as crack-like noise. The next section will describe the integration of two stages involved in crack detection and segmentation. A CNN is first used as a detection model, where it has been trained to recognize all the cracks in image patches while at the same time eliminating artifacts and noise. In the subsequent stage, the crack is segmented at the pixel level within smaller patches rather than the entire original image. Consequently, the amalgamated model inherits the benefits of both detection and segmentation methodologies.

Crack Detection: In order to increase useful features and focus on inconspicuous crack features, various convolutional layers are employed which are responsible for different aspects representation of the image such as form, border or brightness of the images. The desired features of the crack were derived by a five-layer CNN model in this work. Empirical justifications are taken into consideration while choosing the layer count. This demonstrates that, for the specific crack detection problem under investigation, a five-layer CNN architecture yields the highest performance. Padding was not employed in the CNN layers of the detection stage, implying that the image samples processed by the CNN layer did not receive extra padding during convolution through the kernel. By shrinking each layer, we reduce the number of parameters in them, and then attach a max-pooling layer of kernel size 2×2 behind each CNN layer. To pick the highest value in every 2×2 block, max-pooling function is applied splitting down further on every level of convolution network thorough diminution. Number of weights is decreased and overfitting is avoided by max-pool at same time.

In CNNs, Fully Connected layers are typically positioned at the end of the model which gather and consolidate every features learned in preceding layers. For this task, where images are categorised into two classes—positive (crack) and negative (non-crack)—two neurons are employed in the final Fully Connected layer.

The process of combining features is accomplished through two Fully Connected layers. The first layer comprises two hundred neurons, facilitating the flattening of features and build a vector out of them. This vector consolidates all the feature information and significant elements from the preceding CNN layers. The second Fully Connected layer encompasses two neurons, aligning with the crack and non-crack classification.

In order to determine whether there exists a crack or not, the samples require estimation during the final step. To identify crack inside the picture, which eventually leads to a range of 0 – 1 depicting chances that one may have no cracks present or some other number other than zero, a softmax function will be used at the output layer.

In this proposed method, a max-pooling layer is employed following each convolutional layer. This design choice aims to accelerate the scanning of the input image compared to models without pooling layers , while also mitigating the risk of overfitting. The implementation of max pooling involves a straightforward and efficient algorithm. By appending a pooling layer after each layer, the training duration is shortened thus parameter count is reduced, and overfitting is regulated.

After each layer, a technique known as max-pooling is used to lessen the weights thus avoiding overfitting in this model. The ReLU function serves as the activation function in the convolutional layers and first fully connected layer. This choice is due to its common application in CNN classification models and its ability to accelerate training time .IIn the last layer, a Softmax function is used in a fully connected layer of two neurons. This setup makes it possible to determine the class that an input image belongs to; either crack or non-crack. This setup leads to identification of the input image type either as crack or non-crack.

Crack Segmentation: The suggested CNN architecture that is divided into two essential parts; one known as contraction part that is the encoding part while the other is expansion part which is regarded as the decoding one. Such an architecture employs encoder-decoder model that is common in image segmentation. Accordingly, they are ideal with regard to the semantic type of tasks that involve segmentation of cracks in image data.

The contraction phase is made in such a way that it seeks out for objects' feature in images. This phase uses a 5-layer CNN model to extract features. Contraction phase conception is steer by an architecture that increases the number of filters in every layer, as is the case in established segmentation and object detection techniques. Since the network is tailored for segmentation, its objective is to identify and categorize objects not just at the block level but also at the pixel level. Consequently, the number of kernels employed differs from previous work [87]. Unlike U-Net, which features a substantial number of filters in every layer, our approach gradually reduces the number of kernels in every subsequent layer, halving it.

Padding is included in each layer's output to ensure that all samples receive additional padding when convolved with the kernel. Moreover, after each layer, a max-pooling layer does decrease the number of weights and sample size.

The expanding segment serves as counterpart to the contraction phase. While the contraction phase is employed to extract features, the expansion phase is utilized for spatially localizing patterns within image smaples. This architecture features a symmetric structure for both components decoder and encoder. It can be seen that the growing part acts similar to a deconvolutional network, which is an inverse counterpart of the convolutional network, as it expands the input data until it produces a bigger image. Furthermore, upsampling is conducted having size of 2×2 to retain the dimensions to those of the samples. In this design, 1×1 kernel layer acts as a sigmoid activation function, helping to process feature maps resulting into a segmentation map; four concatenation layers on the other side combine two layers at a given axis for increased information flow through various levels in the network.

4.3 Overall Architecture

This section introduces the proposed model, which combines aspects of YOLOv5 and the Faster R-CNN model, operating in two distinct phases: Detection and Segmentation.

4.3.1 Detection and Noise Reduction Using YOLO

Initially, the YOLOv5 (You Only Look Once version 5) model is primarily employed for detecting cracks or damage on runway surfaces. The approach encompasses the following steps:

1. Training on Image Patches: During the training phase, the YOLOv5 test sets the image patches containing runway sections with and without cracks, thereby providing

discriminating details of the crack that will help differentiate it from other crack features.

2. Categorization and Detection: In the Detection stage, the YOLOv5 model processes the full runway images and determines sections with cracks and none. Object detection has been a popular technique among experts because of its speed and accuracy, which is one reason the YOLOv5 design framework permits it to tag regions early.

3. Background Noise Reduction: YOLOv5 is not only a way to detect objects; it is also a way to clean up pictures by eliminating all background noise and unnecessary stuff. This stage ensures that only natural cracks in runway images are examined during detection.

The YOLOv5 model is capable of finding and detecting the damaged spots in the runway images correctly. Several crack regions, which have been identified, form the results of this stage and are ready for more extensive examination.

4.3.2 Pixel-Level Segmentation of Runway Cracks

Once the detection phase is over, the model moves on to the segmentation stage where a more detailed analysis is done. During this stage, the focus is on identifying the exact location of each crack within the defined areas at pixel level.

1. Localization of Detected Areas: In the primary phase, those sections that are said to have cracks are chosen out for further investigation. This is how the customization makes it easier for the computer program to concentrate more on the mentioned specific regions hence improving on how it works better while dividing the image into distinct pieces.

2. Pixel-Level Segmentation: On these localised areas, well-thought segmentation methods are used to differentiate crack pixels from those that are not. To define the size and extent of an individual crack, the segmentation process examines pixel intensities, textures, and other characteristics to precisely delineate the shape and scope of each crack.

3. Schematic Representation and CNN Layers: Each CNN layer in the YOLOv5 architecture is followed by a max pooling layer, as shown in the schematic representation of the proposed model (Figure 4.1). The role of the convolutional layers is to extract features whereas max pooling layers help to reduce spatial dimensions and identify important features detected by the convolutional layers.

   The proposed model integrates detection and segmentation thereby offering a comprehensive approach that not only identifies whether there is a crack or not but also produces detailed segmentation maps indicating the exact location and shape of cracks. By following a two–stage procedure, it is anticipated that the detection rate as well as characterization of runway cracks will be more accurate leading to improved maintenance practices.

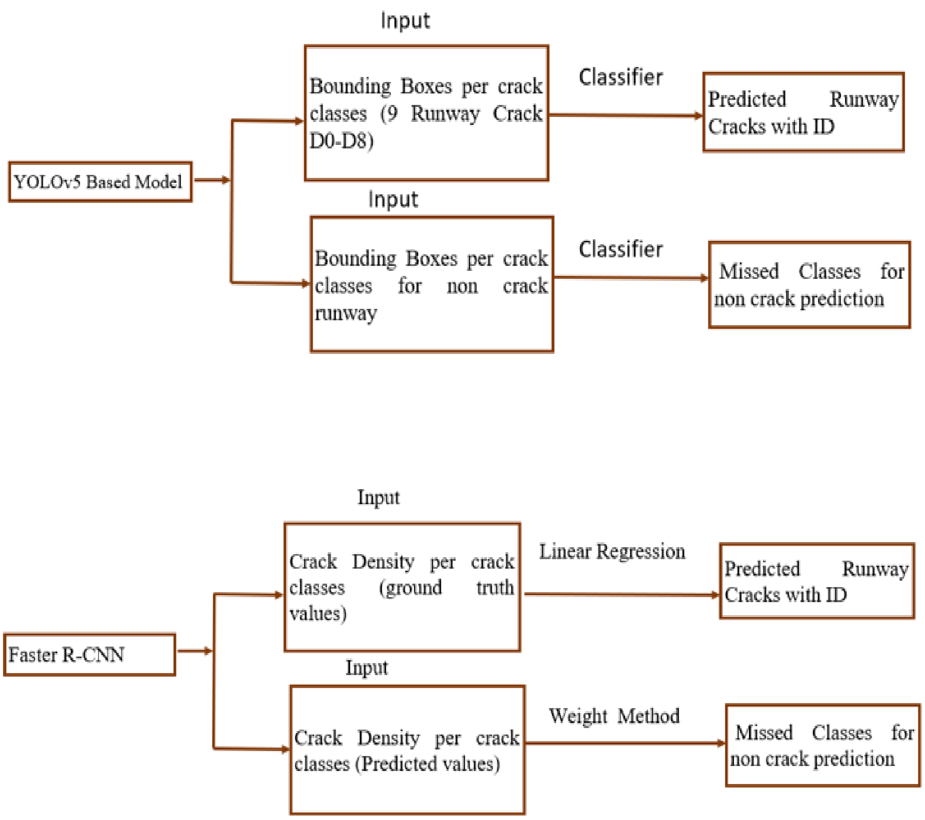


Figure 4.1. Block Diagram of Proposed Model for Detecting Road Cracks and its Segmentation

4.3.3 YOLO

The YOLOv5 model uses deep learning for recognizing different types of cracking automatically using a framework [101]. YOLO stands for “You Only Look Once” and it can be thought of as a relatively new object detecting technique that has gained much popularity due to its ability to achieve high accuracy when applied within systems based on deep learning techniques. Table 4.1 enlists the details of CNN architecture and shows the series of layers along with each kernel size, strides with the output shape of pixel for model implementation.

Key Features and Efficiency of YOLOv5:

Single-Pass Object Detection: The approach of this single pass contrasts with that which might require a number of passages over the same image for identification and classification of objects, hence optimizing object detection by the YOLOv5 model.

Reframing Object Detection: When working with YOLOv5, a single regression problem is believed to be reframed as an object detection problem. It involves predicting bounding boxes and class probabilities from full-images directly in a single evaluation; hence making detection process very fast and efficient

Table 4.1 . CNN Architecture for the Proposed Model

#Layers	Kernel-Size	#Stride	Output shape
Input			[416, 416, 3]
Convolutional Layer	3x3	1	[416, 416, 16]
Max pooling	2x2	2	[205, 205, 16]
Convolutional Layer	3x3	1	[205, 205, 32]
Max pooling	2x2	2	[104, 104, 32]
Convolutional Layer	3x3	1	[104, 104, 64]
Max pooling	2x2	2	[52, 52, 64]
Convolutional Layer	3x3	1	[52, 52, 128]
Max pooling	2x2	2	[26, 26, 128]
Convolutional Layer	3x3	1	[26, 26, 256]
Max pooling	2x2	2	[13, 13, 256]
Convolutional Layer	3x3	1	[13, 13, 512]
Max pooling	2x2	1	[13, 13, 512]
Convolutional Layer	3x3	1	[13, 13, 1024]
Convolutional Layer	3x3	1	[13, 13, 1024]
Convolutional Layer	1x1	1	[13, 13, 35]

Integration of CNN Classifiers: There are Convolutional Neural Network (CNN) classifiers in the model who aid in extracting characteristics and classifying photos, which is critical. Convolutional neural nets are proficient enough to recognize spatial hierarchies in images; therefore, they are suitable for detailed object identification tasks.

Simultaneous Prediction: An ability of YOLOv5 is its stand–out feature allowing prediction of several class probabilities along with bounding boxes at once. The simultaneous prediction allows this model to process multiple objects in an image without sacrificing speed at high accuracy levels

Architectural Diagram and Functionality: Figure 4.2 illustrates the architectural design of the YOLOv5 model. The architecture is designed to streamline the object detection workflow through several key components:

Input Layer: The relevant features are extracted by the subsequent layers that process the input image.

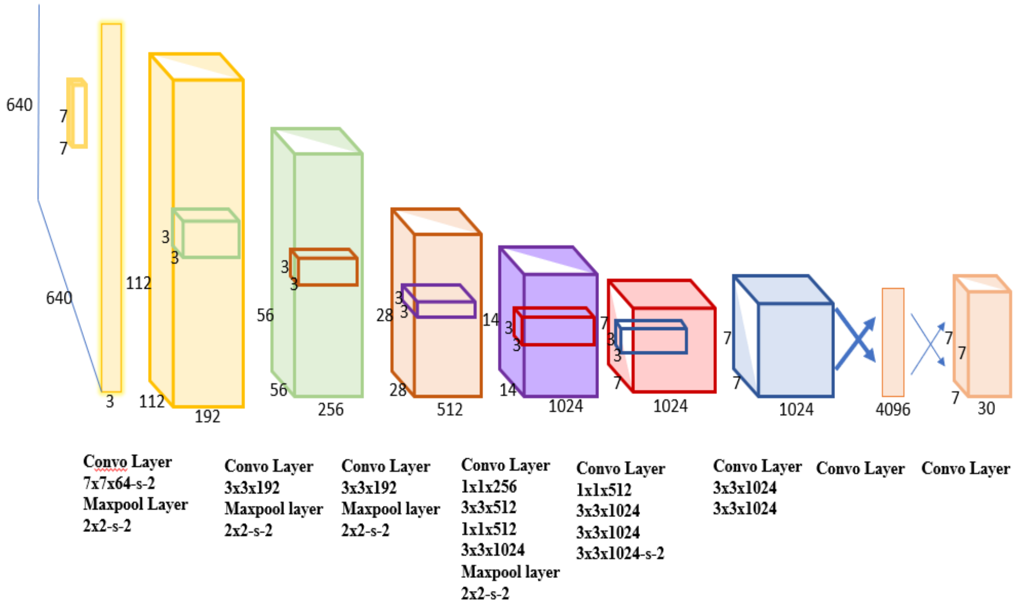


Figure 4.2. YOLO v5 Architectural Diagram

Convolutional Layers: For feature extraction, multiple convolutional layers are utilized. Those layers apply different filters to the input image to detect edges, textures and other important patterns which mean cracks.

Max-Pooling Layers: For each convolutional layer, following one another max-pooling layers are employed to lower the spatial dimensions of feature maps helping in pointing out significant features while keeping computing efficiency in place.

Bounding Box Prediction: In this scenario the application is predicting locations where it can say that cracks are found; within those locations a range of choices are mentioned for accuracy: likelyhood concerning cracks' varieties as classes.

Output Layer: Finally, it submits the detections results represented as the bounding box coordinates and associated class probabilities.

4.3.3 Faster RCNN

The model Faster R-CNN follows a crack-guessed, two-stage detection technique that embeds three major components in it: Feature Extraction Classification (FEC), Location Refinement (LR), and Informative Region Selection (IRS). Through this well-thought-out structure, the system can effectively identify any form of cracks on an image with great precision and detail. Here's what happens:

1. Informative Region Selection (IRS)

Objective: To find and select regions in the image that are potential to carry cracks.

Process: Image Segmentation: Commence by segmenting the complete crack image into smaller portions or regions of interest (ROIs) is the first step. This reduces the size of those areas in question areas that need a closer investigation.

Region Proposal Network (RPN): The RPN is a tool which helps to give areas on the image which are likely to have cracks the feature detection stage coming up with categories of objects. It looks at the whole image and finds regions of crack-prone through some mechanisms..

2. Feature Extraction Classification (FEC)

Objective: In order to extract relevant characteristics for the proposed regions, and use it to identify cracks.

Process: Convolutional Layers, A range of convolutional-filter layers are applied to each segment or candidate area. With the assistance of these layers, diverse filters are utilized for the purpose of extracting significant attributes that are connected to carck like textures, edges and patterns

Feature Maps: The significant features detected in each segment are highlighted by the feature maps produced by convolutional layers.

Classifier: A classifier is employed to process the extracted characteristics and decide on the possibility of cracks. The task of this classification is to determine if there are any cracks present, and identify what type they may be (e.g., longitudinal or transverse).

3. Location Refinement (LR)

Objective: To refine the location and bounding boxes of the detected cracks for precise localization.

Process: Bounding Box Regression, The proposed regions have their coordinates refined through bounding box regression algorithm in the model. This way, the bounding boxes are readjusted in order to better enclose the detected cracks, hence improving the positioning accuracy.

Output: The final output includes the recognized regions with their respective types of cracks as well as the exact coordinates of bounding boxes which have been refined. This accurate localization is very important in order that more detailed examinations followed by repairs may take place.

CHAPTER 5

EXPERIMENTAL EVALUATION

5.1 Implementation Details

In this section, detailed information on how the entire setup is implemented is provided. The sections have practical examples to help in understanding the data used in the new model. Also, section 4.2 provides information on how hyperparameters were selected during training.

Dataset Collection : The explanation above highlights the collection of new data set used to train the model on various runway crack types. It is inclusive of various images depicting different types of runway cracks as well as their conditions in the images; these have been carefully chosen with the intention of capturing many possible scenarios so as to ensure precise labeling which makes training more effective.

Hyperparameter Selection: Optimal hyperparameters selection for model training would be discussed. The hyperparameters optimized were like learning rate and batch size; it also included tuning the number of epochs as well as fine-tuning other architecture-related ones. Explicitly, this part describes what specific values have been chosen in these hyperparameters ’set-up and provides insight into model precision as well as costs issues.

Experimental Setup: We conducted experiments on a device with the hardware specifications mentioned here: - Processor: AMD Ryzen 5 5600H, - Graphics: Radeon with a 3.30 GHz speed - RAM: 8 GB - GPU: NVIDIA GeForce RTX

These hardware components enabled the model to carry out the complicated tasks in training and validating the deep learning algorithm. Selecting a high-speed graphics card such as NVIDIA GeForce RTX was essential in hastening the process of training due to the large volume and intricacy of the data set.

5.2 Training and Testing

In the developing phase, we introduced the model to the freshly accumulated data which she used to understand different patterns and characteristics related to runway breaks. Multiple iterations were made during training where the model’s performance would be continuously reviewed for improvement purposes using selected hyperparameters.

The efficacy of the model was appraised during the testing phase by means of employing another subset of the same dataset. Such an evaluation allowed us to determine how well this model could identify or characterize various types of linear damages within images that had never been observed before.

There is a comprehensive overview of the entire setup inclusive of dataset collection and hyper parameter search until the exact hardware used in conducting experiments is presented in finer details in this section. In this way, it is guaranteed that the model will be able to detect and segment cracks on runway robustly while being efficient too.

5.3 Dataset Description

In general, figure 5.1 shows pavement cracks are categorized into nine types as shown below

1. Reflective Runway Crack
2. Transverse Runway Crack
3. Block Runway Crack
4. Longitudinal Runway Crack
5. Alligator Runway Cracks
6. Sealed-Reflective Runway Crack
7. Lane-Longitudinal Runway Crack
8. Sealed-Longitudinal Runway Crack

ARID Dataset: To implement crack detection on airport runway, we present a novel dataset called ARID (Airport Runway Image Dataset), consisting of 8,228 images gathered from 10 different airports within India. The images were taken by an iPhone 11 camera which had the following specifications: * **Main Camera**: 12 MP, f/1.8, 26mm (wide), 1/2.55", 1.4µm, dual pixel PDAF, OIS * **Secondary Camera**: 12 MP, f/2.4, 120˚, 13mm (ultrawide), 1/3.6”
Moreover, the Google API was used to automatically obtain distress surface images with the help of GPS coordinates, in addition to camera and image parameters. Image capture was enabled by choosing the starting and ending points for every marked runway.

Image Capture Process: We captured various snapshots of the same fractures located in specific positions through cameras inclined at -60° and -90° from the normal, in order to enable a right classification of pavement imperfections. All pictures were resized to 640 x 640 pixels because of uniformity.

Annotation: A software annotation tool was used to annotate wide-view images in such a way that they show all the nine different types of runway cracks (named D0 to D8). Therefore, by annotating this dataset, it was made sure that every picture got correctly labelled for training and testing.

Dataset Composition: A set of two sets makes the data
Training set: 5760 images
Testing set: 2468 pictures.
The ARID dataset provides a whole range of annotated images showing cracks on runways from different airports. As a result, it becomes a useful source for which other models can be trained and tested for the detection and classification of these types of anomalies. Real life applicability guarantees high correct rate and trustworthy models given that it employs high resolution photos with carefully made notes.

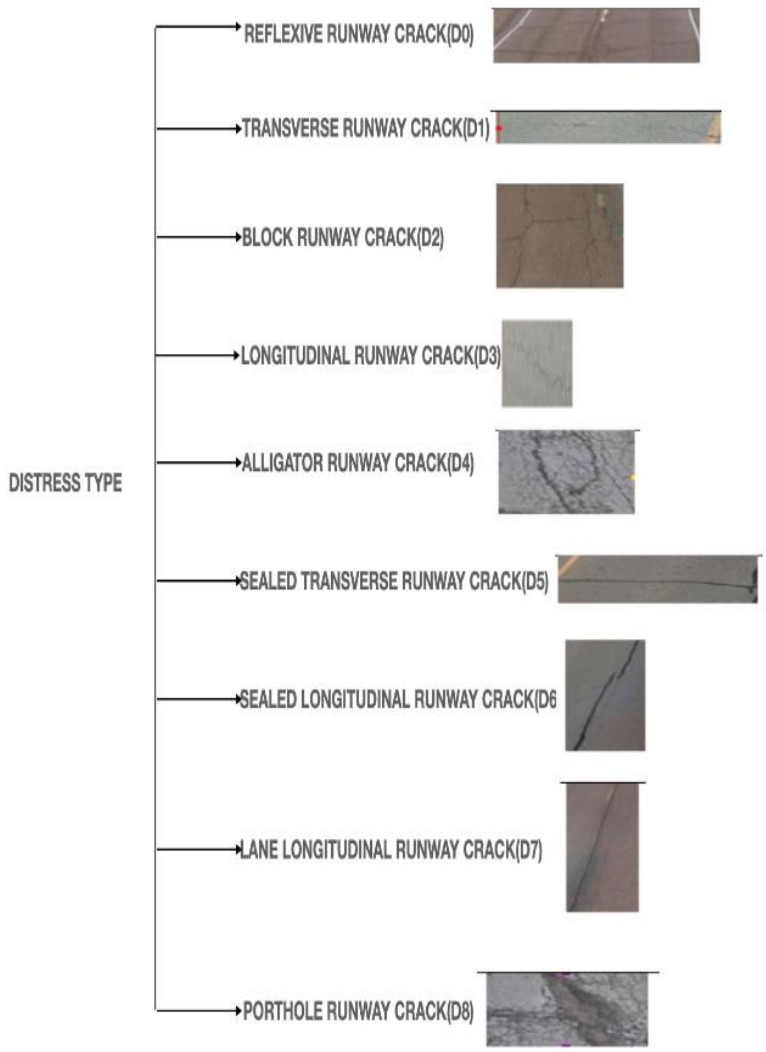


Figure 5.1. Airport Runway Distress Crack Types With its Crack ID.

5.4 Performance Metrics

This part will include the metrics for the evaluation of runway crack classification and detection performance, among them precision, recall, and F1-score. Here are the definitions of these metrics:

$$Percision = \frac{tp}{(tp+fp)} \quad 1)$$

$$Recall = \frac{tp}{(tp+fn)} \quad (2)$$

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall} \quad (3)$$

where:
tp (True Positive) indicates the number of correctly detected cracks.
fp (False Positive) indicates the number of incorrectly detected cracks.

fn (False Negative) indicates the number of undetected cracks.

5.4 Model Training and Evaluation

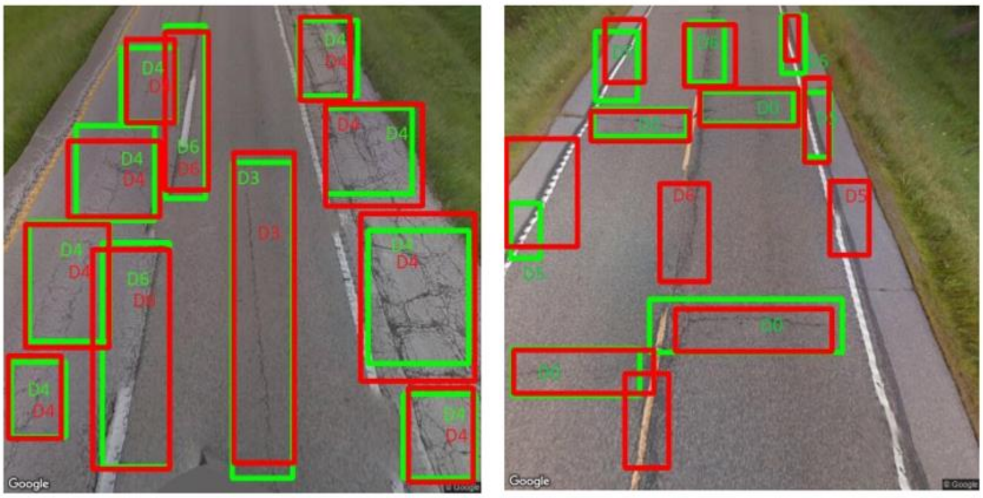
The model proposed for training had a total of 5,760 images used in its training while 2,468 others were used to evaluate it. What happened next was that the training took 20,000 iterations within the space of 10 epochs and this meant the learning rate had to be set at 0.01.

Accuracy Estimation: To estimate accuracy, we calculated overlaping area of predicted bounding boxes wit ground-truth bounding boxes. The rules used in identifying true positives; false positives; and false negatives are as below:

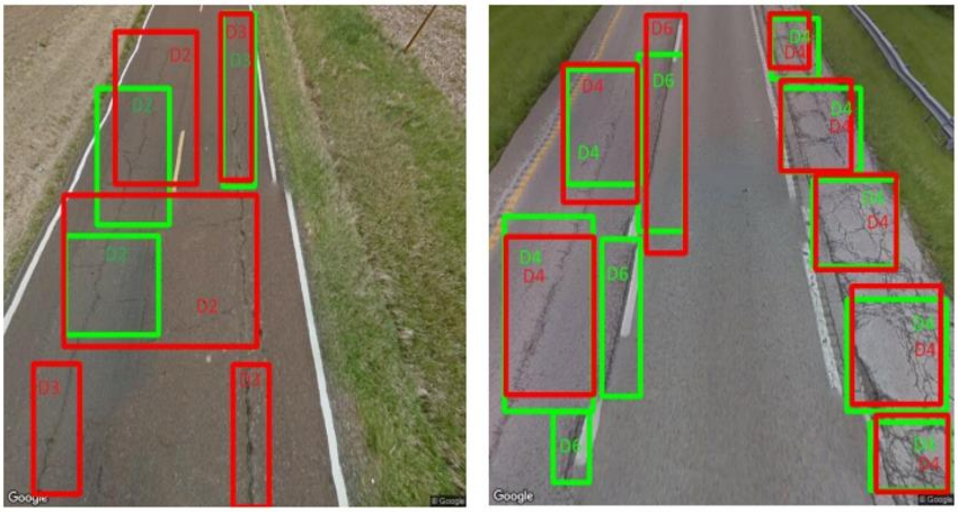
True Positive (TP): If more than 20% of the ground-truth bounding box is overlapped by the predicted bounding box, then it is counted as a right prediction. False Positive (FP): If less than 20% of the ground-truth bounding box is overlapped by the predicted bounding box, it is counted as wrong prediction. False Negative (FN): It is a false negative if a crack is not predicted by the model.

Bounding Box Color Coding: Red bounding box stands for ground truth values. The green bounding box signifies predicted boxes.

Evaluation Results: The cracks that are correctly detected and classified showed accuracy with over 20% Intersection over Union (IoU) in each crack class segment in Figure 5.2. In Figure 5.2, there are parts that have less than 20% IoU overlap when compared to the ground-truth which means they are wrong. Figure 5.2 has less than 20 percent IoU overlap, it shows true negatives. During the manual annotation process Figure 5.2 shows unlabeled cracks that remained.



(a) True positive (b) False positive & False negative



(c) False negative (d) Missed during annotations

Figure 5.2. Classification of Predicted Runway Crack for Validation Set.

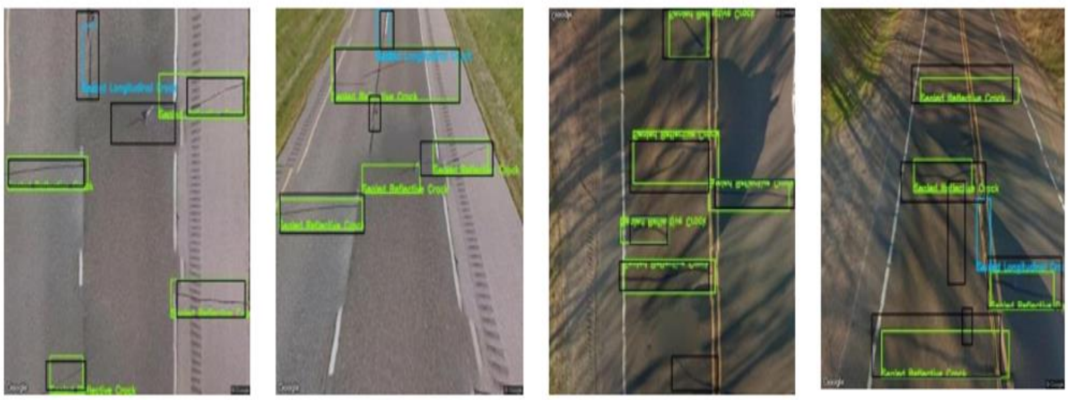
Conclusion: The proposed YOLOv5-based model performs highly well in evaluation metrics and visual representations of the bounding boxes, hence making it apparent. In the practical applications for which it was proffered, this shows that it is indeed effective and dependable in detecting and classifying runway cracks.

Result Comparison: YOLOv5 vs Faster R-CNN
Table 6 shows how YOLOv5 and Faster R-CNN perceive and classify nine crack categories. In the Faster R-CNN model, precision, recall and F1 scores for longitudinal, alligator and longitudinal lane cracks are relatively lower. On the other hand, YOLOv5 has achieved better F1 scores for all classes than Faster R-CNN. To be more specific, the YOLOv5 model has precision of 93% and 77% recall. The principal precision, recall and F1 score of proposed runway crack detection model in YOLOv5 is 84%. This emphasizes the importance of annotated data in developing runway crack detection with type identification powers. The result for detection and classification of the YOLO v5 and Faster R-CNN for 9 crack classes are shown in Table 5.1

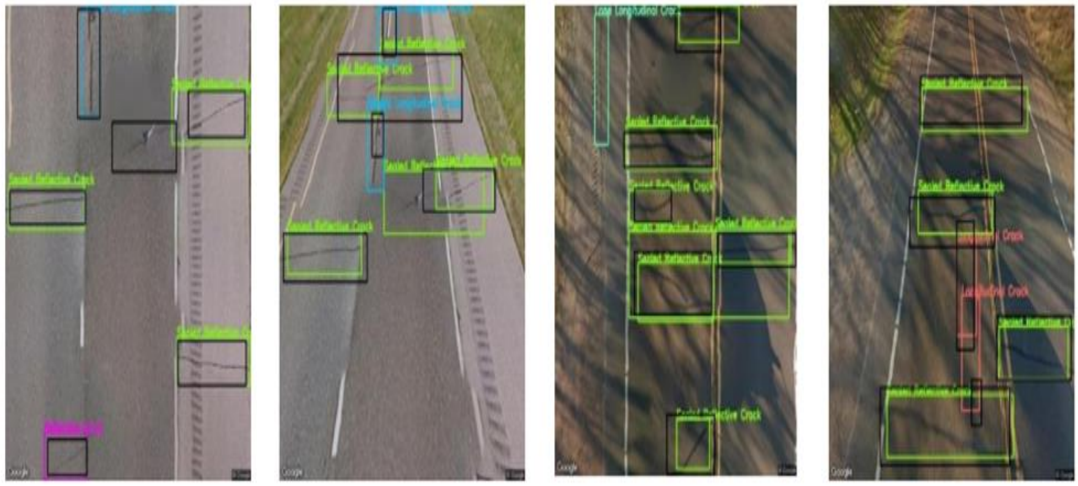
Comparative Analysis: The robustness of both models was tested by deliberately selecting images with obstacles such as sunshine and shadows (e.g., trees, crew buses) for figuring out top-down images if YOLOv5 was better than Faster R-CNN in detecting runway cracks. Ground truth values of cracks are represented by black bounding boxes in Figure 5.3. On the runway, blue and green bounded boxes show the predicted detections of cracks. Despite obstacles being present, both models are able to detect runway cracks with precision; they are thus effective and robust.

Table 5.1 . Result of YOLOv5 and Faster RCNN for Nine Crack Classes

Crack_ID	#Crack_Class	YOLOv5 Model			Faster R-CNN Model		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
D0	Reflective Runway Crack	0.92	0.75	0.83	0.72	0.71	0.71
D1	Transverse Runway Crack	0.89	0.82	0.85	0.74	0.73	0.74
D2	Block Runway Crack	0.92	0.78	0.84	0.81	0.58	0.67
D3	Longitudinal Runway Crack	0.91	0.83	0.87	0.66	0.43	0.52
D4	Alligator Runway Crack	0.91	0.74	0.82	0.81	0.43	0.57
D5	Sealed Transverse Runway Crack	0.93	0.83	0.87	0.83	0.68	0.75
D6	Sealed-longitudinal Runway Crack	0.92	0.78	0.84	0.82	0.53	0.64
D7	Lane longitudinal Runway Crack	0.94	0.57	0.71	0.75	0.30	0.42
D8	Pothole Runway Crack	0.96	0.78	0.86	0.83	0.78	0.80
	Average Mean	**0.92**	**0.76**	**0.83**	**0.77**	**0.57**	**0.64**



top-down image plain wide-view image shadowed top-down image shadowed wide view image.



top-down image | plain wide-view image | shadowed top-down image | shadowed wide view image.

Figure 5.3. Runway Crack Detection using YOLO v5 model and Faster R-CNN model.

In detecting and classifying runway cracks, the YOLOv5 model outperforms Faster R-CNN according to the comparative evaluation. The YOLOv5 model is capable accurately identifying such crack types as alligator, longitudinal and longitudinal lane with better precision, recall and F1 scores. Furthermore, given difficult condition, it also shows that both models can detect even when cracks are on the runway proving they can be relied upon for use in airport upkeep and caution.

CHAPTER 6

CONCLUSION AND FUTURE SCOPIC

The researchers presented Airport Runway Image Dataset (ARID) for image dataset developed to aid classifications, detections and monitoring of surface cracks on airport runways using deep learning methods that will facilitate automation The data set includes two forms of images, wide angle shots as well as overhead views Illustrative examples are given below in order to explain what each picture type is meant for in this case. Mainly wide-view images are used for the classification of runway cracks which helps to categorize observed types of pavement distresses. On the contrary, top-down images are utilized to estimate the density of the cracks, indicating how much destruction has taken place.

The dataset contains nine typical types of airport runway surface defects, which can be used to make an extensive assessment of different types of cracks. This study is about using two types of deep learning models-YOLO model 5 and Faster R-CNN model- to detect and identify cracks. The level of precision standing at 83 percent indicates that YOLO bot model 5 is efficient in recognizing cracks. While it had a marginally lower accuracy rate, the Faster R-CNN model nonetheless performed well in detecting runway cracks as well as other things.

The study is meant to show how surface cracks could be categorized using deep learning techniques together with wide-view imagery. From the results, the authors conclude that the proposed model is dependable since it can detect as well as forecast cracks at different camera views too. Therefore, it is a valuable and budget-friendly tool for assessing surface crack condition, runway surveillance, and maintenance activities.

In future work, we can focus on improving Robustness models, mainly through enhancing the Faster R-CNN analysis. Also, combining payload images and system during Google Map can increase practical applicability of the system in actual real-world scenarios. The improvement seeks to make runway inspections and management easy to do so that it can be made more effective than before.

REFERENCES

[1] Gopalakrishnan, Kasthurirangan. "Deep learning in data-driven pavement image analysis and automated distress detection: A review." Data 3.3 (2018): 28. https://doi.org/10.3390/data3030028

[2] Deleted.Oliveira, Henrique, and Paulo Lobato Correia. "CrackIT—An image processing toolbox for crack detection and characterization." 2014 IEEE international conference on image processing (ICIP). IEEE, 2014. https://doi.org/10.1109/ICIP.2014.7025160

[3] Zou, Qin, et al. "CrackTree: Automatic crack detection from pavement images." Pattern Recognition Letters 33.3 (2012): 227-238 https://doi.org/10.1016/j.patrec.2011.11.004

[4] Zhang, Allen, et al. "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network." Computer-Aided Civil and Infrastructure Engineering 32.10 (2017):805-819. https://doi.org/10.1111/mice.12297

[5] Zalama, Eduardo, et al. "Road crack detection using visual features extracted by Gabor filters." Computer-Aided Civil and Infrastructure Engineering 29.5 (2014): 342-358. https://doi.org/10.1111/mice.12042

[6] Akarsu, Büşra, et al. "A fast and adaptive road defect detection approach using computer vision with real time implementation." International Journal of Applied Mathematics Electronics and Computers Special Issue-1 (2016): 290-295. https://doi.org/10.18100/ijamec.270546

[7] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. doi:10.1038/nature14539.

[8] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. vol. 1. MIT press Cambridge; 2016.

[9] Rosenblatt F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. CORNELL AERONAUTICAL LAB INC BUFFALO NY; 1961.113

[10] Minsky M, Papert SA. Perceptrons: An introduction to computational geometry. MIT press; 2017.

[11] Zou Q, Cao Y, Li Q, Mao Q, Wang S. CrackTree: Automatic crack detection from pavement images. Pattern Recognit Lett 2012;33:227–38. doi:10.1016/j.patrec.2011.11.004.

[12] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Int J Comput Vis1988;1:321–31. doi:10.1007/BF00133570.

[13] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18:1527–1554.

[14] Bengio Y, LeCun Y. Scaling learning algorithms towards AI. Large-Scale Kernel Mach 2007;34:1–41.

[15] Samuel AL. Some Studies in Machine Learning Using the Game of Checkers. IBM J Res Dev 1959;3:210–29. doi:10.1147/rd.33.0210.

[16] Thuan, Do. "Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm." (2021).

[17]]Heidenreich H. What are the types of machine learning? Data Sci 2018. https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f (accessed May 5, 2019).

[18] Shalev-Shwartz S, Ben-David S. Understanding machine learning: From theory to algorithms. Cambridge university press; 2014.

[19] Skinner BF. 'Superstition'in the pigeon. J Exp Psychol 1948;38:168.

[20] Activation functions - deeplearning.ai. Coursera n.d. https://www.coursera.org/learn/neural-networks-deep-learning/lecture/4dDC1/activation-functions (accessed July 26, 2018).
[21] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proc. 27th Int. Conf. Mach. Learn. ICML-10, 2010, p. 807–814.
[22] Zeiler MD, Ranzato M, Monga R, Mao M, Yang K, Le QV, et al. On rectified linear units for speech processing. 2013 IEEE Int. Conf. Acoust. Speech Signal Process., 2013, p.3517–21. doi:10.1109/ICASSP.2013.6638312.
[23] Scherer D, Müller A, Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition. Artif. Neural Networks–ICANN 2010, Springer; 2010, p.92–101.
[24] Spark C. Deep learning for complete beginners: convolutional neural networks with keras n.d. http://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html (accessed July 26, 2018).
[25] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–1958.
[26] Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv Prepr ArXiv14126980 2014.
[27] Reddi SJ, Kale S, Kumar S. ON THE CONVERGENCE OF ADAM AND BEYOND 2018:23.
[28] An Open Source Machine Learning Framework for Everyone: tensorflow/tensorflow. tensorflow; 2019.
[29] Bengio Y. Practical recommendations for gradient-based training of deep architectures.Neural Netw. Tricks Trade, Springer; 2012, p. 437–478.
[30] Masters D, Luschi C. Revisiting Small Batch Training for Deep Neural Networks. ArXiv Prepr ArXiv180407612 2018.
[31] Rafael C Gonzalez, Richard E Woods, et al. Digital image processing, 2002.
[32] Qingquan Li and Xianglong Liu. Novel approach to pavement image segmentation based on neighboring difference histogram method. In Image and Signal Processing, 2008. CISP'08. Congress on, volume 2, pages 792–796. IEEE, 2008.
[33] H Zakeri, Fereidoon Moghadas Nejad, and Ahmad Fahimifar. Image based tech- niques for crack detection, classification and quantification in asphalt pavement: a review. Archives of Computational Methods in Engineering, 24(4):935–977, 2017.
[34] Arun Mohan and Sumathi Poobal. Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal, 57(2):787–798, 2018.
[35] Jun Li. A wavelet approach to edge detection. PhD thesis, Citeseer, 2003.
[36] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In Image Processing (ICIP), 2016 IEEE International Conference on, pages 3708–3712. IEEE, 2016.
[37] Zhun Fan, Yuming Wu, Jiewei Lu, and Wenji Li. Automatic pavement crack detec- tion based on structured prediction with the convolutional neural network. arXiv preprint arXiv:1802.02208, 2018.
[38] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Au- tomatic crack detection from pavement images. Pattern Recognition Letters, 33(3): 227–238, 2012.
[39] Djemel Ziou, Salvatore Tabbone, et al. Edge detection techniques-an overview. Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii, 8:537–559, 1998.
[40] John Canny. A computational approach to edge detection. In Readings in Computer Vision, pages 184–203. Elsevier, 1987.
[41] Huili Zhao, Guofeng Qin, and Xingjian Wang. Improvement of canny algorithm based on pavement edge detection. In Image and Signal Processing (CISP), 2010 3rd International Congress on, volume 2, pages 964–967. IEEE, 2010.

[42] Sos Agaian, Ali Almuntashri, and AT Papagiannakis. An improved canny edge detection application for asphalt concrete. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, pages 3683–3687. IEEE, 2009.
[43] Yi-Chang Tsai, Vivek Kaul, and Russell M Mersereau. Critical assessment of pave- ment distress segmentation methods. Journal of transportation engineering, 136(1): 11, 2010.
[44] Turgut Aydin, Yucel Yemez, Emin Anarim, and Bulent Sankur. Multidirectional and multiscale edge detection via m-band wavelet transform. IEEE Transactions on image processing, 5(9):1370–1377, 1996.
[45] Zhen Zhang, Siliang Ma, Hui Liu, and Yuexin Gong. An edge detection approach based on directional wavelet transform. Computers & Mathematics with Applica- tions, 57(8):1265–1271, 2009.
[46] Piotr Porwik and Agnieszka Lisowska. The haar-wavelet transform in digital image processing: its status and achievements. Machine graphics and vision, 13(1/2):79–98, 2004.
[47] A Cuhadar, K Shalaby, and S Tasdoken. Automatic segmentation of pavement condition data using wavelet transform. In Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on, volume 2, pages 1009–1014. IEEE, 2002.
[48] Javier R Movellan. Tutorial on gabor filters. Open Source Document, 2002.
[49] Lia Ma, Yunhong Wang, Tieniu Tan, et al. Iris recognition based on multichannel gabor filtering. In Proc. Fifth Asian Conf. Computer Vision, volume 1, pages 279– 283, 2002.
[50] Wai Kin Kong, David Zhang, and Wenxin Li. Palmprint feature extraction using 2-d gabor filters. Pattern recognition, 36(10):2339–2347, 2003.
[51] Khagswar Bhoi and Dipesh Kumar Solanki. Texture Segmentation Using Optimal Gabor Filter. PhD thesis, 2011.
[52] M Salman, S Mathavan, K Kamal, and M Rahman. Pavemet crack detection using the gabor filter. In Intelligent Transportation Systems-(ITSC), 2013 16th Interna- tional IEEE Conference on, pages 2039–2044. IEEE, 2013.
[53] Eduardo Zalama, Jaime G´omez-Garc´ıa-Bermejo, Roberto Medina, and Jos´e Llamas. Road crack detection using visual features extracted by gabor filters. Computer- Aided Civil and Infrastructure Engineering, 29(5):342–358, 2014.
[54] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. Journal of graphics tools, 12(2):13–21, 2007.
[55] Kelvin CP Wang, Qiang Li, and Weiguo Gong. Wavelet-based pavement distress image edge detection with a trous algorithm. Transportation Research Record, 2024 (1):73–81, 2007.
[56] Paul Viola and Michael J Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004.
[57] Rui Fan, Mohammud Junaid Bocus, Yilong Zhu, Jianhao Jiao, Li Wang, Fulong Ma, Shanshan Cheng, and Ming Liu. Road crack detection using deep convolutional neural network and adaptive thresholding. arXiv preprint arXiv:1904.08582, 2019.
[58] Nobuyuki Otsu. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1):62–66, 1979.
[59] Jagat Narain Kapur, Prasanna K Sahoo, and Andrew KC Wong. A new method for gray-level picture thresholding using the entropy of the histogram. Computer vision, graphics, and image processing, 29(3):273–285, 1985.
[60] Yong Hu and Chun-xia Zhao. A novel lbp based methods for pavement crack detec- tion. Journal of pattern Recognition research, 5(1):140–147, 2010.
[61] Qingquan Li and Xianglong Liu. Novel approach to pavement image segmentation based on neighboring difference histogram method. In Image and Signal Processing, 2008. CISP'08. Congress on, volume 2, pages 792–796. IEEE, 2008.
[62] HN Nguyen, TY Kam, and PY Cheng. Automatic crack detection from 2d images using a crack measure-based b-spline level set model. Multidimensional Systems and Signal Processing, 29(1):213–244, 2018.

[63] Hoang-Nam Nguyen, Tai-Yan Kam, and Pi-Ying Cheng. An automatic approach for accurate edge detection of concrete crack utilizing 2d geometric features of crack. Journal of Signal Processing Systems, 77(3):221–240, 2014.

[64] Roberto Medina, Jos´e Llamas, Eduardo Zalama, and Jaime G´omez-Garc´ıa-Bermejo. Enhanced automatic detection of road surface cracks by combining 2d/3d image processing techniques. In Image Processing (ICIP), 2014 IEEE International Conference on, pages 778–782. IEEE, 2014.

[65] Henrique Jos´e Monteiro Oliveira. Crack Detection and Characterization in Flexible Road Pavements using Digital Image Processing. PhD thesis, PhD thesis, Universi- dade de Lisboa-Instituto Superior T´ecnico, 2013.

[66] Lee S, Ha J, Zokhirova M, Moon H, Lee J. Background Information of Deep Learning for Structural Engineering. Arch Comput Methods Eng 2018;25:121–9. doi:10.1007/s11831-017-9237-0.

[67] Liu R, Yang B, Zio E, Chen X. Artificial intelligence for fault diagnosis of rotating machinery: A review. Mech Syst Signal Process 2018;108:33–47.doi:10.1016/j.ymssp.2018.02.016.

[68] Ross ZE, Meier M-A, Hauksson E, Heaton TH. Generalized Seismic Phase Detection with Deep Learning. Bull Seismol Soc Am 2018;108:2894–901. doi:10.1785/0120180080.

[69] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. doi:10.1038/nature14539.

[70] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278–2324.

[71] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conf. Comput. Vis. Pattern Recognit., 2009, p. 248–55.doi:10.1109/CVPR.2009.5206848.

[72] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst., 2012, p. 1097–1105.

[73] Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Infrastruct Eng 2017;32:361–378.

[74] Cha Y-J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput-Aided Civ Infrastruct Eng 2017.

[75] Huang H, Li Q, Zhang D. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunn Undergr Space Technol 2018;77:166–176.111

[76] Chen F-C, Jahanshahi MR. NB-CNN: deep learning-based crack detection using convolutional neural network and naive Bayes data fusion. IEEE Trans Ind Electron2018;65:4392–4400.

[77] Zhang A, Wang KC, Li B, Yang E, Dai X, Peng Y, et al. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. Comput-Aided Civ Infrastruct Eng 2017;32:805–819.

[78] Eisenbach M, Stricker R, Debes K, Gross H-M. Crack Detection with an Interactive and Adaptive Video Inspection System. Arbeitsgruppentagung Infrastrukturmanagement 2017:94–103.

[79] Pauly L, Hogg D, Fuentes R, Peel H. Deeper networks for pavement crack detection. Proc.34th ISARC, IAARC; 2017, p. 479–485.

[80] Zhang L, Yang F, Zhang YD, Zhu YJ. Road crack detection using deep convolutional neural network. Image Process. ICIP 2016 IEEE Int. Conf. On, IEEE; 2016, p. 3708–3712.

[81] Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H. Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. ArXiv PreprArXiv180109454 2018.

[82] Vishal Mandal, Lan Uong, and Yaw Adu-Gyamfi. Automated road crack detection using deep convolutional neural networks. In 2018 IEEE International Conference on Big Data (Big Data), pages 5212–5215. IEEE, 2018.

[83] BV Sobol, AN Soloviev, PV Vasiliev, and LA Podkolzina. Deep convolution neural network model in problem of crack segmentation on asphalt images. Vestnik of Don State Technical University, 19(1), 2019.
[84] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, and Hi- roshi Omata. Road damage detection using deep neural networks with images cap- tured through a smartphone. arXiv preprint arXiv:1801.09454, 2018.
[85] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector ma- chines. ACM transactions on intelligent systems and technology (TIST), 2(3):1–27, 2011.
[86] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 14(771-780):1612, 1999.
[87] Nhung Thi Hong Nguyen, Thanh Ha Le, Stuart Perry, and Thi Thuy Nguyen. Pave- ment crack detection using convolutional neural network. In Proceedings of the Ninth International Symposium on Information and Communication Technology, pages 251–256. ACM, 2018.
[88] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
[89] Laha Ale, Ning Zhang, and Longzhuang Li. Road damage detection using retinanet. In 2018 IEEE International Conference on Big Data (Big Data), pages 5197–5200. IEEE, 2018.
[90] Somin Park, Seongdeok Bang, Hongjo Kim, and Hyoungkwan Kim. Patch-based crack detection in black box images using convolutional neural networks. Journal of Computing in Civil Engineering, 33(3):04019017, 2019.
[91] Kaiming He, Georgia Gkioxari, Piotr Doll´ar, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961– 2969, 2017.
[92] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[93] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.
[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[95] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional net- works for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
[96] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolu- tional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12):2481–2495, 2017.
[97] Joseph Bullock, Carolina Cuesta-L´azaro, and Arnau Quera-Bofarull. Xnet: a convo- lutional neural network (cnn) implementation for medical x-ray image segmentation suitable for small datasets. In Medical Imaging 2019: Biomedical Applications in Bibliography 117 Molecular, Structural, and Functional Imaging, volume 10953, page 109531Z. International Society for Optics and Photonics, 2019.
[98] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing, 338:139–153, 2019.
[99] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. IEEE transactions on pattern analysis and machine intelligence, 35(6):1397–1409, 2012.
[100] Mohamed Y Shahin and Starr D Kohn. Development of a pavement condition rating procedure for roads, streets, and parking lots. volume i. conditions rating procedure. Technical report, CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL, 1979.

LIST OF PUBLICATION(S)

1. Abhilasha Sharma, Aryan Bansal, “Airport Runway Crack Detection to Classify and Densify Surface Crack Type”.

OPEN ACCESS

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-13 Issue-3, February 2024

Airport Runway Crack Detection to Classify and Densify Surface Crack Type

Abhilasha Sharma, Aryan Bansal

***Abstract**: With the extensive development in infrastructures, many airports are built in order to satisfy travelling needs of people. The frequent arrival and departure of numerous plans lead to substantial runway damage and related safety concerns. So, the regular maintenance of runway has become an essential task specially for detection and classification of cracks in terms of owing to the intensity heterogeneity of cracks such as low real-time performance and the long time-consuming manual inspection. This paper introduces a new dataset named as ARID with 8 different crack classes. A runway crack detection model based on YOLOv5 and Faster RCNN has been proposed which is annotated on 8,228 collected datasets. Then the model is trained with different parameters for training to obtain the optimal result. Finally, based on experimental result, the crack detection precision has improved from 83% to 92%, while the recall has increased from 62.8% to 76%.*

***Keywords**: Crack Segmentation, Google API, Pavement Detection, Runway Crack, Runway Distresses Detection.*

I. INTRODUCTION

In recent decades extreme travelling and transportation exchanges has been tremendously increased across globe. The aviation industry has witnessed significant advancements in technology, leading to safer and more comfortable flights. Modern aircraft are equipped with state-of-the-art navigation systems, advanced safety features, and improved cabin amenities, making air travel a more enjoyable experience for passengers. While increased transportation activity can indirectly impact the service performance and service life of infrastructure, the development of surface cracks is influenced via some factors. Thus, regular maintenance has become an essential task specially for detection and classification of cracks on the runway. The structural degradation of runway can potentially endanger safety and diminish service life as well as may cause loss in economics growth. Crack-based damages has the potential to impair performance and present safety risks. They become the most common defect that appears on airport runways that lowers the stress state and potentially causes accidents. If this damaged pavement is not repaired timely, the problem will worsen due to recurring environmental or human factors.

Manuscript received on 02 August 2023 | Revised Manuscript received on 19 January 2024 | Manuscript Accepted on 15 February 2024 | Manuscript published on 28 February 2024.
**Correspondence Author(s)*

Dr. Abhilasha Sharma, Department of Software Engineering, Delhi Technological University, Delhi, India. E-mail: abhi16.sharma@gmail.com, ORCID ID: 0000-0003-1493-9070

Aryan Bansal*, Department of Software Engineering, Delhi Technological University, Delhi, India. E-mail: aryanbansal1711@gmail.com, ORCID ID: 0009-0006-0472-8623

Repairing a crack before it deteriorates will decrease the cost of maintenance, reduce the impact on the environment, and lengthen the life of the asphalt. If the maintenance tasks of crack removal are achieved in time, the price for the crack rehabilitation can be kept upto to 80%.

In recent years, the quick growth in Indian economy has a pace of airport development which leads the aviation industry to recover from pre-pandemic levels, and new routes and startup carriers are on the horizon. By 2025, the government hopes to build 220 additional airports. According to Jyotiraditya Scindia, the minister for civil aviation, India would have 1,200 planes and 400 million passengers by 2027. The nation is building new greenfield airports using public financing and public-private partnerships in a market that is expected to experience tremendous growth. 8 of 21 greenfield airports are already operating. As more and more people opt to fly, various types of runway damage will unavoidably result. The runway is extensively tainted with fuel stains and aircraft wheel marks. Moreover, there are often very thin cracks present which can indicate the possibility of significant failure These images are extremely noisy and feature a variety of characteristics including very small fractures, fuel stains, and textured surfaces. Automated crack detection technologies have revolutionized the analysis process in intelligent transportation systems by providing rapid and reliable results, replacing the slow and subjective traditional approaches. An automated crack detection system can efficiently evaluate the condition of a runway and aid airport authorities {(International Civil Aviation Organization (ICAO)} in organizing and prioritizing repair activities aimed at increasing the runway's useful life. Computer vision (CV) fabricates machine by learning from the features of digital images and videos. By using visual data, it improves understanding of features and patterns. For these research domains, there is a vast amount of visual data available via cellphones and digital cameras.

Various researchers have been gone through the concept behind deep architecture-based crack detection approaches as explained further. Gopalakrishna et al. [1] gives a chronicle review on deep-learning approaches grounded on crack detection. To eliminate road markings from the track image, Otsu's enhanced threshold segmentation algorithm is applied. After the markings have been eliminated and the crack has been produced, the enhanced adaptive threshold segmentation algorithm is used to segment the image. Oliveira et al. [2] employed a variety of image analysis techniques to identify and describe cracks on road surfaces.

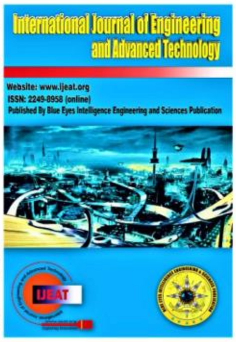
Retrieval Number: 100.1/ijeat.A42731013123
DOI: 10.35940/ijeat.A4273.13030224
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

International Journal of Engineering and Advanced Technology (IJEAT)

Exploring Innovation | ISSN: 2249-8958 (Online) | A Periodical Journal | Reg. No.: C/819981 | Published By BEIESP

The *International Journal of Engineering and Advanced Technology (IJEAT)* has ISSN: 2249-8958 (online), an open-access, peer-reviewed, periodical bi-monthly international journal, which is published by *Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)* in *February, April, June, August, October and December*.



The journal aims to publish high-quality peer–reviewed original articles in the area of *Engineering and Technology* that covers Computer Science and Engineering, Information Technology, Electrical and Electronics Engineering, Electronics and Telecommunication, Mechanical Engineering, Civil Engineering and all interdisciplinary streams of *Engineering and Technology*.

Important Dates-

Articles Submission Open for Volume-13 Issue-6, August 2024

Last Date of Article Submission: 30 July 2024
Date of Notification: 15 August 2024
Date of Publication: 30 August 2024

Article Submission System

Author (s) can submit articles electronically throughout the year. The Editors reserve the right to reject articles without sending them out for review. Submitted articles should not have been previously published or are currently under consideration for publication elsewhere. Briefs and research notes are not published in this journal. Submitted articles must be within the scope of the journal. All articles go through a double-anonymized peer-review process.

Authors can streamline proofreading tools such as Grammarly to enhance their articles. As an open-access journal, Authors must pay an Article Processing Charge (APC) to publish their articles and retain copyright. Furthermore, authors acquaint themselves with the journal's editorial and publishing policies.