

國立臺灣大學工學院土木工程學系

碩士論文



Department of Civil Engineering

College of Engineering

National Taiwan University

Master Thesis

結合深度學習模型與區塊影像技術應用於

混凝土表面裂縫辨識

Automatic Recognition Method for Concrete Surface Cracks:  
Integrating Image Patch Technique and Deep Learning Models

林禹齊

Lin, Yu-Qi

指導教授：張家銘 教授

Advisor: Dr. Chia-Ming Chang

中華民國 112 年 7 月

July, 2023

國立臺灣大學碩士學位論文  
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE  
NATIONAL TAIWAN UNIVERSITY

結合深度學習模型與區塊影像技術應用於  
混凝土表面裂縫辨識

(論文中文題目) (Chinese title of Master's thesis)

Automatic Recognition Method for Concrete Surface Cracks: Integrating Image  
Patch Technique and Deep Learning Models

(論文英文題目) (English title of Master's thesis)

本論文係 林禹齊 (姓名) R10521203 (學號) 在國立臺灣大學 土木工程系 (系/所/學位學程) 完成之碩士學位論文，於民國 102 年 7 月 28 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department / Institute of Civil Engineering on 28 (date) 7 (month) 2023 (year) have examined a Master's thesis entitled above presented by Lin, Yu-Qi (name) R10521203 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

張富銘

(指導教授 Advisor)

楊卓謙

江向榮

賴勇志

系主任/所長 Director: 葛宇甯

## 誌謝



可以完成這篇論文，需要感謝指導老師張家銘教授的指導。從碩一開始就願意固定花時間與我們研究生們討論，碩士第一年，在老師的指導下，做了結構控制和使用 AI 的裂縫辨識，而第二年最終選擇以 AI 裂縫辨識作為論文的研究主題，在兩年的研究過程中，謝謝張家銘教授不厭其煩的將專業知識傳授給我們，在做研究的過程，獲得了專業的知識。在老師指導的兩年多來，獲得的專業的知識雖然重要，但對我來說更重要的是在研究過程中自己思想與老師思想碰撞的過程。老師無論研究的表現是否符合預期，都願意不斷花時間投注在我們每一位研究生身上，對我們來說是一位值得尊敬的老師。

另外需要感謝論文口試期間的汪向榮教授、賴勇安教授與楊卓謙博士仔細專業的指證，提出論文需要修改的地方，以使本論文的內容更加完善。

研究生兩年多的生涯中還要感謝曾經一起在研究室聊天、討論研究內容及一起修課甚至一起玩撲克牌的同學。謝謝肇昱、婕妤、庭維、奕婕、子倫及宜謙學長姐在我們碩一時和我們分享各種修課經驗；謝謝同屆的冠諭、鄭棋及 Nikka，一起經歷研究及修課從無到有的過程，還有各種大大小小的事情；謝謝曾經在 AI 組一起分享研究成果、互相協助的大家；還有一起完成國際健康監測競賽的 MOJI，以及後來加入的各位研究室的夥伴們。

成為研究生兩年半，研究過程經跌跌撞撞，雖然研究成果沒有到很豐碩完美，但至少有做出當初的目標：實用且創新的研究；而與老師與同儕的過程累積的經驗，是未來帶得走的寶貴經驗。

林禹齊 謹誌於  
國立台灣大學土木研究大樓 511 室  
2023 年 7 月

## 中文摘要



對建築結構而言，結構健康檢測是至關重要，其可於災後觀察結構物的受損情形以評估補強方式，或是對老舊建物進行壽命評估，恰當的結構物健康檢測及補強可以延續建築物生命週期甚至避免生命及財產的損失。目前結構健康檢測分為破壞式及非破壞式檢測，為了進行大量的建物健康檢測，例如災後建物受損程度的評估，或是定期性的對老屋進行評估，通常以非破壞式檢測作為主要健康檢測的方式。非破壞性檢測主要觀察建物的外觀受損情形，以評估建築物的受損程度，其中以觀察裂縫之型態作為重要判斷依據，包括裂縫之長度、寬度及走向。過去之檢測方式需要技師至現場實地勘驗，以目視觀察並於劣化評估表上標註破壞，而人力之觀察及紀錄流程通常是耗費時間及精力且不精確的，因此近年來開始以電腦輔助裂縫之判別，期盼以自動化之判別流程節省時間及精力，並利用自動化判讀之優點使判斷標準統一。本研究提出流程化之自動化裂縫判讀流程，使用深度學習之方式作為主要裂縫識別之方式，改善以往以深度學習裂縫辨識之缺點，同時可以達到裂縫訓練資料標註過程簡化，並提高訓練後之模型於辨識時之精度，提出新的辨識判別流程，以計算裂縫可能出現之機率，再進一步利用計算出之裂縫出現機率建立裂縫之走向。

本研究提出之裂縫辨識方法主要分為三個部分，第一部分為深度學習模型之訓練，本研究主要以 VGG19 及 Resnet50 以圖像分類為任務的深度學習模型，作為辨識區塊圖像之裂縫之深度學習模型。為了比較不同訓練模型之精度性，訓練及預測時將原始資料集進行灰階圖片之轉換，及配合限制對比度自適應直方圖均衡化 (CLAHE) 以校正圖片之色階分布，並於訓練過程中使用資料增強 (Data

Augmentation)的方式，提高訓練照片之多元性。第二部分則為提出之裂縫辨識流程，此方法將原始裂縫圖片切割成區塊圖片，同時調整區塊圖片之重疊率，再將區塊圖像放入深度學習模型中預測，以圖像分類之方式達到像素等級的裂縫辨識結果，並進一步利用裂縫辨識結果之機率分布，計算裂縫之粗略位置與走向。第三部分則為裂縫辨識流程之驗證，使用於某停車場所拍攝之裂縫圖像，以驗證模型之可行性。因為自行拍攝之圖像與原始訓練之裂縫照片資料集於拍攝條件、裂縫型態具有一定之差異性，因此可同時驗證本研究提出之裂縫辨識方法之精度。經驗證過程中發現，以 Resnet50 搭配原始裂縫訓練資料集擁有最好的辨識精度，而以相鄰區塊圖片之步長為 14 計算出的裂縫機率分布圖可以有效的顯現出裂縫的位置，最後以 Zhang-Suen 骨架化演算法計算出的裂縫位置與原始圖片之裂縫標註相比，發現其結果可以有效的預測裂縫位置與趨勢，僅有少部分裂縫區域未被辨識出。

關鍵字：結構健康檢測、非破壞式檢測、電腦輔助、自動化裂縫判讀、深度學習、直方圖均衡化、資料增強、圖像分類

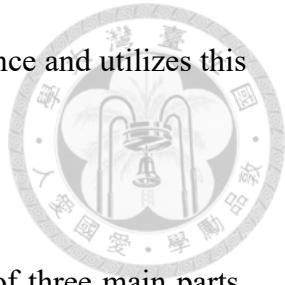
# Abstract

For architectural structures, structural health monitoring is crucial. It can be used to observe the damage to structures after a disaster in order to assess the reinforcement methods. It is also used for evaluating the lifespan of old buildings. Proper structural health monitoring and reinforcement can extend the lifecycle of a building and even prevent loss of life and property. Currently, structural health monitoring can be categorized as destructive and non-destructive testing. Non-destructive testing is usually the preferred method for conducting extensive building health assessments, such as evaluating the extent of damage after a disaster or conducting regular assessments of old houses. Non-destructive testing primarily examines the appearance of the building to assess the level of damage, with a focus on observing crack patterns. This includes measuring the length, width, and direction of cracks.

In the past, inspection methods required technicians to physically visit the site, visually observe and mark the damage on a deterioration assessment form. However, the manual observation and recording process was time-consuming, labor-intensive, and not precise. Therefore, in recent years, there has been a shift towards computer-assisted crack identification to save time and effort. The advantages of automated interpretation are utilized to establish unified criteria for crack assessment.

This study proposes a process-oriented automated crack identification approach that utilizes deep learning as the primary method for crack recognition. It aims to improve upon the limitations of previous deep learning crack recognition methods. The proposed approach simplifies the crack labeling process during training and enhances the generalizability of the trained model to various types of cracks. It introduces a new

identification process that calculates the probability of crack occurrence and utilizes this probability to determine the crack direction.



The crack recognition method proposed in this study consists of three main parts. The first part involves training deep learning models. VGG19 and ResNet50, commonly used deep learning models for image classification, are employed as the crack recognition models for block images. To compare the generalizability of different training models, the original dataset is transformed into grayscale images during training and prediction. Additionally, contrast-limited adaptive histogram equalization (CLAHE) is applied to correct the grayscale distribution of images. Data augmentation techniques are also employed during the training process to increase the diversity of training photos.

The second part introduces the crack recognition process. This method involves segmenting the original crack images into block images and adjusting the overlap rate of these blocks. The block images are then fed into the deep learning models for prediction, enabling pixel-level crack recognition results through image classification. Furthermore, the probability distribution of the crack recognition results is utilized to calculate the approximate position and direction of the cracks.

The third part is the validation of the crack recognition process, using crack images captured in a parking lot to assess the feasibility of the proposed model. As there may be differences in shooting conditions and crack patterns between the self-captured images and the original training crack photo dataset, the generalizability of the crack recognition method proposed in this study can be evaluated simultaneously. During the validation process, it was found that using ResNet50 with the original training crack dataset achieved the best recognition accuracy. The crack probability distribution calculated

using a stride of 14 for adjacent block images effectively displayed the crack positions.

Finally, the crack positions and trends predicted by the Zhang-Suen skeletonization algorithm were compared to the crack annotations in the original images. The results showed that the proposed method effectively predicted the crack positions and trends, with only a small portion of the crack areas remaining undetected.

# 目錄



口試委員會審定書 .....	#
誌謝 .....	i
中文摘要 .....	ii
Abstract .....	iv
目錄 .....	vii
圖目錄 .....	ix
表目錄 .....	xii
<b>Chapter 1 緒論 .....</b>	<b>1</b>
1.1 研究動機 .....	1
1.2 文獻回顧 .....	3
1.2.1 非深度學習方法於裂縫辨識之應用 .....	3
1.2.2 深度學習方法於裂縫辨識之應用 .....	4
1.2.3 深度學習與裂縫的連續性 .....	7
1.3 本文內容 .....	7
<b>Chapter 2 深度學習模型之訓練 .....</b>	<b>9</b>
2.1 前言 .....	9
2.2 訓練資料集準備 .....	9
2.3 彩色與灰階圖像之轉換 .....	12
2.4 限制對比度自適應直方圖均衡化(CLAHE) .....	12
2.5 深度學習模型 .....	17
2.5.1 VGG16 .....	17

2.5.2	ResNet50 .....	18
2.6	深度學習模型訓練結果 .....	20
2.6.1	ResNet50 .....	25
2.6.2	VGG16 .....	28
2.7	小結 .....	30
<b>Chapter 3</b>	<b>裂縫辨識架構 .....</b>	<b>32</b>
3.1	前言 .....	32
3.2	原始圖片與區塊圖片轉換 .....	32
3.3	區塊圖片之辨識 .....	34
3.4	裂縫機率計算 .....	35
3.5	黑白數值圖 .....	36
3.6	骨架化演算法提取裂縫路徑 .....	37
<b>Chapter 4</b>	<b>裂縫辨識應用於實際場域 .....</b>	<b>40</b>
4.1	前言 .....	40
4.2	深度學習模型初步驗證 .....	42
4.3	裂縫機率分布計算 .....	56
4.4	骨架化演算法提取裂縫走向 .....	64
4.5	小結 .....	67
<b>Chapter 5</b>	<b>結論與未來展望 .....</b>	<b>68</b>
5.1	結論 .....	68
5.2	未來展望 .....	69
<b>參考文獻</b>	<b>71</b>	

# 圖目錄



圖 2.1 原始資料集之照片 .....	10
圖 2.2 灰階資料集影像 .....	11
圖 2.3 灰階與灰階 CLAHE 資料集影像 .....	11
圖 2.4 直方圖均衡化前 .....	14
圖 2.5 直方圖均衡化後 .....	14
圖 2.6 限制對比度自適應直方圖均衡化(CLAHE)重分配過程 .....	15
圖 2.7 使用 Clip Limit 為 2 .....	16
圖 2.8 使用 Clip Limit 為 4 .....	16
圖 2.9 使用 Clip Limit 為 8 .....	17
圖 2.10 VGG16 模型架構 .....	18
圖 2.11 ResNet 家族 5 種不同大小模型的架構(圖片引用自 He et al., 2016) .....	18
圖 2.12 殘差學習架構 .....	19
圖 2.13 Resnet50 模型架構 .....	19
圖 2.14 Zoom Range 為 0.7 至 1.3 的裂縫圖片 .....	21
圖 2.15 Rotation Range 為 60 度的裂縫圖片 .....	22
圖 2.16 Width Shift 為 0.2 的裂縫圖片 .....	23
圖 2.17 Height Shift 為 0.2 的裂縫圖片 .....	24
圖 2.18 原始資料集未經資料增強圖片之訓練成果 .....	27
圖 2.19 原始資料集圖片之訓練成果 .....	27
圖 2.20 灰階資料集圖片之訓練成果 .....	27



圖 2.21 灰階與 CLAHE 資料集圖片之訓練成果 .....	28
圖 2.22 原始資料集未經資料增強圖片之訓練成果 .....	29
圖 2.23 原始資料集圖片之訓練成果 .....	29
圖 2.24 灰階資料集圖片之訓練成果 .....	30
圖 2.25 灰階與 CLAHE 資料集圖片之訓練成果 .....	30
圖 3.1 裂縫辨識架構 .....	32
圖 3.2 區塊圖像之分割方法 .....	33
圖 3.3 以區塊圖像之裂縫辨識結果 .....	34
圖 3.4 裂縫機率分布圖 .....	36
圖 3.5 產生裂縫機率分布圖之過程 .....	37
圖 3.6 使用 Zhang-Suen 骨架化演算法 .....	37
圖 3.7 像素 A1 與周圍其他像素之關係 .....	38
圖 4.1 自行拍攝之裂縫圖片 .....	40
圖 4.2 經過灰階轉換的裂縫圖片 .....	41
圖 4.3 經過灰階轉換與直方圖均衡化的裂縫圖片 .....	41
圖 4.4 大型裂縫照片 .....	42
圖 4.5 9 張裂縫圖片標註結果 .....	43
圖 4.6 大型裂縫圖片標註 .....	43
圖 4.7 原始圖像使用 VGG16 模型未經資料增強裂縫辨識結果 .....	46
圖 4.8 原始圖像使用 ResNet50 模型未經資料增強裂縫辨識結果 .....	47
圖 4.9 原始圖像使用 VGG16 模型裂縫辨識結果 .....	48
圖 4.10 原始圖像使用 ResNet50 模型裂縫辨識結果 .....	49



圖 4.11 灰階圖像使用 VGG16 模型裂縫辨識結果 .....	50
圖 4.12 灰階圖像使用 ResNet50 模型裂縫辨識結果 .....	51
圖 4.13 灰階轉換與 CLAHE 使用 VGG16 模型裂縫辨識結果表 .....	52
圖 4.14 灰階轉換與 CLAHE 使用 ResNet50 模型裂縫辨識結果 .....	53
圖 4.15 大型裂縫圖片辨識展示圖片一 .....	55
圖 4.16 大型裂縫圖片辨識展示圖片二 .....	55
圖 4.17 步長為 224 像素之裂縫辨識機率圖 .....	57
圖 4.18 步長為 112 像素之裂縫辨識機率圖 .....	58
圖 4.19 步長為 56 像素之裂縫辨識機率圖 .....	59
圖 4.20 步長為 28 像素之裂縫辨識機率圖 .....	60
圖 4.21 步長為 14 像素之裂縫辨識機率圖 .....	61
圖 4.22 步長為 14 的裂縫黑白數值圖 .....	64
圖 4.23 經骨架化得出的裂縫路徑 .....	65
圖 4.24 比較裂縫標註結果與預測路徑 .....	66

# 表目錄



表 2.1 資料集中照片數量 .....	10
表 2.2 線上資料增強參數 .....	21
表 2.3 ResNet50 深度學習訓練參數 .....	26
表 2.4 VGG16 深度學習訓練參數 .....	29
表 4.1 裂縫精度計算之四種情況 .....	44
表 4.2 原始圖像使用 VGG16 模型未經資料增強裂縫辨識結果 .....	46
表 4.3 原始圖像使用 ResNet50 模型未經資料增強裂縫辨識結果 .....	47
表 4.4 原始圖像使用 VGG16 模型裂縫辨識結果 .....	48
表 4.5 原始圖像使用 ResNet50 模型裂縫辨識結果 .....	49
表 4.6 灰階圖像使用 VGG16 模型裂縫辨識結果 .....	50
表 4.7 灰階圖像使用 ResNet50 模型裂縫辨識結果 .....	51
表 4.8 灰階轉換與 CLAHE 使用 VGG16 模型裂縫辨識結果 .....	52
表 4.9 灰階轉換與 CLAHE 使用 ResNet50 模型裂縫辨識結果 .....	53
表 4.10 六種組合裂縫辨識結果之平均值 .....	54
表 4.11 大型裂縫圖片辨識結果範例 .....	56
表 4.12 大型裂縫圖片平均辨識結果 .....	56
表 4.13 步長、重疊率與總區塊圖像數之關係 .....	61
表 4.14 步長 112 使用 ResNet50 模型裂縫辨識結果 .....	62
表 4.15 步長 56 使用 ResNet50 模型裂縫辨識結果 .....	63
表 4.16 步長 28 使用 ResNet50 模型裂縫辨識結果 .....	63



表 4.17 步長 14 使用 ResNet50 模型裂縫辨識結果 ..... 63

表 4.18 不同步長下使用 ResNet50 模型裂縫辨識結果 ..... 64



# Chapter 1 緒論

## 1.1 研究動機

結構健康檢測對於維持老舊建築物的壽命及評估災後受損建築物之健康狀況至關重要。檢測方式可包括非破壞性檢測及破壞性檢測，而為了大規模的結構健康檢測，如區域性所有老舊建築健檢，或是災後損傷建物平估，通常以非破壞性檢測作為主要進行手段，以加快評估速度。然而現行評估仍以人力評估為主，作業流程需要以人力觀察結構物的表面，並詳細記錄建築物外觀之受損情形，通常評估流程是耗時且費力的，不符合經濟效益。隨著科技的進步，近年來逐漸開始發展以電腦輔助之方式評估建築物外觀的受損情形，尤其近期深度學習領域的發展，更是讓自動且精確的評估流程成為可能，因此本研究即延續先前學者之工作，期望進一步改善目前深度學習於裂縫辨識之方法。

深度學習方法如圖像分類(Image Classification)、物件辨識(Object Detection)、實例分割(Instant Segmentation)等，已應用於裂縫的辨識，而不同的方法具有不同的優勢及缺點。物件辨識依據訓練的標註資料，可以使模型框選出裂縫的位置，然而因為裂縫是細長型且多裂縫可能具有交集，無法清楚定義一個裂縫之實際大小與範圍，因此於訓練資料標註時如何定義一個裂縫之大小及範圍，將會影響模型之訓練及辨識成果，是使用物件辨識模型辨識裂縫的主要缺點；而實例分割模型則可以使用標註過後的裂縫遮罩(MASK)訓練模型以得到裂縫之實際位置，並達到像素等級的裂縫位置預測精度，然而實例分割之訓練資料中的裂縫遮罩由於需要達到像素等級的精度，且標註品質會大幅影響模型預測精度，因此建立實例分割的訓練



資料通常是耗時且費力的；至於圖像分類則可以用來判斷照片中是否具有裂縫，此方法僅需判斷圖片中是否包含裂縫，因此訓練資料建立過程簡單，且模型易於訓練並擁有相當高的精度。然而其缺點在於無法精細的預測裂縫位置，通常僅能簡易的判斷圖像中之該區域是否具有裂縫，而無法達到精確的裂縫位置預測。綜合以上三種模型之優缺點，可知若要得到精確之裂縫位置，則需要花費大量時間進行訓練資料標註，而若要減化訓練資料之標註流程，則無法提供精確之裂縫位置預測，因此如何建立同時具有精確辨識裂縫位置功能且易於生成之訓練資料是本研究所欲解決之問題。

本研究之另一目的在於提高深度學習模型於裂縫辨識之精度，現行以深度學習預測裂縫之方法，由於訓練資料蒐集的限制，通常訓練後之深度學習模型僅能運用於具有相似條件之場域。然而建築形態之多樣性，使得一個訓練後之深度學習模型，難以運用於所有場域之裂縫辨識。為了進行自動化裂縫辨識，通常需要再蒐集該場域之裂縫型態，訓練一個全新的深度學習模型，繁瑣的前置作業使得使用深度學習自動化辨識裂縫之方法無法普及，為深度學習方法無法普及之主要原因。

鑑於深度學習應用於裂縫辨識所具有之優缺點，及考量現行深度學習方法於裂縫辨識之困境，本研究提出之方法嘗試綜合深度學習不同方法之優點，提出可辨識像素等級裂縫位置之深度學習方法，簡化模型訓練資料的準備過程，並使用資料增強的方式及嘗試以灰階轉換與 CLAHE 轉換提高不同場域的辨識精度，期許深度學習裂縫辨識方法可更加普及。



## 1.2 文獻回顧

自動化裂縫辨識技術目前主要可分為兩種方式：使用非深度學習之方法與使用深度學習之方式，早期主要以非深度學習的方式進行，如將灰階圖片二值化以得到裂縫的路徑，近幾年來隨著 AI 科技的發展，自動化裂縫辨識之研究逐漸轉往深度學習方向發展。由於以深度學習辨識裂縫之方式無論是在辨識精度或是對於不同裂縫型態之精度皆能大幅領先以影像處理辨識裂縫之方式(Hsieh & Tsai, 2020 及 Dorafshan et al., 2018) ，因此以深度學習辨識裂縫之方式逐漸成為自動化裂縫辨識的主流[14,29]。

### 1.2.1 非深度學習方法於裂縫辨識之應用

傳統之非深度學習裂縫方法有局部二值化(LBP)或是應用各式的濾波器於灰階圖像以提取裂縫之像素位置，如 Hu et al. 改善局部二值化的方法，以使二值化方法可以應用於不同種類的結構物表面，與辨識不同紋理之裂縫[44]；Salman et al. 使用 Gabor Filter 以提取相片中的裂縫[41]，以上方法通常具有簡單、消耗運算資源低等優點。

現地拍攝之裂縫圖像若沒有經過特殊的篩選，照片本身可能因為光線或是結構物表面之其他髒污使得裂縫照片具有非裂縫但顏色不均勻的區域，為了處理相關的問題，通常裂縫辨識需要一系列的過程，包括影像前處理、裂縫辨識演算法及辨識結果後處理。如 Fujita et al.提出泛用的裂縫辨識流程，首先結合中值濾波器(Median Filter)去除照片中的陰影部分，再使用線型濾波器(Line Filter)以濾除圖像中的污漬，最後以自適應閾值法及概率鬆弛(Probabilistic Relaxation)以判斷裂縫之

最終位置[45]；Cubero-Fernandez et al. 則使用各式的預處理以提取裂縫資訊，如 Canny Detection 及雙邊濾波器，接著再使用決策樹啟發式算法以對裂縫進行分類[32]。綜合以上方法，搭配前處理再使用裂縫辨識流程可以有效提升裂縫之辨識結果精度。

### 1.2.2 深度學習方法於裂縫辨識之應用

隨著 AI 圖像辨識技術的發展，裂縫辨識也開始以深度學習模型取代傳統的裂縫辨識方法，如 Liu et al. 使用 U-net 進行裂縫辨識[23]；Yamane et al. 則將 MASK R-CNN 模型應用於裂縫辨識[13]；Hang et al. 使用全連接卷積網路(FCN)，進行不同型態裂縫的語意分割[13]；而 Dung et al. 及 Golding et al. 使用基於 VGG16 的編碼器的深度全卷積網路(FCN)於特定資料集進行裂縫辨識，並達到 90%以上的精度[25,4]；最後 Özgenel et al. 則使用常見的預訓練的深度學習網路，比較不同的訓練照片數量大小、模型深度及訓練迭帶次數對預測裂縫表現的影響[30]，以上學者開始使用現有的模型架構於裂縫辨識領域，並獲得不錯的辨識成果。

然而現有模型可能會因為裂縫型態與傳統物件辨識模型所針對之物體型不同，因此為了提升辨識成果及效率，需多學者開始自行開發的深度學習模型，如 Zhang et al. 基於 U-net 的模型架構進行修改，並提出新的損失函數(Generalized Dice Loss)以對裂縫進行實例分割；為了適應不同的裂縫尺寸，Zou et al. 提出新的深度學習網絡 DeepCrack，以階層式的模型架構捕捉不同尺寸的照片特徵，並以 SegNet 作為編碼及解碼的主要模型架構[15]；Yang et al. 結合特徵金字塔及分層增強網路提出 FPHBN 模型架構[18]；Pan et al. 則修改了 VGG19 的類神經網路，建立新的

深度學習神經網絡 SCHNet [10]，利用特徵金字塔注意力模塊(Feature Pyramid Attention Module)、空間注意力模塊(Spatial Attention Module)及頻道注意力模塊(Channel Attention Module)實現自注意力機制；Chen et al. 參考 SegNet，建立新的神經網絡 PCSN，同時使用沒有標頭的 VGG16 模型作為編碼器[16]；Alipour et al. 則使用三種不同方式 joint training、sequential training 及 ensemble learning，用以研究相同模型對目前常見裂縫的泛用性[20]。Huyan et al. 提出了 CrackU-net 用以辨識裂縫[19]；Xu et al. 基於卷積神經網絡並結合空洞卷積及空間金字塔池化的優勢建立新的裂縫辨識模型，以適應不同尺寸的裂縫[24]；Zhang et al. 則提出新的類神經網絡模型 ConvNet 以解決強度裂縫的不均勻性和圖像背景的複雜性[34]；Fang 設計新的類神經網絡 CDDS 用以辨識以無人機拍攝的原始圖像並提供像素等級的裂縫辨識[11]；Kumar et al. 使用了修正的 LeNet5 模型辨識橋梁裂縫、瀝青裂縫及混凝土裂縫共三類資料集的裂縫[9]；Ren et al. 提出基於深層全連接卷積網絡改進的 CrackSegNet，以改善全連接卷積網絡應用於裂縫的效能[21]，說明已有多名學者開始研發是用於裂縫辨識的深度學習模型，並期望可以透過自行修改的模型，進一步提高深度學習模型於裂縫辨識的辨識精度。

使用深度學習方法與非深度學習之裂縫辨識流程相似，建立系統化的深度學習辨識流程是至關重要的，於使用深度學習辨識前去除圖像中的背景雜訊或是於模型訓練前或訓練過程中增強訓練資料集之多樣性以增加模型的泛用性是提升模型辨識裂縫精確度的方法，如 Prasanna et al. 提出自動化裂縫辨識流程 STRUM[36]，使用線型偵測器搭配空間調諧多特徵計算並深度學習分類網路，運用於橋梁自動掃描系統；Rajadurai et al. 則以 AlexNet 模型，結合遷移學習與訓練圖像資料增強方

式以進行裂縫辨識[6]；Protopapadakis et al. 先使用卷積神經網絡對裂縫進行辨識，再使用啟發式後處理，達到高精度、低操作時間的裂縫辨識[27]，意味許多系統化的裂縫流程已被提出，由於系統化的辨識流程經過前處理與後處理，因此可以應付不同的裂縫型態，並計算出裂縫資訊，因此更完善的架構使其更適合於應用於現場場域。

而許多學者也提出結合不同深度學習模型或是電腦影像辨識方法，許多複合式辨識裂縫的方法已被提出。如 Fang et al. 提出的架構中，首先使用深度學習模型辨識圖像中是否有裂縫，再以雙邊過濾消除圖片中的雜訊，最後以自適應閾值法提取裂縫的位置，是結合深度學習與電腦影像處理的方式，而此二階段的辨識方法可以運用於大面積的結構體辨識，以增加裂縫辨識的效率與精度[11]；Kim et al. 等人則首先使用類神經網絡進行的圖像分類及實例分割，再以細化及跟縱演算法計算照片中裂縫的長度與寬度[8]；Laxman et al.[1] 提出的架構先使用卷積神經網絡辨識混凝土表面的裂縫，再使用結合卷積特徵提取層和回歸模型的卷積神經網絡以辨識裂縫的深度；而為了聚焦於照片中具有裂縫的位置，Liu et al. 先以 YOLOv3 模型辨識裂縫的區域，接著再以修正後的 U-net 模型對裂縫區域進行實例分割[21]；相似的，Zhao et al. 則以 YOLOv5 先對裂縫區域進行辨識，接著再以自行提出的 Crack-FPN 模型對裂縫進行實例分割[3]；而 Kalfarisi et al. 提出的方式則包括以 FRCNN 先辨識出裂縫區域接著再使用結構隨機生成邊緣辨識對辨識為裂縫的區域進行實例分割[41]，這些方法由於使用兩步驟的辨識流程，因此可以分兩階段將裂縫於圖片中提取出來，被驗證可以提高裂縫的辨識成果。



### 1.2.3 深度學習與裂縫的連續性

然而，目前的自動化辨識流程中，其中一個主要的問題是裂縫辨識結果的不連續性，辨識結果的不連續性將會影響後續對裂縫尺寸，因此許多學者已經提出以後處理為方式之解決方法。如 Wu et al. 提出新的裂縫辨識架構 MorphLink-C 以連接分散的裂縫，其首先膨脹變換，再以細化變換連接不連續的裂縫[37]；Fang et al. 提出結合深度學習模型及 Bayesian 機率分析的方法，首先使用 CNN 模型辨識裂縫的寬度，再根據周圍被辨識為裂縫的圖像區塊依據空間的接近程度、方向連續性計算累積的 Bayesian 機率。周圍若無其他裂縫區塊的支持，將被視為辨識錯誤，以以後處理的方式排除辨識錯誤之裂縫部分，以增加模型的精度[11]。Zou et al. 則開發 CrackTree 使用測地線陰影移除演算法移除圖片上的陰影，並以向量投票法建立裂縫機率分布圖，最終勾勒出裂縫的走向，以建立裂縫的連續性[26]，以上方法為使用各種技巧應用於辨識後的裂縫成果，並都取得不錯的成果，可以使用後處理的方式解決裂縫不連續的問題。

## 1.3 本文內容

本文之內容主要分為五個章節，依序為第一章緒論、第二章深度學習模型之訓練、第三章提出之裂縫辨識流程、第四章裂縫辨識流程驗證及第五章的結論與未來展望，希望可以藉由本文之內容將提出之裂縫辨識方法流程從研究動機、提出方法之詳細流程及最終之成果展示清楚描述，其中提出之新的裂縫辨識方法流程主要描述於第二章及第三章。

第一章主要說明提出此裂縫辨識方法之緣由，描述過去研究學者之研究成果，

嘗試進行改進並提出本文之裂縫辨識方法。

第二章內容聚焦於模型訓練所經歷之流程，內容從裂縫資料集的蒐集、資料集之圖像轉換、深度學習模型之架構、如何訓練深度學習模型及最終之模型訓練結果。以本章描述之深度學習模型訓練方法，訓練出之深度學習模型，將應用於第三章所述之裂縫辨識流程中。

第三章之目的在於描述提出之裂縫辨識流程，將清楚描述從原始影像之輸入、切割成區塊圖像、放入訓練完成之深度學習模型中辨識至最後辨識出的裂縫走向之詳細流程，此章結為本文之主要貢獻。

第四章之目的則為驗證本文提出方法之可行性，該章節以自行拍攝之裂縫照片進行驗證，首先選用表現最好的模型與訓練資料，再使用第三章所述之裂縫辨識架構進行驗證，並取得裂縫位置辨識成果。

第五章則為結論與展望，總結本文提出之裂縫辨識方法之流程設計結果、實際應用可行性並提出未來可改進之處。

## Chapter 2 深度學習模型之訓練



### 2.1 前言

本文欲提出之裂縫辨識方法主要使用深度學習模型作為裂縫判別的方式，因此選取並訓練出一個適用的深度學習模型是至關重要的。為了訓練模型需先找出合適之資料集作為訓練資料，本文提出之裂縫辨識方法其中一個目的為增加模型於不同型態裂縫之精度，因此本章結同時包括使用線上資料(Online Augmentation)增強、彩色與灰階圖像之轉換、自適應直方圖均衡化等方法，以提升模型應用於測試資料時的精度。

### 2.2 訓練資料集準備

用於訓練深度學習模型之裂縫資料集來源為 Özgenel & Çağlar Fırat 於 2019 年發布之混凝土裂縫公開資料集[47]，原始之資料集共有 40000 張裂縫照片，包括 20000 張含有裂縫的照片與 20000 張未含有裂縫的照片。本研究共有三種型態的裂縫資料集需要建立，包括原始的裂縫照片、經過灰階處理的裂縫照片與同時經過灰階處理與自適應直方圖均衡化(CLAHE)的裂縫照片，其中訓練資料集與驗證資料集的比例為 8 比 2，所有資料集的照片數量如表一，圖 2.1 至圖 2.3 展示三種資料集中的照片。

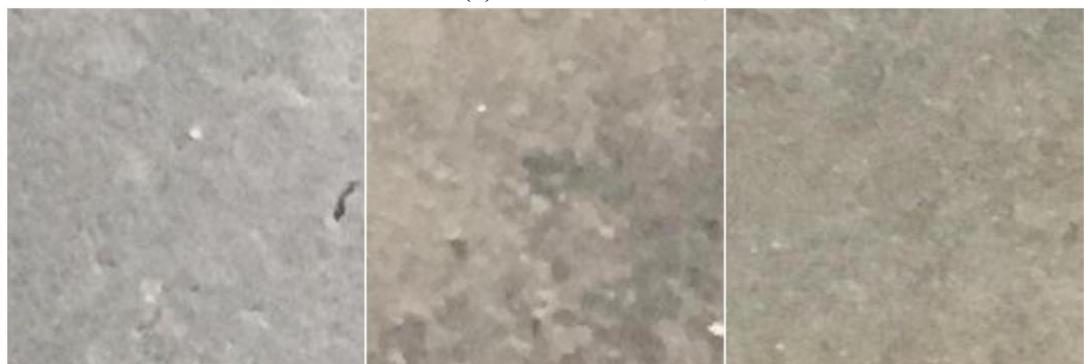
表 2.1 資料集中照片數量



	訓練資料集		驗證資料集	
	含裂縫	不含裂縫	含裂縫	不含裂縫
原始資料集	16,000	16,000	4,000	4,000
灰階	16,000	16,000	4,000	4,000
灰階與 CLAHE	16,000	16,000	4,000	4,000

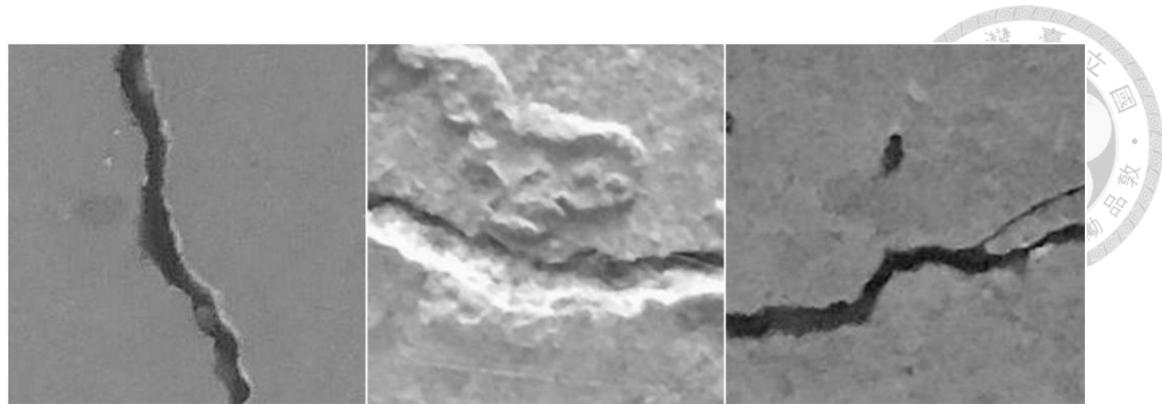


(a)具有裂縫之影像

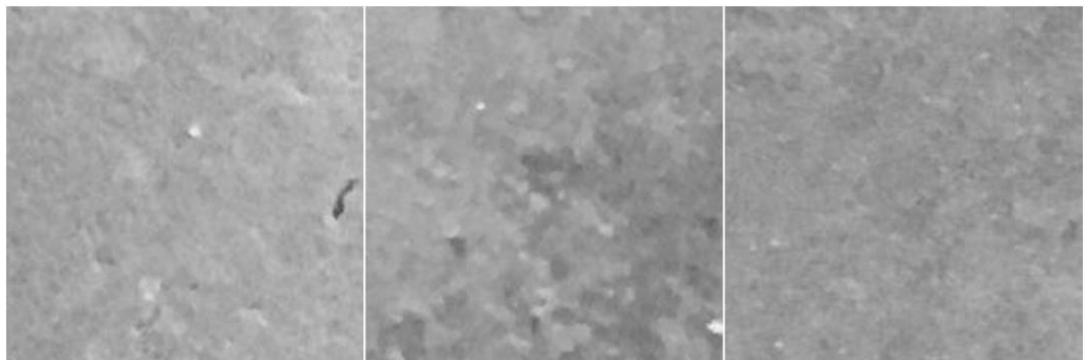


(b)不具有裂縫之影像

圖 2.1 原始資料集之照片

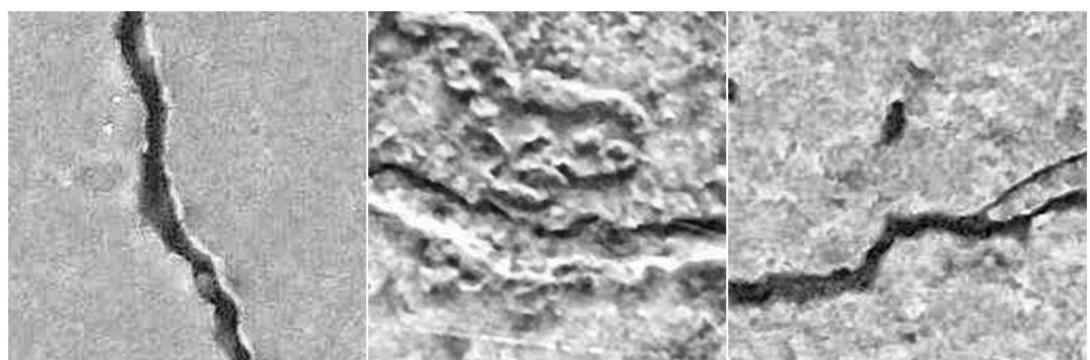


(a)具有裂縫之影像

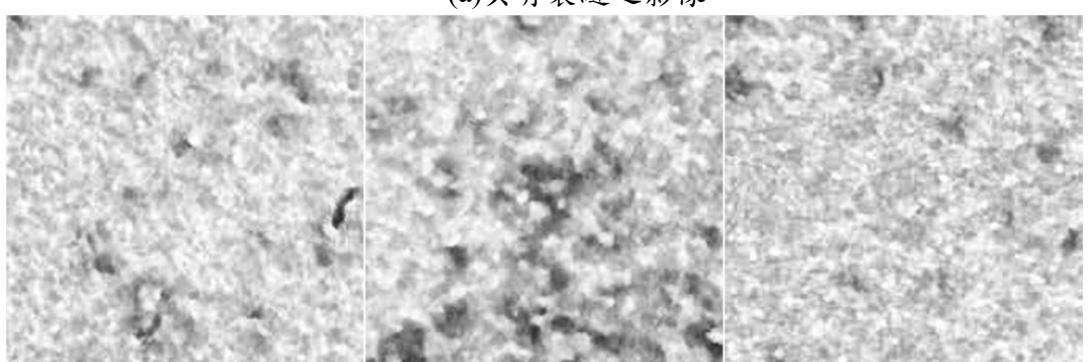


(b)不具有裂縫之影像

圖 2.2 灰階資料集影像



(a)具有裂縫之影像



(b)不具有裂縫之影像

圖 2.3 灰階與灰階 CLAHE 資料集影像



## 2.3 彩色與灰階圖像之轉換

相片可以分為彩色與黑白，其以 2D 像素矩陣組合而成，每個像素擁有各自的值以表現出該像素之色彩，當每個像素表現出各自的色彩時，即可組成一張具有意義的圖片。

以彩色圖片為例，每個像素都具有 3 個值以分別表示該像素中紅色、綠色與藍色之值(R、G、B)，若要將彩色影像轉換為灰階影像，則需要以(2.1)式進行轉換，轉換結果可比較圖 2.1 與圖 2.2。

$$Gray = R \times 0.299 + G \times 0.587 + B \times 0.114 \quad (2.1)$$

其中  $R$ 、 $G$ 、 $B$  分別為原始彩色影像中該像素之 3 個通道之值， $Gray$  為轉化為灰階之單一通道。

## 2.4 限制對比度自適應直方圖均衡化(CLAHE)

自適應直方圖均衡化是常見的影像處理技巧，用來改善圖片色彩的對比度，藉由這個方法，色彩的分配可以在直方圖上分布的更平均[49]。其轉換方式為：

考慮一個原始灰階圖像，以  $n_i$  代表圖像中灰階  $i$  出現的次數，則該灰階  $i$  出現的概率為式(2.2)：

$$p_x(i) = p(x = i) = \frac{n_i}{n}, 0 \leq i < L \quad (2.2)$$

其中  $n$  是圖像中所有的像素總數， $L$  是圖像中灰階數值的範圍，通常為 256，而  $p_x(i)$

即可代表該圖像之直方圖，並且正規化至 0 與 1 間。

接著使用(2.3)式的轉換公式，可以將直方圖的分布轉換為平均分布於 0 至  $L$  之間。

$$g_{i,j} = \text{floor} \left[ (L-1) \sum_{i=0}^{f_{i,j}} p_x(i) \right] \quad (2.3)$$

其中  $f_{i,j}$  是原始圖像位於  $i,j$  位置之像素數值(Pixel Intensity)，而  $g_{i,j}$  為轉換後之像素數值， $\text{floor}$  函數則會無條件捨去到最接近的整數， $p_x(i)$  則定義於(2.2)式中。

此外(2.3)式可改寫為：

$$Y = (L-1) \int_0^X p_X(x) dx \quad (2.4)$$

其中  $Y$  是轉換後圖像之新的像素強度，而  $X$  為原始圖像欲轉換像素強度之值。

轉換後新圖像之所有像素強度  $Y$  之像素強度分布，可由(2.2)式推導並寫為  $p_Y(y)$ 。而其灰階概率式  $p_Y(y)$  之分布如(2.5)式，因此新的像素機率分布將會以定值平均的分布在像素強度 0 至  $L$  之間，(2.5)式之詳細推導過程可參考 Gonzalez 及 Woods 於 2008 年出版的著作[49]。

$$p_Y(y) = \frac{1}{L-1}, 0 \leq y < L \quad (2.5)$$

以原始圖像 2.4(b)為例，其所有像素之累積分布如圖 2.4(a)，當使用自適應直方圖均衡化之方法後的結果為圖 2.5(a)，其影像之所有像素累積分布如圖 2.5(b)。

觀察圖 2.4(b)與 2.5(b)，可以發現圖中影像具有更高的對比度，而由圖 2.4(a)與 2.5(a)可知其實現方式為改變直方圖的分布，將直方圖更平均的分布在 0 到 255 之間，而 cdf 之分布曲線由曲線轉換為斜直線，其中 cdf 之定義如(2.6)式，為(2.2)式累加之結果，或是代表圖 2.4 中紅色直方圖累加之結果。

$$cdf(i) = \sum_{j=0}^i p(x = j) \quad (2.6)$$

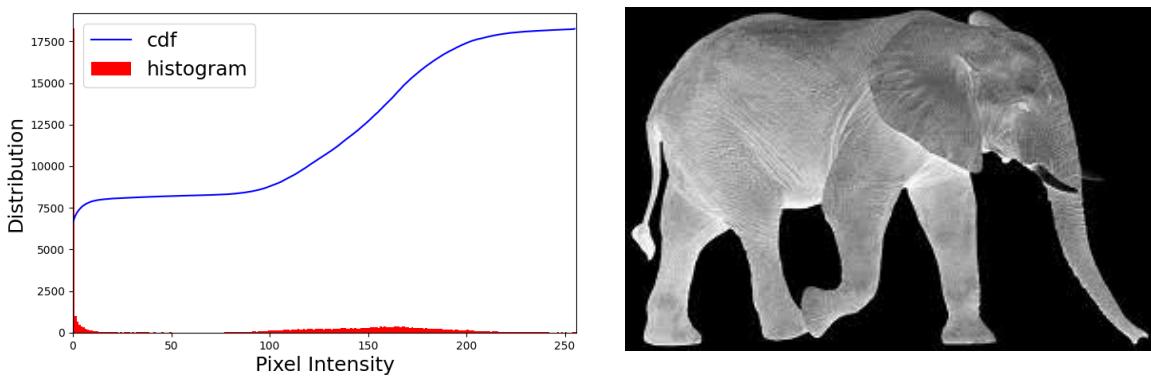


圖 2.4 直方圖均衡化前

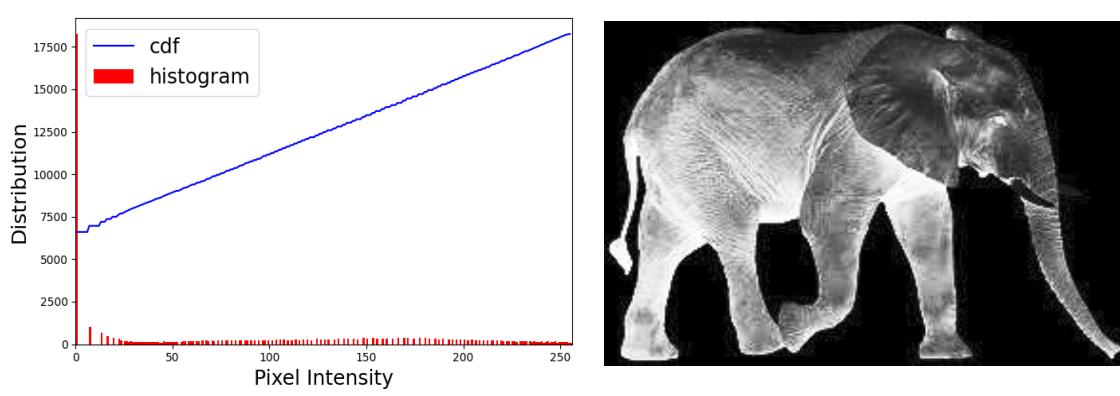


圖 2.5 直方圖均衡化後

直方圖均衡化雖然可以增加全域圖片的對比度，卻無法針對圖像中部分區域增加對比度，因此自適應直方圖均衡化 (Adaptive Histogram Equalization, AHE) 被提出，其實現方式為針對不同區域進行各自的直方圖均衡化計算，並使用這些直方圖進行亮度重分配，以提高區域的對比度。

然而，自適應直方圖均衡化因為過度強調區域內的直方圖均衡化，因此若該區域具有相似的顏色分布，卻沒有邊緣特徵，自適應直方圖均衡化將會使圖片產生雜訊，因此限制對比度自適應直方圖均衡化(Contrast Limited AHE，CLAHE)被提出。

限制對比度自適應直方圖均衡化首先將原圖像計算出之機率分布直方圖限制至設定數值以下，如圖 2.6 左圖為例，原始之直方圖為一錐形曲線，其中虛線部分為設置之對比度限制，圖中橘色區域因超過設定之數值，因此被重分平均分配到右圖中之橘色區域，重新分配後之直方圖仍有可能有超出對比度限制，因此需重複迭代此過程當直方圖無區域超越此限制。經過上述之轉換後，接著再使用 (2.3)式將原始圖像之像素素值進行轉換，此操作可以限制轉換函數的幅度，以減少生成圖片雜訊的產生。

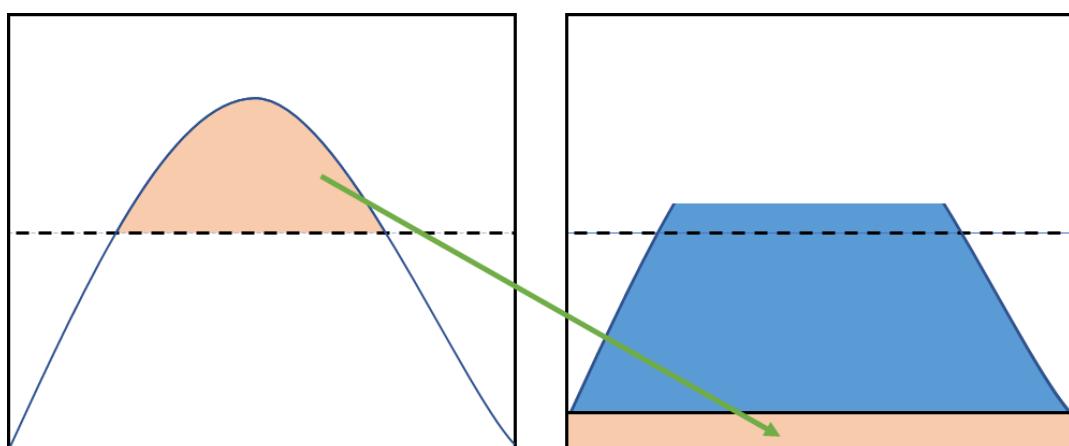


圖 2.6 限制對比度自適應直方圖均衡化(CLAHE)重分配過程

本研究使用 OpenCv 提供的 CLAHE 函數[51]，函數中具有兩個參數，分別為 Tile Grid Size 即 Clip Limit。其中 Grid Size 之定義為區域性直方圖均衡化的尺寸大小，輸入的圖片將會被裁切成該尺寸的的區塊，接著對每個區塊進行直方圖均衡化；而 Clip Limit 為局部對比度變化的對比度限制。為了尋找最適合的函數變數，在增加對比度與減少雜訊間取得平衡，比較圖 2.7 至 2.9 中不同函數對圖片結果的差異，選擇雜訊最少與對比度最明顯之圖片，最終以 Clip Limit 為 4 與 Tile Grid Size 為 16 作為最終之參數。

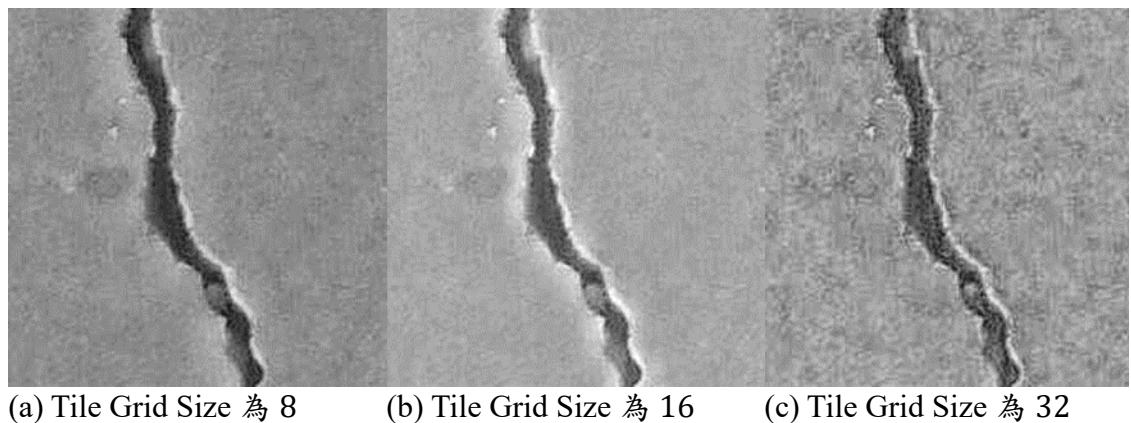


圖 2.7 使用 Clip Limit 為 2

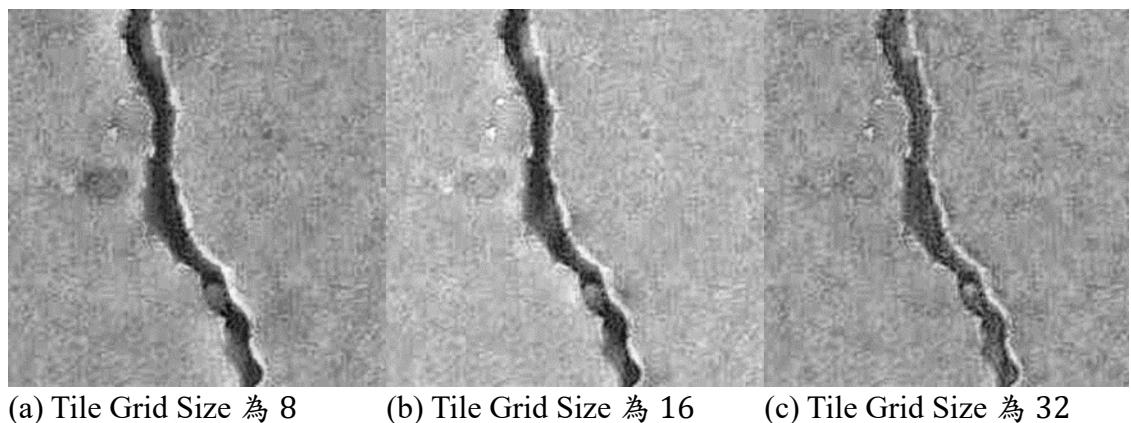


圖 2.8 使用 Clip Limit 為 4



(a) Tile Grid Size 為 8

(b) Tile Grid Size 為 16

(c) Tile Grid Size 為 32

圖 2.9 使用 Clip Limit 為 8

## 2.5 深度學習模型

本研究使用兩種不同的深度學習模型以用來辨識區塊圖像是否為裂縫，兩種深度學習模型分別為 VGG19 及 ResNet50，皆為常見用來進行影像分類的深度學習模型。

### 2.5.1 VGG16

VGG16 模型是 2014 年由 ImageNet 舉辦的競賽(ILSVRC)的亞軍，由牛津大學提出，由於競賽中達到的高精確度因此受到重視[48]。VGG16 模型是一種卷積神經網路，其模型架構延續 AlexNet，並同時建立更多的隱藏層，其隱藏層數目達到 16 層，模型架構如圖 2.10。

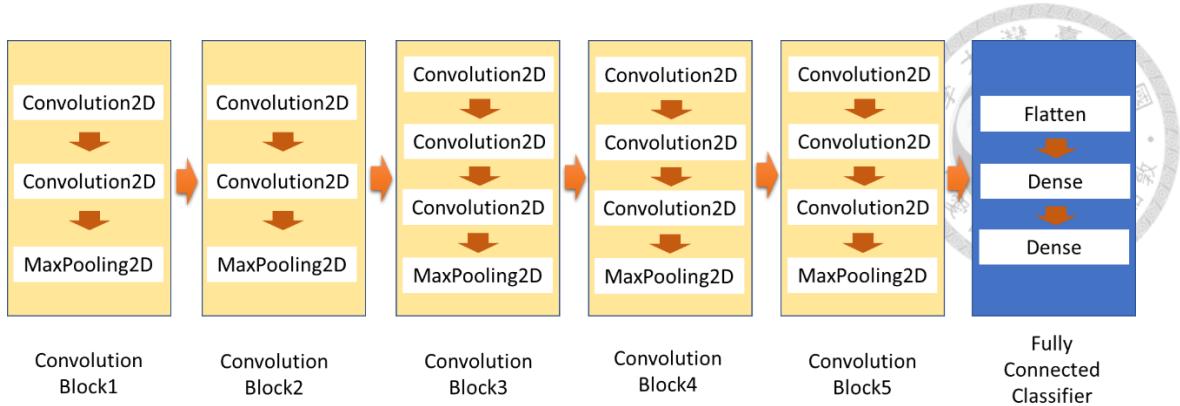


圖 2.10 VGG16 模型架構

### 2.5.2 ResNet50

ResNet50 模型同樣出自 ImageNet 舉辦的 ILSVRC 競賽，獲得 2015 年的冠軍 [35]。ResNet50 源自於 ResNet 家族的其中一個模型，其他模型包括 ResNet18、ResNet34、ResNet50、ResNet101 及 ResNet152，5 種模型主要的架構差異在於卷積層區塊堆疊次數多寡，如圖 2.11 所示。以 ResNet50 與 ResNet101 為例，兩者主要差異在第四個卷積區域，卷積層區塊堆疊數量分別為 6 層及 23 層，本研究選用 ResNet50 因其為最常見的 ResNet 家族模型，並已經實踐於 Python 中的 Keras 套件中。

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56			3×3 max pool, stride 2		
conv3_x	28×28	$\left[ \begin{array}{c} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array} \right] \times 2$	$\left[ \begin{array}{c} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array} \right] \times 3$	$\left[ \begin{array}{c} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$	$\left[ \begin{array}{c} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$	$\left[ \begin{array}{c} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$
conv4_x	14×14	$\left[ \begin{array}{c} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array} \right] \times 2$	$\left[ \begin{array}{c} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array} \right] \times 4$	$\left[ \begin{array}{c} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 4$	$\left[ \begin{array}{c} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 4$	$\left[ \begin{array}{c} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 8$
conv5_x	7×7	$\left[ \begin{array}{c} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array} \right] \times 2$	$\left[ \begin{array}{c} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array} \right] \times 3$	$\left[ \begin{array}{c} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 6$	$\left[ \begin{array}{c} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 23$	$\left[ \begin{array}{c} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 36$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

圖 2.11 ResNet 家族 5 種不同大小模型的架構(圖片引用自 He et al., 2016)

作者發現隨著深度學習模型深度的增加，訓練後模型的精度由於 Gradient Vanishing 和 Exploding 的問題反而下降，因此作者提出圖 2.11 的殘差學習架構 (Residual Learning)，藉由跨層連結的神經網路，創造衡等映射(Identity Mapping)的函數於深度學習架構，由於跨層連接的特性使得此架構更容易訓練，同時不會增加模型中的參數量。圖 2.12 是 ResNet50 的架構圖，殘差學習架構主要應用於中間的 4 種卷積神經網路。

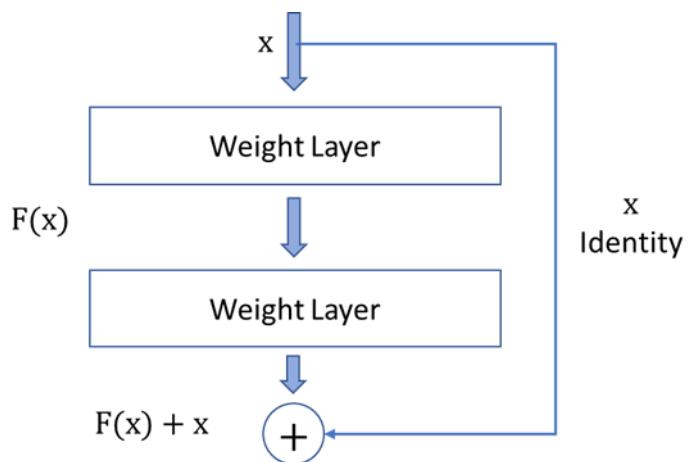


圖 2.12 殘差學習架構

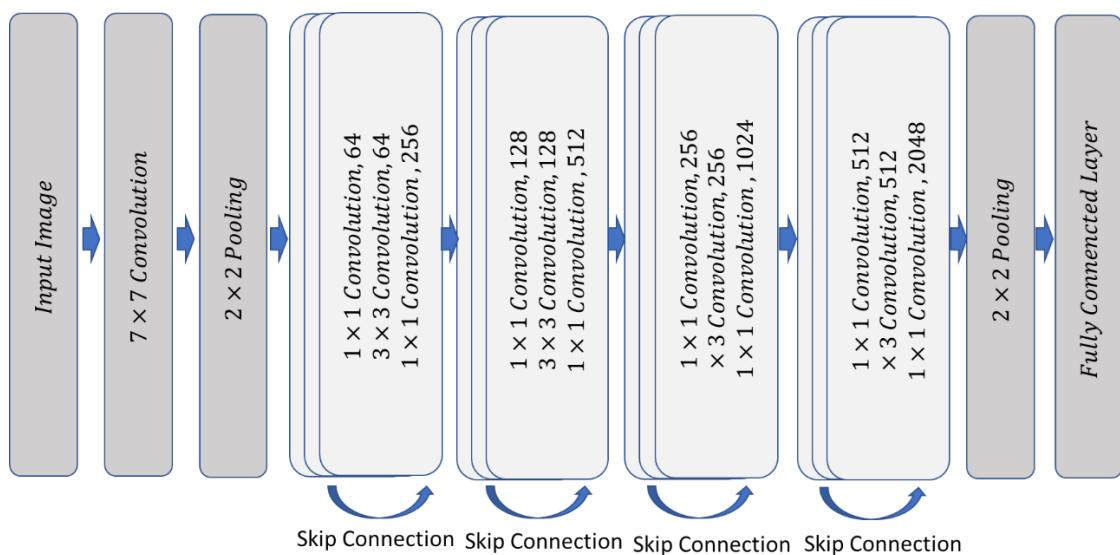


圖 2.13 Resnet50 模型架構



## 2.6 深度學習模型訓練結果

選用的深度學習模型共有兩種，包括 ResNet50 及 VGG16，而選擇使用的資料集照片型態共有三種，包括原始之裂縫資料集、經過自適應直方圖均衡化後之資料集與依序經過灰階轉換與自適應直方圖均衡化後之資料集。

訓練模型時，使用線上資料增強訓練方式(Online Augmentation)，於訓練及驗證時會於每個迭代數重新生成增強的訓練及驗證資料集，以增強模型的精度，為了驗證線上資料增強訓練方式，將原始資料集以無資料增強及有資料增強的方式進行訓練，以進行比較。其中資料增強參數包括縮放、旋轉、左右平移、上下平移、左右翻轉、上下翻轉及亮度調整，兩種模型使用同樣的資料增強設置參數，前四種參數，包括 Zoom Range 為設定圖片之縮放比例、Rotation Range 為設定圖片之旋轉角度、Width Shift Range 設定圖片的左右平移幅度、Height Shift Range 設定圖片上下平移的幅度。

其中 Zoom Range 的 0.3 為增強後之照片在原始照片 0.7 至 1.3 倍的範圍內縮放；Rotation Range 60 為增強後之照片將會在  $\pm 60^\circ$  間旋轉；Width Shift Range 0.2 為在原始照片寬度之 0.2 倍範圍內左右平移；Height Shift Range 0.2 為在原始照片高度之 0.2 倍範圍內上下平移，考慮資料集照片之裂縫圖案的裂縫寬度、位置及照片的明暗度後，所有參數值之設置如表 2.2。

資料增強的參數代表隨機參數選擇的範圍，而因為本研究結合許多不同的參數設置，因此每一項參數均會同時隨機挑選，並生成一張擁有隨機資料增強過後的裂縫圖片，依此方式於每個迭代次數後生成完整的增強資料集。此外，資料增強時如平移或是旋轉均可能使資料增強過後的圖片無法由原始圖片填滿，因此會設置填充模式，並將其設置為使用最近的像素點填充剩餘圖片的空白像素，而此模式可

以生成相對自然而不突兀的增強圖片。



表 2.2 線上資料增強參數

	Zoom Range	Rotation Range	Width Shift Range	Height Shift Range	Brightness Range	Horizontal Flip	Vertical Flip
Value	0.3	60	0.2	0.2	0.8-1.2	True	True

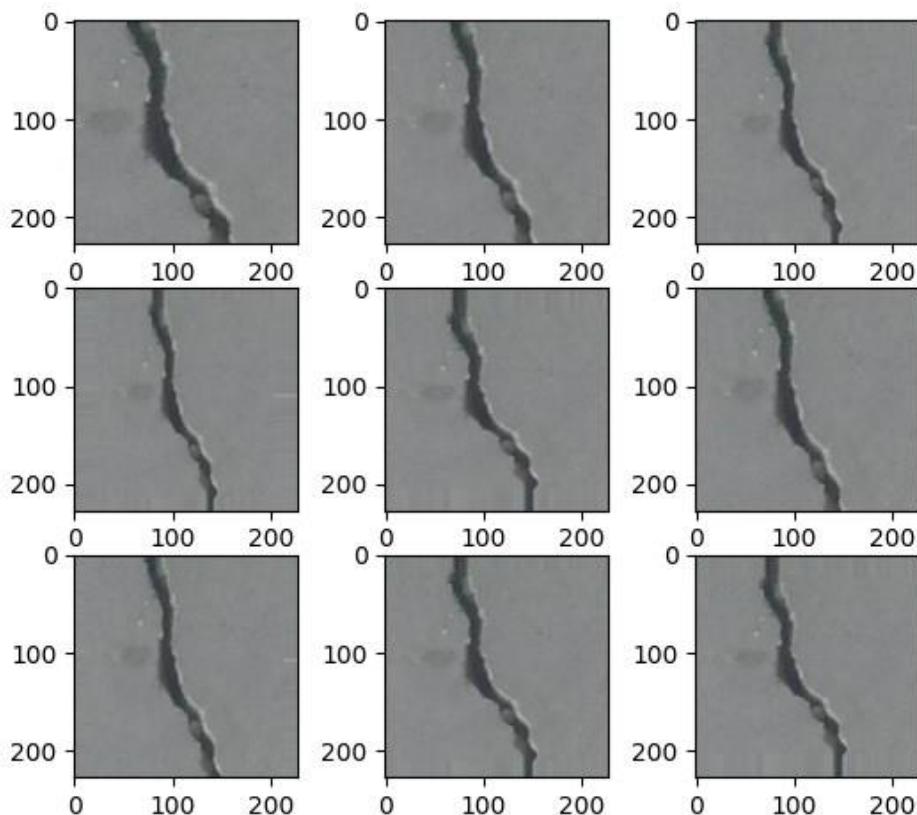


圖 2.14 Zoom Range 為 0.7 至 1.3 的裂縫圖片

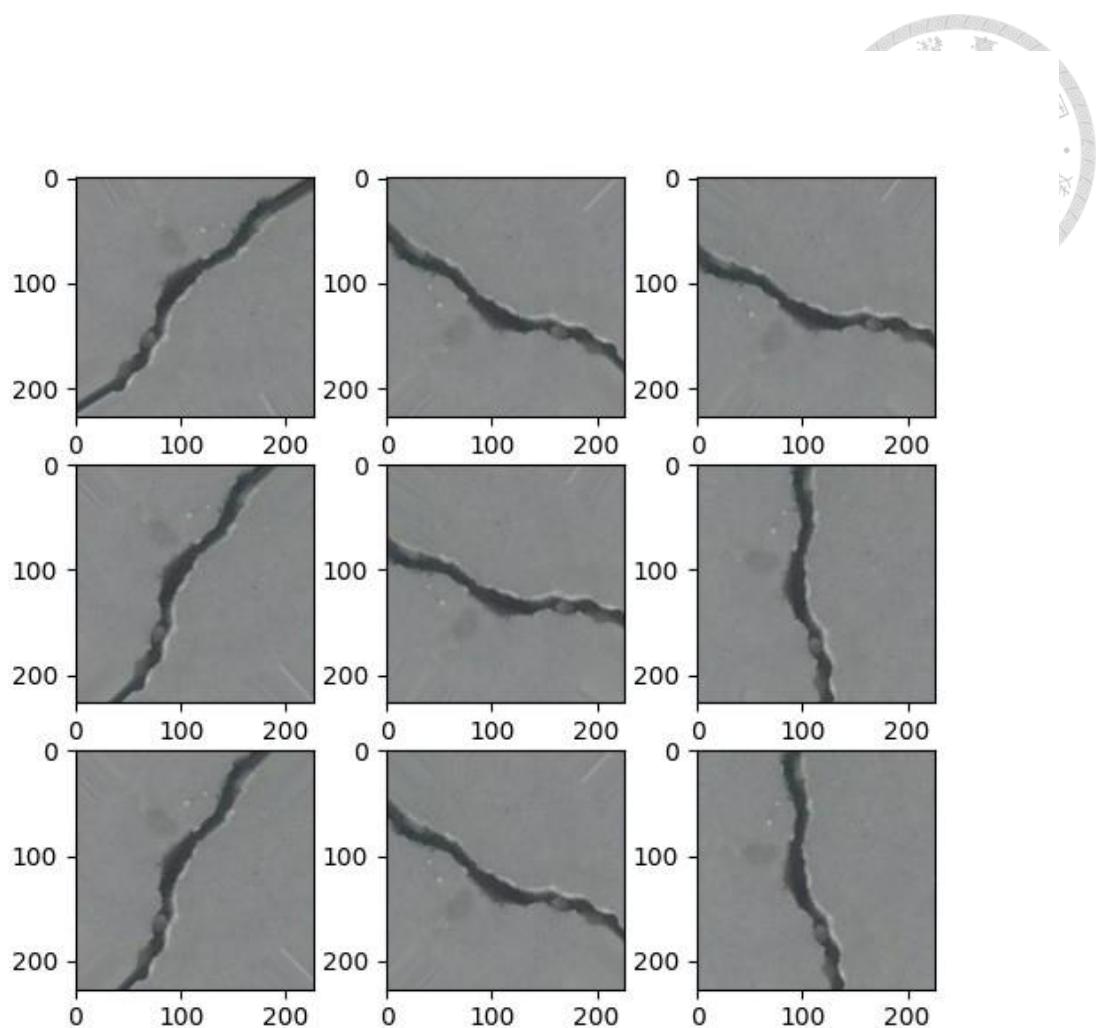


圖 2.15 Rotation Range 為 60 度的裂縫圖片

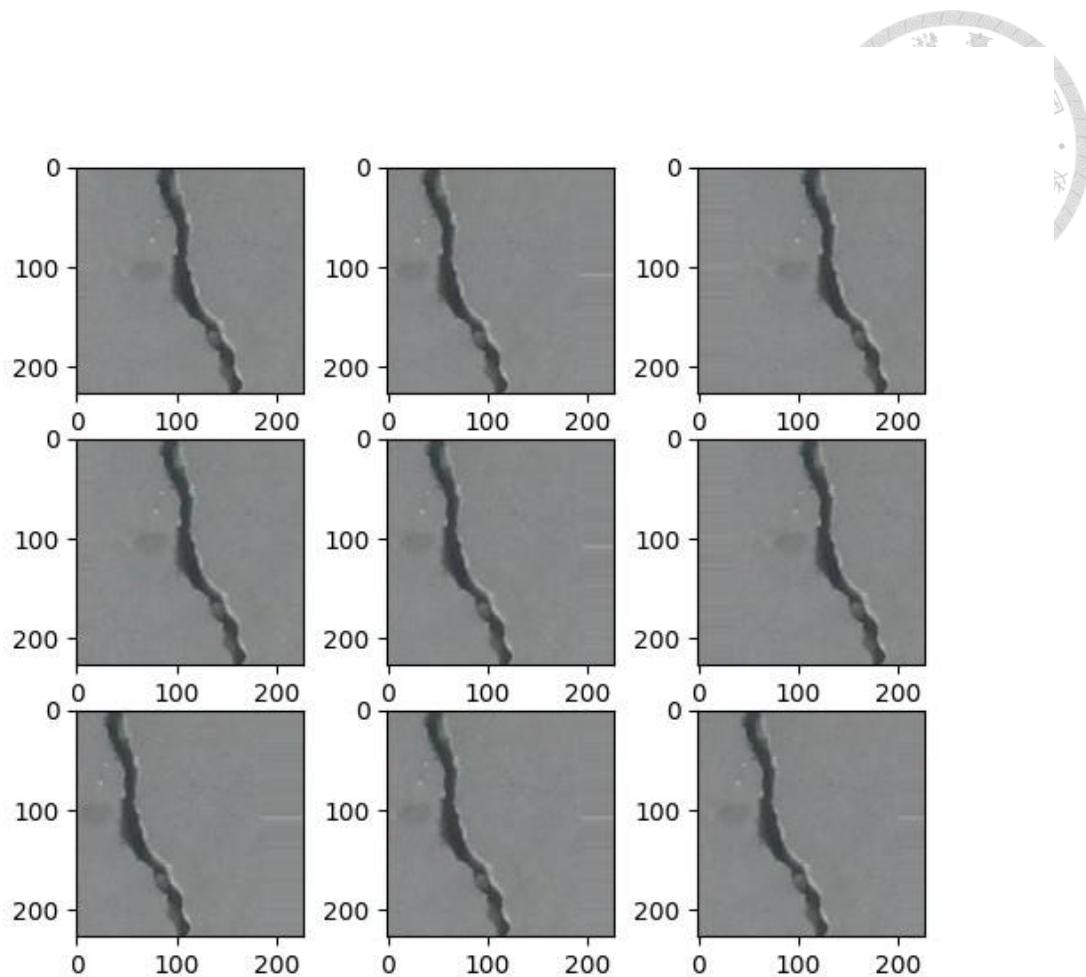


圖 2.16 Width Shift 為 0.2 的裂縫圖片

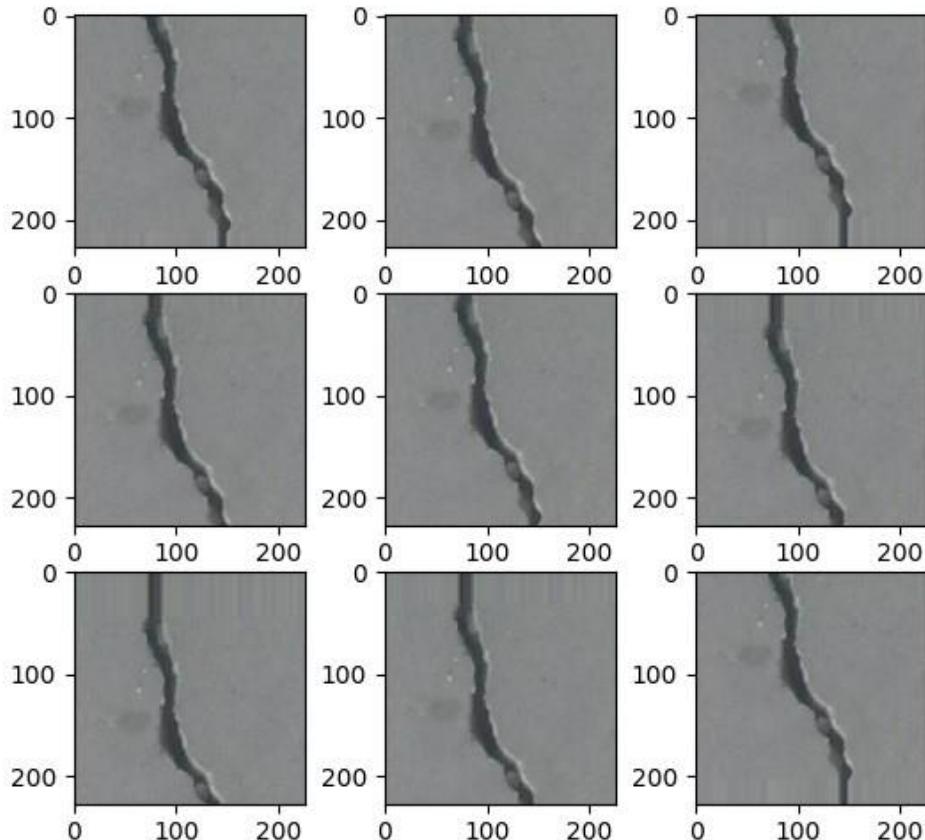


圖 2.17 Height Shift 為 0.2 的裂縫圖片

模型之精確度(Accuracy)以(2.5)式表示因為本資料集中僅有裂縫與非裂縫圖片，模型之精確度可理解為模型是否正確判斷圖像中是否存在裂縫。

$$Accuracy = \frac{\text{Correct Predictions}}{\text{All Predictions}} \quad (2.5)$$

兩種模型中之 Loss Function 均選用 Cross-Entropy，其目的為將分類模型為了判斷圖像屬於何種類別，通常產生的機率介於 0 至 1 之間，將此機率結果與實際

之標籤進行比較，以評估模型的表現。Cross-Entropy 可以  $L_{CE}$  表示為(2.6)式：

$$L_{CE} = - \sum_{c=1}^M y_{o,c} \log (p_{o,c}) \quad (2.6)$$

其中  $L_{CE}$  為計算出的損失， $M$  為總類別的數量， $y_{o,c}$  為僅具有 0 或 1 之值的變數，對於觀察次數  $o$ ，判斷標籤  $c$  時，當標籤  $c$  是正確的類別時， $y_{o,c}$  之值為 1，而  $p_{o,c}$  則為當對於標籤  $c$ ，觀察次數為  $o$  時模型判斷屬於該類別的機率。

若為僅有兩個類別的分類，則 Binary Cross-Entropy 可以表示為(2.7)式，為本次研究所使用之 Loss Function。

$$L_{CE} = -(y \log(p) + (1 - y) \log(1 - p)) \quad (2.7)$$

其中  $y$  之值為 0 或 1 的標籤， $p$  則為該模型判斷屬於標籤 1 的機率。

### 2.6.1 ResNet50

ResNet5 模型的訓練參數列於表 2.3，其中 Learning Rate 的選擇為嘗試後，考量模型可訓練程度與訓練速度後之選擇。4 種訓練組合中，包括 3 種資料型態及 1 種無資料增強的訓練結果展示於圖 2.18 至 2.21。觀察圖 2.18(a)、2.19(a)、2.20(a) 與 2.21(a)，藍色實線為訓練資料集的模型準確度驗證，而橘線為驗證資料集的準確度驗證，VGG16 模型以原始圖像進行訓練時，最多約需 25 個迭代次數達到收斂，為了使模型的訓練曲線趨於穩定，並提取其最終之訓練參數，選擇以 50 個迭代次數進行訓練後，發現無論是訓練資料集或是驗證資料集，訓練曲線都呈現放緩

趨勢，並且可以形成一個穩定的平台，代表模型之準確度最終都達到收斂，同時比較訓練資料集與驗證資料集的訓練成果，確認模型不會有過度擬合(Over Fitting)的情況發生。

觀察圖 2.18 與其他未經資料增強的圖片相比，其收斂速度明顯快於其他經過資料增強的資料集，而其模型之訓練及驗證精度也很快的趨於穩定，並擁有相對高的精度。

另外可以發現到圖 2.19 至圖 2.21 因為模型使用線上的資料增強方式訓練，橘色實線之收斂速度明顯慢於藍色實線，因為模型使用線上資料增強的方式，因此訓練資料集與驗證資料集與每個迭代次數都會產生歧異，模型需要藉由每個迭代過程修正模型的參數，才得以增加模型於不同裂縫型態的辨識率以提高驗證資料集的精確度。

對於 Loss 部分，其趨勢均與 Accuracy 相同，達到收斂後均擁有較低的值，意味模型於此資料集擁有不錯的準確性。此外可以觀察圖 2.19 與圖 2.20、2.21 之差別，圖 2.19 彩色資料集的收斂速度明顯快於後面二者，代表彩色資料集可能於訓練時具有快速收斂的優勢。

表 2.3 ResNet50 深度學習訓練參數

Optimizer	Learning Rate	Loss Function	Training Epochs	Input Size	Total Parameters
Adam	0.0001	Cross-Entropy	50	224 × 224	23,850,242

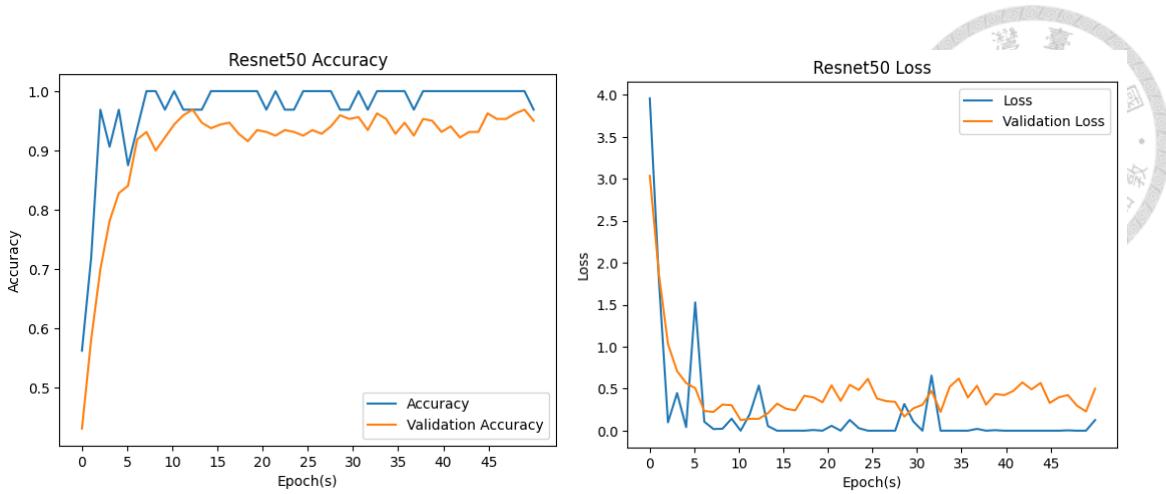


圖 2.18 原始資料集未經資料增強圖片之訓練成果

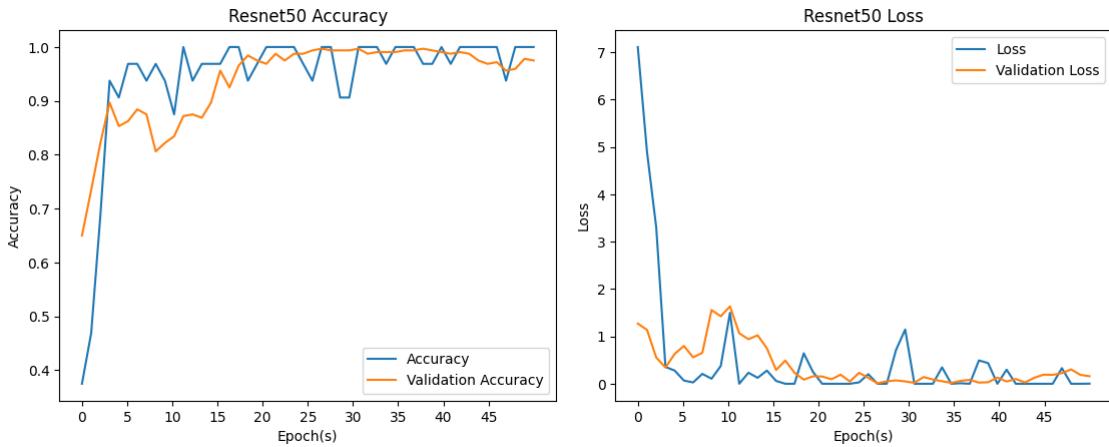


圖 2.19 原始資料集圖片之訓練成果

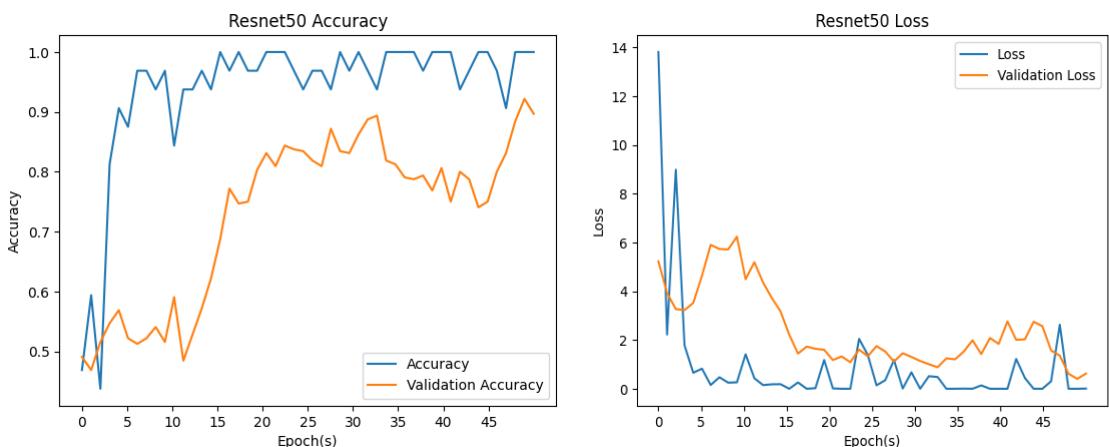


圖 2.20 灰階資料集圖片之訓練成果

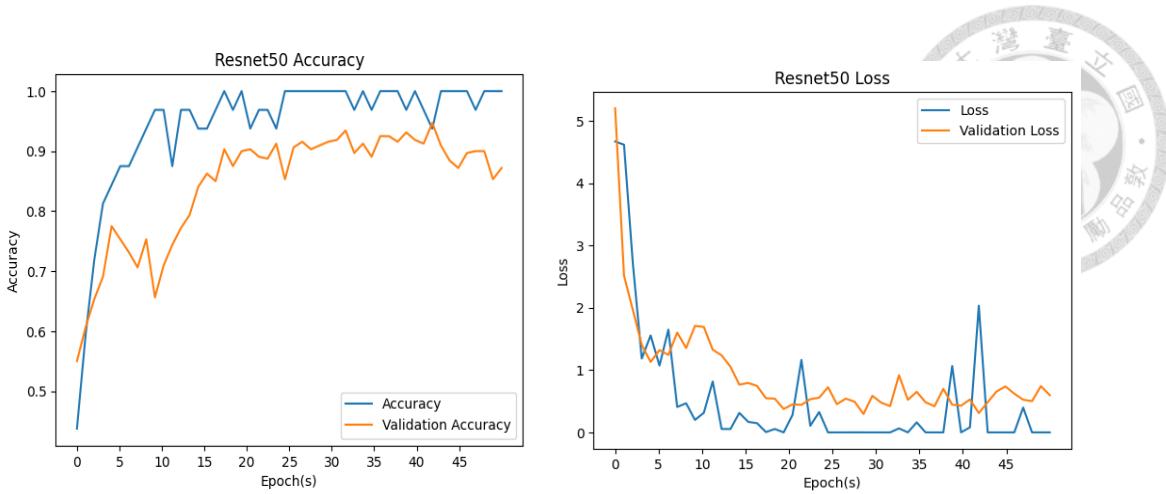


圖 2.21 灰階與 CLAHE 資料集圖片之訓練成果

## 2.6.2 VGG16

VGG16 模型的參數列於表 2.4，同樣的，Learning Rate 為考量模型可訓練程度與訓練速度後之最終選擇，訓練成果列於圖 2.22 至圖 2.25。觀察圖 2.23 至圖 2.25，ResNet50 模型於迭代次數最多約需 30 個迭代次數才會達到收斂，因此選擇以 50 個迭代次數作為總訓練迭代次數，使模型的 Training Accuracy 及 Validation Accuracy 可以趨於穩定，並以最後的迭代次數調整完的模型作為最終的模型。4 種模型，包括 3 種資料集的模型及 1 種未經資料增強的模型，經過 50 個迭代次數後都達到收斂，無過擬合的情況發生，且無論是訓練資料集或是驗證資料集的準確度均高於 0.9。

圖 2.22 因為無資料增強，因此其收斂速度快於圖 2.23 至圖 2.25，且其平台段穩定且具有相當高的精度。而圖 2.23 至圖 2.25 的訓練曲線則相對的具有波動性，於約 10 個迭代次數後，loss 函數也可下降至低點，並趨於收斂。

表 2.4 VGG16 深度學習訓練參數

Optimizer	Learning Rate	Loss Function	Training Epochs	Input Size	Total Parameters
Adam	0.00001	Cross-Entropy	50	224 × 224	33,605,442

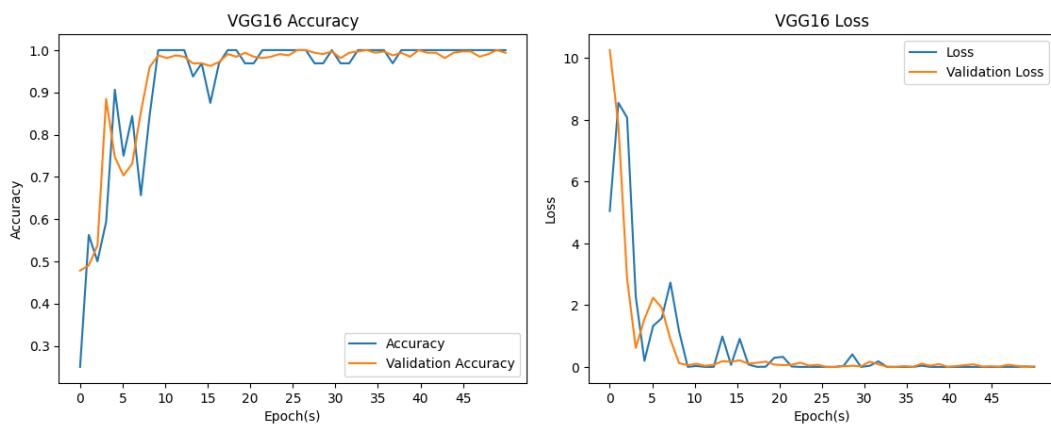


圖 2.22 原始資料集未經資料增強圖片之訓練成果

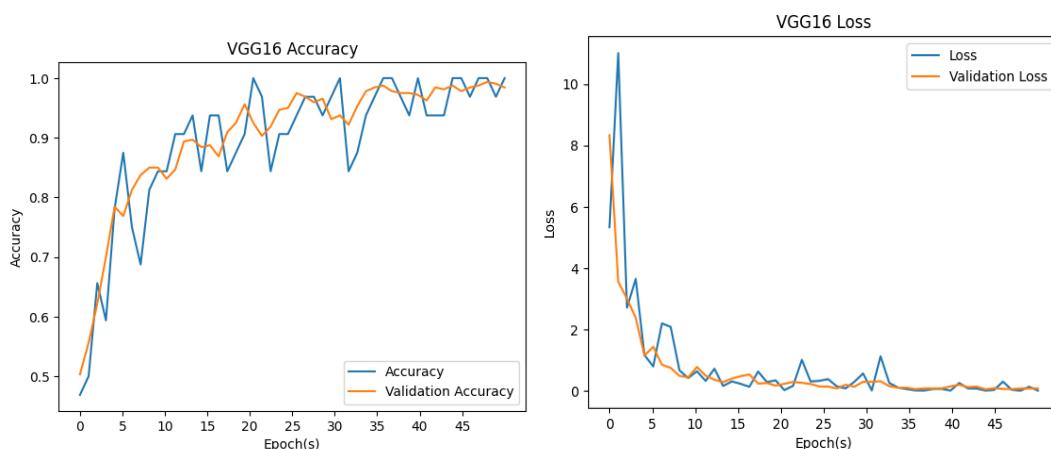


圖 2.23 原始資料集圖片之訓練成果

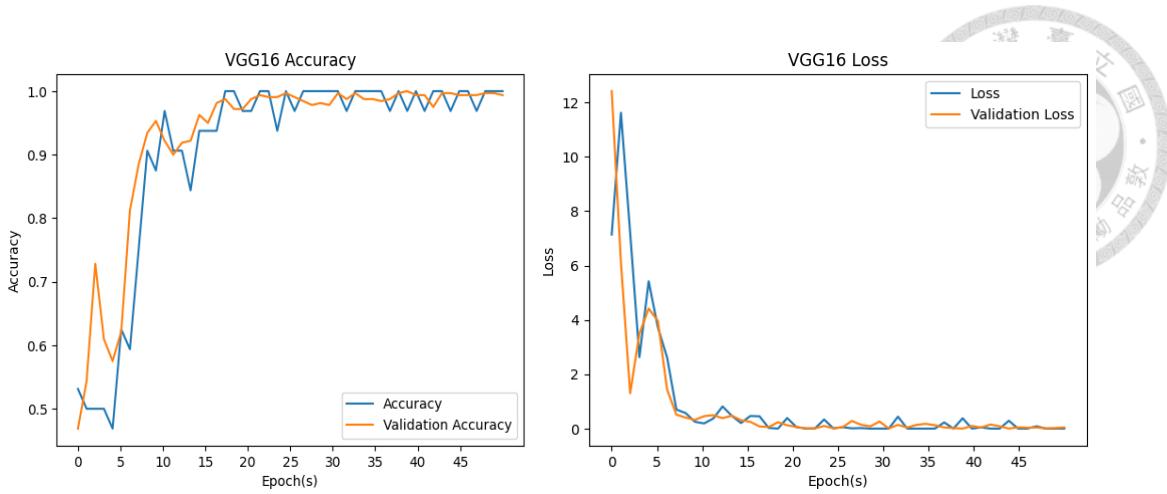


圖 2.24 灰階資料集圖片之訓練成果

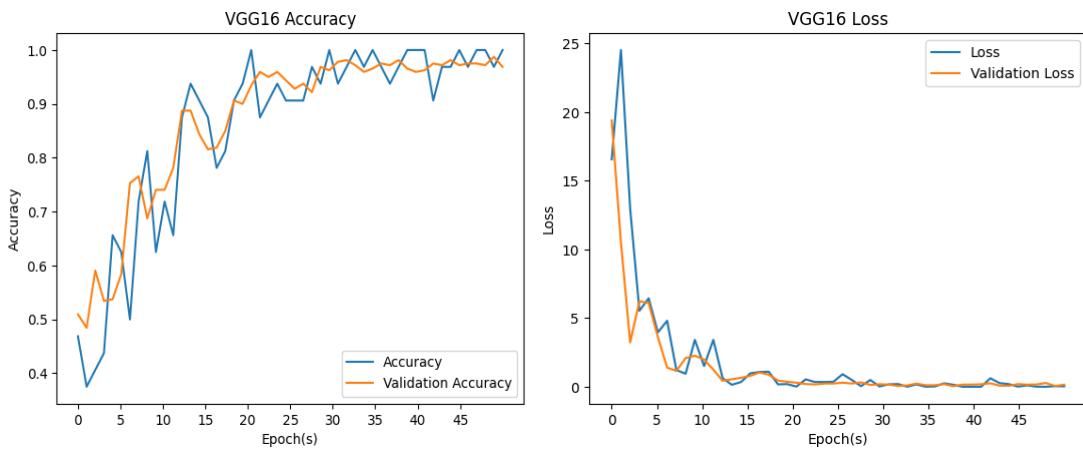


圖 2.25 灰階與 CLAHE 資料集圖片之訓練成果

## 2.7 小結

本章節主要介紹深度學習模型，並將其用於之後的裂縫辨識架構中。為了增強模型的精度，以線上資料增強增加圖片的多樣性；以彩色與灰階轉換將圖像轉換成灰階，以減少不同型態裂縫照片間的差異；以自適應直方圖均衡化(CLAHE)，提高裂縫於相片中的對比度，希望可以藉此提升模型應用時的辨識精度。

研究使用兩種模型作為裂縫辨識模型，分別為 ResNet50 及 VGG16，其中無線上資料增強的模型可以快速達到收斂，並有穩定的平台與相對高的精度，相較之下

經過線上資料增強的模型於訓練時則相對收斂較慢，但仍能趨於收斂，訓練結果如圖 2.18 至 2.25，訓練後的模型將可應用於接下來的裂縫辨識架構中。



# Chapter 3 裂縫辨識架構



## 3.1 前言

本研究提出之內容主要為建立使用深度學習之架構進行裂縫的辨識，因此本章節將詳述架構中各流程的方法與過程。裂縫辨識詳細架構如圖 3.1，以原始拍攝現場之照片作為輸入，主要步驟包括前處理、圖像區塊分割、深度學習辨識與最終的裂縫機率分佈與後處理。其中前處理的部分與深度學習訓練資料集之前處理步驟相同，目的為使辨識時之輸入照片與訓練時之訓練照片具有一致之前處理流程，而此前處理流程可以使深度學習模型具有更好的精度，以適合各種型態之裂縫，而其餘架構流程將詳細描述於此章節。

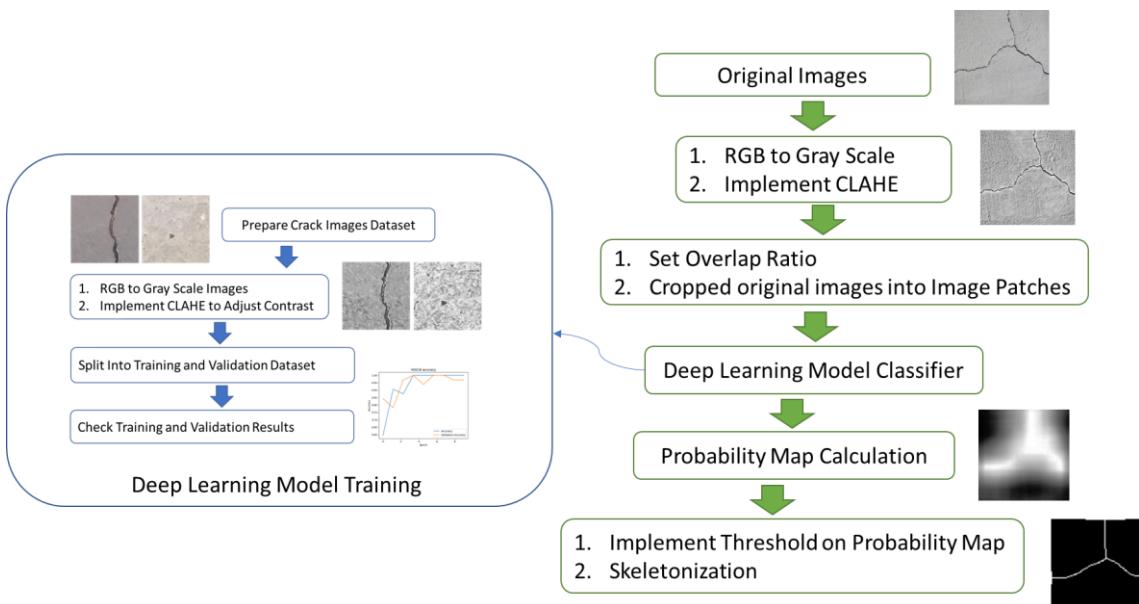


圖 3.1 裂縫辨識架構

## 3.2 原始圖片與區塊圖片轉換

本研究分別使用 VGG16 及 ResNet50 模型作為深度學習辨識器，因兩者之模

型架構之輸入影像大小皆為 224 像素，區塊圖像之大小固定為  $224 \times 224$  平方像素，以符合深度學習模型之輸入圖片尺寸。切割區塊圖像時，區塊圖像間之重疊率為其中一個重要的參數，其調整方式為改變相鄰區塊間之移動步長，如圖 3.2(a)所示，以一張原始大小為  $640 \times 640$  之原始影像為例，當移動步長為 224 時，無論左右方向或上下方向之區塊圖像均無重疊，此外若原始圖像之大小非區塊圖像長度之整數倍時，則最右邊或是最下方切割後之區塊圖像可以為非完整尺寸之區塊圖像，因此原始照片可以裁減為 9 塊區塊圖像，於模型辨識前再將其縮放至  $224 \times 224$ ，而使用的縮放方式為使用三次差值法(Cubic Interpolation)進行放大。

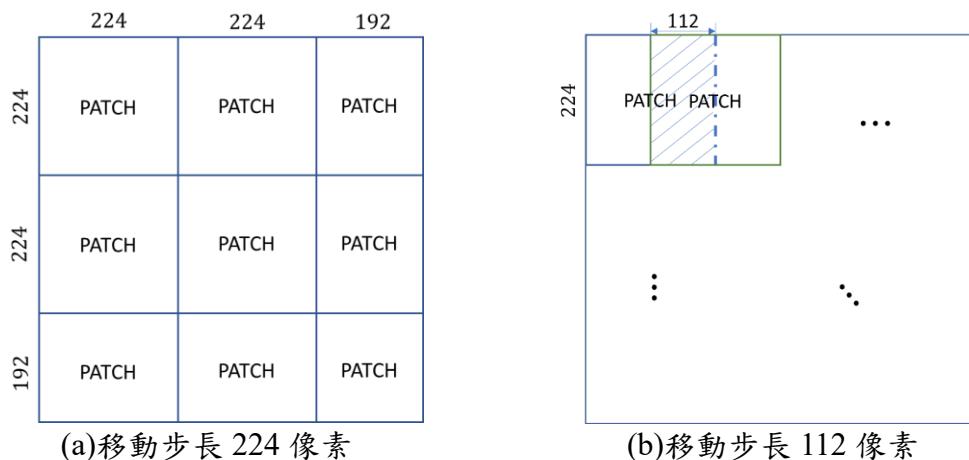


圖 3.2 區塊圖像之分割方法

無重疊率之區塊影像切割方法已由先前之學者 Alipour et al. 實踐[22]，可用於大尺寸的原始照片，並區分該區塊是否具有裂縫。然而若需要更加精確之裂縫位置，則中使用本研究之具重疊率區塊圖像辨識方法進行最佳化。

圖 3.2(b)中綠色框線為與左邊區塊圖片相鄰之區塊圖像，兩者之重疊區域為圖中斜線部分，本例中圖中重疊率為二分之一長度之區塊圖像，移動步長為 112 像素。需注意圖中僅展示左右相鄰之重疊，實際上下區塊圖像也會相鄰，依據二分之一原始區塊圖像尺寸之重疊率，與原始無重疊率之區塊圖像切割方法相比總

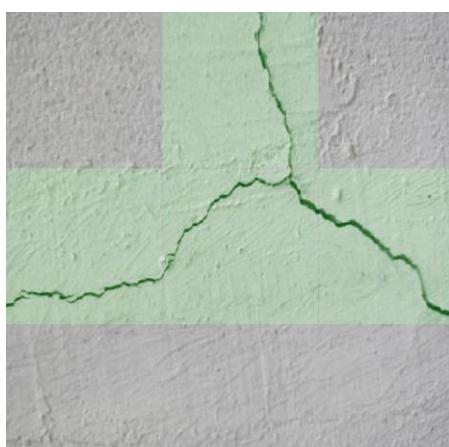
區塊圖像數量為無重疊率之 4 倍，因此 36 張區塊圖像將會被生成。

本研究之裂縫重疊長度除了前述的移動步長為 224 像素與 112 像素，還進行了 56、28 與 14 像素之測試，希望藉由減少移動步長提高區域裂縫辨識的精度，獲得更精確的裂縫位置。

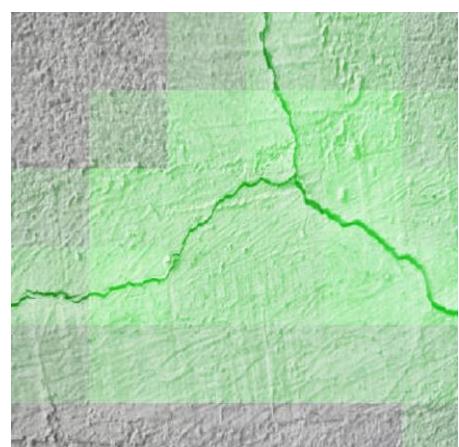
### 3.3 區塊圖片之辨識

於第三章已說明深度學習模型之訓練方式，訓練完成之深度學習辨識器將應用於架構中之裂縫辨識部分，此部分為將 3.1 節所述之裁切後之區塊圖像放入深度學習模型，以讓深度學習模型辨識該區塊圖像是否為裂縫。

圖 3.3(a)及 3.3(b)分別為無重疊率之區塊圖像辨識完成之結果與二分之一重疊率之區塊圖像之辨識結果，其中圖片中之綠色部分為辨識為裂縫之區域，可以觀察到綠色區域可以有效區分裂縫於原始圖片中的位置。觀察圖 3.3(b)的區塊圖辨識結果，相鄰區塊重疊部分可能因為相鄰區塊皆有被辨識為裂縫，因此會有重疊之綠色區塊，當綠色越深則代表該重疊區域被辨識為裂縫之次數越多，若持續將區塊圖像間之移動步長降低以提高重疊率，則可以提供更精細的裂縫位置關係，在之後之架構中可以用來計算裂縫於該區塊出現的機率與計算裂縫的實際位置。



(a) 移動步長 224 像素



(b) 移動步長 112 像素

圖 3.3 以區塊圖像之裂縫辨識結果

### 3.4 裂縫機率計算

使用深度學習模型進行裂縫辨識時，於第二章訓練的深度學習模型可以用來辨識該區塊圖像是否為裂縫，以圖 3.4(a)為例，此裂縫機率分布圖由圖 3.3(a)之辨識結果轉換而來，由於所有區塊圖片均未重疊，因此只要該區塊圖像被辨識為裂縫或非裂縫，則該區域之裂縫機率為 0 或是 1。而圖 3.4(b)則為重疊率為二分之一區塊圖像之長度之裂縫分布機率圖，其機率分布圖由 3.3(b)之辨識結果轉換而來，當重疊率為二分之一時，由於相同照片之區塊圖像次數增加為 4 倍，因此單位面積之裂縫辨識次數同樣增加為 4 倍，當該區域之照片於某個區塊圖像之辨識中被辨識為裂縫時，該區域之裂縫機率增加 0.25。

若區塊圖像間之步長持續減少，意即重疊率持續增加，則單位面積之辨識次數也會隨著增加，單位面積之重疊次數(Overlaps per unit area)可以(3.1)式表示。

$$\text{Overlaps per unit area} = \left( \frac{224(\text{pixel})}{\text{Moving step(pixel)}} \right)^2 \quad (3.1)$$

而當單位區塊被辨識為裂縫時該區域之機率增加量(Probability increment)為單位面積重疊次數之倒數，如式 3.2 所示。

$$\text{Probability increment} = \frac{1}{\text{Overlaps per unit area}} \times 100\% \quad (3.2)$$

藉由將深度學習模型對於區塊圖像中是否有裂縫之辨識結果轉換為機率，相較於單純有或無的分類結果，

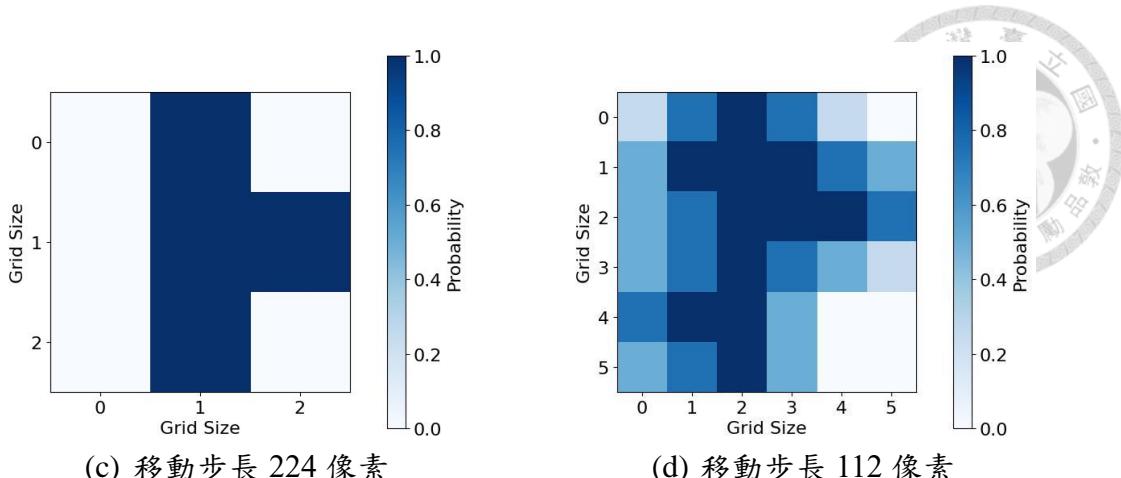


圖 3.4 裂縫機率分布圖

### 3.5 黑白數值圖

於 3.3 節中已以(3.2)式計算出裂縫的機率分布圖，機率分布圖雖可以看出裂縫的大致位置，然而若需要進行近一步裂縫位置的處理，則需要將機率分布圖轉換為比較方便處理的黑白數值圖。黑白數值圖每一個像素的值為 0 至 255，當值為 0 時，則顏色為黑色，若值為 255 則顏色為白色，若希望當裂縫機率為 0 時其值為 0，當裂縫機率為 1 時其值為 255，則可由(3.3)式轉換。

$$\text{Gray value} = \text{Probability map} \times 255 \quad (3.4)$$

其中 Gray value 為該像素轉換後的黑白數值，而 Probability map 則為該像素的屬於裂縫的機率，其值介於 0 與 1 之間。

將計算圖片中每個像素後即可將裂縫機率圖轉換為裂縫黑白數值圖，圖 3.5(a)與 3.5(b)為轉換前與轉換後的結果。

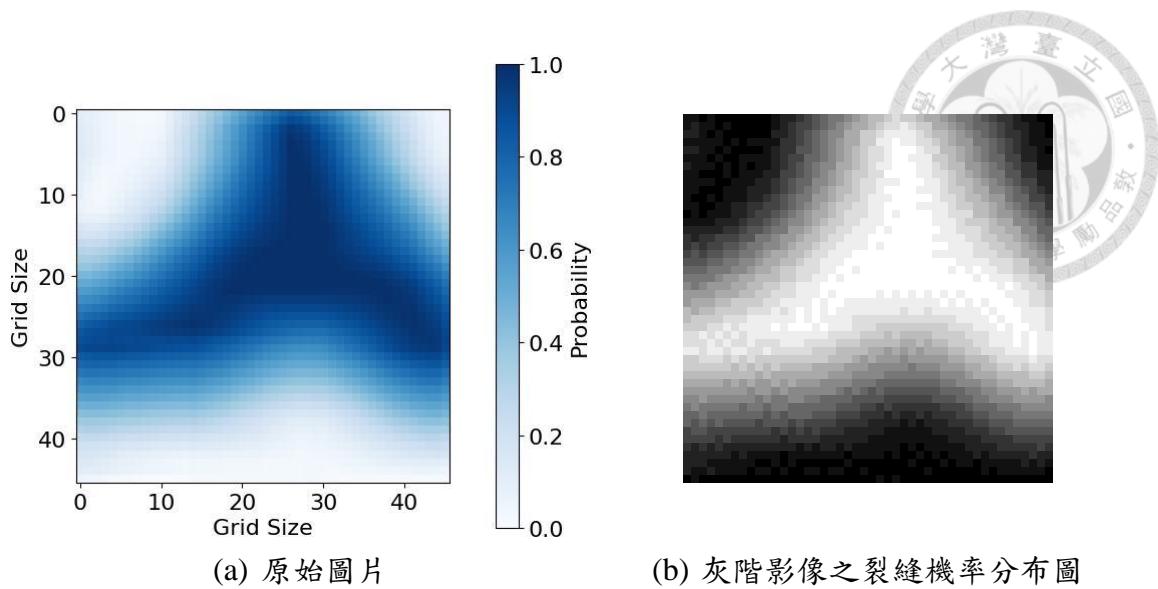


圖 3.5 產生裂縫機率分布圖之過程

### 3.6 骨架化演算法提取裂縫路徑

已計算出的裂縫黑白數值圖，如圖 3.5(b)，為了使黑白數值圖可以細化，以提取裂縫的路徑，本架構使用 Zhang-Suen 骨架化演算法 (Zhang-Suen Thinning Algorithm)[50]，此方法為最常見的骨架化演算法，提出於 1984 年，為了達到細化圖片之目的，此演算法於每個迭代此數會使用兩個步驟來移除圖片中多餘的白色像素部分，以圖 3.6(a)中的馬為例，當使用 Zhang-Suen 骨架化演算法後，圖片中大部分的白色區域都被移除，只剩下圖片中馬的輪廓。

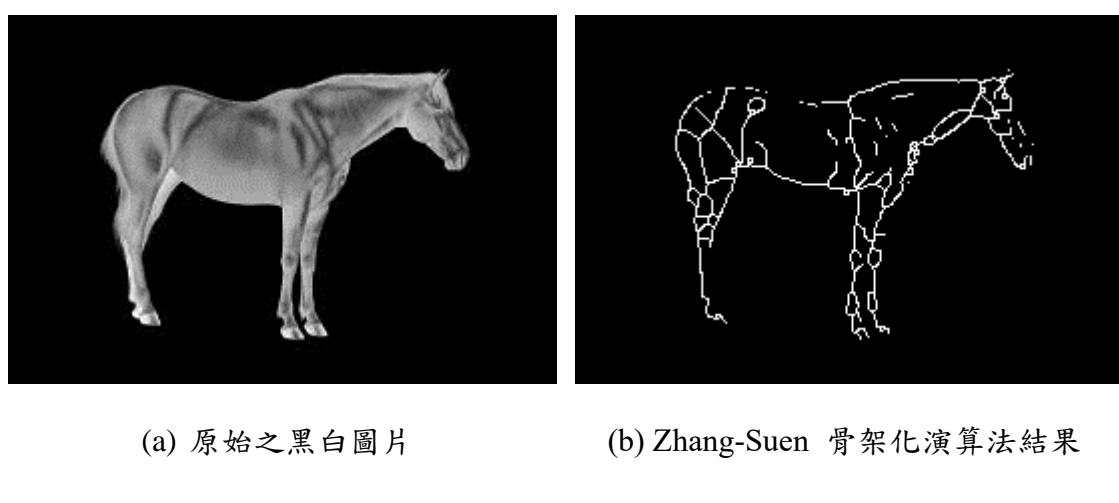


圖 3.6 使用 Zhang-Suen 骨架化演算法

Zhang-Suen 骨架化演算法是藉由不斷的迭代將圖片中白色區域移除的過程，圖 3.7 為 9 個像素組成，而目前正在處理的白色像素為 A1，為了使圖片之白色部分細化，此演算法正考量是否需要將 A1 轉換為黑色。若需要解釋此演算法則需要先介紹兩個參數，分別為  $I(A1)$  與  $J(A1)$ 。

其中  $I(A1)$  之定義為由環繞 A1，由 A2 順時針出發，經過 A3、A4、A5、A6、A7、A8、A9 最終回到 A2 時，像素顏色由黑至白的轉換次數。而  $J(A1)$  則為環繞 A1 的像素中白色像素的數量，意即 A2 至 A9 中白色像素的數量。

A9	A2	A3
A8	A1	A4
A7	A6	A5

圖 3.7 像素 A1 與周圍其他像素之關係

接著，需每個迭代次數中，Zhang-Suen 骨架化演算法都會使用到兩個步驟，分別為 STEP1 與 STEP2。

STEP1：

若同時滿足以下 5 個條件，則 A1 會由白色轉換為黑色，此步驟會將所有圖像中的白色像素判斷一次，條件為：

1. 當此像素是白色時(A1 為白色)，且其周圍擁有 8 個其他像素(A2 至 A9)。

2. 周圍像素微白色的數量為 2 至 6 之間，以(3.6)式表示為：

$$2 \leq J(A1) \leq 6$$

(3.6)

3. 周圍像素由黑轉換為白的次數為 1 次，以(3.7)式表示：

$$I(A1) = 1$$

(3.6)

4. A2、A4 與 A6 中其中一個像素為黑色。

5. A4、A6 與 A8 中其中一個像素為黑色。

若同時滿足以上之 5 個條件，則 A1 轉換為黑色。

STEP2：

步驟 1 至 3 與 STEP1 相同，其中步驟 4 與 5 為：

4. A2、A4 與 A8 中其中一個像素為黑色。

5. A2、A6 與 A8 中其中一個像素為黑色。

同樣的，若同時滿足以上之 5 個條件，則 A1 轉換為黑色。

STEP1 與 STEP2 會於所有圖像中的白色像素中重複操作，直到圖像中之所有像素值都不再因為 STEP1 或 STEP2 更改為止，此時演算法收斂。

# Chapter 4 裂縫辨識應用於實際場域



## 4.1 前言

於本節首先使用 9 張自行拍攝之裂縫圖片及 Kaggle 公開資料集(Concrete Crack Segmentation Dataset)的 445 張大型裂縫圖片進行驗證。於此步驟中用以驗證的 9 張照片及大型裂縫圖片可以視為除了二章中除了用來訓練模型的 Training Set 及來驗證模型表現的 Validation Set 外的 Test Set。

9 張裂縫的原始圖片如圖 4.1 所示，由左至右，由上至下編號依序為 1 至 9，其大小均為  $640 \times 640$ 。為了驗證 3 種照片型態(原始、灰階、灰階與 CLAHE)於辨識結果精度上的差異，9 張照片也會經過前處理，將其轉換為灰階照片與灰階與經過 CLAHE 之圖片，圖 4.1 為 9 張原始照片，圖 4.2 為彩色圖案轉換為灰階的照片，圖 4.3 則為轉換為灰階圖案並使用 CLAHE 的照片。

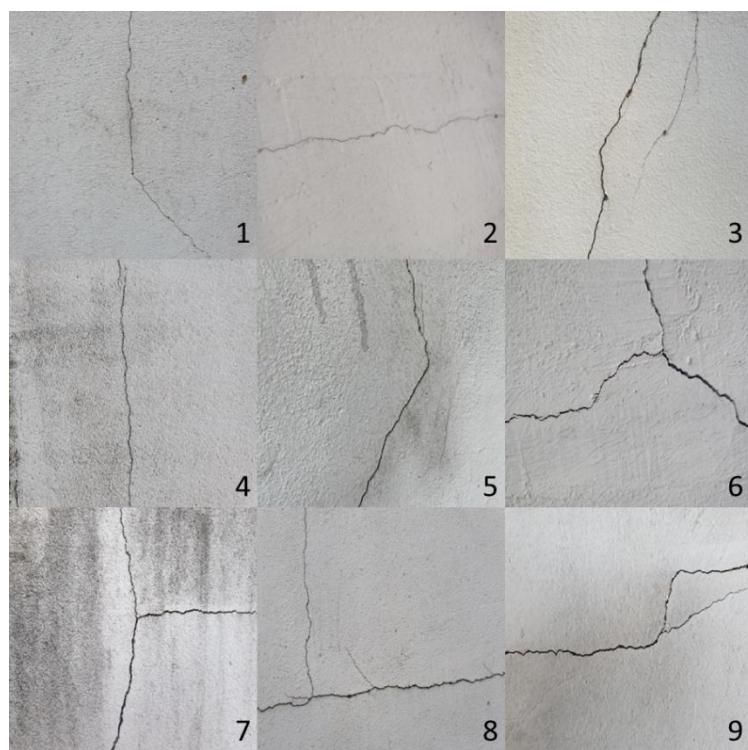


圖 4.1 自行拍攝之裂縫圖片

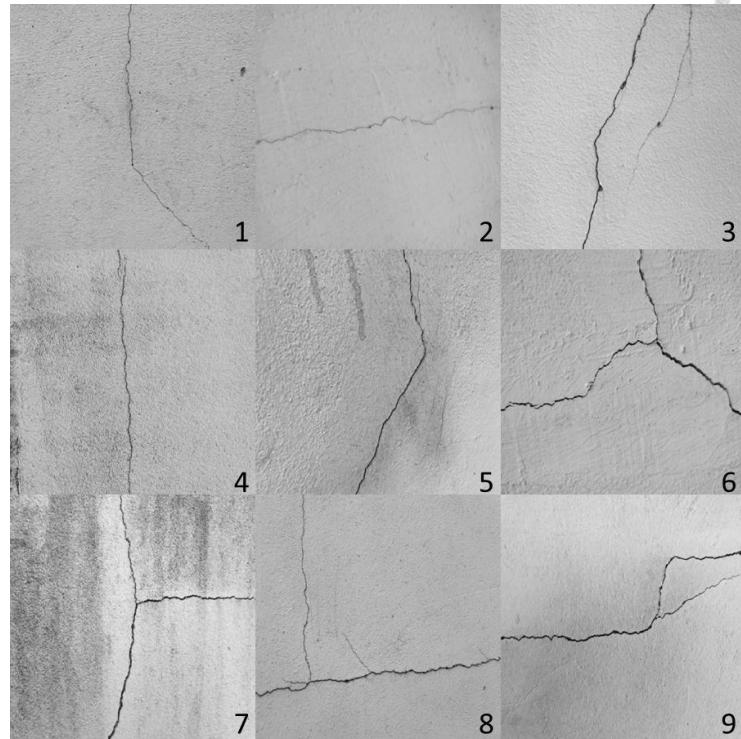


圖 4.2 經過灰階轉換的裂縫圖片

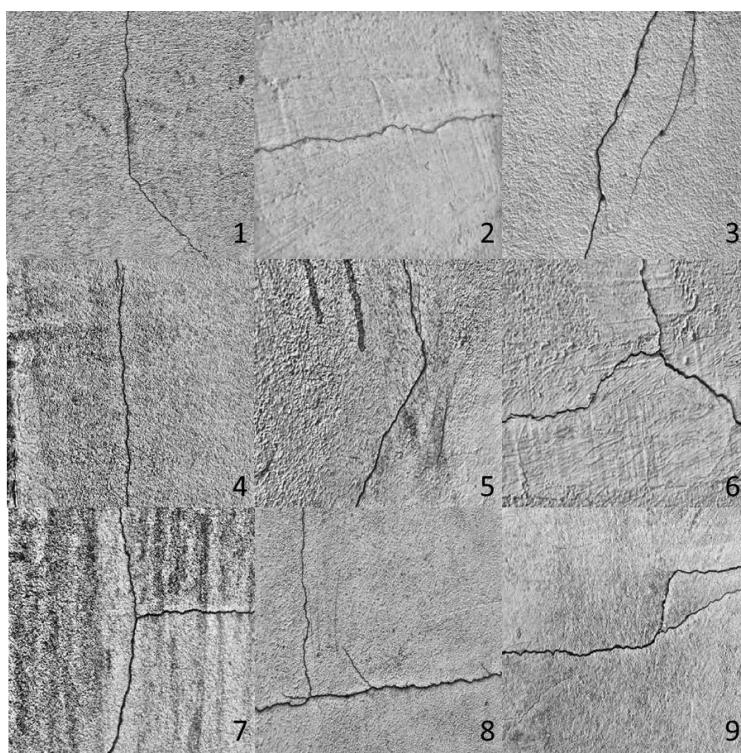
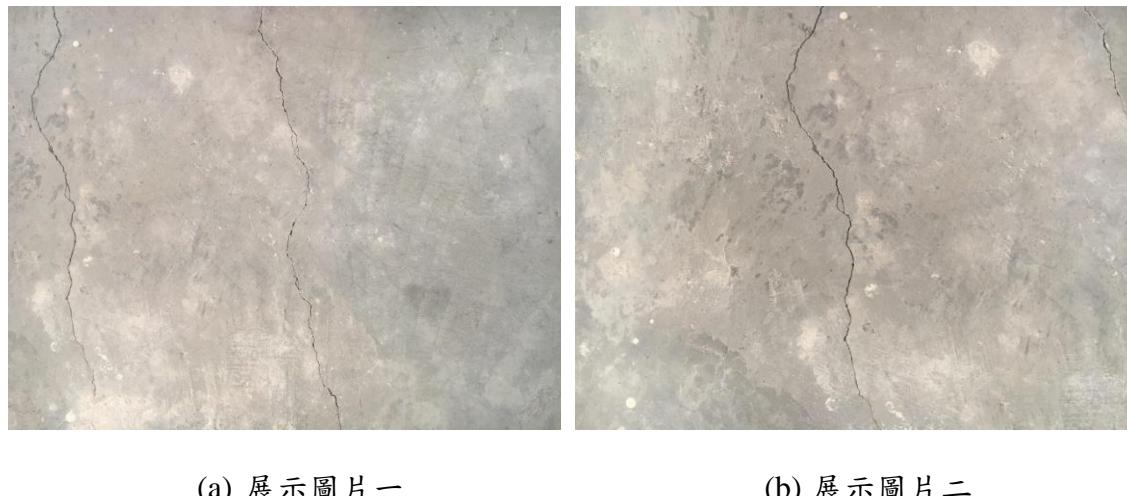


圖 4.3 經過灰階轉換與直方圖均衡化的裂縫圖片

9張現場拍攝之照片將以三種型態的測試照片以兩種深度學習模型進行驗證，  
共 6 種組合之辨識成果，並使用無重疊率的相鄰區塊圖像設置選取最佳的組合。  
以 9 張現場照片評估過何種模型及照片型態擁有最高的精度後，以該照片型態與  
深度學習模型進行驗證。完成初步的深度學習模型驗證後，將再以 9 張現場拍攝  
之照片進行本研究提出之後續裂縫辨識流程，包含裂縫機率分布與裂縫實際位置  
之辨識。

大型裂縫照片的圖像展示於圖 4.4，該資料集擁有 445 張大型裂縫圖片，其尺  
寸包含  $4032 \times 3024$  像素與  $2448 \times 3264$  像素，以無重疊率之相鄰區塊圖像進行切  
割，切割完成後擁有區塊圖像總數 110,400，此資料集僅應用於無重疊率時之裂縫  
精度量化。



(a) 展示圖片一

(b) 展示圖片二

圖 4.4 大型裂縫照片

## 4.2 深度學習模型初步驗證

為了驗證深度學習模型於訓練完成後於區塊圖像辨識裂縫的能力，以單位步

長為 224 無重疊率之區塊圖像進行驗證並計算模型之精度，並先對裂縫照片進行位置標註，本研究使用 LabelMe 作為標註工具，9 張自行拍攝裂縫照片標註結果如圖 4.5，而大型裂縫照片的標註如圖 4.6。由於本研究僅需判斷裂縫是否位於區塊圖像中，因此裂縫位置僅大略標註，以利後續的驗證模型驗證。

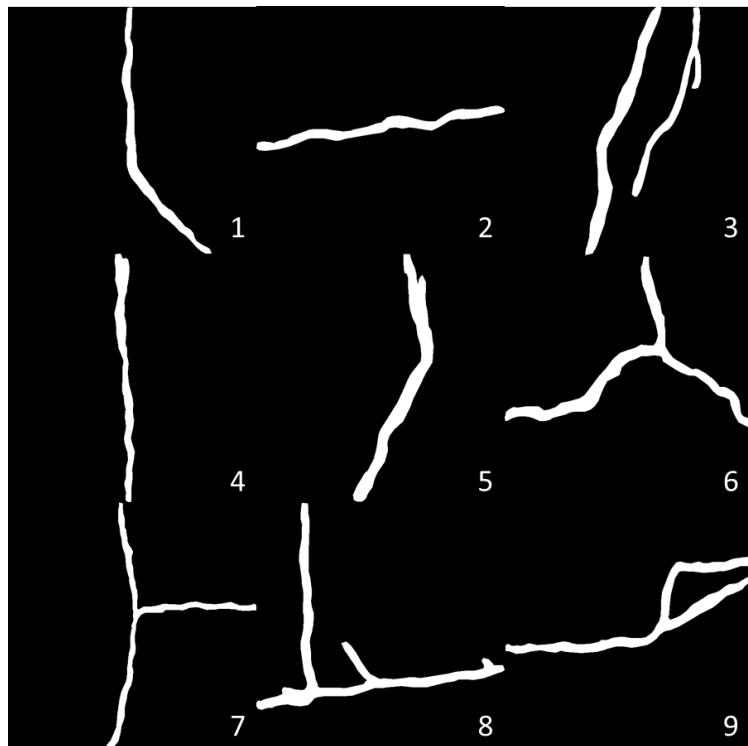
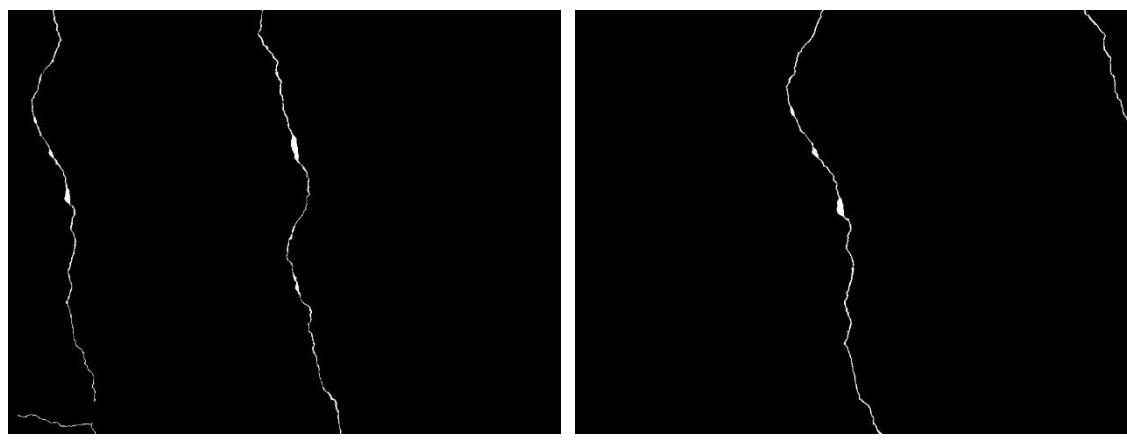


圖 4.5 9 張裂縫圖片標註結果



(a) 展示圖片一

(b) 展示圖片二

圖 4.6 大型裂縫圖片標註

將裂縫位置標註結果與深度學習模型對區塊圖像中是否含有裂縫之辨識結果進行比較，以驗證模型之精確性。兩者比較之結果可區分為4種類別，分別為 TP(true positive)、TN(true negative)、FP(false positive)與 FN(false negative)，下表詳述四種情況之情形。

表 4.1 裂縫精度計算之四種情況

Conditions	區塊圖像是否辨識為裂縫 (預測結果)	區塊圖像是否有裂縫標註 (真值)
TP(True Positive)	Yes	Yes
TN(True Negative)	No	No
FP(False Positive)	Yes	No
FN(False Negative)	No	Yes

裂縫辨識結果使用三種計算方式量化，分別為 Precision1、Precision2 與 Recall，如(4.1)、(4.2)及(4.3)式，其中 Precision1 為考量TN(True Negative)之計算結果，而 Precision2 則為不考量TN之計算結果，計算 Precision2 之目的為當裂縫照片相當大時，排除非裂縫之區塊圖像，以避免計算 Precision 時因TN數量過多，造成 FP(False Positive)或 FN(False Negative)對 Precision 的影響過低，而 Recall 則為考量裂縫回收率的指標。

$$Precision1 = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Precision2 = \frac{TP}{TP + FP + FN}$$



$$Recall = \frac{TP}{TP + FN}$$

(4.3)

於先前章節提及，本研究使用 2 種模型、4 種訓練資料型態(包括有 3 種有資料增強與 1 種無資料增強)，共八種組合進行比較，其中模型架構分別為 VGG16 及 ResNet50，而訓練資料型態分別為，原始無進行資料增強之裂縫圖案、原始之裂縫圖案、灰階之裂縫圖案與依序經過灰階與自適應直方圖均衡化(CLAHE)之資料集進行訓練及驗證，表 4.2 至表 4.9 為八種組合之辨識精度計算結果，圖 4.7 至圖 4.14 則為裂縫之視覺化辨識成果。為了觀察深度學習辨識結果，可以觀察辨識後並加上綠色區塊的裂縫圖片，以圖 4.8 的第一張裂縫圖片為例，原始之裂縫照片裂縫為垂直走向，而辨識出的綠色方塊也同樣的覆蓋了大多數的裂縫區塊，這些區塊定義為 TP(True Positive)，然而右下角仍有一塊裂縫區塊未被成功辨識，因此區塊會被定義為 FN(False Negative)，而此圖中之 TN(True Negative)為 5 張，FN(False Negative)為 0 張，因此結果紀錄於表 4.3 中，並以(4.1)、(4.2)、(4.3)式計算 Precision 與 Recall。

表 4.2 原始圖像使用 VGG16 模型未經資料增強裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	0	0	3	0	0	3	3	4	3
<b>TN</b>	5	6	3	6	4	5	5	2	4
<b>FP</b>	0	0	0	0	0	0	0	0	0
<b>FN</b>	4	3	3	3	5	1	1	3	2
<b>Precision1</b>	0.44	0.67	0.67	0.67	0.44	0.89	0.89	0.67	0.78
<b>Precision2</b>	0.00	0.00	0.50	0.00	0.00	0.75	0.75	0.57	0.60
<b>Recall</b>	0.00	0.00	0.50	0.00	0.00	0.75	0.75	0.57	0.60

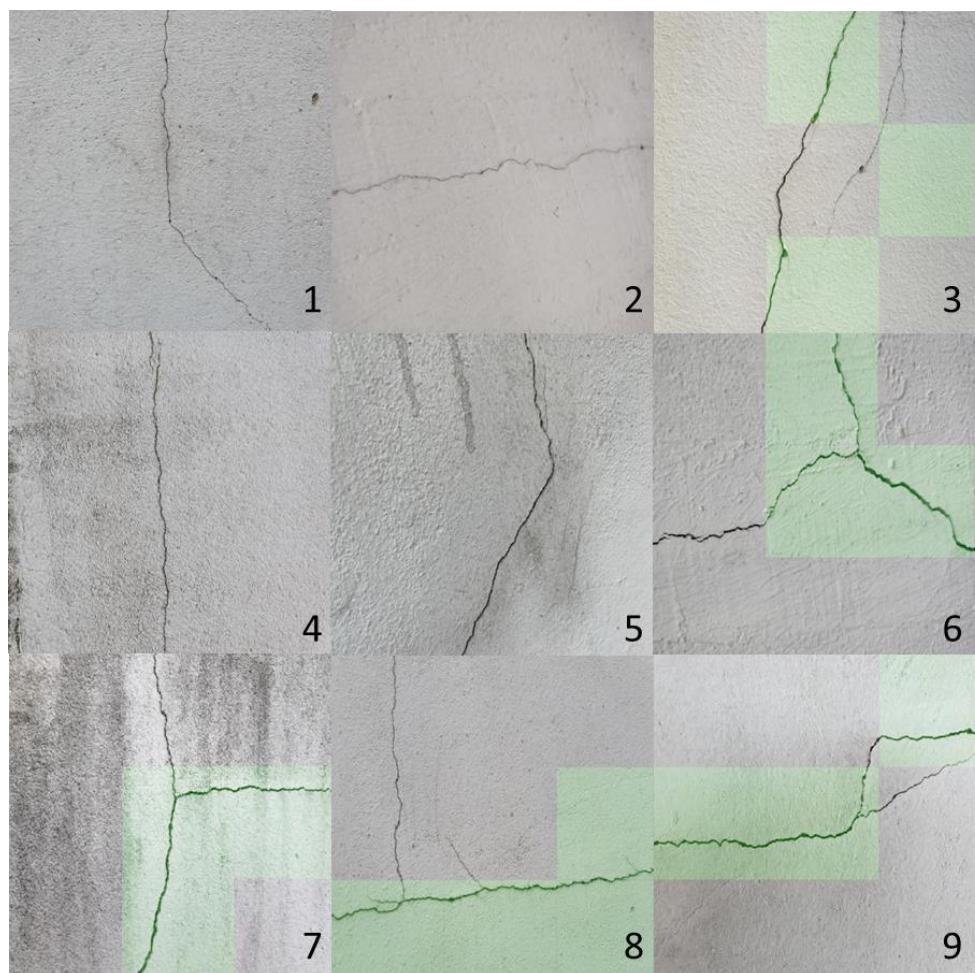


圖 4.7 原始圖像使用 VGG16 模型未經資料增強裂縫辨識結果

表 4.3 原始圖像使用 ResNet50 模型未經資料增強裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	2	2	4	3	3	4	4	6	5
<b>TN</b>	5	6	3	6	4	5	5	2	4
<b>FP</b>	0	0	0	0	0	0	0	0	0
<b>FN</b>	2	1	2	0	2	0	0	1	0
<b>Precision1</b>	0.78	0.89	0.78	1.00	0.78	1.00	1.00	0.89	1.00
<b>Precision2</b>	0.50	0.67	0.67	1.00	0.60	1.00	1.00	0.86	1.00
<b>Recall</b>	0.50	0.67	0.67	1.00	0.60	1.00	1.00	0.86	1.00

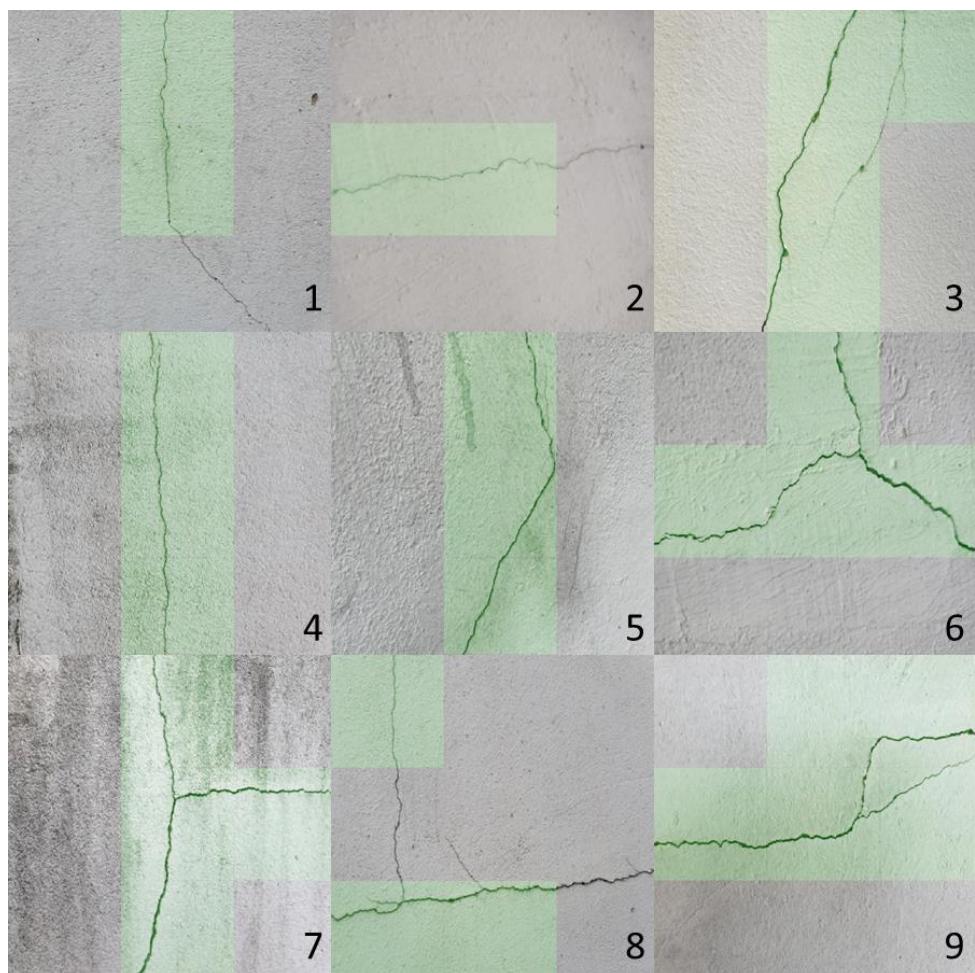


圖 4.8 原始圖像使用 ResNet50 模型未經資料增強裂縫辨識結果

表 4.4 原始圖像使用 VGG16 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	3	3	4	1	2	4	4	5	3
<b>TN</b>	5	6	3	6	4	5	5	2	4
<b>FP</b>	0	0	0	0	0	0	0	0	0
<b>FN</b>	1	0	2	2	3	0	0	2	2
<b>Precision1</b>	0.89	1.00	0.78	0.78	0.67	1.00	1.00	0.78	0.78
<b>Precision2</b>	0.75	1.00	0.67	0.33	0.40	1.00	1.00	0.71	0.60
<b>Recall</b>	0.75	1.00	0.67	0.33	0.40	1.00	1.00	0.71	0.60

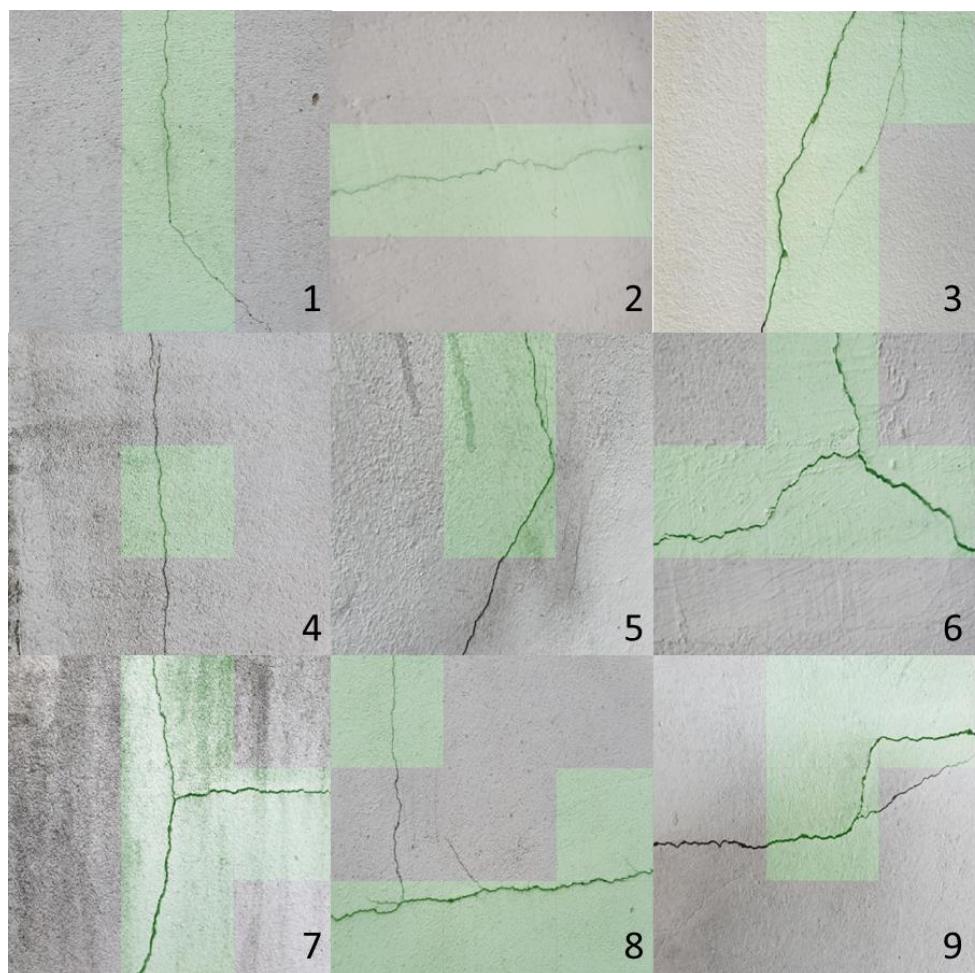


圖 4.9 原始圖像使用 VGG16 模型裂縫辨識結果

表 4.5 原始圖像使用 ResNet50 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	4	2	5	3	3	4	4	7	5
<b>TN</b>	5	6	3	5	4	4	5	2	4
<b>FP</b>	0	0	0	1	0	1	0	0	0
<b>FN</b>	0	1	1	0	2	0	0	0	0
<b>Precision1</b>	1.00	0.89	1.00	0.89	0.78	0.89	1.00	1.00	1.00
<b>Precision2</b>	1.00	0.67	0.83	0.75	0.60	0.80	1.00	1.00	1.00
<b>Recall</b>	1.00	0.67	0.83	1.00	0.60	1.00	1.00	1.00	1.00

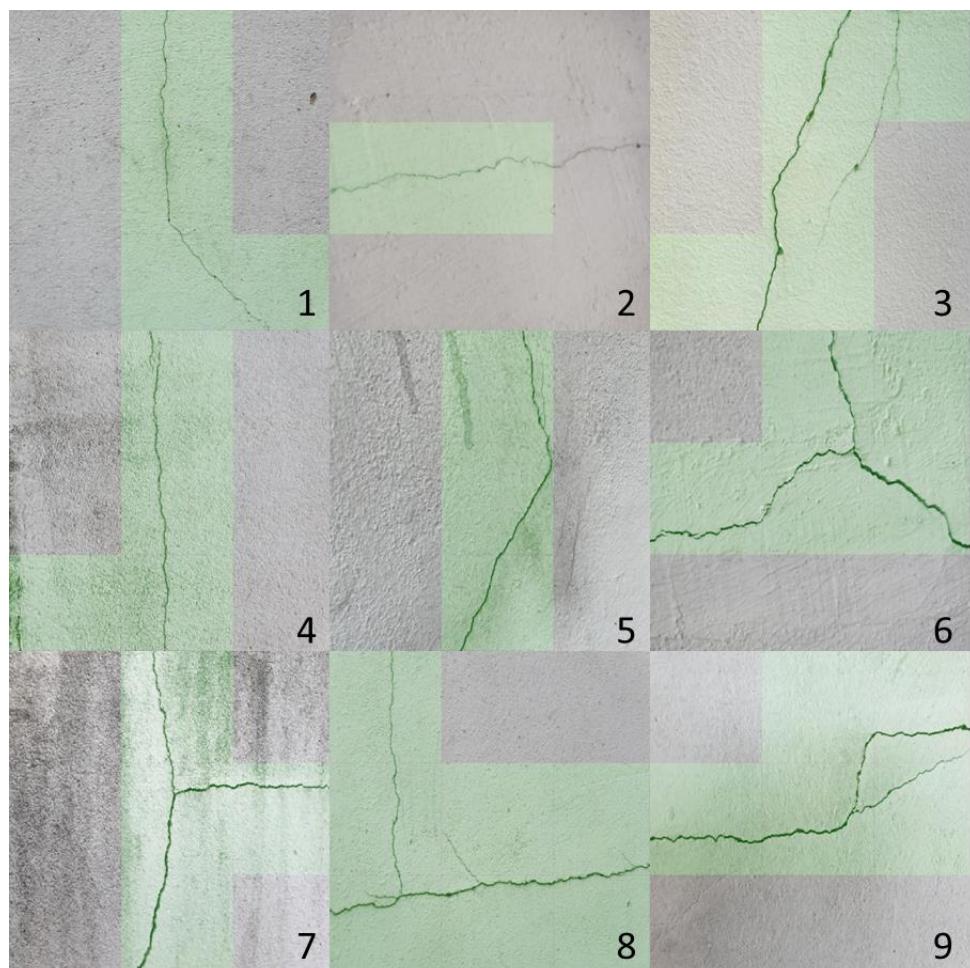


圖 4.10 原始圖像使用 ResNet50 模型裂縫辨識結果

表 4.6 灰階圖像使用 VGG16 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	1	3	6	0	1	4	3	4	5
<b>TN</b>	5	6	3	6	4	5	5	2	4
<b>FP</b>	0	0	0	0	0	0	0	0	0
<b>FN</b>	3	0	0	3	4	0	1	3	0
<b>Precision1</b>	0.67	1.00	1.00	0.67	0.56	1.00	0.78	0.67	1.00
<b>Precision2</b>	0.25	1.00	1.00	0.00	0.20	1.00	0.75	0.57	1.00
<b>Recall</b>	0.25	1.00	1.00	0.00	0.20	1.00	0.75	0.57	1.00

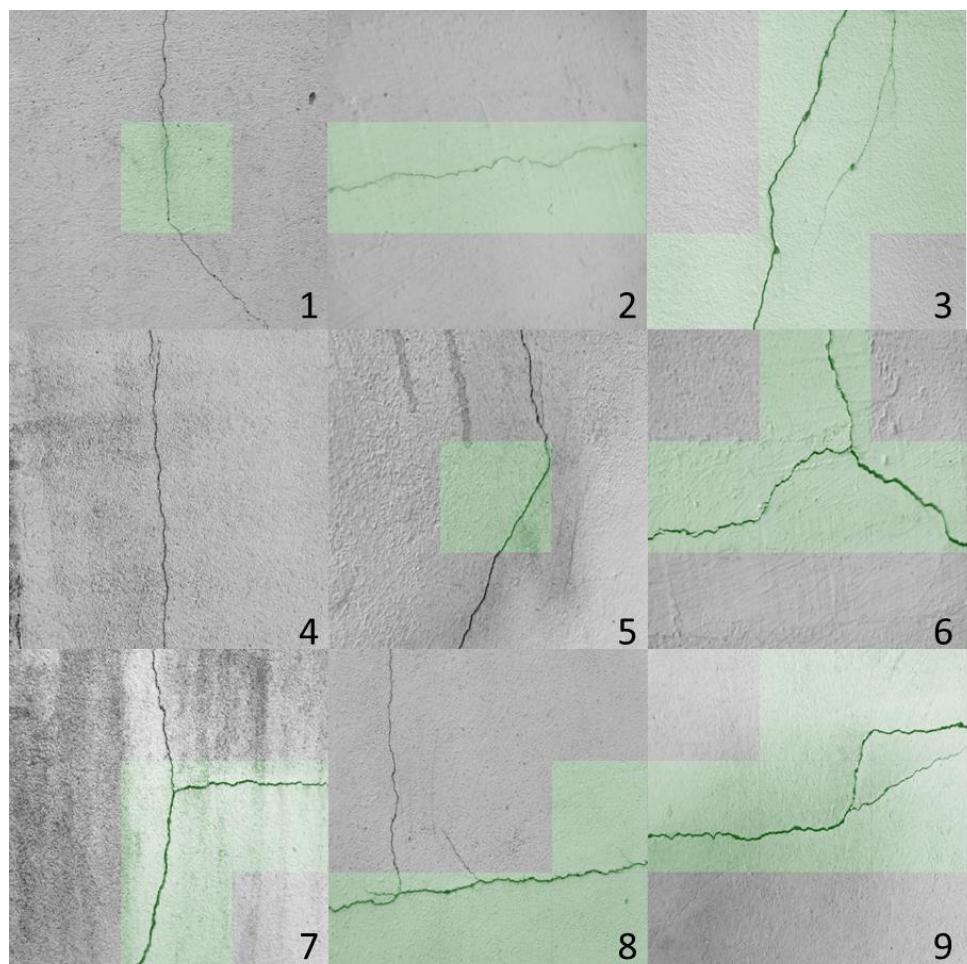


圖 4.11 灰階圖像使用 VGG16 模型裂縫辨識結果

表 4.7 灰階圖像使用 ResNet50 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	4	2	4	3	3	4	4	6	5
<b>TN</b>	5	6	3	2	0	5	1	2	2
<b>FP</b>	0	0	0	4	4	0	4	0	2
<b>FN</b>	0	1	2	0	2	0	0	1	0
<b>Precision1</b>	1.00	0.89	0.78	0.56	0.33	1.00	0.56	0.89	0.78
<b>Precision2</b>	1.00	0.67	0.67	0.43	0.33	1.00	0.50	0.86	0.71
<b>Recall</b>	1.00	0.67	0.67	1.00	0.60	1.00	1.00	0.86	1.00

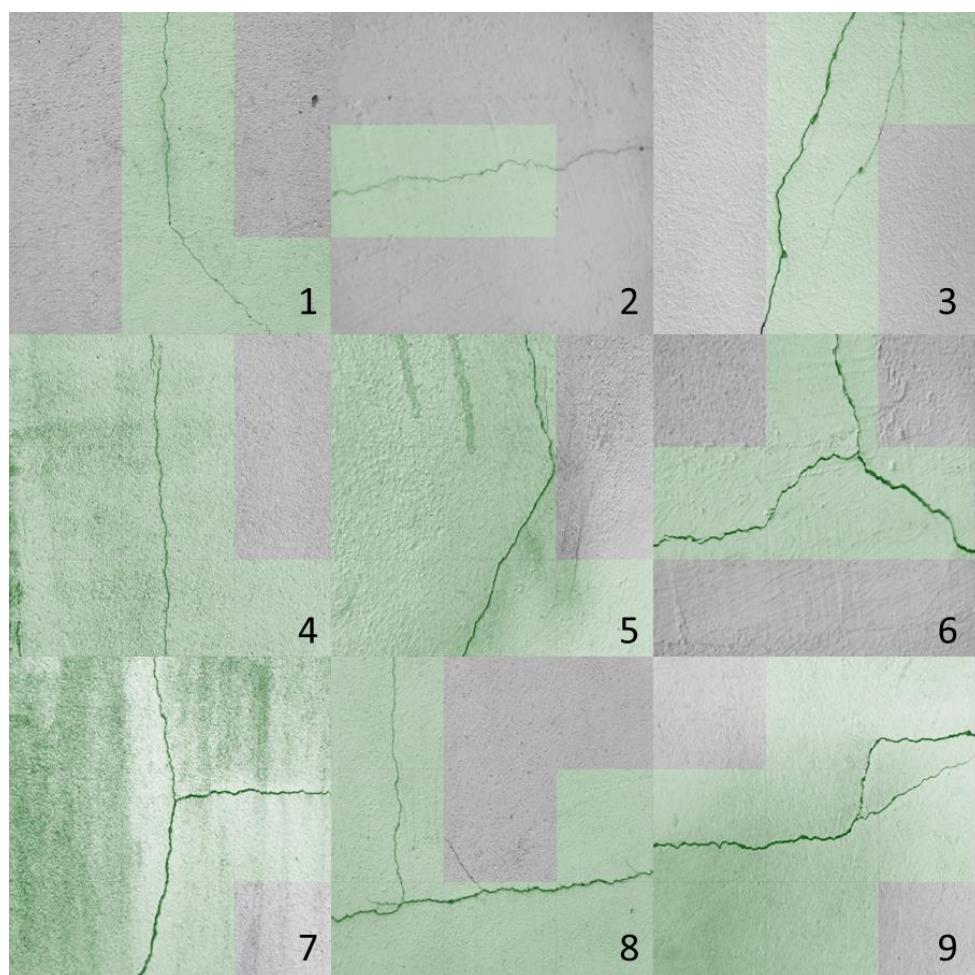


圖 4.12 灰階圖像使用 ResNet50 模型裂縫辨識結果

表 4.8 灰階轉換與 CLAHE 使用 VGG16 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	4	3	5	2	3	4	4	7	5
<b>TN</b>	2	6	3	6	4	5	4	2	4
<b>FP</b>	3	0	0	0	0	0	1	0	0
<b>FN</b>	0	0	1	1	2	0	0	0	0
<b>Precision1</b>	0.67	1.00	0.89	0.89	0.78	1.00	0.89	1.00	1.00
<b>Precision2</b>	0.57	1.00	0.83	0.67	0.60	1.00	0.80	1.00	1.00
<b>Recall</b>	1.00	1.00	0.83	0.67	0.60	1.00	1.00	1.00	1.00

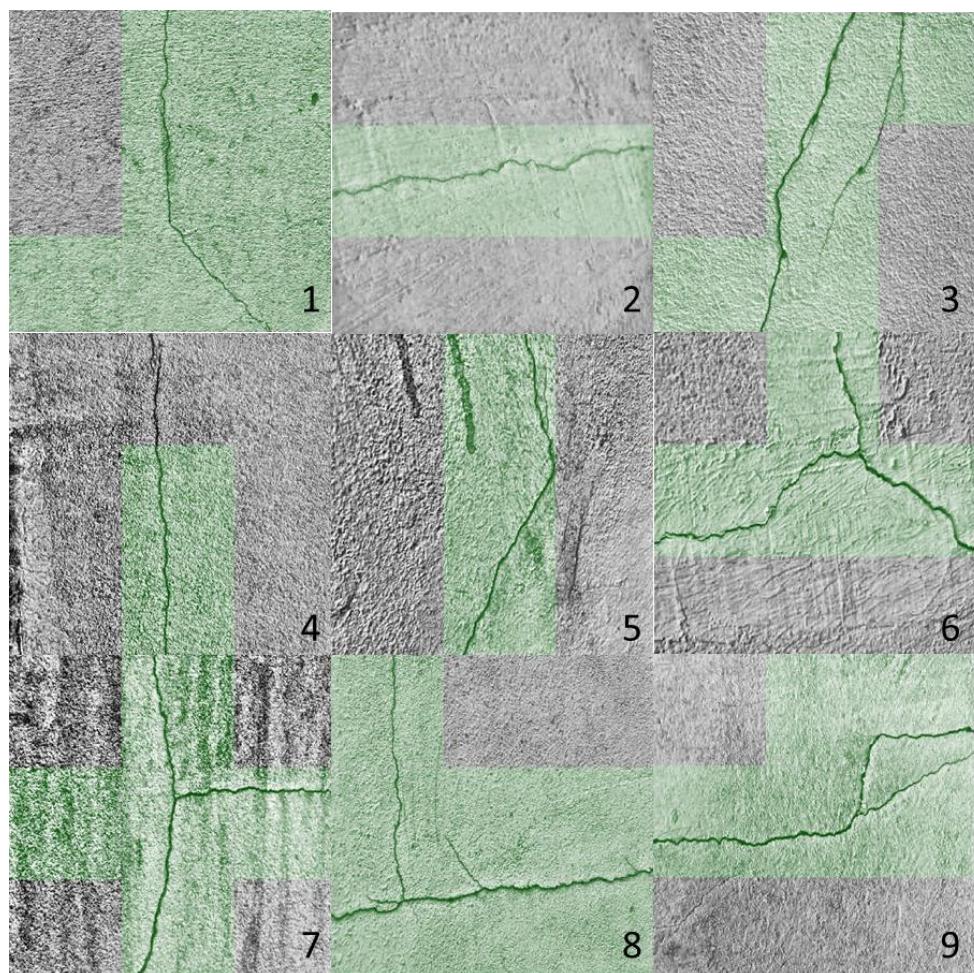


圖 4.13 灰階轉換與 CLAHE 使用 VGG16 模型裂縫辨識結果表

表 4.9 灰階轉換與 CLAHE 使用 ResNet50 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	4	3	6	3	4	4	4	7	5
<b>TN</b>	1	2	1	1	0	3	2	1	3
<b>FP</b>	4	4	2	5	4	2	3	1	1
<b>FN</b>	0	0	0	0	1	0	0	0	0
<b>Precision1</b>	0.56	0.56	0.78	0.44	0.44	0.78	0.67	0.89	0.89
<b>Precision2</b>	0.50	0.43	0.75	0.38	0.44	0.67	0.57	0.88	0.83
<b>Recall</b>	1.00	1.00	1.00	1.00	0.80	1.00	1.00	1.00	1.00

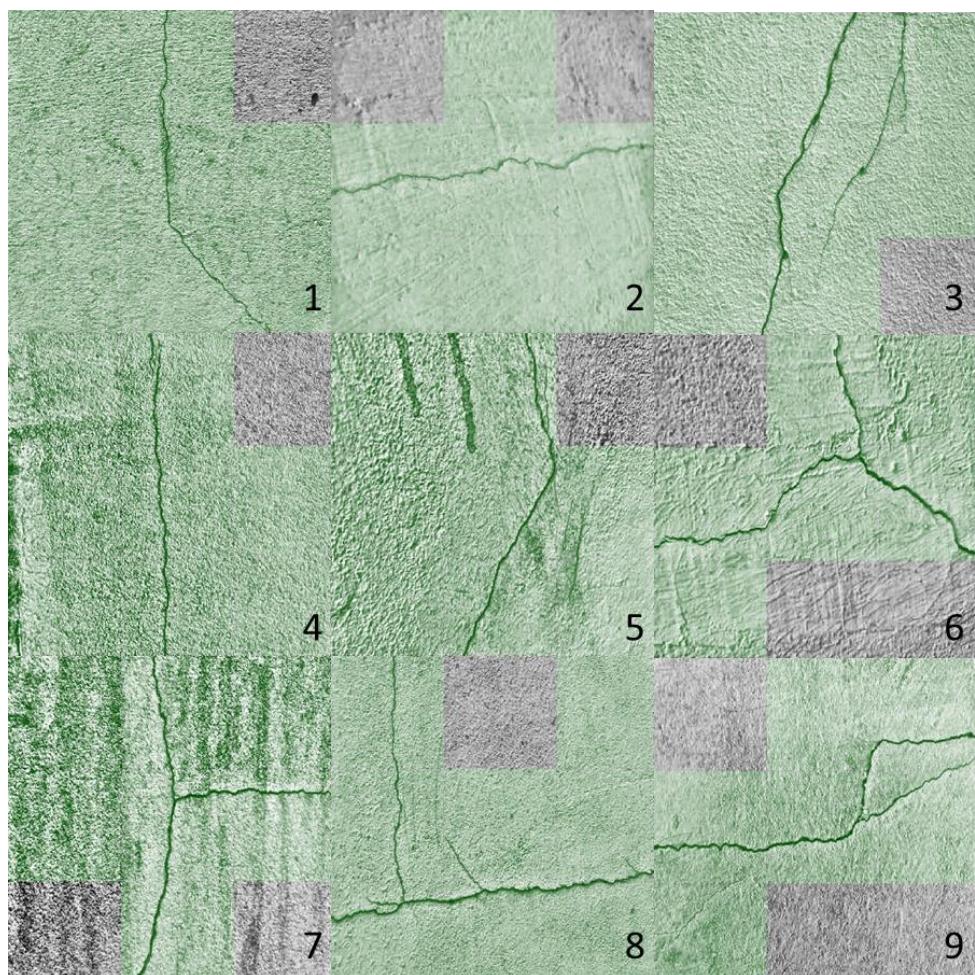


圖 4.14 灰階轉換與 CLAHE 使用 ResNet50 模型裂縫辨識結果

最後將八張圖片 Precision1、Precision2 與 Recall 的平均結果列於表 4.10，觀察八種組合之計算結果發現，訓練時未經資料增強模型的辨識精度無論是 VGG16 或是 ResNet50 模型均低於訓練時有線上資料增強的模型，說明線上資料增強可以讓模型於辨識時擁有較高的精度。

於所有的模型中，ResNet50 模型以原始圖像的方式去訓練及辨識會有最好的辨識成果，其 Precision1、Precision2 與 Recall 的值為 0.94、0.85 與 0.90；而 VGG16 則是以灰階與 CLAHE 進行訓練及辨識會有最好的辨識成果，其 Precision1、Precision2 與 Recall 的值為 0.90、0.83 與 0.90。兩者的 Precision1 均可以超越或至少等於 0.9 且 Recall 均大於 0.90，意味模型於測試裂縫照片具有不錯的辨識成果，且可辨識高比例的裂縫區域。

表 4.10 六種組合裂縫辨識結果之平均值

	原始圖像未經資料增強		原始圖像		灰階		灰階與 CLAHE	
	VGG16	ResNet50	VGG16	ResNet50	VGG16	ResNet50	VGG16	ResNet50
Average Precision1	0.69	0.90	0.85	0.94	0.82	0.75	0.90	0.67
Average Precision2	0.30	0.81	0.72	0.85	0.64	0.69	0.83	0.61
Recall	0.31	0.81	0.72	0.90	0.64	0.87	0.90	0.98

由於 ResNet50 模型搭配原始圖像經資料增於 9 張現場裂縫圖片辨識擁有最佳之精度，因此將其用於大型裂縫照片辨識。同樣的，其中兩張大型裂縫的辨識成果如圖 4.15 及 4.16，精度計算於表 4.11，而所有大型圖片的辨識結果也應過累計後列於表 4.12。

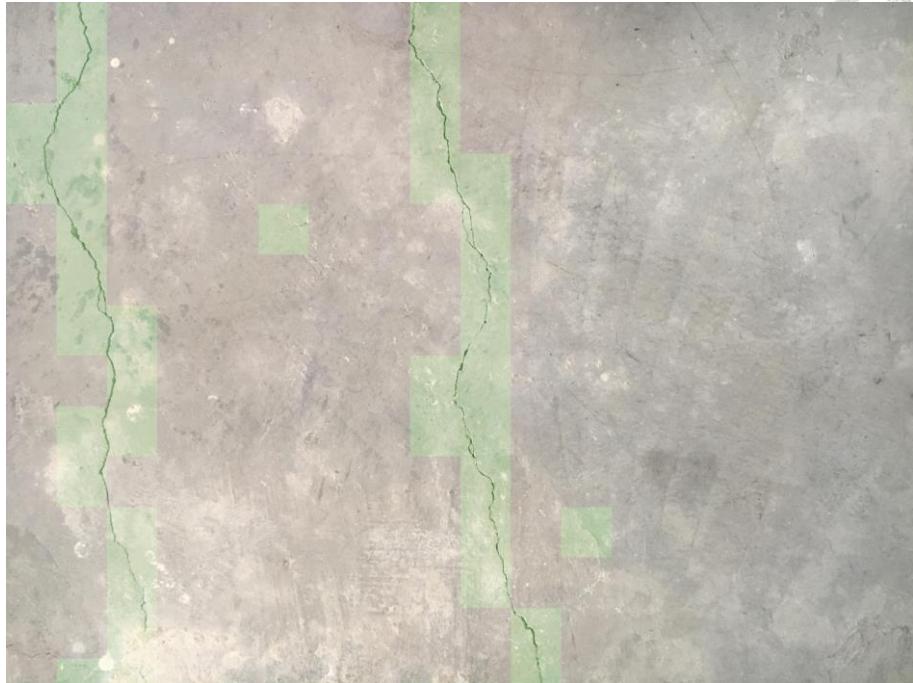


圖 4.15 大型裂縫圖片辨識展示圖片一



圖 4.16 大型裂縫圖片辨識展示圖片二

表 4.11 大型裂縫圖片辨識結果範例

	TP	TN	FP	FN	Precision1	Precision2	Recall
展示圖片一	36	209	2	5	0.97	0.84	0.88
展示圖片二	25	220	5	2	0.97	0.78	0.93

表 4.12 大型裂縫圖片平均辨識結果

總照片數	總區塊圖像	TP	TN	FP	FN	Precision1	Precision2	Recall
445	110,400	12,597	92,455	3,095	2,253	0.95	0.70	0.85

觀察圖 4.15 與 4.16，發現深度學習模型已能很好的辨識區塊圖像中的裂縫，兩張示範圖片的結果列於表 4.11，而總共 445 張的大型裂縫圖片結果列於表 4.12。觀察表 4.11 與表 4.12，發現 Precision1 均可以大於 0.95 以上，而平均 Recall 可以達到 0.85 以上，說明本研究選用的深度學習模型與照片型態應用於大型裂縫照片資料集，可以應用於該大型資料集的裂縫圖片並得到不錯的精度。

### 4.3 裂縫機率分布計算

根據 4.2 節的辨識成果，因使用原始圖像訓練的 ResNet50 模型擁有最高的 Precision 結果，因此被選用進行後續之裂縫辨識。初始相鄰照片間的步長為 224，接著我們更改相鄰區塊圖像間的移動步長，本研究中相鄰區塊圖像的步長的改變量總共有 5 種，分別為 224、112、56、28 及 14，將經過區塊圖像的使用(3.2)式將所有生成的裂縫機率之辨識與計算成果展示於圖 4.17 至 4.21。

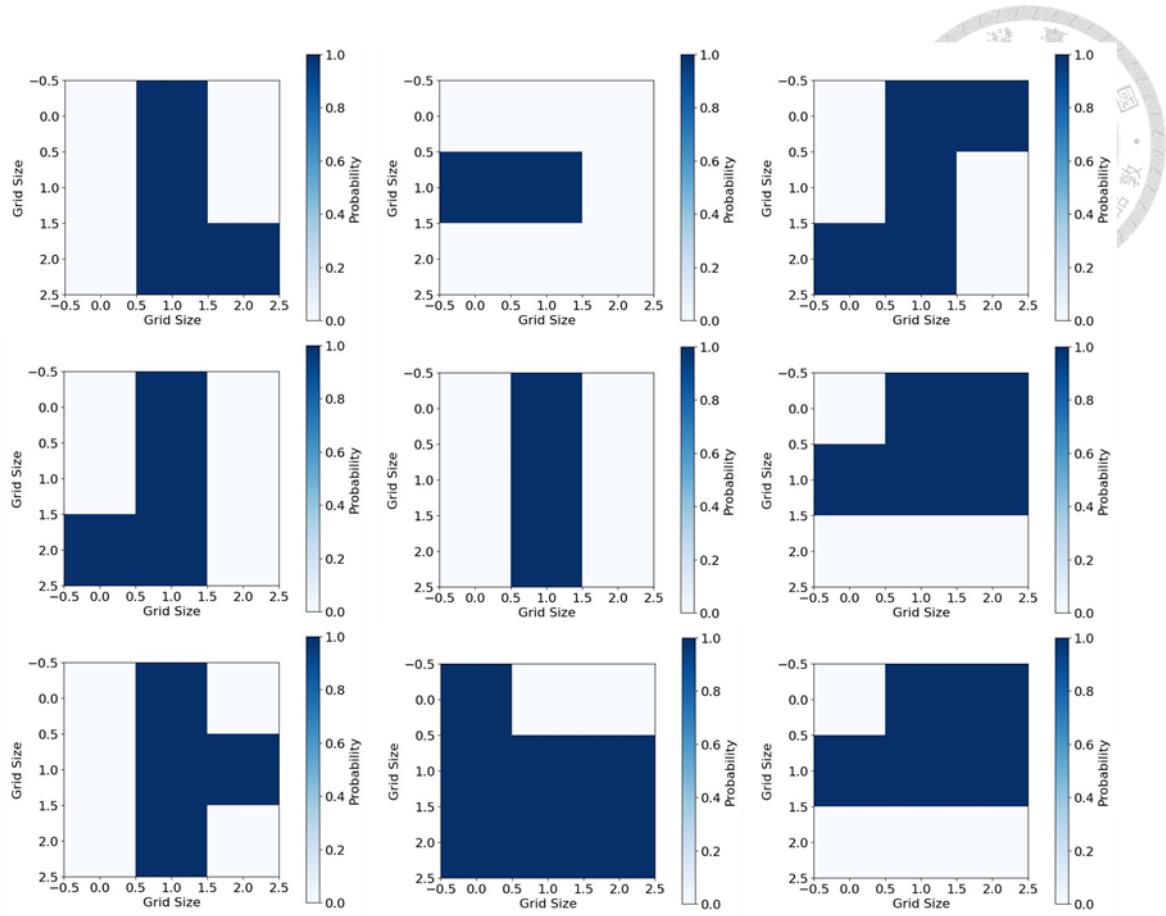


圖 4.17 步長為 224 像素之裂縫辨識機率圖

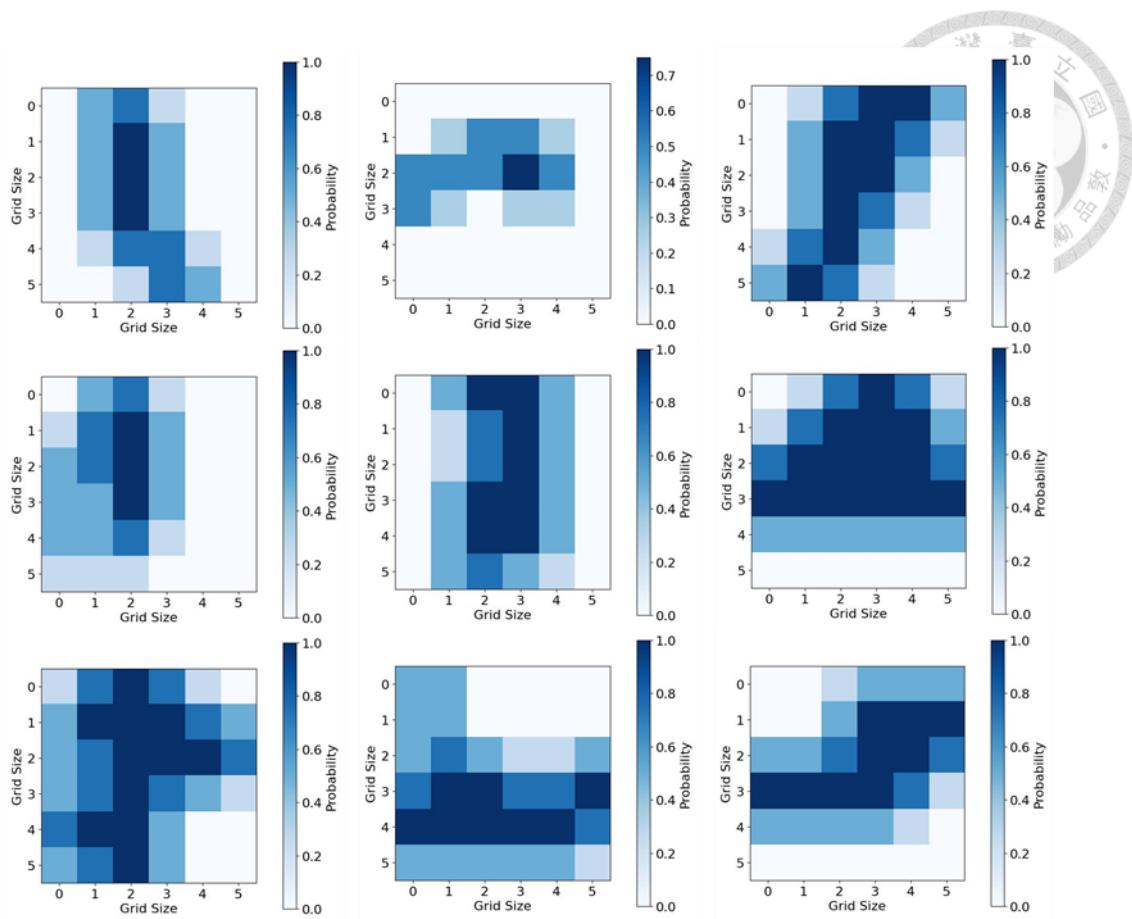


圖 4.18 步長為 112 像素之裂縫辨識機率圖

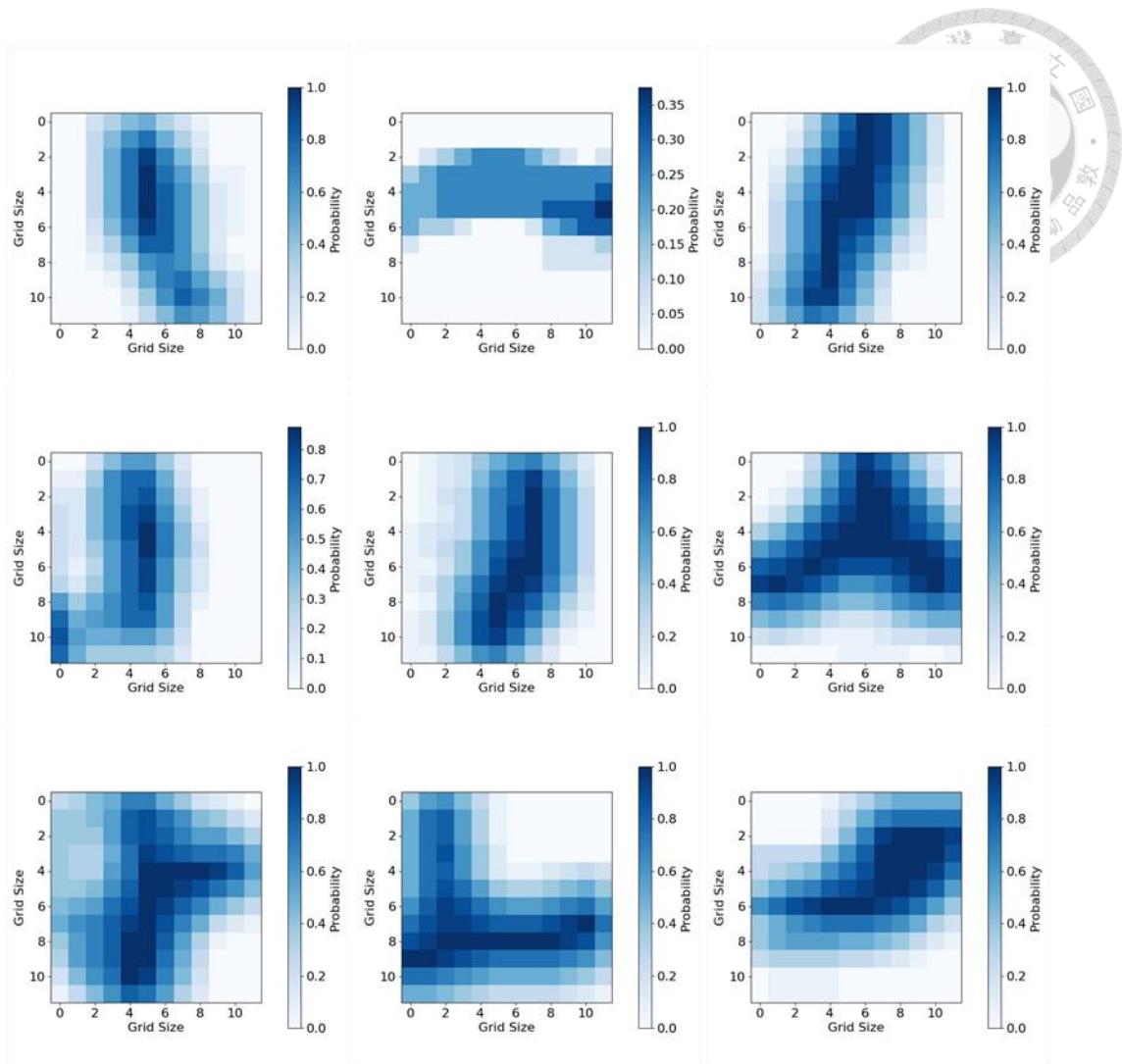


圖 4.19 步長為 56 像素之裂縫辨識機率圖

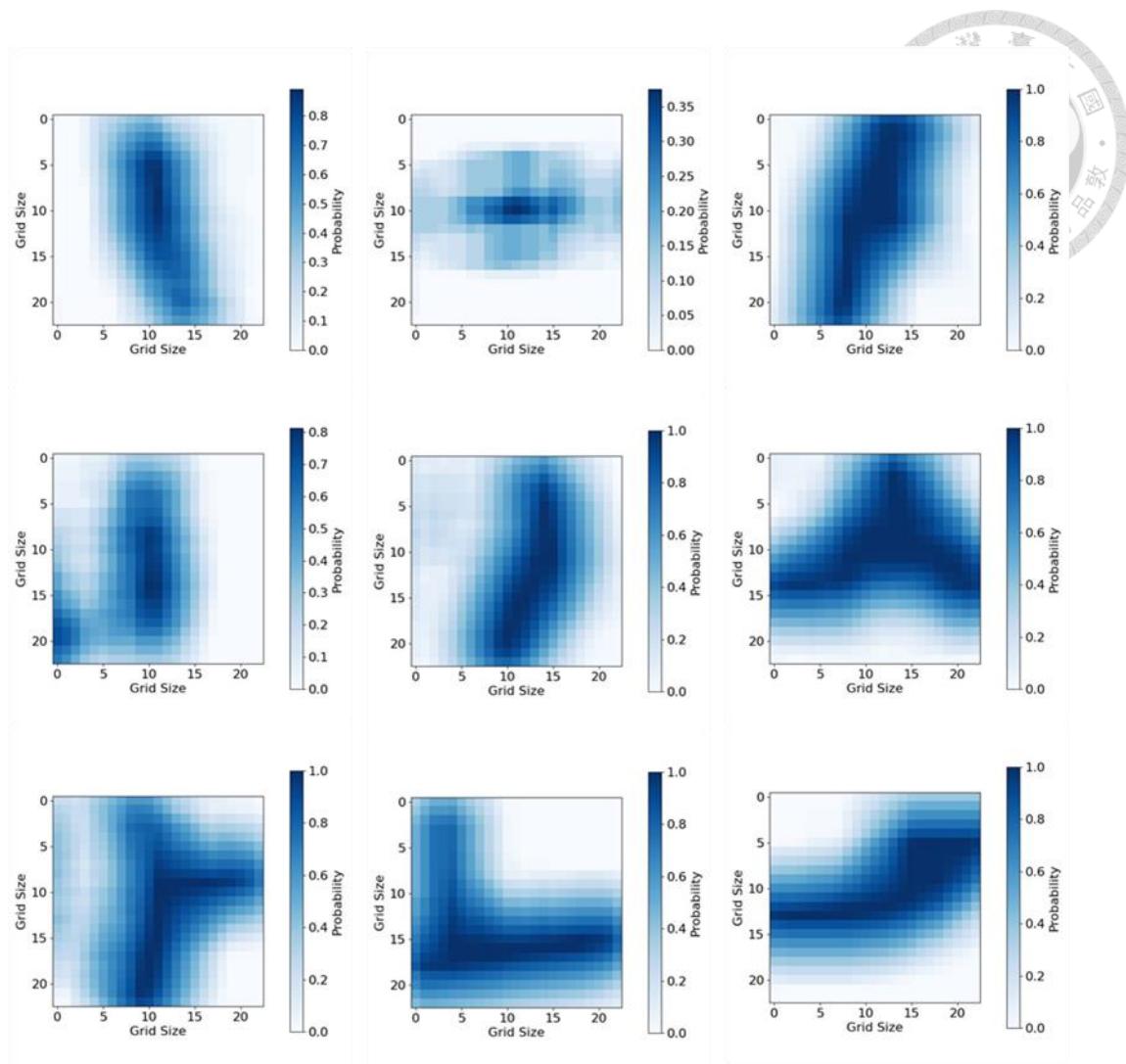


圖 4.20 步長為 28 像素之裂縫辨識機率圖

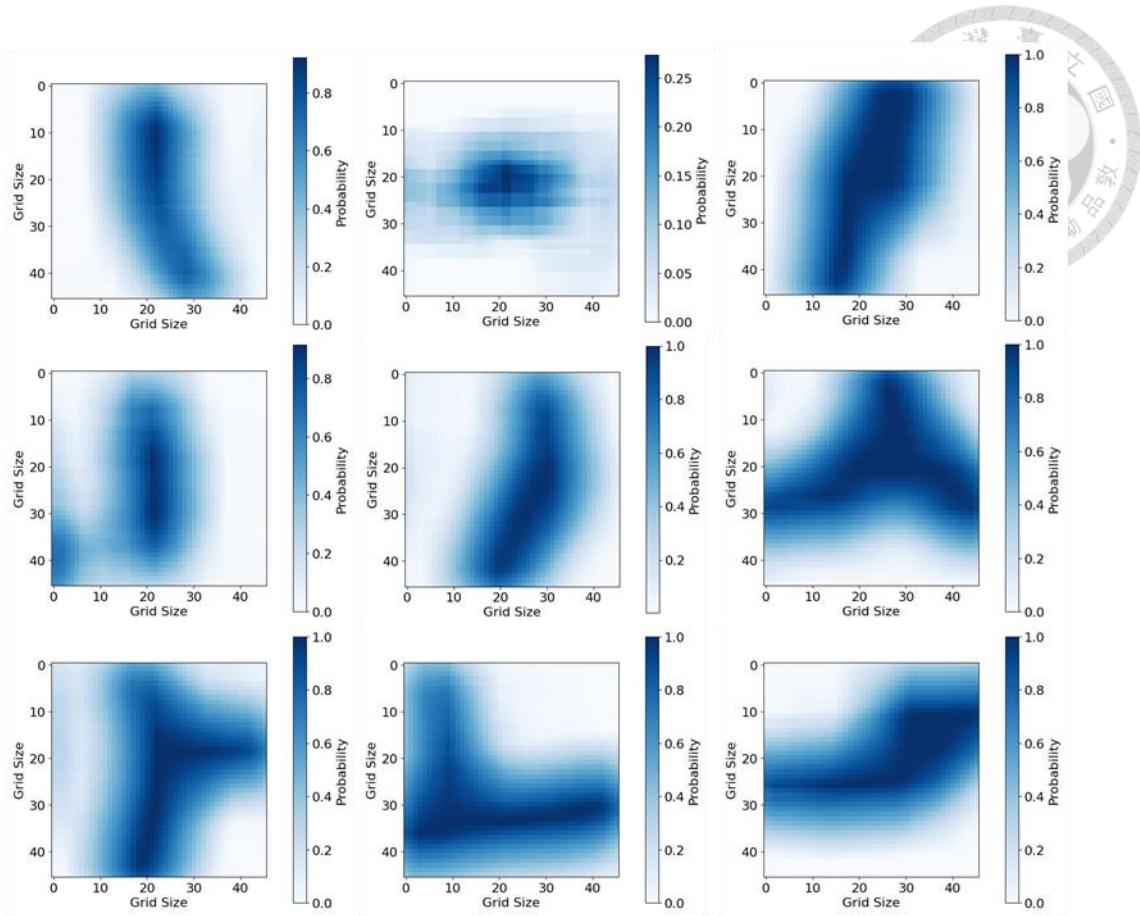
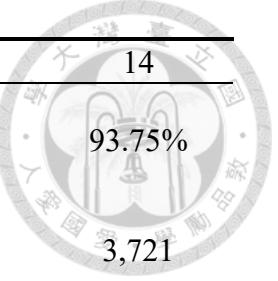


圖 4.21 步長為 14 像素之裂縫辨識機率圖

觀察圖 4.17 至 4.21，當步長為 224 像素時，模型已可以準確的辨識裂縫位於那些區塊圖像，如先前表 4.3 與圖 4.6 的結果，其辨識精度可以達到 0.9 以上，而圖 4.17 中，由於區塊圖像間不重疊，使用(3.2)式計算每個區域中含有裂縫的機率後，每個區域之裂縫機率為 0 或是 1。雖然圖 4.17 已能辨識大致的裂縫位置，其提供的裂縫位置由於單一圖像區塊的面積太大，仍無法提供裂縫的精準位置，因此圖 4.18 至 4.21 使相鄰區塊圖像得以重疊，其步長、重疊率與裁切後總區塊圖像數量列於表 4.13。

表 4.13 步長、重疊率與總區塊圖像數之關係



步長(pixel)	224	112	56	28	14
重疊率	0%	50%	75%	87.5%	93.75%
總區塊圖像數	9	49	225	900	3,721

觀察圖 4.17 至圖 4.21，隨著步長減少，相鄰區塊圖像重疊率增加，裂縫的特徵於機率圖中的位置越加明顯，意味增加區塊圖像間的重疊率有助於使裂縫位置辨識結果更加精確，觀察圖 4.21 當步長為 14 時，已能看到明顯的裂縫路徑位置。圖 4.17 至圖 4.21 的辨識使用精度如表 4.5 與表 4.14 至 4.17，而 9 張照片平均之結果如表 4.18。使用(4.1)式計算之Precision1在不同步長下均能維持大於 0.8 的精度，而隨著步長下降，Precision1 也有下降的趨勢，然而由於使用(4.2)式計算之Precision2為扣除 TN 之結果，當 TN 數量眾多時，其精度會因此而下降。藉由觀察圖 4.21 裂縫區塊圖像之辨識結果可知，使用深度學習模型區塊圖像之辨識方法成果可以大致辨識出裂縫的位置，因此本研究將繼續使用骨架化演算法以計算出之裂縫機率圖提取裂縫的實際位置。

表 4.14 步長 112 使用 ResNet50 模型裂縫辨識結果

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5	Fig.6	Fig.7	Fig.8	Fig.9
<b>TP</b>	13	8	18	9	13	18	17	19	18
<b>TN</b>	29	30	23	30	28	27	23	21	29
<b>FP</b>	0	1	0	5	1	0	6	0	0
<b>FN</b>	7	10	8	5	7	4	3	9	2
<b>Precision1</b>	0.86	0.78	0.84	0.80	0.84	0.92	0.82	0.82	0.96
<b>Precision2</b>	0.65	0.42	0.69	0.47	0.62	0.82	0.65	0.68	0.86

表 4.15 步長 56 使用 ResNet50 模型裂縫辨識結果

	<b>Fig.1</b>	<b>Fig.2</b>	<b>Fig.3</b>	<b>Fig.4</b>	<b>Fig.5</b>	<b>Fig.6</b>	<b>Fig.7</b>	<b>Fig.8</b>	<b>Fig.9</b>
<b>TP</b>	47	12	59	30	45	72	63	66	57
<b>TN</b>	145	150	119	149	134	116	124	113	136
<b>FP</b>	1	2	0	9	3	2	10	0	3
<b>FN</b>	32	61	47	37	43	35	28	46	29
<b>Precision1</b>	0.85	0.72	0.79	0.80	0.80	0.84	0.83	0.80	0.86
<b>Precision2</b>	0.59	0.16	0.56	0.39	0.49	0.66	0.62	0.59	0.64

表 4.16 步長 28 使用 ResNet50 模型裂縫辨識結果

	<b>Fig.1</b>	<b>Fig.2</b>	<b>Fig.3</b>	<b>Fig.4</b>	<b>Fig.5</b>	<b>Fig.6</b>	<b>Fig.7</b>	<b>Fig.8</b>	<b>Fig.9</b>
<b>TP</b>	160	25	241	99	205	289	246	267	251
<b>TN</b>	574	603	474	602	552	451	511	439	551
<b>FP</b>	1	6	1	14	8	5	27	0	3
<b>FN</b>	165	266	184	185	135	155	116	194	95
<b>Precision1</b>	0.82	0.70	0.79	0.78	0.84	0.82	0.84	0.78	0.89
<b>Precision2</b>	0.49	0.09	0.56	0.33	0.59	0.64	0.63	0.58	0.72

表 4.17 步長 14 使用 ResNet50 模型裂縫辨識結果

	<b>Fig.1</b>	<b>Fig.2</b>	<b>Fig.3</b>	<b>Fig.4</b>	<b>Fig.5</b>	<b>Fig.6</b>	<b>Fig.7</b>	<b>Fig.8</b>	<b>Fig.9</b>
<b>TP</b>	680	80	921	434	828	1171	963	1048	996
<b>TN</b>	2396	2536	1991	2473	2349	1930	2047	1844	2280
<b>FP</b>	22	9	6	122	20	6	161	5	12
<b>FN</b>	623	1096	803	692	524	614	550	824	433
<b>Precision1</b>	0.83	0.70	0.78	0.78	0.85	0.83	0.81	0.78	0.88
<b>Precision2</b>	0.51	0.07	0.53	0.35	0.60	0.65	0.58	0.56	0.69

表 4.18 不同步長下使用 ResNet50 模型裂縫辨識結果

步長 (pixels)	224	112	56	28	14
Precision1	0.94	0.85	0.81	0.81	0.80
Precision2	0.85	0.58	0.52	0.51	0.50

#### 4.4 骨架化演算法提取裂縫走向

圖 4.17 至圖 4.21 之結果已能看出裂縫的大致分布，為了提取裂縫的位置，需先將每個像素的機率分布圖以(3.4)式轉換為 0 至 255 的數值，將裂縫機率圖 4.21 轉換為黑白數值圖結果如圖 4.22。

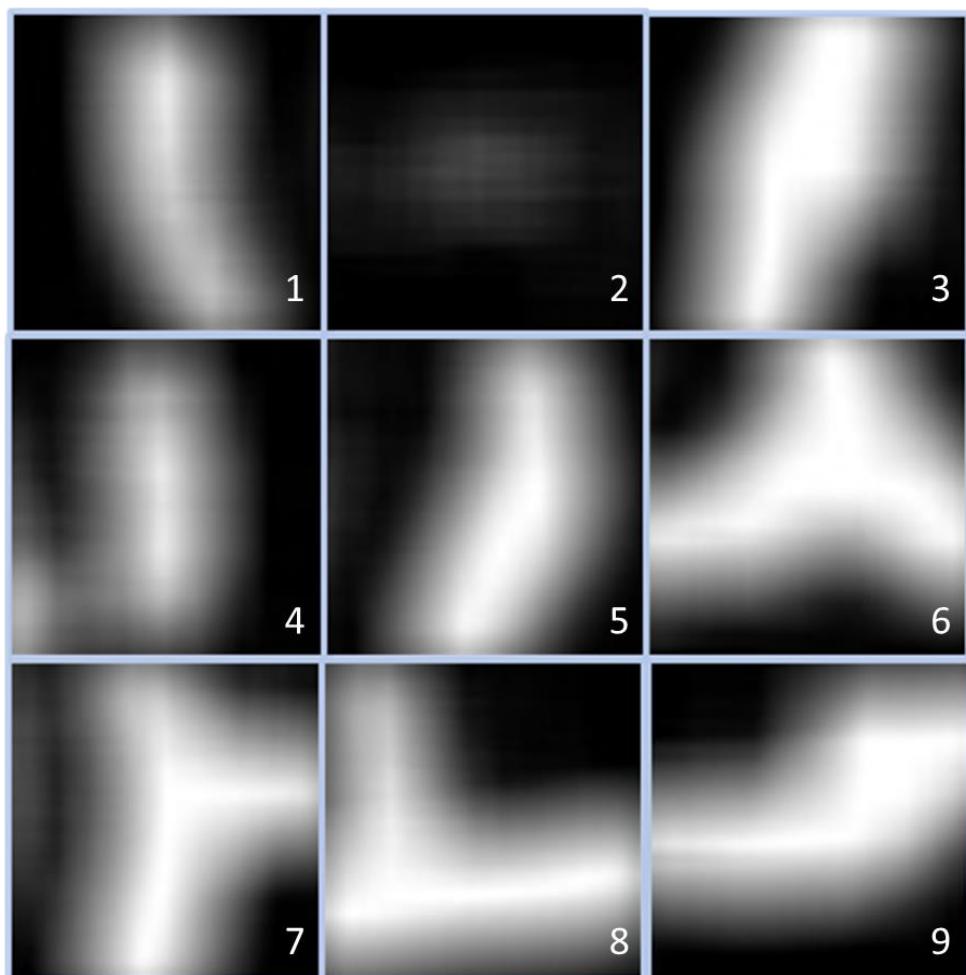


圖 4.22 步長為 14 的裂縫黑白數值圖

接著將 3.6 節中的 Zhang-Suen 骨架化演算法應用於圖 4.22，經過細化後的圖片展示於圖 4.23 此裂縫之特徵為使用一連串的裂縫辨識流程，再使用最後的骨架化演算法得出的最後結果。

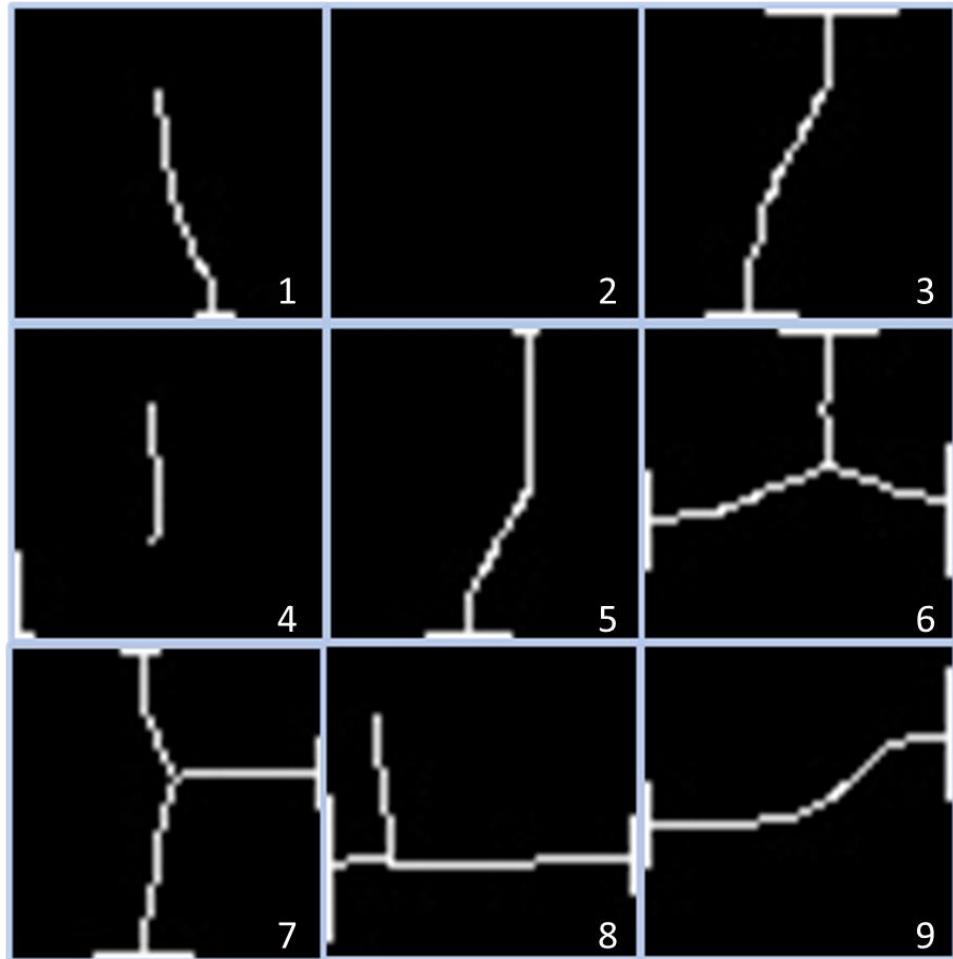


圖 4.23 經骨架化得出的裂縫路徑

將先前之裂縫標註結果圖 4.4 與圖使用裂縫辨識框架辨識出的裂縫位置圖 4.23 進行比較後，原先為裂縫但未辨識為裂縫之區域標註於圖 4.24 中。觀察圖 4.24，之結果，紅色方框為是裂縫但未被辨識為裂縫的區域，其中裂縫圖 3、5、6、7、9 幾乎可以完美預測裂縫位置，而裂縫圖 1 中少了正上方的裂縫位置、裂縫圖 4 中少了正上方與正下方的裂縫位置，而裂縫圖 8 中則於裂縫的分支部分少了上方的

裂縫位置，最後裂縫圖 2 則完全缺少辨識結果，觀察機率分布圖 4.21 中的裂縫圖 2 可知，在原先辨識裂縫時，裂縫圖 2 即擁有相對低的裂縫辨識機率，意味大多數該裂縫圖的裂縫區域不被深度學習模型辨識為裂縫，因此擁有較低的裂縫機率導致無法於骨架化演算法中呈現出裂縫。此外可以發現由於 Zhang-Suen 骨架化演算法的結果，會使得圖片邊界會有骨架化的白線產生，因此於未來的裂縫辨識時應予以移除。

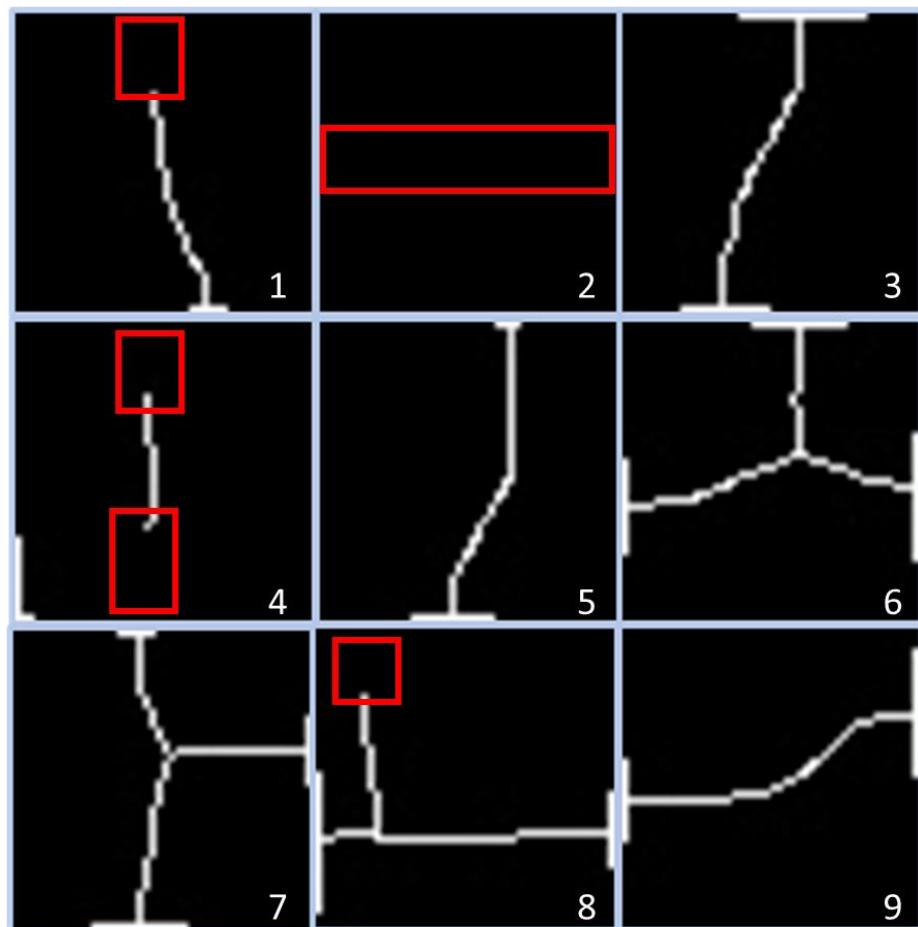


圖 4.24 比較裂縫標註結果與預測路徑

## 4.5 小結

本節首先比較六種組合的深度學習模型與訓練資料型態，對測試裂縫圖片辨識精確度的影響，發現使用線上資料增強可以提升深度學習模型的精度，而所有訓練的模型中，以 ResNet50 搭配原始未經轉換的裂縫資料集擁有最高的精確度，因此被選用為接下來步驟使用的深度學習模型。

將深度學習模型應用於接下來的裂縫辨識框架中，使用不同的鄰近區塊步長作為參數，發現當步長減少，相鄰區塊之重疊率增加、總區塊圖像數增加時，有助於使裂縫的機率圖呈現出更精確的裂縫位置，接著將裂縫機率分布圖轉換為黑白數值圖，經 Zhang-Suen 骨架化演算法後，裂縫的位置可以被找出，並呈現於圖 4.23，與原始的裂縫標註資料比較發現，辨識成果可以呈現出裂縫的位置與趨勢，唯少數圖片缺少部分裂縫區塊，而裂縫圖 2 則因原始的辨識結果精度差，而無法找出裂縫的位置。

# Chapter 5 結論與未來展望



## 5.1 結論

本研究提出了使用深度學習進行分類任務(Classification)的模型進行裂縫位置辨識任務。提出的裂縫辨識架構首先訓練深度學習模型 VGG16 與 ResNet50，並使用 4 種資料型態進行訓練，其中以 ResNet50 搭配原始訓練資料型態的訓練結果擁有最高的精度，接著裂縫辨識架構使用上述組合進行後續區塊圖像的辨識，其中包含調整相鄰區塊圖像間的重疊率，並使用 Zhang-Suen 骨架化演算法由裂縫黑白數值圖中提取裂縫的精確位置。由第二章節的深度學習模型訓練成果與四章節的模型實際場域驗證結果可以得出以下結論：

1. 由實驗結果得知，即使沒有使用資料增強的模型於訓練時擁有較高的驗證精度且收斂較快，然而有使用線上資料增強的模型在辨識現場的裂縫時則擁有較高的精度，且由於現場場域裂縫與訓練時的裂縫具有一定的差異性，說明資料增強可以提升模型的精度。
2. 研究中使用 VGG16 及 ResNet50 模型並搭配 4 種資料型態進行訓練，所有的模型訓練成果於 Training Set 或是 Validation Set 的精度接大於 0.9，因此兩種模型已經足夠用於裂縫分類任務。
3. 第四章節中使用 9 張現場照片進行方法驗證，8 種模型與訓練資料型態組合中，ResNet50 模型搭配原始資料集擁有最高的驗證精度，而 VGG16 模型搭配經過灰階與 CLAHE 轉換後的資料集擁有次高的驗證精度。
4. 將 ResNet50 模型搭配原始資料訓練的模型應用於後續的裂縫辨識，並調整相鄰區塊的重疊率可以有效的辨識裂縫所在圖像區塊，且當重疊率越高，

觀察生成之裂縫機率圖，可以發現裂縫位置越明顯。

5. 最後以 Zhang-Suen 骨架化演算法對裂縫位置進行提取，生成的裂縫位置圖，已可以判斷大多數的裂縫位置與趨勢，唯少部分區域無法辨識出裂縫，或是少數圖片因深度學習模型無法精確判斷裂縫，而使得缺少部分裂縫位置。



## 5.2 未來展望

本研究提出新的裂縫辨識方法，使用深度學習分類模型，搭配影像處理方法，以簡化目前裂縫辨識模型之訓練辨識及後處理流程。於第四章節已經驗證提出之裂縫辨識流程之可行性，並由最後之裂縫走向看出已可以辨識裂縫之粗略位置及走向，然而本架構仍有可以改進或增加之步驟，因此以下給予相關建議，以精進本裂縫辨識流程。

1. 目前區塊圖像之步長含有 224、112、56、28、14，而使用不同步長進行區塊圖像切割，將能獲得不同精細程度的裂縫位置機率圖，考量辨識時間與辨識精度要求，需評估何種以何種步長作為區塊圖像的切割參數，才得以同時滿足以上要求，為後續可進行的研究。
2. 於第四章時用來驗證的現場場域裂縫照片與訓練模型時使用的裂縫照片具有相當大的差異，其最主要的差異為裂縫寬度，雖然於場域驗證時有不錯的精度，且已於訓練模型時增加縮放訓練圖像之參數，然而未來仍可以增加資料集的豐富度，以應付不同寬度裂縫的圖案。
3. 本次研究使用的測試資料集僅擁有 9 張現場照片，若要進行更嚴謹的模型精度驗證，應該使用更多且更多元的裂縫照片，才能使驗證結果更有信

服度。此外驗證照片尺寸只有  $640 \times 640$ ，可以選擇更大尺寸的照片進行區塊圖像切割，呈現的辨識結果會更完整。

- 
4. 此次使用的線上資料增強參數為粗略設置，目標為使訓練及驗證資料集的裂縫照片擁有更高的豐富性，然而實際參數設定應該更嚴謹，應考慮目前資料集的裂縫照片型態，再使用不同的縮放比例、平移量，並考慮灰階與 CLAHE 之設定再使用資料增強參數中的亮度調整，應會使模型會有更好的辨識成效。
  5. 可以將辨識結果與實例分割模型進行比較，如 U-Net 及 Mask R-CNN，討論本裂縫辨識框架與實例分割模型，根據辨識時間、辨識精度與訓練資料準備的複雜度進行比較。
  6. 於使用 Zhang-Suen 骨架化演算法時會使圖片邊緣有多餘的白色線條產生，於之後的研究中可以予以移除。

## 參考文獻



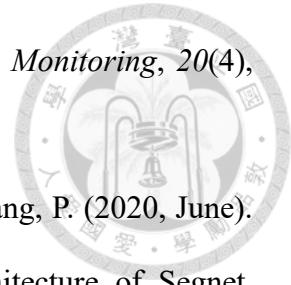
- [1] Laxman, K. C., Tabassum, N., Ai, L., Cole, C., & Ziehl, P. (2023, March). Automated Crack Detection and Crack Depth Prediction for Reinforced Concrete Structures Using Deep Learning. *Construction and Building Materials*, 370, 130709. <https://doi.org/10.1016/j.conbuildmat.2023.130709>
- [2] Hang, J., Wu, Y., Li, Y., Lai, T., Zhang, J., & Li, Y. (2023, January 5). A Deep Learning Semantic Segmentation Network with Attention Mechanism for Concrete Crack Detection. *Structural Health Monitoring*, 147592172211261. <https://doi.org/10.1177/14759217221126170>
- [3] Golding, V. P., Gharineiat, Z., Munawar, H. S., & Ullah, F. (2022, July 2). Crack Detection in Concrete Structures Using Deep Learning. *Sustainability*, 14(13), 8117. <https://doi.org/10.3390/su14138117>
- [4] Zhao, W., Liu, Y., Zhang, J., Shao, Y., & Shu, J. (2022, April 26). Automatic Pixel-Level Crack Detection and Evaluation of Concrete Structures Using Deep Learning. *Structural Control and Health Monitoring*, 29(8). <https://doi.org/10.1002/stc.2981>
- [5] Munawar, H. S., Hammad, A. W. A., Haddad, A., Soares, C. A. P., & Waller, S. T. (2021, August 14). Image-Based Crack Detection Methods: A Review. *Infrastructures*, 6(8), 115. <https://doi.org/10.3390/infrastructures6080115>
- [6] Rajadurai, R. S., & Kang, S. T. (2021, June 4). Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning. *Applied Sciences*, 11(11), 5229. <https://doi.org/10.3390/app11115229>
- [7] Kumar, A., Jha, A. K., & Trivedi, A. (2020, December 3). Crack Detection of Structures using Deep Learning Framework. *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*.

<https://doi.org/10.1109/iciss49785.2020.9315949>



- [8] Kim, J. J., Kim, A. R., & Lee, S. W. (2020, November 16). Artificial Neural Network-Based Automated Crack Detection and Analysis for the Inspection of Concrete Structures. *Applied Sciences*, 10(22), 8105. <https://doi.org/10.3390/app10228105>
- [9] Kumar, F., Li, L., Gu, Y., Zhu, H., & Lim, J. H. (2020, November). A Novel Hybrid Approach for Crack Detection. *Pattern Recognition*, 107, 107474. <https://doi.org/10.1016/j.patcog.2020.107474>
- [10] Pan, Y., Zhang, G., & Zhang, L. (2020, November). A Spatial-Channel Hierarchical Deep Learning Network for Pixel-Level Automated Crack Detection. *Automation in Construction*, 119, 103357. <https://doi.org/10.1016/j.autcon.2020.103357>
- [11] Fang, F., Li, L., Gu, Y., Zhu, H., & Lim, J. H. (2020, November). A Novel Hybrid Approach for Crack Detection. *Pattern Recognition*, 107, 107474. <https://doi.org/10.1016/j.patcog.2020.107474>
- [12] Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V. C., & Ding, L. (2020, September 17). Automated Pavement Crack Detection and Segmentation Based on Two-Step Convolutional Neural Network. *Computer-Aided Civil and Infrastructure Engineering*, 35(11), 1291–1305. <https://doi.org/10.1111/mice.12622>
- [13] Yamane, T., & Chun, P. J. (2020, September 16). Crack Detection from a Concrete Surface Image Based on Semantic Segmentation Using Deep Learning. *Journal of Advanced Concrete Technology*, 18(9), 493–504. <https://doi.org/10.3151/jact.18.493>
- [14] Hsieh, Y. A., & Tsai, Y. J. (2020, September). Machine Learning for Crack Detection: Review and Model Performance Comparison. *Journal of Computing in Civil Engineering*, 34(5). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000918](https://doi.org/10.1061/(asce)cp.1943-5487.0000918)
- [15] Zhang, L., Shen, J., & Zhu, B. (2020, July 21). A Research on An Improved Unet-

Based Concrete Crack Detection Algorithm. *Structural Health Monitoring*, 20(4), 1864–1879. <https://doi.org/10.1177/1475921720940068>



- [16] Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T., & Wang, P. (2020, June). Pavement Crack Detection and Recognition Using the Architecture of Segnet. *Journal of Industrial Information Integration*, 18, 100144. <https://doi.org/10.1016/j.jii.2020.100144>
- [17] Kalfarisi, R., Wu, Z. Y., & Soh, K. (2020, May). Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization. *Journal of Computing in Civil Engineering*, 34(3). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000890](https://doi.org/10.1061/(asce)cp.1943-5487.0000890)
- [18] Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2020, April). Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1525–1535. <https://doi.org/10.1109/tits.2019.2910595>
- [19] Huyan, J., Li, W., Tighe, S., Xu, Z., & Zhai, J. (2020, March 18). Cracku-Net: A Novel Deep Convolutional Neural Network for Pixelwise Pavement Crack Detection. *Structural Control and Health Monitoring*, 27(8). <https://doi.org/10.1002/stc.2551>
- [20] Alipour, M., & Harris, D. K. (2020, March). Increasing The Robustness Of Material-Specific Deep Learning Models For Crack Detection Across Different Materials. *Engineering Structures*, 206, 110157. <https://doi.org/10.1016/j.engstruct.2019.110157>
- [21] Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., & Shen, X. (2020, February). Image-Based Concrete Crack Detection in Tunnels Using Deep Fully Convolutional Networks. *Construction and Building Materials*, 234, 117367.

<https://doi.org/10.1016/j.conbuildmat.2019.117367>



- [22] Liu, K., Han, X., & Chen, B. M. (2019, December). Deep Learning Based Automatic Crack Detection and Segmentation for Unmanned Aerial Vehicle Inspections. *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

<https://doi.org/10.1109/robio49542.2019.8961534>

- [23] Liu, Z., Cao, Y., Wang, Y., & Wang, W. (2019, August). Computer Vision-Based Concrete Crack Detection Using U-Net Fully Convolutional Networks. *Automation In Construction*, 104, 129–139. <https://doi.org/10.1016/j.autcon.2019.04.005>

- [24] Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., & Chen, X. (2019, July 18). Automatic Bridge Crack Detection Using a Convolutional Neural Network. *Applied Sciences*, 9(14), 2867. <https://doi.org/10.3390/app9142867>

- [25] Dung, C. V., & Anh, L. D. (2019, March). Autonomous Concrete Crack Detection Using Deep Fully Convolutional Neural Network. *Automation in Construction*, 99, 52–58. <https://doi.org/10.1016/j.autcon.2018.11.028>

- [26] Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2019, March). DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Transactions on Image Processing*, 28(3), 1498–1512. <https://doi.org/10.1109/tip.2018.2878966>

- [27] Protopapadakis, E., Voulodimos, A., Doulamis, A., Doulamis, N., & Stathaki, T. (2019, February 7). Automatic Crack Detection for Tunnel Inspection Using Deep Learning and Heuristic Image Post-Processing. *Applied Intelligence*, 49(7), 2793–2806. <https://doi.org/10.1007/s10489-018-01396-y>

- [28] Mandal, V., Uong, L., & Adu-Gyamfi, Y. (2018, December). Automated Road Crack Detection Using Deep Convolutional Neural Networks. *2018 IEEE International Conference on Big Data (Big Data)*. <https://doi.org/10.1109/bigdata.2018.8622327>

[29] Dorafshan, S., Thomas, R. J., & Maguire, M. (2018, October). Comparison Of Deep Convolutional Neural Networks and Edge Detectors for Image-Based Crack Detection in Concrete. *Construction and Building Materials*, 186, 1031–1045.

<https://doi.org/10.1016/j.conbuildmat.2018.08.011>

[30] Özgenel, F., & Sorguç, A. G. (2018, July 22). Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*.

<https://doi.org/10.22260/isarc2018/0094>

[31] Pauly, L., Peel, H., Luo, S., Hogg, D., & Fuentes, R. (2017, July 1). Deeper Networks for Pavement Crack Detection. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*.

<https://doi.org/10.22260/isarc2017/0066>

[32] Cubero-Fernandez, A., Rodriguez-Lozano, F. J., Villatoro, R., Olivares, J., & Palomares, J. M. (2017, June 13). Efficient Pavement Crack Detection and Classification. *EURASIP Journal on Image and Video Processing*, 2017(1).

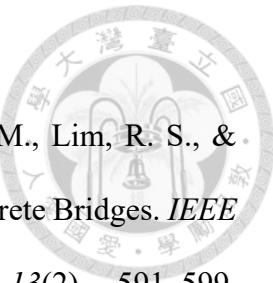
<https://doi.org/10.1186/s13640-017-0187-0>

[33] Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016, December). Automatic Road Crack Detection Using Random Structured Forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12), 3434–3445.

<https://doi.org/10.1109/tits.2016.2552248>

[34] Zhang, L., Yang, F., Daniel Zhang, Y., & Zhu, Y. J. (2016, September). Road Crack Detection Using Deep Convolutional Neural Network. *2016 IEEE International Conference on Image Processing (ICIP)*. <https://doi.org/10.1109/icip.2016.7533052>

[35] He, K., Zhang, X., Ren, S., & Sun, J. (2016, June). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern*



- [36] Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., & Parvardeh, H. (2016, April). Automated Crack Detection on Concrete Bridges. *IEEE Transactions on Automation Science and Engineering*, 13(2), 591–599. <https://doi.org/10.1109/tase.2014.2354314>
- [37] Lins, R. G., & Givigi, S. N. (2016, March). Automatic Crack Detection and Measurement Based on Image Analysis. *IEEE Transactions on Instrumentation and Measurement*, 65(3), 583–590. <https://doi.org/10.1109/tim.2015.2509278>
- [38] Wu, L., Mokhtari, S., Nazef, A., Nam, B., & Yun, H. B. (2016, January). Improvement of Crack-Detection Accuracy Using a Novel Crack Defragmentation Technique in Image-Based Road Assessment. *Journal of Computing in Civil Engineering*, 30(1). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000451](https://doi.org/10.1061/(asce)cp.1943-5487.0000451)
- [39] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*. <https://doi.org/10.1109/iccv.2015.123>
- [40] Oliveira, H., & Correia, P. L. (2014, October). CrackIT #2014; An Image Processing Toolbox for Crack Detection and Characterization. *2014 IEEE International Conference on Image Processing (ICIP)*. <https://doi.org/10.1109/icip.2014.7025160>
- [41] Salman, M., Mathavan, S., Kamal, K., & Rahman, M. (2013, October). Pavement Crack Detection Using the Gabor Filter. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC) 2013*. <https://doi.org/10.1109/itsc.2013.6728529>
- [42] Prasanna, P., Dana, K., Gucunski, N., & Basily, B. (2012, April 26). Computer-Vision Based Crack Detection and Analysis. *SPIE Proceedings*.

<https://doi.org/10.1117/12.915384>



- [43] Gavilán, M., Balcones, D., Marcos, O., Llorca, D. F., Sotelo, M. A., Parra, I., Ocaña, M., Aliseda, P., Yarza, P., & Amírola, A. (2011, October 12). Adaptive Road Crack Detection System by Pavement Classification. *Sensors*, 11(10), 9628–9657. <https://doi.org/10.3390/s111009628>
- [44] Hu, Y., & Zhao, C. X. (2010). A Novel LBP Based Methods for Pavement Crack Detection. *Journal of Pattern Recognition Research*, 5(1), 140–147. <https://doi.org/10.13176/11.167>
- [45] Fujita, Y., & Hamamoto, Y. (2010, February 2). A Robust Automatic Crack Detection Method from Noisy Concrete Surfaces. *Machine Vision and Applications*, 22(2), 245–254. <https://doi.org/10.1007/s00138-009-0244-5>
- [46] Fujita, Y., Mitani, Y., & Hamamoto, Y. (2006). A Method for Crack Detection on a Concrete Structure. *18th International Conference on Pattern Recognition (ICPR '06)*. <https://doi.org/10.1109/icpr.2006.98>
- [47] Özgenel, Çağlar Fırat (2019), “Concrete Crack Images for Classification”, Mendeley Data, V2, <https://doi.org/10.17632/5y9wdsg2zt.2>
- [48] Simonyan, K. (2014, September 4). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv.org*. <https://doi.org/10.48550/arXiv.1409.1556>
- [49] Gonzalez, R. C., & Woods, R. E. (2018, January 1). *Digital Image Processing*.
- [50] Zhang, T. Y., & Suen, C. Y. (1984, March). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236–239. <https://doi.org/10.1145/357994.358023>
- [51] OpenCV: cv::CLAHE Class Reference. (n.d.). [https://docs.opencv.org/4.x/d6/db6/classcv\\_1\\_1CLAHE.html](https://docs.opencv.org/4.x/d6/db6/classcv_1_1CLAHE.html)