國立臺灣大學工學院土木工程研究所

碩士論文

Department of civil engineering

College of Engineering

National Taiwan University

Master's Thesis

# 新型多尺度磁磚剝落分割模型與無監督式對比學習策略之研發

# Developments of Multi-Scale Feature Fusion Network and Unsupervised Contrastive Learning Strategy for Tile Spalling Segmentation

王海威

Hai-Wei Wang

指導教授: 吳日騰 博士

Advisor: Rih-Teng Wu, Ph.D.

中華民國 113 年 8 月

August, 2024

doi:10.6342/NTU202403549

# Acknowledgements

首先我要感謝凱米颱風於 2024/7/24 侵台造成颱風假，讓我在電子檔上傳死線前兩天還在改論文，接下來我要感謝在螢幕前看著這篇論文的你，如果不是螢幕而是紙本的話就更感謝了，以下進入正題。

首先我要感謝的是指導教授吳日騰老師，還記得當初大四上剛決定要推 CAE 組時剛好看到老師剛來台大要找專題生，於是就去試試累積點經驗，畢竟那時候成績大概只有 PR25，除了變魔術甚麼都不會，連暑期實習的面試都進不去。在老師的指導下年底我奇蹟似的推上了 CAE 組，雖然是最低分錄取，而上了研究所後老師也帶我們參與了一系列的活動，例如美國的 EMI 研討會，ICSHM 的比賽，這些經歷讓我的寫作能力及報告能力大幅提升，在比賽甚至拿了冠軍，最後也成功地通過口試準時畢業，除此之外老師還讓我參與一些產學計畫賺點錢養活自己，這邊也感謝衣食父母兼口試委員韓仁毓老師，以及甘翊萱學姊多買的便當。

接著要感謝 R11 及 R12 的夥伴們，不論是球場上的球友，或是計畫的研究夥伴，亦或是實驗室的難兄難弟，你們的存在豐富了我枯燥乏味的研究生活，希望以後大家能一展長才，要出過的能申請到好學校，留國內工作的能發財，學弟們研究也能順利。

最後要感謝我的爸媽，雖然大學搬出來後較少見面，不過他們還是常常關心我的身體狀況，研究順不順利，讓我每次回家都能好好放鬆，很抱歉大學時玩太



doi:10.6342/NTU202403549

兒讓你們擔心，接下來我會實現夢想讓你們過上好日子。



doi:10.6342/NTU202403549

# 摘要

磁磚為台灣建物常見之外牆飾材，但其經常會因老化和環境因素而產生表面劣化，造成磁磚碎片剝落等問題。而剝落的碎片對鄰近人行道上的行人和車輛造成損傷的案例屢見不鮮。近年來不少研究使用深度學習及電腦視覺的方式來對外牆進行結構健檢，其中不乏針對磁磚外牆的脫落以及破損的檢測，而現今的主流方法為使用具人工標記的圖片資料集進行監督式訓練，然而監督式方法需要大量破壞的資料及相應的標記數據，使得資料集的構築十分費時費力，尤其是需要像素級標註的語意分割資料集，這也使得相較於傳統的語意分割資料集，該領域的資料較為有限。因此在本研究中，我們設計了一個名為 Multi-Scale Branch Fusion UNet (MBF-UNet) 的深度學習模型，該模型使用多個不同感受野 (receptive field) 的分支來增加影像資訊以解決由於資料量過少而引起的過度擬合問題，數據分析顯示此模型架構在我們設計的六個指標中都能有著最好的表現及較低較穩定的標準差。此外我們還提出了一種新式無監督訓練方法，使得我們在訓練過程中不需要任何剝落的標註資料，甚至不需要剝落的圖像，就能使模型辨識出建物圖片上的磁磚剝落，此法基於不確定性評估的概念，以一個不含剝落的已標注房屋資料集對模型進行訓練使其熟悉正常房屋外觀，訓練後模型將對剝落區域產生較高不確定性因為訓練資料不包含剝落，該不確定性即可被評估並輸出異常分數，另外我們還研發了一種剝落生成技術，能夠在正常的房屋影像中添加人工生成剝落以促進模型訓練，數據分析顯示在 AUC, AP 及 FPR95 分數中，此方法可優於其他方



doi:10.6342/NTU202403549

法 18.4%, 46.6% 及 31.7%。對比於傳統的監督式學習，此無監督式學習法能將資料及建構時間從 200 小時減少至 1 小時，換算成工資約台幣 36600 元。此研究有助於基礎設施監測的應用，並提高磁磚剝落檢測的資料效率。

**關鍵字**：結構健檢、監督式學習、標籤效率、建物外牆檢測、無監督深度對比式學習、不確定性評估、磁磚剝落檢測



doi:10.6342/NTU202403549

# Abstract

Exterior wall deterioration, particularly tile spalling, is a common consequence of aging and environmental degradation in urban environments. These structural impairments pose significant threats to public safety, particularly for pedestrians and vehicles on sidewalks. In recent years, deep learning-based approaches have been leveraged in autonomous methods for building condition assessments owing to their capacity to identify structural anomalies. However, training a supervised learning model typically requires a large labeled dataset, which is often unavailable in engineering domain tasks. Therefore, in this study, we design a novel model called the Multi-Scale Branch Fusion UNet (MBF-UNet) for semantic segmentation of tile spalling. The MBF-UNet incorporates additional branches with different receptive fields and self-attention mechanisms to extract meaningful representations of surface damage. Statistical measures have demonstrated that the proposed MBF-UNet outperforms the state-of-the-art segmentation models in 6 general segmentation metrics. Additionally, we propose a new unsupervised learning framework



doi:10.6342/NTU202403549

for anomaly detection of tile spalling. There is no need of images that contains spalling or corresponding labels. The framework leverage uncertainty estimation by training a segmentation model with a labeled dataset consisting of known and given classes excluding spalling. After training, the model identifies spalling area as outlier pixels, i.e., the anomaly, due to higher uncertainty score. Besides, we develop a synthetic pattern namely Spalling Craft for outlier exposure to add some anomaly patterns into the inlier building images to enhance model performance. The proposed approach outperforms the state-of-the-art baselines by approximately 18.4%, 46.6% and 31.7% in AUC, AP and FPR95 score, respectively. Compared to supervised learning, our approach reduces the time of dataset construction from 200 hours to merely 1 hour, which is only 0.5% of the original labeling time, and saves approximately 36600 NTD in cost. The outcomes of our study will benefit practical application in infrastructure monitoring, enhancing data efficiency in tile spalling segmentation.

**Keywords:** structural health monitoring, supervised learning, label efficiency, façade anomaly detection, unsupervised deep contrastive learning, uncertainty estimation, tile spalling segmentation



doi:10.6342/NTU202403549

# Contents





doi:10.6342/NTU202403549





doi:10.6342/NTU202403549

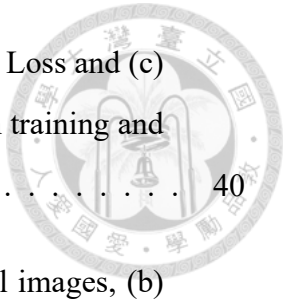
doi:10.6342/NTU202403549

# List of Figures





doi:10.6342/NTU202403549







doi:10.6342/NTU202403549

doi:10.6342/NTU202403549

# List of Tables





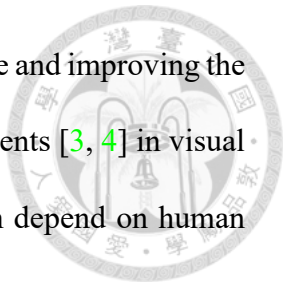
doi:10.6342/NTU202403549

# Chapter 1 Introduction

## 1.1 Motivation and Relevant Works

The use of tile exteriors in architectural design offers great versatility in both aesthetics and functionality. Ensuring the integrity of tiled surfaces is essential, making the detection of tile spalling a key aspect of structural health monitoring. This is crucial due to its effects on the safety, durability, and visual uniformity of buildings. Tiles act as protective barriers for various architectural elements such as floors, walls, and facades, safeguarding the underlying structures from harmful environmental factors like moisture and chemicals. When tiles deteriorate through spalling, their protective capability is compromised, potentially leading to problems like water infiltration, material decay, and even structural instability [1]. Additionally, falling tile pieces present a significant safety risk, particularly when spalling occurs at high elevations. A 2010 survey [2] found that about 85% of 15-floor buildings in Taipei have tile exteriors, with 53% experiencing spalling issues. There is a noticeable trend showing that spalling rates increase as buildings age, indicating that this problem will likely become more severe over time. Early detection of tile spalling allows engineers to address underlying issues promptly, preventing further damage and expensive repairs, thereby extending the lifespan and ensuring the safety of structures. Moreover, fixing spalling tiles not only meets functional requirements but also


doi:10.6342/NTU202403549



enhances the visual appeal of buildings, preserving their aesthetic value and improving the overall perception of the built environment. Despite recent advancements [3, 4] in visual inspection techniques for detecting tile spalling, these methods often depend on human expertise, which can be time-consuming and prone to errors.

Recent advancements in Artificial Intelligence (AI) have been driven by enhancements in computational power, the availability of vast amounts of data, and innovative algorithmic techniques. AI systems now possess the capability to perform tasks traditionally done by humans, such as learning, reasoning, and problem-solving. One significant advantage of AI is its ability to automate intricate and time-consuming operations, thereby increasing efficiency and productivity across various industries. Deep Learning (DL), a branch of AI, focuses on training artificial neural networks to independently learn and make predictions. DL is extensively utilized in computer vision tasks, particularly in object segmentation, which is essential for dividing data or images into segments or regions for analysis [5–7]. Image segmentation is a critical task within computer vision, involving the partitioning of an image into multiple segments to simplify its representation and facilitate analysis. The rise of DL in the 2010s introduced a new framework for image segmentation through Convolutional Neural Networks (CNNs), enabling end-to-end learning and achieving state-of-the-art results [8–10]. Notable advancements include Fully Convolutional Networks (FCN) [11], which in 2015 enabled pixel-wise predictions without the need for manual feature engineering, surpassing traditional methods in performance. Subsequent research focused on refining CNN architectures, leading to innovations like U-Net [12], which improved segmentation accuracy with skip connections. Over time, CNN-based image segmentation has continued to evolve, incorporating techniques such as atrous convolutions [13], attention mechanisms [14–16], and encoder-decoder archi-


doi:10.6342/NTU202403549

tectures [17]. These advancements have culminated in state-of-the-art approaches like DeepLab [18] and Mask R-CNN [19]. These DL models have diverse applications, including object recognition, autonomous driving, medical imaging, and video surveillance.

DL has emerged as a focal point in civil engineering due to its versatile applications [20–27]. One significant application involves applying computer vision to evaluate structural elements in infrastructure projects such as bridges, roads, and buildings. Through the analysis of images depicting these structures, engineers can accurately detect and quantify various defects such as cracks, corrosion, and material degradation. This capability is crucial for developing targeted maintenance plans, thereby enhancing the safety and durability of infrastructure [28–34]. Automated crack detection has garnered substantial interest, enabling engineers to assess infrastructure conditions using automated algorithms [35–39]. Similarly, DL techniques are applied to identify tile defects. For instance, Okeke et al. [40] developed a CNN-based binary classifier to distinguish between tiny cracks and normal tile surfaces, while Santos et al. [41] employed U-Net and LinkNet for ceramic crack segmentation, highlighting DL's potential in this area. However, existing studies primarily focus on tile cracking and often use indoor tile features rather than exterior surfaces. Addressing this gap, Kung et al. [42] utilized a modified VGG-16 [43] network to classify defect patterns, including tile spalling on exterior surfaces, albeit with less precise localization results, as shown in Figure 1.1. Furthermore, the lack of open datasets tailored to train DL-based models for tile spalling segmentation exacerbates the challenge. Cao [44] introduced an instance segmentation model, YOLOM, designed for segmenting spalling regions on building façade and proposed a framework using YOLOM with drones for inspection tasks, as shown in Figure 1.2. A team constructed a dataset by capturing images of tile spalling on various structures such as university dormitories, apartment


doi:10.6342/NTU202403549

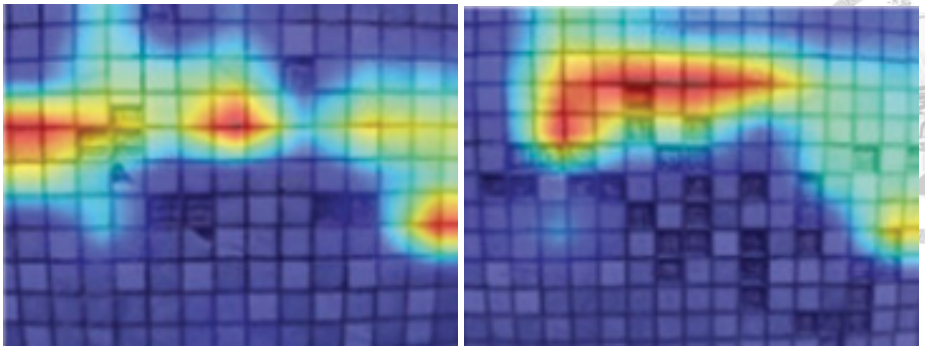


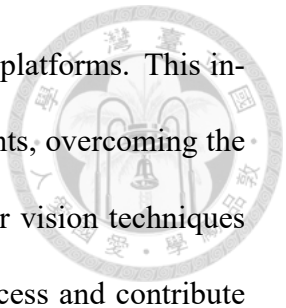
Figure 1.1: Classification results that are

Figure 1.2: Results of instance segmentation

buildings, hospitals, and government offices in Taiwan. These images were meticulously labeled and verified by team members to ensure high-quality annotations, resulting in a dataset containing 1458 images with 4595 instances of tile spalling. This process was time-consuming and labor-intensive due to the extensive effort required for both image collection and labeling. Thus, there are some existing approaches dedicate to train a deep-learning model with small dataset to mitigate the demand on labeled dataset. Katsigiannis et al. [45] tackled crack classification on masonry façades using data augmentation [46] to increase variability and employing transfer learning [47] to enhance model performance on relatively small datasets. This study aims to develop a framework for automatic tile spalling identification using limited data, providing civil engineers with a crucial tool for regular inspection and maintenance of exterior tile surfaces.

In this study, we propose integrating segmentation mechanisms specifically tailored

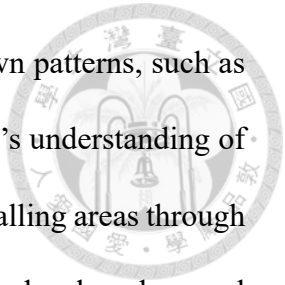

doi:10.6342/NTU202403549



for identifying tile spallings in exterior images sourced from diverse platforms. This innovative approach enhances the accuracy and efficiency of assessments, overcoming the constraints of traditional visual inspections. By harnessing computer vision techniques for spalling segmentation, our goal is to advance the evaluation process and contribute to improved maintenance practices in civil engineering projects. However, the lack of publicly available datasets dedicated to tile spalling poses a significant challenge for developing robust segmentation models in this field. To address this gap, we curated our own dataset by collecting and manually annotating 364 images from sources including Google Street View, social media, and photographs taken in Taiwan over a four-month period. While this dataset is relatively small compared to existing segmentation datasets, its construction underscored the considerable economic and time investments required for gathering labeled spalling images. In response, we devised the Multi-Scale Branch Fusion UNet (MBF-UNet), a novel model that integrates additional branches with distinct receptive fields. This architecture enriches feature extraction by capturing information from images at various scales, thereby maximizing contextual understanding from limited samples. MBF-UNet is tailored to accommodate the diverse sizes of tile spallings, with larger receptive fields effectively handling larger spallings and smaller ones addressing smaller anomalies. As a result, our approach achieves more precise and resilient segmentation outcomes, even when faced with data constraints that challenge conventional models.

Additionally, we propose an unsupervised learning framework designed to identify spallings in pixel-level detail from noisy Google Street View images without relying on labeled spalling data. Our approach leverages uncertainty estimation techniques to train a segmentation model using an existing dataset that includes various inlier classes, not limited to defect-like patterns. The model learns to distinguish these inlier patterns, en-


doi:10.6342/NTU202403549



abling it to assign high uncertainty scores when encountering unknown patterns, such as spallings. By training on spalling-free images, we enhance the model's understanding of typical inlier patterns, thereby facilitating accurate identification of spalling areas through high uncertainty scores. To further improve the model's robustness, we developed a novel outlier exposure method that integrates anomaly patterns into the inlier scenes. This technique guides the model to assign high uncertainty scores specifically to these anomalies, thereby refining its training process. Additionally, we propose a framework for generating spalling patterns that synthesizes realistic spalling images. This unsupervised learning method minimizes the need for extensive spalling image collection and labeling, thereby streamlining the efficiency of model training.

## 1.2 Contribution and Scope

In this work, our main contributions are summarized as follows.

- The proposed DL-based approach effectively identifies and segments spalling regions on tile exteriors. This holds significant potential for infrastructure monitoring, allowing for efficient detection of spalling occurrences. By installing the model in vehicles for regular inspections by authorities, public safety can be improved while reducing the cost of detection.

- We facilitate the research on label efficiency training in both model architecture design and training framework, which is crucial for tile spalling segmentation due to the time-consuming and labor-intensive nature of image collection and spalling labeling. In model architecture, we proposed a novel network named MBF-UNet. This model features three additional branches that capture information at multiple


doi:10.6342/NTU202403549

scales, maximizing the amount of information extracted from a limited dataset. This approach mitigates the lack of contextual features, and improves the identification of spallings of varying sizes. In the training framework, we proposed an unsupervised training approach that uses uncertainty estimation to encourage the model to exhibit high uncertainty when encountering spalling patterns. Additionally, we developed an innovative spalling synthesis approach to achieve the outlier exposure process. This framework allows the model to be trained on spalling-free images without any labels, significantly reducing the cost of dataset construction.

- We build a pixel-level labeled tile spalling dataset with a total of 364 images from various sources to keep the diversity in data for four months. This dataset can be served as a valuable resource for further investigations related to tile spalling identifications.

Section 2 provides a comprehensive overview of the dataset, including our labeled dataset in supervised learning and the two datasets in unsupervised learning. In Section 3, we illustrate our tile spalling identification approach, elucidating the design and the architecture of the proposed MBF-UNet model, and the unsupervised training framework, along with the employed evaluation metrics. Section 4 presents the results, offering an interpretation and discussions about the model performance on the tile spalling dataset. The concluding Section 5 summarizes key contributions and outlines avenues for future research and developments.


doi:10.6342/NTU202403549

# Chapter 2 Dataset

In this study, we propose two approaches: supervised learning and unsupervised learning. Section 2.1 will detail the construction of the labeled dataset used in our supervised learning experiments, including the image acquisition and labeling process. Section 2.2 will introduce the datasets used in the proposed unsupervised learning framework, which consists of two datasets: the source dataset and the target dataset, described in Section 2.2.1 and Section 2.2.2, respectively.

## 2.1 Supervised Learning

### 2.1.1 Labeled Dataset

There are a total of 364 damaged images in this dataset. In order to maintain the high variability of building and tile spalling, we collect the images from different place and different source. This collection includes all seasons in 10 cities located in Taiwan, 90% of our images were obtained from Google Street View, 6.5% were captured using smartphones, and the remaining 3.5% were sourced from various social media platforms. The images from Google Street View were taken from bottom to top to capture high spallings. For single spallings, the region was scaled up before taking a screenshot, as shown in


doi:10.6342/NTU202403549

Figure 2.1a. When multiple spallings were present on a building, the entire building was captured to maintain its appearance, as illustrated in Figure 2.1b. Building photos were taken from a distance of 10 to 30 meters to preserve the overall appearance, as depicted in Figure 2.1c. Samples from social media platforms included images with extreme conditions, such as Figure 2.1d, which shows many small, widely distributed spallings.

And there is no degree or lightness calibration performed after data collection to simulate real-world application conditions, where the model encounters random environmental factors. For the annotation process, we utilized the VGG Image Annotator (VIA) tool [48] to manually create binary masks that indicate the presence of spalling in each image. Figure 2.1 showcases some original images and the corresponding ground-truth masks from our dataset. In addition to the scale of the image and the appearance of the building, the size and appearance of the spalling also vary greatly, demonstrating the diversity of tile spalling instances included in our collected dataset.

Certainly, the dataset poses several challenges. Firstly, supervised learning typically demands a large volume of labeled data, yet annotating tile spalling masks is both time-consuming and labor-intensive. Consequently, we were restricted to using a relatively limited dataset comprising only 364 images for training our segmentation model. This is notably smaller compared to widely-used semantic segmentation datasets like Cityscapes [7], ADE20K [49], and PASCAL VOC [50], each containing at least 5,000 images. Secondly, certain image areas may resemble spallings, such as gray exterior regions and the cement joints between tiles. Lastly, tile spalling exhibits considerable variability in shape and size. Some spallings appear as large, singular areas, while others consist of numerous small patches. These variations pose a significant challenge to model performance. If the model prioritizes learning large-scale features, it may overlook crucial small details. Con-


doi:10.6342/NTU202403549

(a) (b) (c) (d)

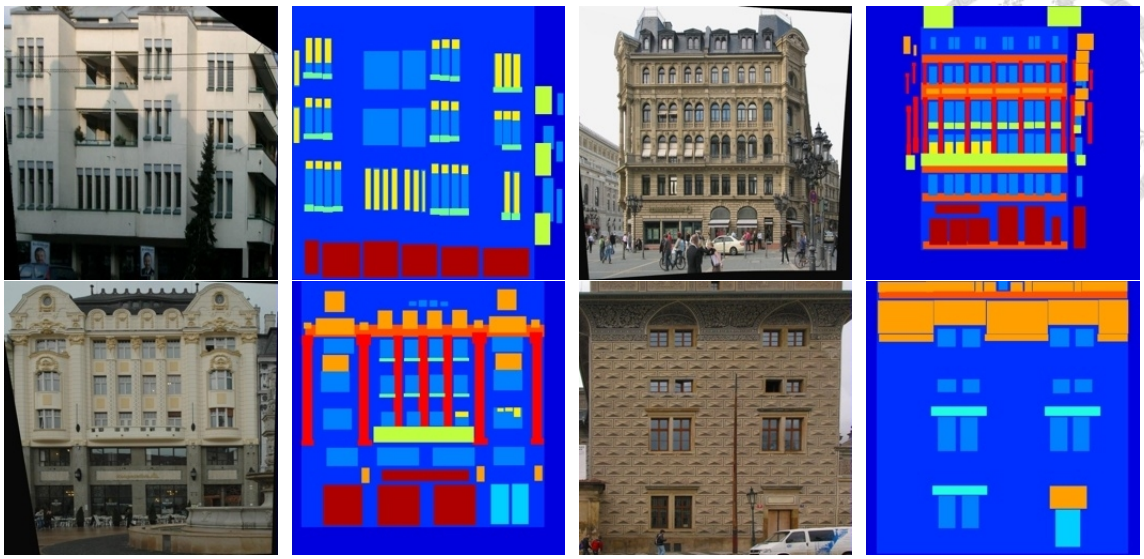
Figure 2.1: Samples of our labeled dataset: (a, c) spalling images, (b, d) ground-truth masks

versely, focusing on fine details might lead to misidentifying regions resembling spallings, especially when trained with limited data resources.

## 2.2 Unsupervised Learning

Our unsupervised learning approach involves two datasets: the source dataset and the target dataset. The source dataset serves as an inlier dataset, teaching the segmentation model the knowledge of those inlier classes within the dataset. The target dataset contains spalling-free tile exterior images from Taiwan, enhancing the model performance on realistic tile images due to the absence of objects in the source dataset. Below, we will detail the process of determining and building these two datasets.


doi:10.6342/NTU202403549



(a) (b) (c) (d)

Figure 2.2: Samples of façades dataset: (a, c) building images, (b, d) ground-truth masks

### 2.2.1 Source Dataset

The source dataset trains the model to recognize inlier classes within the dataset, resulting in a high uncertainty score when the model encounters anomaly patterns, such as tile spallings. Consequently, the source dataset can be any segmentation dataset, not necessarily defect-like, but with images that resemble our target scenes. Therefore, we selected an open dataset known as façades [51], which comprises images of building façades alongside their corresponding annotated segmentations. Each image in the dataset is paired with a label map that delineates various architectural elements such as windows, doors, balconies, and walls, as depicted in Figure 2.2.

### 2.2.2 Target Dataset

Since most building images in the source dataset lack tile exteriors, we created a target dataset to enhance the model performance in identifying tile spallings. We scraped building images from Google Street View using addresses in Taipei City, specifically ex-



doi:10.6342/NTU202403549

(a) (b) (c) (d)

Figure 2.3: Samples of target dataset

cluding samples without tile exteriors. The pitch angle during the scraping process was set to 20 degrees, and the image resolution was 640x640 pixels. This effort resulted in a target dataset comprising a total of 2,540 images, as depicted in Figure 2.3

### 2.2.3 Validation and Test Dataset

We constructed the validation and test datasets for our unsupervised learning approach by sampling 60 labeled real-world images from section 2.1. To simulate the scene in the source dataset and enhance model performance, these images are large-scale, containing the full appearance of buildings rather than just the spalling regions. Examples of these spalling instances are shown in Figure 2.4, highlighting significant variations in shapes and sizes. Furthermore, the images include non-tile exterior elements such as windows and air conditioning units, which could influence model predictions. We divided the dataset into a validation set of 20 images to prevent overfitting and a test set of 40 images to assess model performance.


doi:10.6342/NTU202403549

(a) (b) (c) (d)

Figure 2.4: Samples of the validation and test datasets for our unsupervised learning approach: (a, c) building images, (b, d) ground-truth masks



doi:10.6342/NTU202403549

# Chapter 3 Methodology

## 3.1 Supervised Learning

### 3.1.1 Overview

This section focuses on developing a framework for the autonomous identification of tile spalling, an essential tool for civil engineers conducting routine inspections and maintaining the integrity of tile exteriors, as illustrated in Figure 3.1. The inspection process starts with an initial image of tile spalling in the top-left. Our proposed model can accurately identify spalling in building images, providing engineers with crucial information about spalling distribution and the exterior condition, aiding in the assessment of necessary repairs. Here, we detail the design and training process of the proposed model for tile spalling segmentation. In section 3.1.2, we describe the architecture of our multi-branch model, as depicted in Figure 3.2. The MBF-UNet features multiple branches, labeled $B_1$ to $B_m$, each designed to capture features of different sizes within the image. For our experiments, we set $m$ = 3, allowing the model to detect features at three distinct levels. These branches operate in parallel, sharing feature information extracted by the encoder and producing the spalling mask from concatenated feature maps via the segmentation head. In section 3.1.3 and section 3.1.4, we explain the key components integrated into


doi:10.6342/NTU202403549

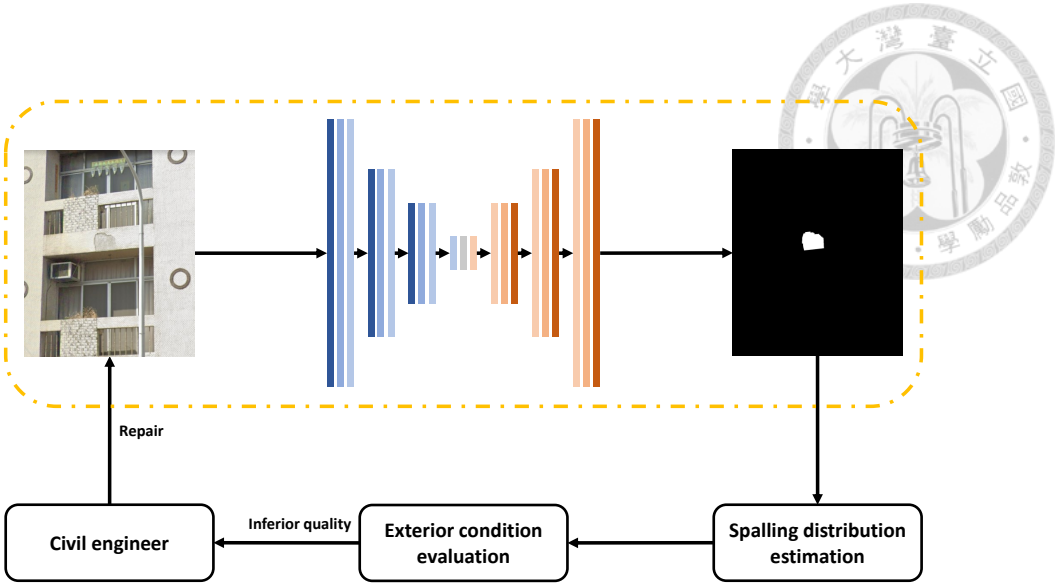



Figure 3.1: The autonomous tile exterior inspection process of the proposed approach, the dashed lines indicate the scope of this study.

the model, namely Squeeze-and-Excitation (SE) Blocks [52] and Atrous convolution, respectively. In section 3.1.5, we compare the baseline optimization method with our proposed optimization approach. Sections 3.1.6 to 3.1.9 provide detailed information on our experiments, covering hyperparameters, the encoder, data augmentation techniques, and other essential training settings. And section 3.1.10 discusses the evaluation metrics used in segmentation, such as mean intersection over union and the Hausdorff distance.

### 3.1.2 Network Architecture

To capture features of varying sizes, we employ a technique known as multi-scale information extraction. This approach is essential for incorporating comprehensive contextual features into image segmentation tasks. Recent studies [53–56] have effectively used multi-scale information to enhance contextual aggregation. To leverage this, our model decoder includes three parallel branches, as depicted in Figure 3.3, each branch having different kernel sizes. This design allows the model to extract features at various scales,


doi:10.6342/NTU202403549

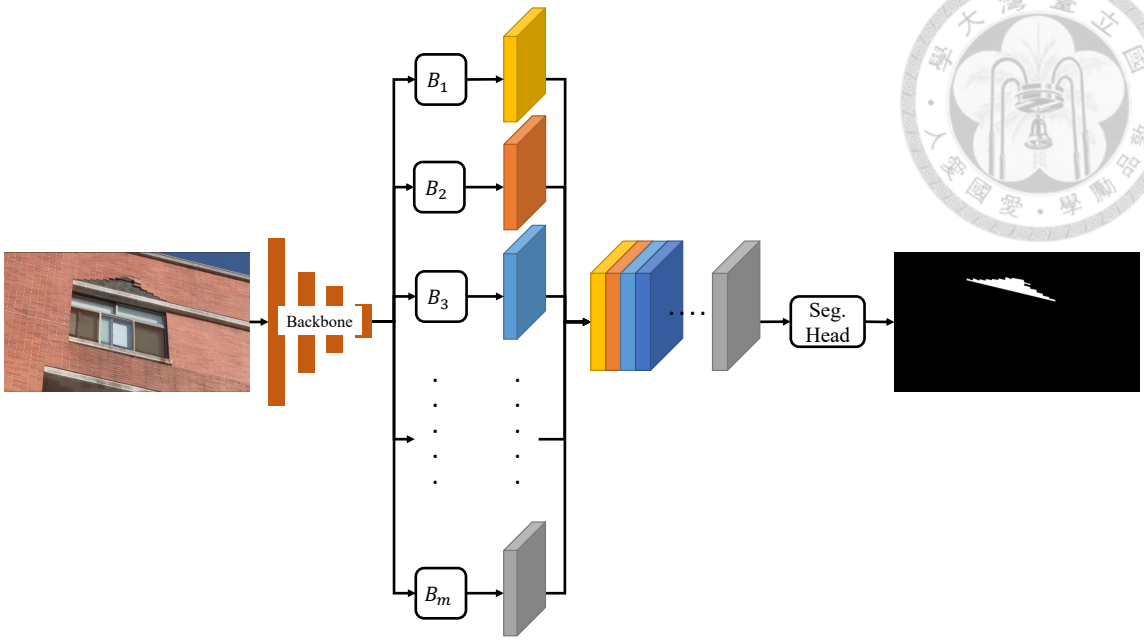



Figure 3.2: The architecture of the proposed MBF-UNet

with the branches utilizing the same feature information from the encoder. The backbone of our decoder is based on the U-Net++ [57] architecture, which integrates SE blocks to achieve self-attention. To prevent the loss of fine image details during the down-sampling and up-sampling stages, U-Net uses skip connections between the encoder and decoder. These connections help maintain detailed information in the images. U-Net++ extends this concept by using nested and dense skip connections, adding more nodes to the skip pathways to enhance semantic similarity between the feature maps of the encoder and decoder networks. To improve the model's performance further, SE blocks are incorporated. Atrous convolution is introduced to mitigate the increase in training parameters that come with larger convolution kernels. The parallel branches use three different dilation rates: 1, 4, and 8. Branches with higher dilation rates create broader receptive fields, capturing more extensive contextual information, while branches with smaller dilation rates focus on finer details and local features. This multi-scale approach enables the model to capture information at different levels, addressing both global and local dependencies within the data. The information from these branches is then aggregated and combined through a


doi:10.6342/NTU202403549

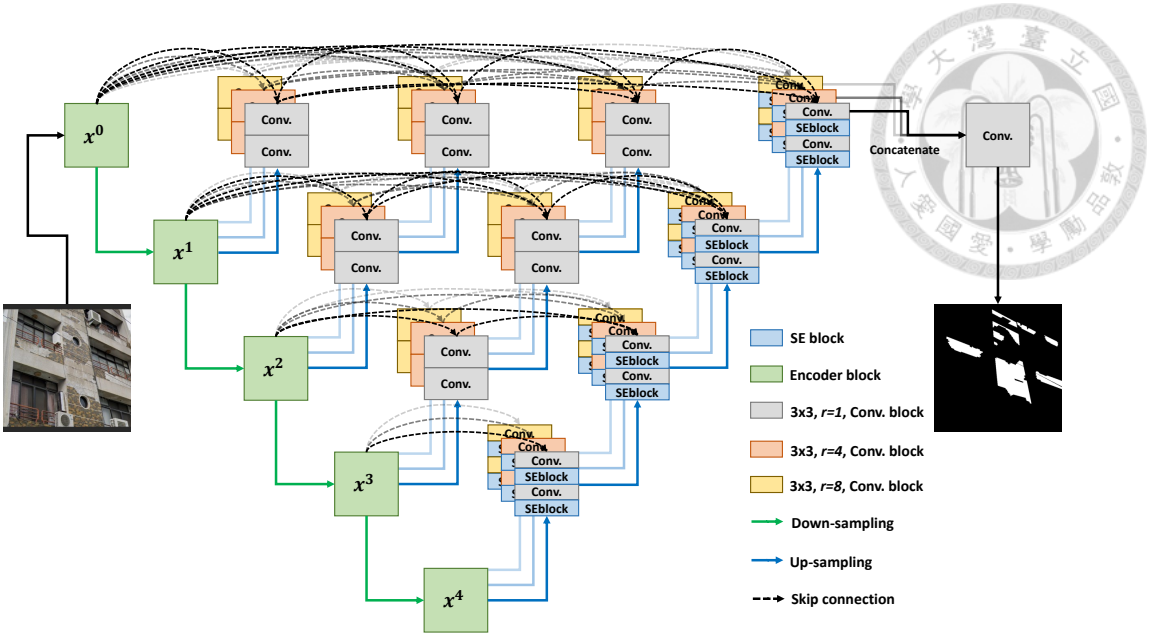



Figure 3.3: The complete architecture of the proposed MBF-UNet

convolution layer, resulting in a robust and comprehensive feature extraction process.

### 3.1.3 Squeeze-and-Excitation Blocks

Squeeze-and-Excitation (SE) Blocks are mechanisms designed to enhance the learning of convolutional features by explicitly modeling channel interdependencies. This process increases the network's sensitivity to informative features. The SE block operates in two steps: squeeze and excitation, as illustrated in Figure 3.4. In the squeeze step, the proposed model aggregates global spatial information into a channel descriptor. This is done by applying a global pooling layer to the input, resulting in channel-wise statistics represented by $z \in \mathbb{R}^C$, where the input is denoted as $u \in \mathbb{R}^{H \times W \times C}$. The $C$-th element of $z$ is calculated by:

$$z_C = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_C(i,j). \tag{3.1}$$

The excitation step is designed to capture channel-wise dependencies comprehensively. In this study, the model generates channel weights denoted as $s \in \mathbb{R}^C$ by incorporating


doi:10.6342/NTU202403549

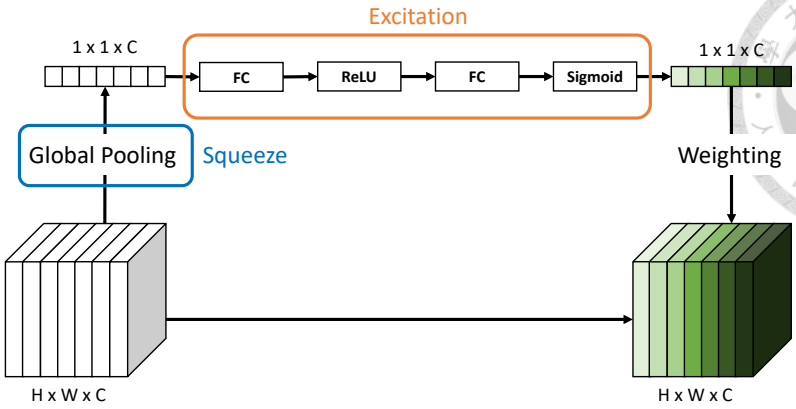



Figure 3.4: The Squeeze-and-Excitation block

two sets of fully connected layers (FC) and an activation function:

$$s = \sigma(\mathbf{W_2}\delta(\mathbf{W_1}z)). \tag{3.2}$$

where $\delta$ refers to the ReLU [58] function, and $\sigma$ refers to the Sigmoid function, The weights of the first and second FC layer are represented by $\mathbf{W_1}$ and $\mathbf{W_2}$, respectively. To generate the final output, the model re-weights the input $u_C$ by performing an element-wise multiplication with the corresponding channel weight $s_C$. By doing this, the model emphasizes channels according to their importance, enhancing the feature representation and improving the learning process.

### 3.1.4 Atrous Convolution

Atrous convolution employs dilation or gaps between the convolution kernel weights, which enables the kernels to enlarge receptive field without increasing the number of training parameters. This is illustrated in Figure 3.5, where the hyperparameter $r$ represents the dilation rate. The size of the receptive field can be designed by:

$$H_{kernel} = W_{kernel} = (r-1) \times (n-1) + n. \tag{3.3}$$


doi:10.6342/NTU202403549

(a) (b)

Figure 3.5: (a) The conventional $9 \times 9$ convolution kernel, and (b) the $3 \times 3$, $r$=4 atrous convolution kernel

where $n$ refers to the size of the non-zero filter. In this work, we design the largest kernel as a $3 \times 3$, $r$=4 atrous convolution kernel, which is 9 times fewer training parameters than the conventional $9 \times 9$ convolution kernel.

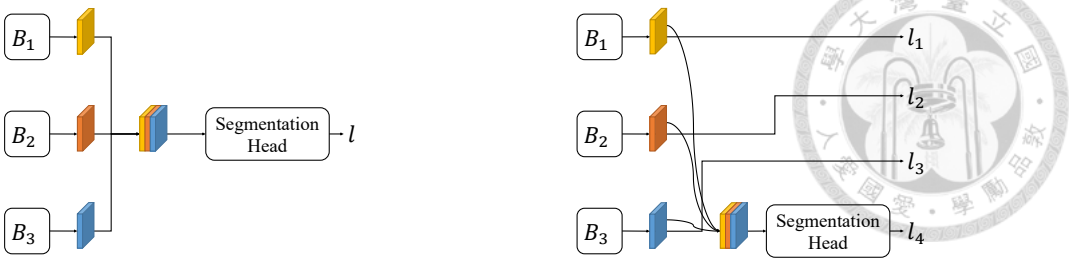
### 3.1.5 Optimization Schemes

To search potential improvements in model performance by enhancing the distinctive capabilities of each branch, we explore two different optimization schemes. The first scheme contains optimizing the model solely based on the loss $l = L(x, y)$ between the final prediction $\hat{y}(w \mid x)$ and the ground-truth $y$:

$$l = L(x, y) = \min_{w} J(\hat{y}(w \mid x), y) \tag{3.4}$$

where $J$ represents the loss function, $w$ means the model weights, and $x$ is the inputs. In this scenario, the focus is primarily on ensuring the accuracy of the final output without interfering with the specific objectives of each branch, as shown in Figure 3.6a. The second scheme incorporates the loss calculated by taking the average of the losses between


doi:10.6342/NTU202403549




(a) (b)

Figure 3.6: (a) The direct optimization scheme, and (b) the individual optimization scheme

the three branches, the final output, and the ground-truth, denoted as $l_1$ to $l_4$:

$$l_i = L_i(x, y) = \min_{\mathbf{w_i}} J(B_i(\mathbf{w_i} \mid x), y) \qquad i = 1 \sim 3 \tag{3.5}$$

$$l_4 = L_4(x, y) = \min_{\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_3}, \mathbf{w_4}} J(\hat{y}(\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_3}, \mathbf{w_4} \mid x), y) \tag{3.6}$$

Where $B_i$ are the $i$-th branch in the decoder, $\mathbf{w_i}$ represents the corresponding model weights, and $\mathbf{w_4}$ is the weights of the $3 \times 3$ convolution layer that produces the prediction. The motivation is to encourage each branch to focus on spalling detection, thereby enhancing the individual performance of each branch, as illustrated in Figure 3.6b. In summary, the former strategy prioritizes the accuracy of the final output, while the latter strategy aims to improve each branch's performance by enhancing its ability to distinguish spallings.

### 3.1.6 EfficientNet

In this study, we adopt EfficientNet [59] as our model encoder. EfficientNet was developed using neural architecture search (NAS) [60], which incorporates compound scaling, a technique that uniformly scales the depth, width, and resolution of the model using a compound coefficient. The baseline model optimizes the product $ACC(m) \times [FLOPS(m)/T]^w$ during NAS, where $ACC(m)$ and $FLOPS(m)$ denote the accuracy and floating-point operations per second (FLOPS) of model $m$, respectively. Here, $T$


doi:10.6342/NTU202403549

represents the target FLOPS, and $w = -0.07$ serves as a hyperparameter controlling the trade-off between accuracy and FLOPS, defining the base model as EfficientNet-B0. The compound scaling method utilizes a compound coefficient $\phi$ to systematically scale the width, depth, and resolution of the network. This scaling process is performed as follows:

$$d = \alpha^{\phi} \quad w = \beta^{\phi} \quad r = \gamma^{\phi} \tag{3.7}$$

$$\text{s.t.} \qquad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{3.8}$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{3.9}$$

where $d$, $w$, and $r$ means depth, width, and resolution; $\alpha$, $\beta$, $\gamma$ are constants that can be determined by a small grid search. For EfficientNet-B0, the optimal values are $\alpha$ = 1.2, $\beta$ = 1.1, $\gamma$ = 1.15 when $\phi$ = 1. Once these values are fixed as constants, EfficientNet-B0 is scaled up with different values of $\phi$ to create models from EfficientNet-B1 to EfficientNet-B7.

In this study, we utilize EfficientNet-B6 as our model encoder. This decision is driven by its optimal balance of computational efficiency and high performance, which are particularly advantageous for our segmentation tasks. The compound scaling ensures that the model maintains high accuracy while remaining computationally efficient, aligning well with the demands of our application in spalling segmentation. By leveraging the capabilities of EfficientNet-B6, our goal is to attain robust and accurate segmentation results, bolstered by our tailored application focus and training methodologies.


doi:10.6342/NTU202403549

### 3.1.7 Baseline Approaches in Segmentation

We select 4 state-of-the-art semantic segmentation models to compare with our proposed model as follows:

**U-Net [12]:** U-Net is a convolutional neural network designed for biomedical image segmentation. It employs a encoder-decoder architecture to capture context and a symmetric expanding path for precise localization. To prevent the loss of high-resolution features during the convolution process, U-Net utilizes skip connections between corresponding layers in the contracting and expanding paths. These connections help retain high-resolution features, ensuring more accurate segmentation results.

**U-Net++ [57]:** An extension of U-Net, U-Net++ introduces a series of nested and dense skip connections aimed at enhancing the accuracy and efficiency of segmentation. These redesigned skip pathways and dense connections help bridge the semantic gap between the encoder and decoder sub-networks, resulting in improved feature fusion and more precise segmentation outcomes. The nested architecture allows for better gradient flow and more effective learning of complex patterns.

**MA-Net [15]:** MA-Net (Multi-scale Attention Network) is designed to capture features at multiple scales while applying attention mechanisms to improve segmentation performance. It integrates multi-scale feature extraction with attention mechanisms, which enhances the segmentation of objects at different scales and helps the model focus on the most relevant parts of the image. This combination allows MA-net to effectively identify and segment objects of varying sizes within an image.

**DeepLabV3+ [13]:** DeepLabV3+ is an advanced version of DeepLabV3, designed


doi:10.6342/NTU202403549

for semantic image segmentation. It integrates atrous spatial pyramid pooling (ASPP) to capture multi-scale information and includes a decoder module for improved delineation of object boundaries. The use of atrous convolution allows it to control the resolution of feature responses, while the straightforward yet effective decoder refines the segmentation results.

### 3.1.8 Baseline Approaches in Addressing Limited Data

The challenge of limited datasets is a significant issue in deep learning approaches, especially in image segmentation problems. One major hurdle is the extensive labeling required, involving manual pixel-level annotation of ground-truth data. Moreover, acquiring defect images is often time-consuming and labor-intensive. These factors contribute to low data variability, increasing the risk of model overfitting. To address the limitations of limited training data, two common baseline approaches are employed: data augmentation and transfer learning. Data augmentation aims to increase dataset variability by applying various image processing techniques such as horizontal or vertical flipping, random cropping, brightness and contrast adjustments, and adding Gaussian noise. In this study, we implemented augmentations including horizontal flipping, distortion, random contrast and brightness adjustments, and Gaussian noise. These augmentations diversify the data, thereby mitigating overfitting during model training.

On the other hand, transfer learning involves using a pre-trained model from a related task instead of starting from scratch. The pre-trained model has learned general features from a large dataset, such as ImageNet [61], which contains millions of labeled images across numerous categories. This approach is particularly beneficial when the new task has limited labeled data available. Transfer learning not only reduces the computational


doi:10.6342/NTU202403549

resources and training time but also often enhances model performance on the target task. It is widely used in structural health monitoring, since the data is often hard to obtain [46, 62]. In our study, we adopt the pre-trained EfficientNet encoder from ImageNet as our baseline, leveraging its universal features for our spalling segmentation task. This approach is compared with our proposed MBF-UNet model and the two baseline methods —data augmentation and transfer learning—are evaluated in section 4.1.3.

### 3.1.9 Experiments

To mitigate model bias that may result from a limited test dataset, all models, including the baseline references, are trained seven times using different training, validation, and testing sets. This approach ensures a more robust assessment of model performance. The data splitting is conducted as follows: 291 samples are randomly selected for training, 36 for validation, and 37 for testing, adhering to an 8:1:1 ratio based on the Pareto principle that 20 percentage of causes are responsible for 80 percentage of the effects or outcomes. Model performance is evaluated on the test set using the weights that achieved the highest mean intersection over union (mIoU) score on the validation set. The training pipeline involves data pre-processing by resizing the images to 864 x 864 pixels and applying various data augmentations. The loss function employed during training is a weighted combination of Lovász-Softmax loss $L_{Lov}(\hat{y}, y)$ [63] and cross entropy (CE) loss $L_{CE}(\hat{y}, y)$ to mitigate gradient saturation. The loss function is expressed as follows:

$$L(\hat{y}, y) = \alpha \cdot L_{Lov}(\hat{y}, y) + (1 - \alpha) \cdot L_{CE}(\hat{y}, y) \tag{3.10}$$

Where $\alpha = 0.5$ represents the ratio of the Lovász-Softmax loss to the cross-entropy loss. The model is implemented in Pytorch [64] and trained on a Linux server equipped with


doi:10.6342/NTU202403549

four Intel Xeon E5-2620 CPUs, 256 GB DDR4 RAM, and eight NVIDIA RTX Quadro 8000 GPUs with 48 GB memory. Table 3.1 summarizes the training setting used in this study.

Table 3.1: The training setting is used in the unsupervised experiment

| Training Setting | Value |
|---|---|
| Optimizer | Adam [65] |
| Max. learning rate | 0.0001 |
| Weight decay | 0.0001 |
| Max. epoch | 80 |
| Batch Size | 3 |

### 3.1.10 Evaluation Metrics

To effectively evaluate the segmentation performance of the proposed for tile spalling, we utilize some general evaluation metrics in semantic segmentation models evaluation as follow:

**Pixel Accuracy:** Pixel accuracy is the proportion of correctly classified pixels out of the total number of pixels in the image. This metric gives an overall measure of how many pixels in the image are classified correctly. While useful, it can be misleading if the dataset is imbalanced (i.e., one class dominates the others), as the model can achieve high accuracy by simply predicting the majority class.

**Precision:** Precision is the ratio of true positive predictions to the sum of true positives and false positives. It measures the accuracy of the positive predictions. High precision indicates that when the model predicts a pixel as belonging to a particular class, it is often correct. This metric is particularly important in applications where the cost of false


doi:10.6342/NTU202403549

positives is high. It is defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.11}$$

**Recall:** Recall is the ratio of true positive predictions to the sum of true positives and false negatives. It measures the model ability to capture all relevant instances (true positives). High recall means the model successfully identifies most of the positive instances, which is crucial in applications where missing a positive instance (false negative) is costly such as our work. It is defined as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negative}} \tag{3.12}$$

**F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a single metric that balances the two. It is useful when we need a balance between precision and recall, especially in cases of imbalanced datasets. It provides a more comprehensive measure of the model performance, taking both false positives and false negatives into account. It is defined as follows:

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.13}$$

**mIoU:** mIoU is a widely used metric for evaluating pixel-level quality in image segmentation tasks [12, 13, 57]. It quantifies model performance by calculating the intersection over union (IoU) between the predicted segmentation and the ground-truth masks, averaged across all classes. The mIoU score ranges from 0 to 1, with higher scores indi-


doi:10.6342/NTU202403549

cating better performance. It is defined as follows:

$$\text{mIoU}(y, \hat{y}) = \frac{y \cap \hat{y}}{y \cup \hat{y}} \tag{3.14}$$

where $y$ and $\hat{y}$ denote the ground-truth and the predicted segmentation masks, respectively.

**HD95:** The Hausdorff distance measures dissimilarity between sets of points or shapes, offering insight into their spatial deviations. Unlike the mIoU score, higher HD95 scores indicate poorer performance. For two sets $A$ and $B$, the Hausdorff distance is defined as the maximum distance from a point in set $A$ to its nearest point in set $B$. It is given by:

$$\text{HD}(A, B) = \max\{\max_{s_A \in S(A)} d(s_A, S(B)), \max_{s_B \in S(B)} d(s_B, S(A))\} \tag{3.15}$$

$$d(v, S(A)) = \min_{s_A \in S(A)} (\|v - s_A\|) \tag{3.16}$$

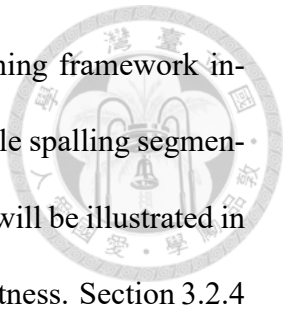
where $v$ denote the arbitrary point in the boundary. The HD95 metric is a region quality evaluation that focuses on the 95th percentile of Hausdorff distances, providing a more robust assessment less affected by outliers or extreme cases. This metric is commonly used in segmentation studies and competitions [66–68] to quantify performance.

## 3.2 Unsupervised Learning

### 3.2.1 Overview

In this section, we provide detailed information about the design of our unsupervised learning training approach, including the training framework, Spalling Craft, and other


doi:10.6342/NTU202403549



training details. In section 3.2.2, we describe the unsupervised training framework inspired by the uncertainty estimation mechanism, which can train the tile spalling segmentation model using only spalling-free building images. Spalling Craft will be illustrated in section 3.2.3, which achieves outlier exposure to enhance model robustness. Section 3.2.4 presents how the contrastive loss improves the model performance in real-world images and how it is designed. The model architecture used in the experiments will be shown in section 3.2.5. Section 3.2.6 to 3.2.8 will cover more details about our experiments, including the hyperparameters, the encoder, evaluation metrics, and the baseline approaches we design.

### 3.2.2 Training Framework

The proposed training framework is designed based on uncertainty estimation. We train a model using predefined categories so that when it encounters a pattern outside those categories, it exhibits high uncertainty compared to the known categories due to its lack of familiarity with the new pattern. Therefore, we can estimate the uncertainty to determine whether a pattern is an anomaly. The source dataset trains the model to recognize the normal appearance of buildings, while the target dataset improves the model performance on tiled exteriors. Additionally, outlier exposure modules are employed with spalling-free building images from both datasets to enhance the model robustness in anomaly detection. The segmentation model is trained using images with synthetic spalling patterns. Consequently, the model is constrained with three loss functions, as illustrated in Figure 3.7. Images ($x^S$) from the labeled source dataset are fed into an outlier exposure module ($OE^S$) to synthesize spalling patterns. The spalling area is defined as the outlier region, and the rest of the image is the inlier region. These images then pass through the segmentation model,


doi:10.6342/NTU202403549

generating logits ($\hat{y}^S$) and feature embeddings ($e^S$) , which are used in contrastive loss, detailed in section 3.2.4. The model learns the inlier classes using an inlier loss ($\mathcal{L}_{inlier}$), comparing the logits of the inlier region to the labels of the inlier dataset. Cross-entropy loss is employed as the inlier loss function. The logits of the outlier region are calculated using an outlier loss, implemented as positive energy loss in the experiment as follow:

$$\mathcal{L}_{outlier}(\hat{y}) = \sum_{\hat{y} \in \hat{y}_{out}} (max(0, -E(\hat{y}))) \tag{3.17}$$

$$E(x) = -log \sum_{i=0}^{K} exp(x_i) \tag{3.18}$$

$K$ is the number of categories in the source dataset, it is 13 in this experiment. The uncertainty score obtained by Eq. 3.18 is increased in regions belonging to the spalling pattern (outlier), resulting in a relatively high uncertainty score compared to the spalling-free region. The model also has a branch to project image features to a latent space to implement contrastive loss. This branch is used only during training and not during inference, where Eq. 3.18 is leveraged to estimate the uncertainty score to determine whether a pixel is spalling. The same process is applied to the target dataset, excluding the inlier loss due to the absence of labels. A batch of source images pairs with a batch of target images, sharing the same model, but the outlier exposure modules ($OE^S$ and $OE^T$) are different. This allows adjustments based on the appearance of the datasets. The following results will demonstrate the model performance in different settings.

### 3.2.3 Spalling Craft

Spalling Craft serves as our designed spalling synthetic approach, functioning as an outlier exposure module in Figure 3.7. Since the building images in our datasets are all


doi:10.6342/NTU202403549

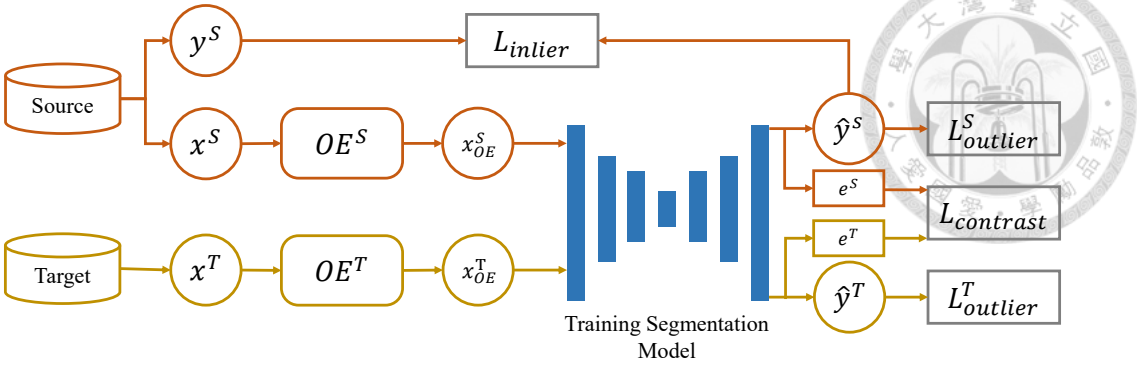



Figure 3.7: The complete unsupervised learning framework

spalling-free, we synthesize fake spalling patterns onto the buildings to induce higher uncertainty scores in those regions. Spalling Craft utilizes image processing techniques to generate realistic spalling patterns, as depicted in Figure 3.8. The process begins with localization: we generate spalling positions ($y_{OE}$) by randomly generating $k$ polygons as candidate regions (k is 10 in this experiment). To ensure spalling occurs only on exterior surfaces and not on non-exterior areas like windows or trees, we employ a noise filter $f$ to exclude those region. This filter utilizes two pre-trained segmentation models from the Cityscape dataset [7] and the Façades dataset. Regions identified as buildings in the Cityscape dataset and façades in the Façade dataset are considered valid regions.

The next step involves content generation: we combine Perlin Noise ($\epsilon_p$), known for its ability to add natural-looking variations, with the gray-scale image ($x_{gray}$). This combination captures details such as shadows and textures on building façades. Perlin Noise is preferred for its capability to generate continuous, smooth variations, avoiding the abrupt edges and repetitive patterns often associated with simpler noise types like Gaussian noise. As a result, the content ($C$) is designed as follow:

$$C = \alpha * x_{gray} + (1 - \alpha) * \epsilon_p \tag{3.19}$$

$\alpha$ = 0.5 in this study represents the ratio between the gray-scale image ($x_{gray}$) and Perlin


doi:10.6342/NTU202403549

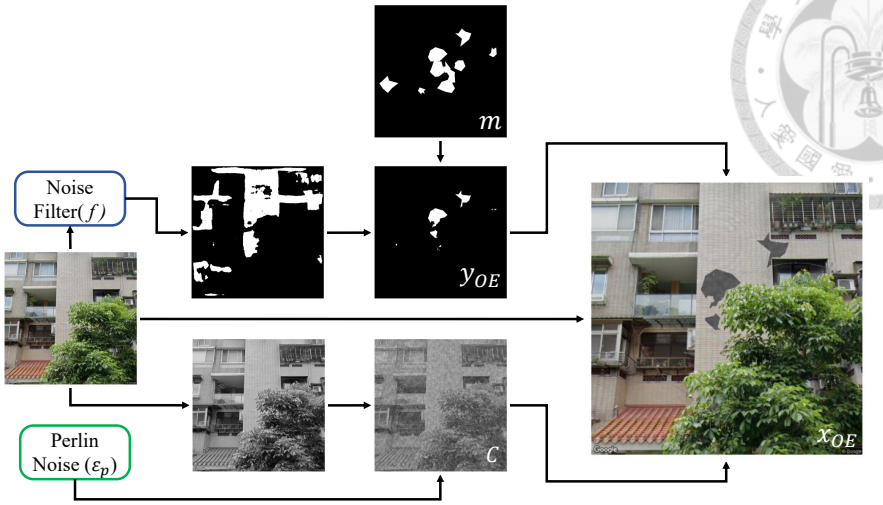



Figure 3.8: The process of Spalling Craft

Noise ($\epsilon_p$). Therefore, the synthetic image ($x_{OE}$) is defined as follow:

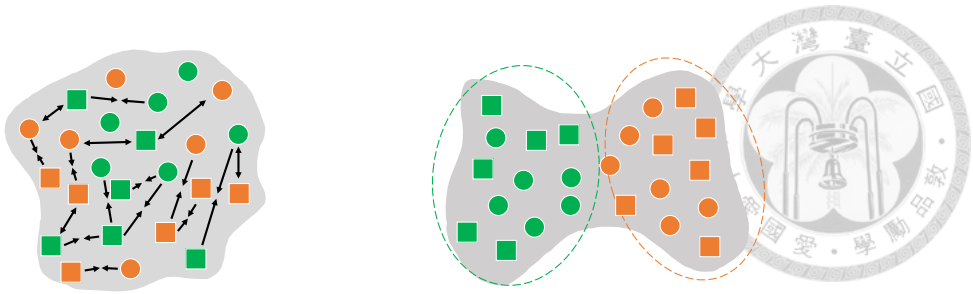
$$x_{OE} = (1 - y_{OE}) * x + y_{OE} * C \tag{3.20}$$

This equation combines the gray-scale image and Perlin Noise according to the specified ratio, resulting in the synthetic image with added spalling patterns.

### 3.2.4 Contrastive Loss

The contrastive loss is designed to increase the distance between inlier features and outlier features, thereby facilitating the model ability to classify them correctly. It achieves this by pulling together feature embeddings with the same classes (inlier and outlier) while pushing apart those from different regions. Inspired by pixel-wise embedding learning [69], we define an anchor set $A = \{x_i \mid x_i \sim x, x \in D_S\}$ to randomly sample an embedding $x_i$ from the source dataset($D_S$). Additionally, we design a contrastive set $C = \{x_i \mid x_i \sim x, x \in (D_S \cup D_T)\}$ from either the source dataset $D_S$ or the target dataset


doi:10.6342/NTU202403549



(a) (b)

Figure 3.9: The impact of contrastive loss, the embedding distribution in latent space will be from (a) to (b).

$D_T$, creating a contrastive embedding for comparison with the anchor embedding. The contrastive learning loss is then defined as follows:

$$\mathcal{L}_{CL} = \sum_{x_i \in A} \sum_{e^+ \in P(x_i)} -log \frac{exp(x_i * e^+/\tau)}{exp(x_i * e^+/\tau) + \sum_{e^- \in N(x_i)} exp(x_i * e^-/\tau)} \tag{3.21}$$

where $P(x_i) = \{x_j \mid x_j \sim C, m_i = m_j\}$ is a set that randomly samples another embedding $x_j$ from $C$ with the same class $m$ (inlier and outlier). And $N(x_i) = \{x_j \mid x_j \sim C, m_i \neq m_j\}$ is a set that randomly samples another embedding $x_j$ from $C$ with different class, $\tau = 1$ in the experiment is a hyper-parameter. Ideally, the impact of contrastive loss is shown as Figure 3.9, yellow represents the embeddings belonging to outliers, blue represents those from inliers. Triangles denote embeddings from the source dataset, and circles denote embeddings from the target dataset. The source embeddings serve as anchors to pull together the embeddings of the same class and push apart embeddings of different classes. This approach transforms the embedding distribution in the latent space from the arrangement shown in Figure 3.9a to that in Figure 3.9b, indicating that the model effectively distinguishes between outliers and inliers.


doi:10.6342/NTU202403549

### 3.2.5 Model Architecture

Due to limitations of our computing platform, we select DeepLabV3+ [13] as the training segmentation model instead of our proposed MBF-UNet, as depicted in Figure 3.7, and apply a 1x1 convolution layer to project the feature as embedding [70] in latent space. The complete model architecture is depicted in Figure 3.10. The backbone of the model, which we have chosen to be EfficientNet-B6 [59], extracts features from the image. Following this, the atrous spatial pyramid pooling (ASPP) layer, consisting of a 1x1 convolution layer, an image pooling layer, and three 3x3 convolution layers with dilation rates of 12, 24, and 36, extracts multi-scale features. These feature maps are then concatenated and projected to form a low-resolution feature map. Simultaneously, the model retains a high-resolution feature map from an intermediate layer of the backbone. This high-resolution feature map is concatenated with the low-resolution feature map, and two individual branches are employed on the combined features. One branch predicts the logits to evaluate the probability distribution of each pixel, and subsequently uses $E(x)$ as defined in Eq. 3.18 to estimate the uncertainty representing anomaly score. The other branch, containing a 1x1 convolution layer, projects the features to an embedding $e$ for contrastive learning, where $e \in \mathbb{R}^{H*W*304}$.

### 3.2.6 Experiments

The training dataset consists of a source dataset and a target dataset. The source dataset contains 606 labeled images, while the target dataset contains 2540 unlabeled images. The model will be pre-trained in source dataset before training. During each epoch of the training process, 606 target images are randomly selected and paired with source im-


doi:10.6342/NTU202403549

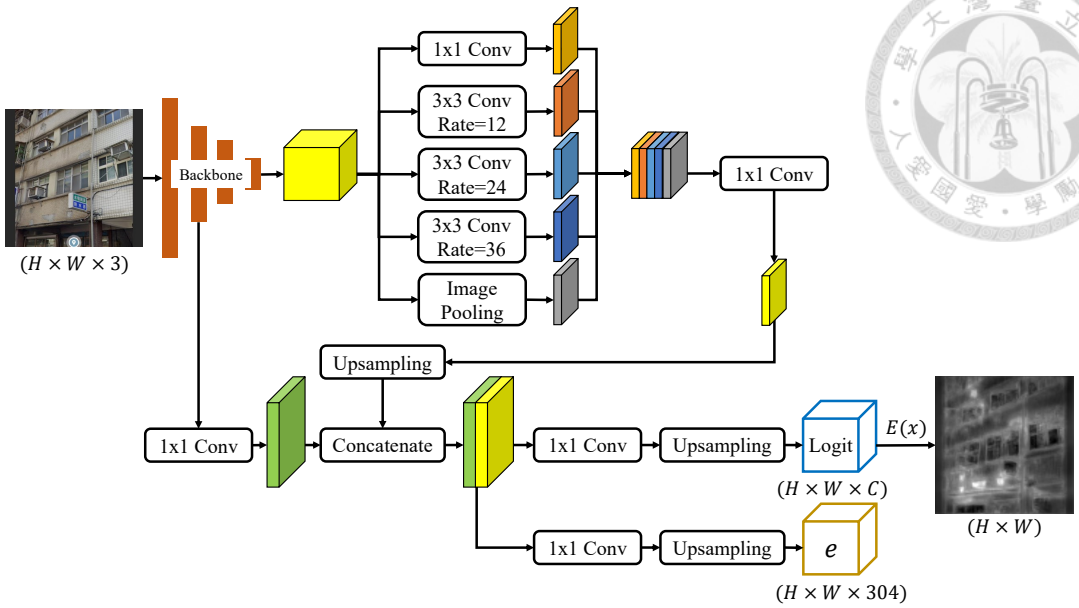



Figure 3.10: The complete model architecture

ages and their labels to meet the requirements of contrastive learning. The model weights that achieved the highest average precision (AP) score on the validation dataset were selected as the final model weights to avoid overfitting. Model will be evaluated in test dataset using some general metrics in anomaly detection. To prevent image distortion, we applied image padding to convert images to square shapes and then resized them to 512 x 512 for training. Table 3.2 summarizes the training setting used in this study.

Table 3.2: The training setting is used in the unsupervised experiment

| Training Setting | Value |
|---|---|
| Optimizer | Adam [65] |
| Scheduler | one-cycle learning rate scheduler |
| Max. learning rate | 0.00005 |
| Weight decay | 0.0001 |
| Max. epoch | 10 |
| Batch Size | 3 |
| Backbone | Efficientnet-B6 |



doi:10.6342/NTU202403549

### 3.2.7 Evaluation Metrics

Different from segmentation maps in Section 3.1, our outputs are anomaly maps. We incorporate several general evaluation metrics from road anomaly detection studies [70–76] to assess the model performance different from Section 3.1.10. These metrics include Average Precision (AP), Area Under the Curve (AUC), and False Positive Rate at 95% True Positive Rate (FPR95). Their definitions and functions are as follows:

**AP:** AP is a measure used to evaluate the performance of a binary classifier in terms of precision and recall. Precision is the ratio of true positive predictions to the total number of positive predictions, while recall is the ratio of true positive predictions to the total number of actual positives. The precision and recall are defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.22}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.23}$$

AP summarizes the precision-recall curve, which plots precision versus recall at varying thresholds settings. It is calculated as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{3.24}$$

where $P_n$ and $R_n$ are the precision and recall at the $n$-th threshold.

**AUC:** AUC represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at


doi:10.6342/NTU202403549

varying thresholds settings. The TPR and FPR are defined as:

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.25}$$

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{3.26}$$

The AUC value ranges from 0 to 1, where 1 indicates a perfect classifier and 0.5 indicates a classifier that performs no better than random chance.

**FPR95:** FPR95 is a metric used to evaluate the robustness of a binary classifier. It specifically measures the False Positive Rate (FPR) when the classifier achieves a True Positive Rate (TPR) of 95%. FPR95 helps in understanding the trade-off between detecting true positives and the rate of false positives at a high level of sensitivity (95% TPR). This metric is particularly useful in applications where it is crucial to maintain a high detection rate while controlling the number of false alarms.

### 3.2.8 Baseline Approaches

Due to the absence of existing anomaly detection and unsupervised learning methods for tile spalling segmentation, we designed some baselines for comparison with our approach. These baselines utilize the same segmentation model architecture, DeepLabV3+, to ensure consistent training parameters and computing time:

**Synthetic Supervised:** We employ supervised learning using the synthetic images and masks generated by the Spalling Craft method described in section 3.2.3. All 2540 target images are fed into the Spalling Craft module to generate fake spalling patterns on tile exteriors, paired with the corresponding synthetic spalling masks to train a DeepLabv3+ model.


doi:10.6342/NTU202403549

**RPL [70]:** RPL is a abbreviation of Residual Pattern Learning, a state-of-the-art approach in road anomaly detection. RPL synthesizes fake patterns by cutting outlier objects from other datasets, such as COCO [77], and pasting them into inlier images. Additionally, it also implements contrastive learning in inlier and outlier images.

**Supervised:** Unlike other baselines, the upper bound model is trained using supervised learning on a real-world spalling image dataset consisting of 304 labeled images sourced from Google Street View in Taiwan. This dataset construction process, involving manual collection and labeling, is the same as our validation and test dataset described in section 2.2.3. This construction process took about four months, highlighting the high cost of supervised learning. We define the anomaly score map $y \in \mathbb{R}^{H*W}$ for the supervised method as follows:

$$y = \text{Sigmoid}(L)^{+} \tag{3.27}$$

where $L \in \mathbb{R}^{H*W*2}$ is the logits output of the binary segmentation model, with two channels: one for the positive class and one for the negative class. We select the positive channel after apply the sigmoid function.


doi:10.6342/NTU202403549

# Chapter 4 Results and Discussions

## 4.1 Supervised Learning

### 4.1.1 Segmentation

Our dataset exhibits a relatively limited proportion of the foreground pixels (i.e., tile spalling) across the entire image, leading to an imbalanced dataset, as illustrated in Figure 4.1a. This imbalance causes the intersection over union (IoU) for the background class to approach values close to 1. Consequently, our evaluation focuses exclusively on the IoU score of the foreground class for all models considered in this study. The learning curves for the validation set, depicted in Figure 4.1b and 4.1c, show steady convergence in the final epochs, indicating that the proposed MBF-UNet model has achieved a stable and consistent performance level.

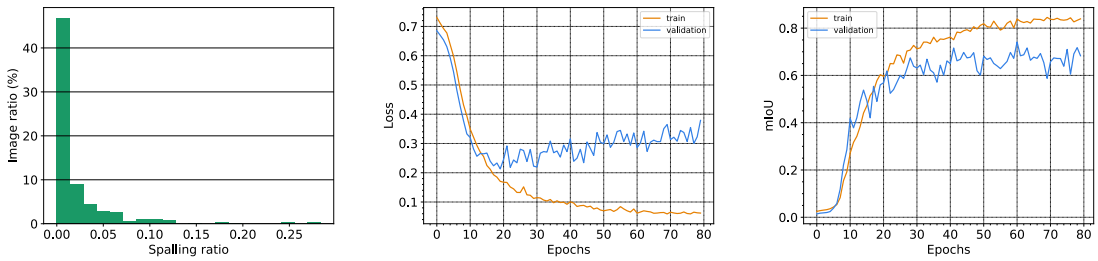
Figure 4.2 offers a visual comparison of segmentation results, including the original images, the corresponding ground-truth spalling masks, and predictions from various baseline models: U-Net [12], MA-Net [15], U-Net++ [57], DeepLabV3+ [13], and the proposed MBF-UNet. In cases involving particularly large spalling, such as images (1) and (2), the baseline models struggle to accurately delineate the spalling areas, even failing to segment anything in image (2). Conversely, the MBF-UNet successfully captures


doi:10.6342/NTU202403549

the spalling regions in both scenarios. For images (3) and (4), where the tile spalling sizes vary significantly with larger spalling areas in the upper left and upper right corners, respectively, U-Net, MA-Net, and U-Net++ perform relatively well in segmenting small spallings. However, their performance diminishes when dealing with larger spallings. DeepLabV3+, on the other hand, excels in segmenting large spallings, achieving near-perfect segmentation in image (4) due to its multi-scale architecture. Nonetheless, DeepLabV3+ over-predicts small spallings in image (3). The proposed MBF-UNet demonstrates superior performance across both large and small spalling areas. In cases (5) and (6), which feature multiple spallings, models such as U-Net, U-Net++, and MBF-UNet exhibit strong performance due to their skip connections, which preserve essential features from previous layers. Particularly, nested and dense skip connections enhance the performance of U-Net++ and MBF-UNet compared to U-Net. In case (6), the multi-scale mechanism proves advantageous as DeepLabV3+ and MBF-UNet outperform others, demonstrating their effectiveness in addressing multiple spallings.

To rigorously assess the proposed model's superiority, we conduct a comparative analysis with state-of-the-art models using the six evaluation metrics outlined in Section 3.1.10. The outcomes of seven repeated experiments are summarized in Table 4.1. Our MBF-UNet achieves the highest mean accuracy score and the lowest standard deviation, as shown in the first row. However, there is a small performance gap compared to other baselines, indicating that our dataset may be imbalanced. The precision and recall scores, presented in the second and third rows, reflect the significance of false positives and false negatives, respectively. In our study, recall is particularly important. Although the recall score of our model is slightly lower than U-Net++ by 0.1%, it has a lower standard deviation. For precision and the integrated F1 score, our model demonstrates the

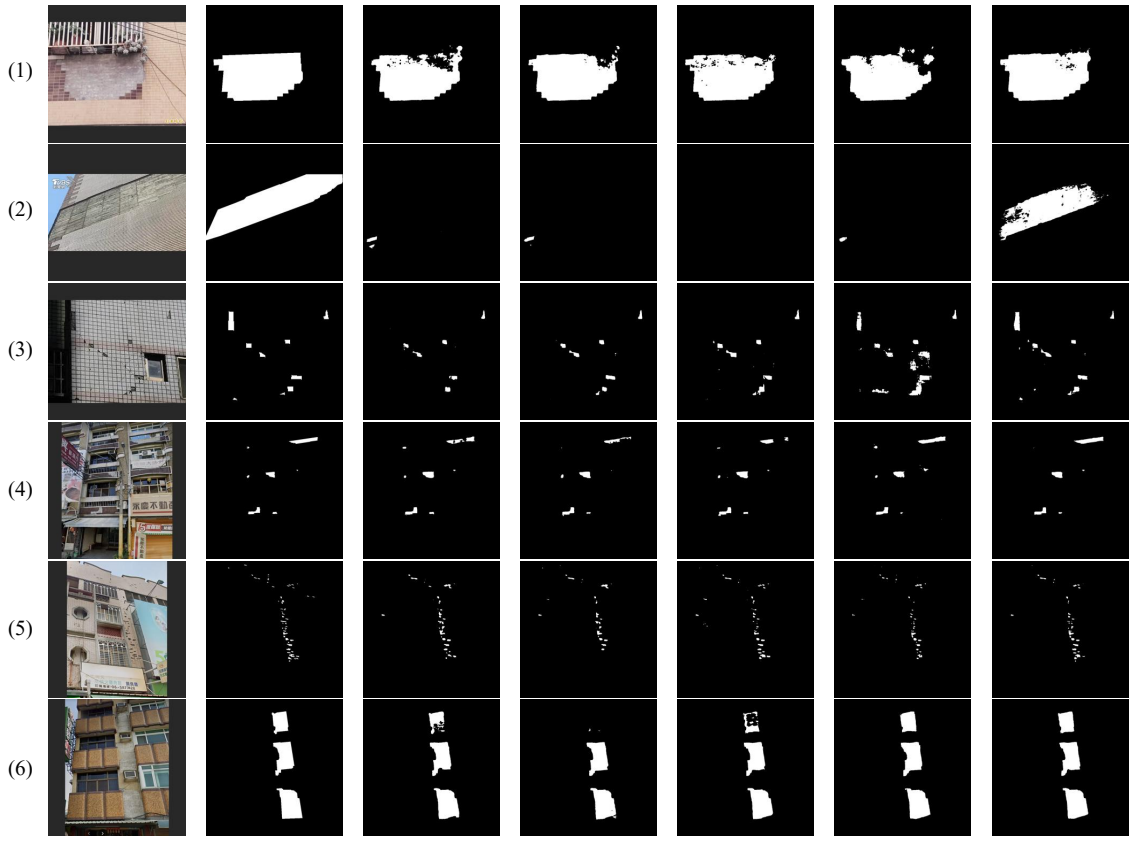

doi:10.6342/NTU202403549




(a) (b) (c)

Figure 4.1: (a) The histogram of proportion of the foreground pixels (b) Loss and (c) mIoU curves of the proposed model during training for both training and validation samples.

best performance in terms of both mean and standard deviation. Furthermore, the highest mIoU score and the lowest HD95 score highlight that our MBF-UNet outperforms other baselines in both model performance and stability.




(a) (b) (c) (d) (e) (f) (g)

Figure 4.2: The spalling segmentation results, including (a) the original images, (b) the corresponding ground-truth masks, and predictions generated by different models, namely (c) U-Net, (d) MA-Net, (e) U-Net++, (f) DeepLabV3+, and (g) the proposed MBF-UNet.



doi:10.6342/NTU202403549

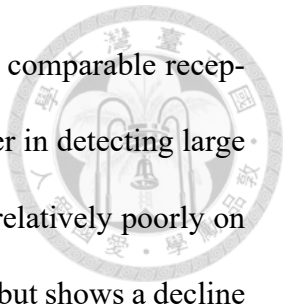
Table 4.1: The model performance comparison between the propose MBF-UNet and other state-of-the-art models. The best scores are in **boldface**. (STD.: standard deviation)

| Metric | | Method | | | | |
|---|---|---|---|---|---|---|
| | | U-Net [12] | U-Net++ [57] | MA-Net [15] | DeepLabV3+ [13] | **MBF-UNet** |
| Accuracy | Mean | 99.3 | 99.3 | 99.3 | 99.3 | **99.5** |
| | STD. | 0.3 | 0.3 | 0.4 | 0.3 | **0.2** |
| Precision | Mean | 85.1 | 83.9 | 84.3 | 87.6 | **90.7** |
| | STD. | 10.8 | 11.8 | 10 | 9.8 | **6.3** |
| Recall | Mean | 89.2 | **90.5** | 89.1 | 84.7 | 90.4 |
| | STD. | 2.2 | 2.5 | 2.2 | 3.7 | **2.1** |
| F1 | Mean | 86.7 | 86.5 | 86.3 | 85.8 | **90.4** |
| | STD. | 6.1 | 6.2 | 5.0 | 5.7 | **3.2** |
| mIoU | Mean | 77.1 | 76.7 | 76.2 | 75.6 | **82.7** |
| | STD. | 9.1 | 9.3 | 7.6 | 8.5 | **5.2** |
| HD95 | Mean | 54.9 | 49.9 | 47.7 | 51.9 | **35.9** |
| | STD. | 14.3 | 16.2 | 10.7 | 15.6 | **9.9** |

### 4.1.2 Intermediate Layer Outputs

Additionally, we analyze the intermediate layer outputs of U-Net++ and the proposed MBF-UNet to evaluate the impact of the multi-branch structure on model performance. Figure 4.3 includes the original images, the corresponding ground-truth masks, the intermediate layer output of U-Net++, the predictions generated by U-Net++, the intermediate layer outputs of MBF-UNet for the small, middle, and large branches, and the final predictions by MBF-UNet. The upper row represents an image with small, scattered spallings, while the lower row shows a case with large spallings. By converting the intermediate outputs to heatmaps, we can observe the focus of each model. The intermediate layer output of U-Net++ demonstrates its ability to concentrate on small spallings. However, U-Net++ struggles to identify large spallings, depicted in pure white in the lower image. This limitation arises from U-Net++'s restricted receptive field, which prevents it from acquiring sufficient contextual information to accurately detect large spallings, resulting in inferior final predictions. In contrast, the intermediate layer output of MBF-UNet for


doi:10.6342/NTU202403549



the small branch shows similar performance to U-Net++ due to their comparable receptive fields. However, the middle branch of MBF-UNet performs better in detecting large spallings, effectively identifying white spallings, though it performs relatively poorly on small spallings. The large branch excels in identifying large spallings but shows a decline in performance for small spallings. By consolidating information from these multi-level branches, the final prediction produced by MBF-UNet surpasses that of U-Net++. The multi-branch architecture of MBF-UNet proves effective in enhancing model performance by allowing each branch to specialize in capturing features of different sizes. This specialization enables MBF-UNet to achieve superior segmentation results, particularly in scenarios involving spallings of varying sizes.

### 4.1.3 Comparisons of Baseline References in Addressing Limited Data

The MBF decoder is designed to extract rich features by utilizing branches with varying receptive fields to capture details at multiple scales, thereby enhancing feature diversity. This approach improves model performance, especially when working with a limited dataset, by enriching feature extraction and mitigating the overfitting issues caused by low data variability. We conducted an ablation study to evaluate several techniques intended to enhance training with a limited dataset and to analyze their effects, including data augmentation and transfer learning. These experiments were repeated seven times across four settings, as detailed in Table 4.2, with results summarized in Table 4.3. Models without the MBF decoder used a standard U-Net architecture for simplicity and comparison. Thus, the first row represents a U-Net model trained from scratch without any data augmentation. Initially, we assessed the impact of transfer learning by using pre-trained weights from ImageNet. This approach significantly improved model performance across all six met-


doi:10.6342/NTU202403549

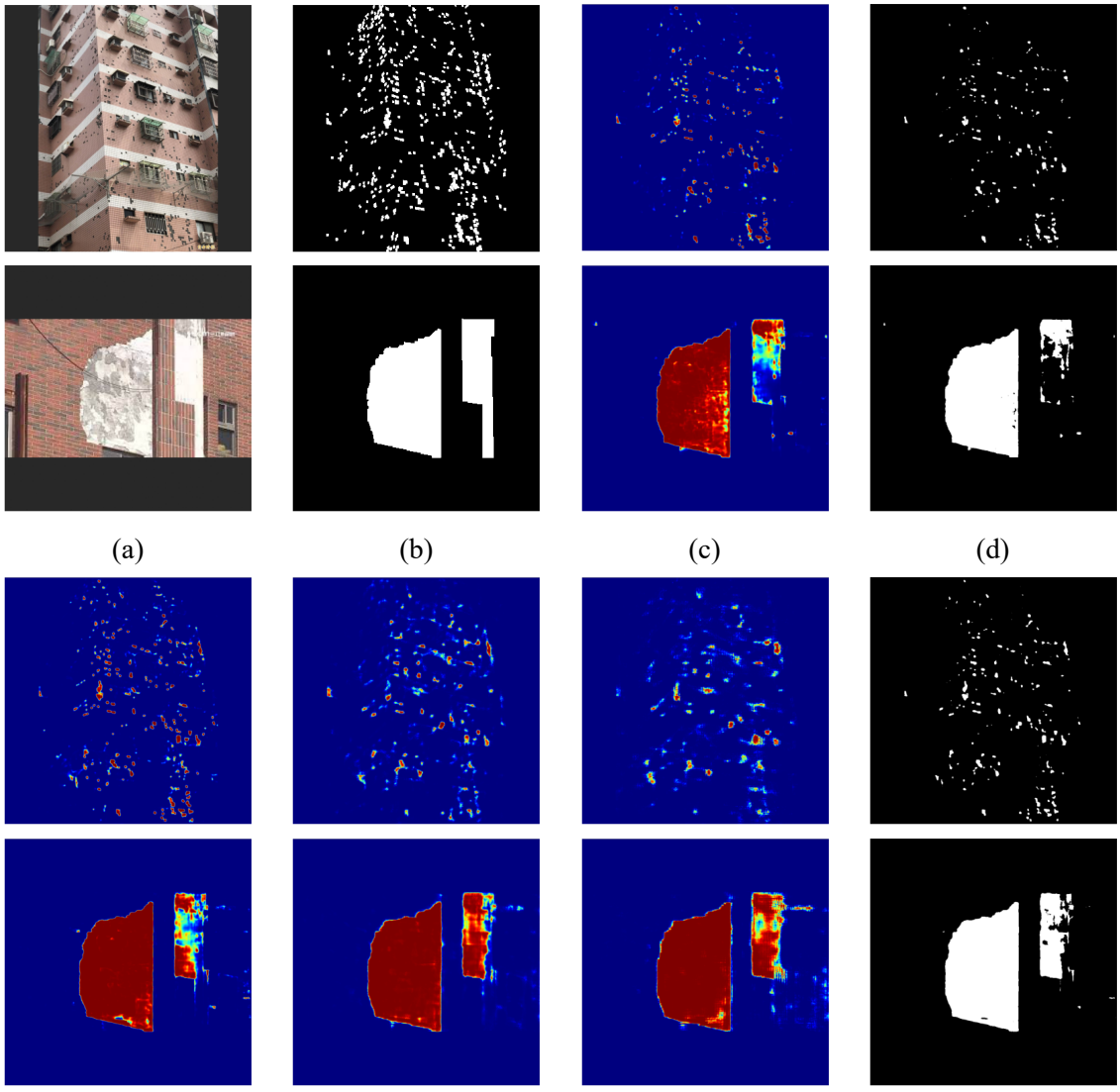



Figure 4.3: Results of network interpretation are presented: (a) The original images, (b) the corresponding ground-truth masks, (c) the intermediate layer output of U-Net++, and (d) the predictions generated by U-Net++. Moreover, we showcase the intermediate layer outputs of (e) the small branch, (f) the middle branch, (g) the large branch, and (h) the final prediction from the proposed MBF-UNet.



doi:10.6342/NTU202403549

rics. Next, we incorporated data augmentation techniques, which led to a modest increase in precision but also caused a decline in the other five metrics. Finally, we integrated our MBF decoder into the model architecture. The MBF-UNet demonstrated the best performance, achieving top results in five out of six metrics, excluding recall. The low standard deviation indicates the robustness and consistency of the proposed approach across multiple runs.

Table 4.2: The ablation settings compare with the baseline references in addressing limited data.

| Setting | Approach | | |
|---|---|---|---|
| | Transfer Learning | Data Augmentation | MBF Decoder |
| 1 | - | - | - |
| 2 | ✓ | - | - |
| 3 | ✓ | ✓ | - |
| 4 | ✓ | ✓ | ✓ |

Table 4.3: The ablation study compares with the baseline references in addressing limited data. The best scores are in **boldface**. (STD.: standard deviation)

| Metric | | Setting | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| Accuracy | Mean | 96.7 | 99.4 | 99.3 | **99.5** |
| | STD. | 1.6 | 0.3 | 0.3 | **0.2** |
| Precision | Mean | 53.4 | 84.9 | 85.1 | **90.7** |
| | STD. | 18.9 | 11.9 | 10.8 | **6.3** |
| Recall | Mean | 42.0 | **93.0** | 89.2 | 90.4 |
| | STD. | 12.0 | 2.4 | 2.2 | **2.1** |
| F1 | Mean | 44.6 | 88.4 | 86.7 | **90.4** |
| | STD. | 10.0 | 7.5 | 6.1 | **3.2** |
| mIoU | Mean | 29.2 | 79.9 | 77.1 | **82.7** |
| | STD. | 8.1 | 11.0 | 9.1 | **5.2** |
| HD95 | Mean | 271.8 | 39.0 | 54.9 | **35.9** |
| | STD. | 21.5 | 13.2 | 14.3 | **9.9** |

### 4.1.4 Effects of Optimization Scheme

Table 4.4 presents a comparison between the direct and individual optimization schemes introduced in Section 3.1.5. The direct optimization approach outperforms in four out


doi:10.6342/NTU202403549

of six metrics, excluding precision, and ties in accuracy. Additionally, the lower standard deviations indicate the robustness of the direct optimization approach. This outcome suggests that we do not need to supervise each branch individually for every sample; instead, we can focus on the final outputs, allowing the branches to attend to the appropriate regions. Both strategies outperform the baseline U-Net++ model, demonstrating the effective utilization of diverse receptive fields in enhancing segmentation accuracy. It is important to note that the performance of each optimization method can vary depending on factors such as the specific dilation rates used. Overall, these findings underscore the effectiveness of the direct optimization strategy in improving segmentation performance and leveraging multi-scale information effectively for tile spalling identification tasks.

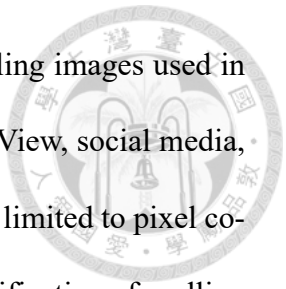
Table 4.4: The comparison between the individual and direct optimization scheme. The best results are in **boldface**.

| Metric | | Method | |
|---|---|---|---|
| | | Individual | Direct |
| Accuracy | Mean | 99.5 | 99.5 |
| | STD. | 0.2 | 0.2 |
| Precision | Mean | **90.8** | 90.7 |
| | STD. | 6.7 | **6.3** |
| Recall | Mean | 89.8 | **90.4** |
| | STD. | 2.2 | **2.1** |
| F1 | Mean | 90.2 | **90.4** |
| | STD. | 3.2 | 3.2 |
| mIoU | Mean | 82.2 | **82.7** |
| | STD. | 5.2 | 5.2 |
| HD95 | Mean | 40.7 | **35.9** |
| | STD. | 17.4 | **9.9** |

### 4.1.5 Limitations

While the proposed MBF-UNet demonstrates effectiveness in segmenting tile spallings of varying sizes from building images, there are several limitations that need addressing before its integration into real-world applications. Currently, our focus lies in developing


doi:10.6342/NTU202403549



novel network architectures for semantic segmentation. The tile spalling images used in this study were collected from diverse sources such as Google Street View, social media, and mobile phones. As a result, the segmentation results are currently limited to pixel coordinates rather than real-world dimensions. To enable practical quantification of spalling areas, it will be necessary to establish a scaling factor that relates pixel coordinates to real-world lengths, possibly through the use of a camera projection matrix or similar techniques. Future studies will focus on resolving these issues to ensure robust quantification of spalling areas in real-world applications.

## 4.2 Unsupervised Learning

### 4.2.1 Comparison of Spalling-Synthetic Approaches

This section we compare the model performance across different spalling-synthetic approaches to evaluate the impact of the proposed Spalling Craft method, and find a proper spalling-synthetic setting for the source dataset and the target dataset. We design several baseline approaches, as depicted in Figure 4.4, which differ in the content of the spallings while maintaining the same spalling localization method. Constant method in Figure 4.4a represents that the spalling regions are filled with a random gray value. Perlin method in Figure 4.4b represents that the spalling regions are filled with non-processed Perlin noise. Our proposed approach for generating realistic spalling patterns is shown in Figure 4.4c. These spalling-synthetic approaches are integrated into our training framework, which requires two synthetic modules, one for the source dataset and one for the target dataset. We design four settings including our approach. The evaluation results for these settings are presented in Table 4.5. Observations from setting 1 and setting 2 indicating that the Perlin


doi:10.6342/NTU202403549

(a) Constant (b) Perlin (c) Spalling Craft

Figure 4.4: The designed spalling-synthetic approaches, (a) Constant method fills the spalling regions by assigning a random gray value. (b) Perlin method fills them using unprocessed Perlin noise. (c) Spalling Craft method is our approach, which generates realistic spalling patterns.

noise method generates more realistic spalling patterns compared to the Constant method, resulting in better model performance. If we employ different synthetic approaches for the source and target datasets such as setting 3 that Constant for source and Perlin for target. It will enrich variation and help prevent model overfitting. Our approach, which uses Perlin for the source dataset and Spalling Craft for the target dataset, significantly outperforms all other settings by 19.8% to 37.1% in AP score. The reason we only implement Spalling Craft in the target dataset is that it is specifically designed for tile exteriors, whereas most of the building images in the source dataset do not feature tile exteriors. Consequently, we apply the Perlin method in the source dataset.

Table 4.5: The comparison of different spalling-synthetic approach settings includes four distinct configurations. The best scores are in **boldface**.

| Setting | | | Metrics | | |
|---|---|---|---|---|---|
| Number | $Syn^S$ | $Syn^T$ | AUC(%)↑ | AP(%)↑ | FPR95(%)↓ |
| 1 | Const. | Const. | 84.3 | 15.3 | 54.2 |
| 2 | Perlin | Perlin | 87.7 | 29.0 | 54.0 |
| 3 | Const. | Perlin | 89.8 | 32.6 | 46.6 |
| 4 | Perlin | S-Craft | **94.7** | **52.4** | **29.5** |


doi:10.6342/NTU202403549

### 4.2.2 Contrastive Learning

The contrastive learning module can pull the embeddings of the same class (inliers or outliers) together and separate them from embeddings of different classes. To analyze the effectiveness of contrastive learning, we project the 304-dimensional embeddings onto a 2D plot using t-SNE. [78]. The projection result are illustrated in Figure 4.5. We train two models: one trained with $\mathcal{L}_{CL}$ in Eq. 3.21 for loss estimation, and the other without it. Figure 4.5a the performance of the non-contrastive model on the training datasets, including the source and target datasets. The crosses and circles represent embeddings from the source and target datasets, respectively, with blue and red indicating the embedding classes. We observe that embeddings of different classes (blue and red) are mixed, while the contrastive model in Figure 4.5b can separate them more clearly. These distributions are also reflected in the test set in Figures 4.5c and Figure 4.5d. Although the contrastive model cannot separate the test embeddings as clearly as the training embeddings due to the domain gap in Figure 4.5d, it still performs better than the non-contrastive model in Figure 4.5c. The estimation results are shown in Table 4.6. The contrastive model achieves better performance than the non-contrastive model by 4.5% in AUC score, 29.1% in AP score, and 9.3% in FPR95 score. This improvement is attributed to the increased distance between inliers and outliers, facilitating better classification.

### 4.2.3 Anomaly Segmentation

Figure 4.6 provides a visual comparison of anomaly score results, showcasing the original images, the corresponding ground-truth spalling masks, and predictions generated by different baseline approaches as described in section 3.2.8. The approaches compared


doi:10.6342/NTU202403549

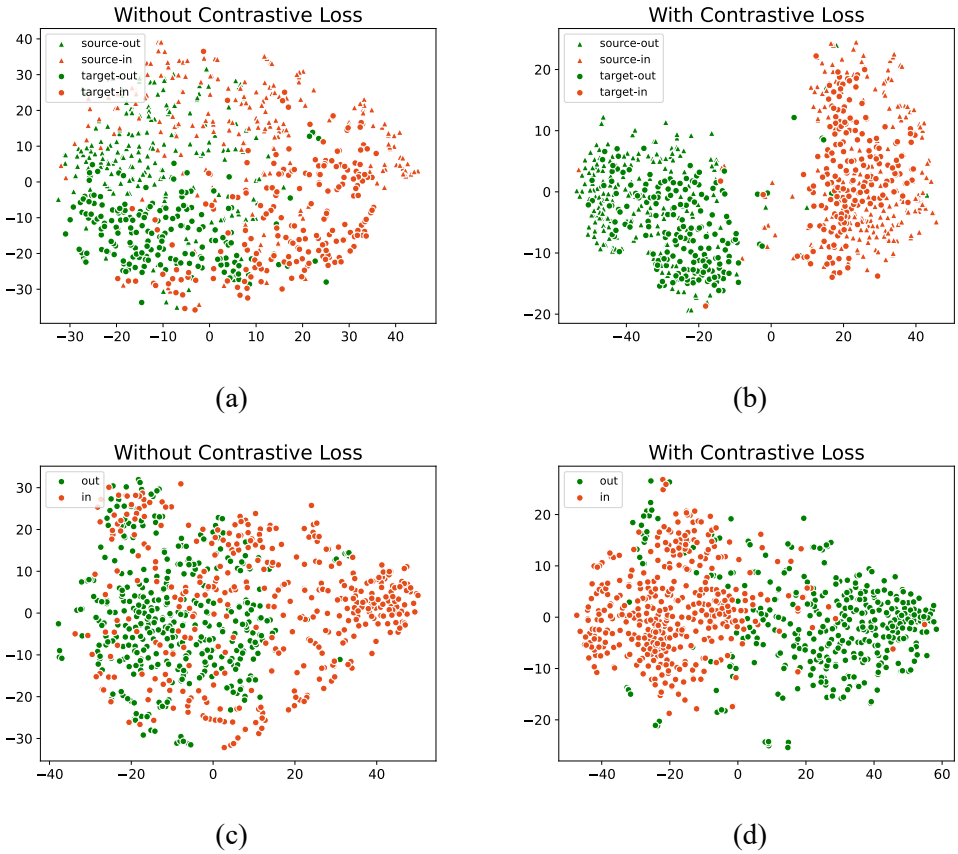
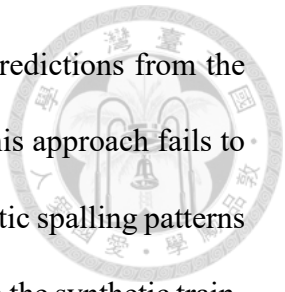



Figure 4.5: The t-SNE projection results of the embeddings generated by the contrastive model and non-contrastive model. (a) The non-contrastive model generates mixed embeddings in the training dataset. (b) The contrastive model separates the training embeddings from different classes (inliers and outliers). (c) The non-contrastive model generates mixed embeddings in the test dataset. (d) The contrastive model separates the test embeddings from different classes.



doi:10.6342/NTU202403549



are synthetic supervised learning, RPL, and the proposed method. Predictions from the synthetic supervised learning model are inferior across all images. This approach fails to identify any spalling in real-world scenarios, indicating that the synthetic spalling patterns used for training are not sufficiently realistic. The domain gap between the synthetic training data and real-world data leads to excellent performance on the training dataset but poor performance on real-world data. The RPL approach performs better, especially in images where the spalling is distinct from the rest of the image, such as image (3). However, the gap between spalling and non-spalling regions is not pronounced, making it difficult to distinguish between them. In more complex scenes, such as those with trees in images (1) and (4), the model assigns relatively high uncertainty scores to these areas, leading to errors. This suggests that using outlier objects from the COCO dataset as spalling patterns does not work well for our task, as these objects are far from typical building appearances. Consequently, the model tends to regard regions that are most different as anomaly, such as trees, leading to incorrect predictions. Our proposed model clearly outperforms the baselines. It can accurately identify the spalling regions compared to other spalling-free areas. The trees in images (1) and (4) have relatively low anomaly scores, and small spallings are also recognized effectively. Both small and large spallings in images (2) and (3) are identified well. Spallings that are similar in color to the tile in image (5), as well as those on high-level buildings in image (6), achieve relatively high anomaly scores. This demonstrates the robustness and accuracy of our model in various real-world scenarios, highlighting its effectiveness in practical applications.

To rigorously evaluate the proposed approach, we conduct a comprehensive analysis with those baselines in the test dataset and estimate the model performance by Average Precision (AP), Area Under the Curve (AUC), and False Positive Rate at 95% True


doi:10.6342/NTU202403549

Positive Rate (FPR95). The outcomes of our experiments are summarized in Table 4.6. Besides, we also conduct a traditional supervised learning method as an upper bound in the last row. Different from the visualization results, the synthetic supervised approach has better performance than RPL in those three metrics. It achieves 80.3% in AUC score, 11.9% in AP score, and 55.3% in FPR95 score, better than the RPL approach by 4.0%, 6.1%, and 5.9%. respectively. Our approach is much more robust than those two baselines in all three metrics. It yields 94.7% in AUC score, 52.4% in AP score, and 29.5% in FPR95 score, better than the synthetic supervised approach by about 14.4%, 40.5%, and 25.8%. Compared to traditional supervised learning, our approach can achieve 59.0% performance in AP score and 91.6% performance in AUC score (50% is the lower bound) while we have no spalling image and label during training.

We apply a threshold to the anomaly maps evaluated by our approach in order to generate a segmentation map. The threshold is determined by the best intersection over union (IoU) score in the validation dataset, as depicted in Figure 4.8. Observations reveal that the predictions tend to be larger than the ground truth and contain some noise. This is likely due to the significant variation in the appearance of houses, leading to parts of the image not being represented in the target dataset. Consequently, the model exhibits increased uncertainty about these regions.

### 4.2.4 Label Efficiency

Compared to traditional supervised learning, our approach only uses normal building images during training, significantly reducing the dataset construction time, both in terms of image searching and spalling labeling. Therefore, we analyze the label efficiency of the two approaches by comparing the cost of dataset construction, as depicted in Table 4.7.


doi:10.6342/NTU202403549

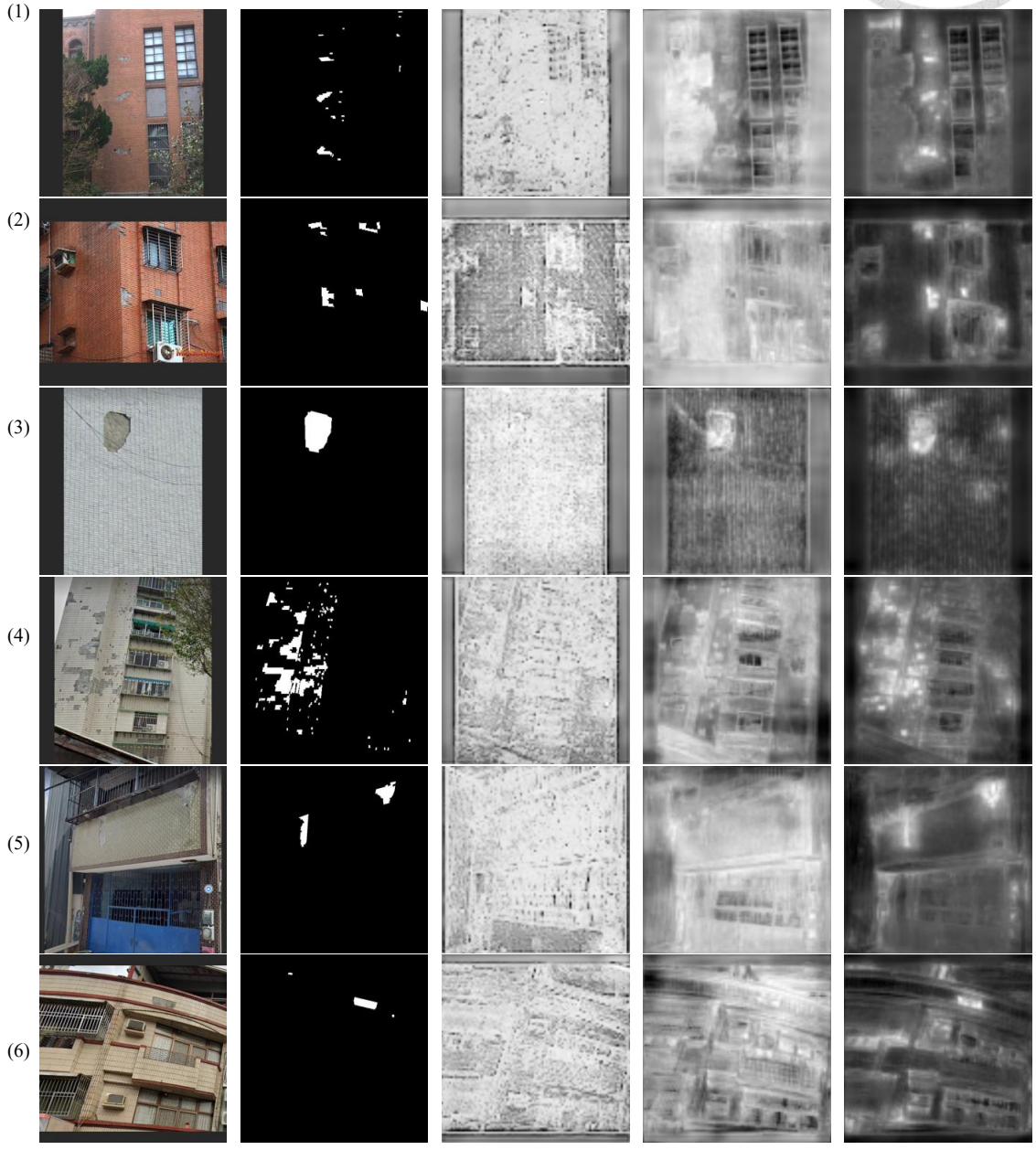
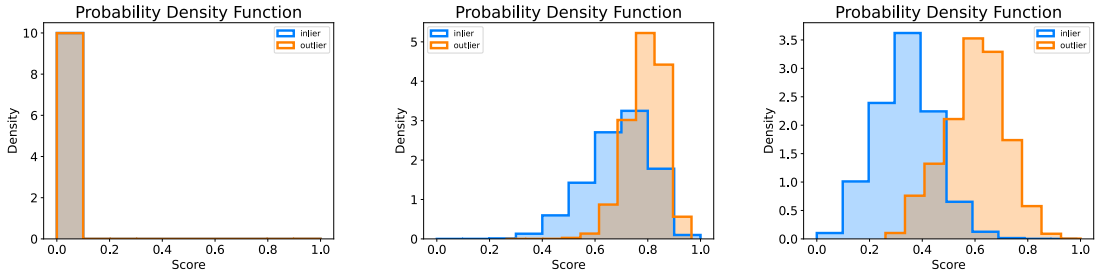



Figure 4.6: The spalling anomaly score results, including (a) the original images, (b) the corresponding ground-truth masks, and predictions generated by different approach, namely (c) synthetic supervised learning, (d) RPL, (e) our approach.



doi:10.6342/NTU202403549




(a) Synthetic supervised (b) RPL (c) Our approach

Figure 4.7: The inlier and outlier probability density function (PDF) of these approaches. (a) The synthetic supervised approach showcases the same distribution for inlier and outlier. (b) The RPL separates the inlier and outlier distributions slightly, but there is still considerable overlap. (c) Our approach wildly separates the inlier and outlier distributions.

(a) Image (b) Mask (c) Segmentation

Figure 4.8: The segmentation results, including (a) the original images, (b) the corresponding ground-truth masks, and (c) the segmentation.



doi:10.6342/NTU202403549

Table 4.6: The model performance comparison between our approach and other baselines. The best scores are in **boldface**, $*$ denote the model without contrastive learning and † denote the upper bound of this experiment, which uses labeled real-world dataset implementing supervised learning.

| Method | Metrics | | |
|---|---|---|---|
| | AUC(%)↑ | AP(%)↑ | FPR95(%)↓ |
| Synthetic supervised | 80.3 | 11.9 | 55.3 |
| RPL [70] | 76.3 | 5.8 | 61.2 |
| *Ours | 90.2 | 23.3 | 38.8 |
| **Ours (proposed)** | **94.7** | **52.4** | **29.5** |
| †Supervised | 98.8 | 88.8 | 3.7 |

The training dataset for the supervised learning model contains 304 spalling images from various sources such as Google Street View, social media, and photos taken in Taiwan over four months. We manually labeled the spalling areas in these images at the pixel level, with the total construction time amounting to about 200 hours. In contrast, the training dataset for our approach contains 2540 normal building images scraped from Google Street View using building addresses in Taiwan. The process takes one hour, including image selection to remove some invalid samples, which is only 0.5% of the time required for the supervised learning model. If we convert the time to money using the minimum hourly wage in Taiwan, which is 183 NTD per hour, the cost is 36600 NTD for the supervised learning model compared to 183 NTD for our approach, saving about 36417 NTD. This saving scales with the quantity of images in the dataset. For instance, if the labeled dataset is scaled to the size of a typical segmentation model dataset such as the Cityscape dataset, which contains 2975 images, our approach can save approximately 356383 NTD.

Table 4.7: Comparison between traditional supervised learning and our approach

| Method | Quantity | Time(hr) | Cost (183$/hr) |
|---|---|---|---|
| Supervised learning | 304 | 200 | 36600 |
| Our approach | 2540 | 1 | 183 (0.5%) |


doi:10.6342/NTU202403549

### 4.2.5 Limitations

While the proposed unsupervised learning approach demonstrates significant label efficiency and decent performances, there are still limitations that require further investigation. Firstly, the anomaly map provides an uncertainty score, which means a threshold is necessary to distinguish outlier pixels from inlier pixels using a small number of labeled images. Additionally, to ensure comprehensive capture of the entire structure, training images in the Façade dataset are taken from a distance, directly facing the building façade. This constraint limits the required camera shooting angle to a perpendicular one. If the spalling is located on a tall building, other data collection platforms like drones may ensure an appropriate shooting angle rather than using Google Street View alone. Lastly, as the proposed approach aims to identify the anomaly based on the source dataset that contains samples without anomaly, it is critical to have a source database that is representative enough for the inlier samples if the proposed approach is adopted to other applications.

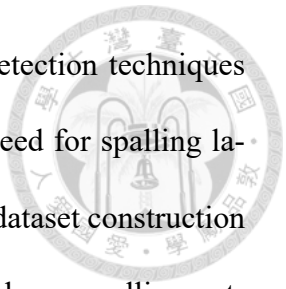

doi:10.6342/NTU202403549

# Chapter 5 Conclusion

## 5.1 Conclusion

In this study, we facilitate the tile spalling segmentation work for autonomous building façades detection. Our work dedicates in data efficiency including two approaches which are supervised and unsupervised, respectively. The supervised approach incorporates a multi-branch architecture with different receptive fields to capture features of varying sizes. This mechanism enhances the discriminative ability of the model, particularly when dealing with objects of greatly varying sizes. We manually labeled the tile spalling dataset in pixel-level, and the dataset consists of 364 images collected from Google street view, social media and mobile phones for four months. This dataset can serve as a valuable resource for future endeavors focusing on tile spalling detection and segmentation. According to seven repeated trials, our proposed MBF-UNet achieves the best performance in five out of six metrics, both in terms of mean and standard deviation. The recall is 0.1% lower in mean compared to U-Net++, but the MBF-UNet still has a lower standard deviation. Additionally, the designed strategies for limited data improve model performance across all six metrics and contribute to more stable training. Finally, by optimizing the predictions directly, our model demonstrates better and more stable performance in four out of six metrics.


doi:10.6342/NTU202403549



The new unsupervised approach inspired from road anomaly detection techniques employing uncertainty estimation. The training process avoids the need for spalling labels or images during training, thereby significantly streamlining the dataset construction process, including spalling image collection and labeling. We develop a spalling pattern synthesize approach namely Spalling Craft to produce realistic spalling patterns onto spalling-free (i.e., anomaly-free) images to facilitate outlier exposure. In addition, a contrastive learning module is integrated into the training framework to effectively group embeddings of the same class while separating those of different classes. In other words, the distributions of inlier and outlier features are more distinguishable after the incorporation of contrastive loss. Results indicate that utilizing Perlin noise in the source dataset and the proposed Spalling Craft in the target dataset yields superior performances, outperforming other settings by 4.9% to 10.4% in AUC score, 19.8% to 37.1% in AP score, and 17.1% to 24.7% in FPR95 score. This finding suggests that varying spalling content effectively mitigates domain gaps between training and test datasets. Furthermore, our approach achieves 94.7% in AUC score, 52.4% in AP score, and 29.5% in FPR95 score, outperforming the baseline method by approximately 18.4%, 46.6%, and 31.7%, respectively. Compared to supervised learning, although the AUC and AP scores drop 4.1% and 36.4%, respectively, our unsupervised approach requires only 0.5% of time for dataset construction, leading to cost savings estimated at 36600 NTD. The benefit in budgeting will scale rapidly with the increasing size of dataset.

## 5.2 Future Work

Our future work will prioritize quantifying the exterior condition of tiled surfaces. We plan to gather spalling images that include detailed camera settings such as shooting an-


doi:10.6342/NTU202403549

gle, camera position, and depth information. These images will help translate pixel-level segmentation into real-world scenarios, enabling us to design an assessment framework to evaluate exterior conditions. This framework will facilitate the development of an autonomous system for façade quantification. This framework will integrate our proposed model to establish a standardized pattern for monitoring building conditions regularly.

Additionally, we aim to investigate optimal camera settings for our segmentation model, including ideal shooting angles and distances. By replicating scenes from our dataset, we can enhance the model performance. To further bolster the effectiveness of our unsupervised model, we intend to expand our dataset by surveying additional building image datasets. This expansion will make our dataset more representative, thereby improving the model ability to identify anomalies in typical building structures.


doi:10.6342/NTU202403549

# References


[1] Clive Briffett. The performance of external wall systems in tropical climates. Energy and Buildings, 16(3-4):917–924, January 1991.

[2] Ting-Jui Lu. A research study on the solution to spalling of exterior wall tiles of high-rise apartment buildings. Technical report, Architecture and Building Research Institute, Ministry of the Interior, ROC (Taiwan), 2011.

[3] Yao-Tsu Chang. The research of building external wall tiles deterioration diagnosis –case study of national taiwan university school building. Master's thesis, National Taiwan University, Taiwan, 2013.

[4] Sy-Jye Guo. The research of building siding health check and renovation assessment system. Technical report, Architecture and Building Research Institute, Ministry of the Interior, ROC (Taiwan), 2011.

[5] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, June 2014.

[6] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detec-




doi:10.6342/NTU202403549

tion and semantic segmentation in the wild. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 891–898, 2014.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016.

[8] Yann. Lecun, Léon. Bottou, Yoshua. Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[9] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks, 2014.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.



doi:10.6342/NTU202403549

[14] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation, 2019.

[15] Tongle Fan, Guanglei Wang, Yan Li, and Hongrui Wang. Ma-net: A multi-scale attention network for liver and tumor segmentation. IEEE Access, 8:179656–179665, 2020.

[16] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation, 2019.

[17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[18] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[20] Stephen L. H. Lau, Edwin K. P. Chong, Xu Yang, and Xin Wang. Automated pavement crack segmentation using u-net-based convolutional neural network. IEEE Access, 8:114892–114899, 2020.

[21] Chengjia Han, Tao Ma, Ju Huyan, Xiaoming Huang, and Yanning Zhang. Crackw-net: A novel pavement crack image segmentation convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 23(11):22135–22144, 2022.

[22] Dongho Kang, Sukhpreet S. Benipal, Dharshan L. Gopal, and Young-Jin Cha. Hybrid pixel-level concrete crack segmentation and quantification across complex



doi:10.6342/NTU202403549

backgrounds using deep learning. Automation in Construction, 118:103291, October 2020.

[23] Yuki Kondo and Norimichi Ukita. Crack segmentation for low-resolution images using joint learning with super- resolution. In 2021 17th International Conference on Machine Vision and Applications (MVA), pages 1–6, 2021.

[24] Ting-Yan Wu, Rih-Teng Wu, Ping-Hsiung Wang, Tzu-Kang Lin, and Kuo-Chun Chang. Development of a high-fidelity failure prediction system for reinforced concrete bridge columns using generative adversarial networks. Engineering Structures, 286:116130, July 2023.

[25] Rih-Teng Wu, Ankush Singla, Mohammad R. Jahanshahi, Elisa Bertino, Bong Jun Ko, and Dinesh Verma. Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures. Computer-Aided Civil and Infrastructure Engineering, 34(9):774–789, May 2019.

[26] Jacob J. Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3d mapping, defect detection, analysis, and reporting. Journal of Computing in Civil Engineering, 35(2), March 2021.

[27] Atiqur Rahman, Zheng Yi Wu, and Rony Kalfarisi. Semantic deep learning integrated with rgb feature-based rule optimization for facility surface corrosion detection and evaluation. Journal of Computing in Civil Engineering, 35(6), November 2021.

[28] Yu Xie, Fangrui Zhu, and Yanwei Fu. Main-secondary network for defect segmen-



doi:10.6342/NTU202403549

tation of textured surface images. In 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 3520–3529, 2020.

[29] Bo Chen, Hua Zhang, Guijin Wang, Jianwen Huo, Yonglong Li, and Linjing Li. Automatic concrete infrastructure crack semantic segmentation using deep learning. Automation in Construction, 152:104950, August 2023.

[30] Chao Xiang, Jingjing Guo, Ran Cao, and Lu Deng. A crack-segmentation algorithm fusing transformers and convolutional neural networks for complex detection scenarios. Automation in Construction, 152:104894, August 2023.

[31] Wenjun Wang and Chao Su. Semi-supervised semantic segmentation network for surface crack detection. Automation in Construction, 128:103786, August 2021.

[32] Zhi-hong Wang, Shao-bo Wang, Li-rong Yan, and Yu Yuan. Road surface state recognition based on semantic segmentation. Journal of Highway and Transportation Research and Development (English Edition), 15(2):88–94, June 2021.

[33] Bin Yu, Xiangcheng Meng, and Qiannan Yu. Automated pixel-wise pavement crack detection by classification-segmentation networks. Journal of Transportation Engineering, Part B: Pavements, 147(2):04021005, June 2021.

[34] Yuhan Jiang, Sisi Han, and Yong Bai. Building and infrastructure defect detection and visualization using drone and deep learning technologies. Journal of Performance of Constructed Facilities, 35(6), December 2021.

[35] Seung-Nam Yu, Jae-Ho Jang, and Chang-Soo Han. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. Automation in Construction, 16(3):255–261, May 2007.



doi:10.6342/NTU202403549

[36] Kai Zhu, Wenjing Cao, Chenhao Ran, and Bohong Gu. A novel automatic crack classification algorithm of 3-d woven composites based on deep-learning u-net model. Engineering Fracture Mechanics, 289:109488, September 2023.

[37] Feng Guo, Jian Liu, Chengshun Lv, and Huayang Yu. A novel transformer-based network with attention mechanism for automatic pavement crack detection. Construction and Building Materials, 391:131852, August 2023.

[38] Carlos Canchila, Shanglian Zhou, and Wei Song. Hyperparameter optimization and importance ranking in deep learning‐based crack segmentation. Journal of Computing in Civil Engineering, 38(2), March 2024.

[39] Sudhir Babu Patel, Pranjal Bisht, and Krishna Kant Pathak. Semantic segmentation of cracks on masonry surfaces using deep-learning techniques. Practice Periodical on Structural Design and Construction, 29(2), May 2024.

[40] Okeke Stephen, Uchenna Joseph Maduh, and Mangal Sain. A machine learning method for detection of surface defects on ceramic tiles using convolutional neural networks. Electronics, 11(1), 2022.

[41] Gerivan Santos Junior, Janderson Ferreira, Cristian Millán-Arias, Ramiro Daniel, Alberto Casado Junior, and Bruno J. T. Fernandes. Ceramic cracks segmentation with deep learning. Applied Sciences, 11(13), 2021.

[42] Ren-Yi Kung, Nai-Hsin Pan, Charles C.N. Wang, and Pin-Chan Lee. Application of deep learning and unmanned aerial vehicle on building maintenance. Advances in Civil Engineering, 2021:1‐12, April 2021.

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.



doi:10.6342/NTU202403549

[44] Minh-Tu Cao. Drone-assisted segmentation of tile peeling on building façades using a deep learning model. Journal of Building Engineering, 80:108063, 2023.

[45] Stamos Katsigiannis, Saleh Seyedzadeh, Andrew Agapiou, and Naeem Ramzan. Deep learning for crack detection on masonry façades using limited data and transfer learning. Journal of Building Engineering, 76:107105, 2023.

[46] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.

[47] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.

[48] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In Proceedings of the 27th ACM International Conference on Multimedia. ACM, October 2019.

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018.

[50] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, June 2010.

[51] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.



doi:10.6342/NTU202403549

[52] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[53] Zaiwang Gu, Jun Cheng, Huazhu Fu, Kang Zhou, Huaying Hao, Yitian Zhao, Tianyang Zhang, Shenghua Gao, and Jiang Liu. CE-net: Context encoder network for 2d medical image segmentation. IEEE Transactions on Medical Imaging, 38(10):2281–2292, oct 2019.

[54] Leonardo Rundo, Changhee Han, Yudai Nagano, Jin Zhang, Ryuichiro Hataya, Carmelo Militello, Andrea Tangherloni, Marco S. Nobile, Claudio Ferretti, Daniela Besozzi, Maria Carla Gilardi, Salvatore Vitabile, Giancarlo Mauri, Hideki Nakayama, and Paolo Cazzaniga. Use-net: incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets, 2019.

[55] Huazhu Fu, Jun Cheng, Yanwu Xu, Damon Wing Kee Wong, Jiang Liu, and Xiaochun Cao. Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE Transactions on Medical Imaging, 37(7):1597–1605, jul 2018.

[56] Li Huang, Cheng Chen, Juntong Yun, Ying Sun, Jinrong Tian, Zhiqiang Hao, Hui Yu, and Hongjie Ma. Multi-scale feature fusion convolutional neural network for indoor small target detection. Frontiers in Neurorobotics, 16, May 2022.

[57] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation, 2018.

[58] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019.



doi:10.6342/NTU202403549

[59] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[60] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning, 2017.

[61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[62] Yulong Chen, Zilong Zhu, Zhijie Lin, and Youmei Zhou. Building surface crack detection using deep learning technology. Buildings, 13(7), 2023.

[63] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, 2018.

[64] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[65] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[66] Zoé Lambert, Caroline Petitjean, Bernard Dubray, and Su Ruan. Segthor: Segmentation of thoracic organs at risk in ct images, 2019.



doi:10.6342/NTU202403549

[67] Davood Karimi and Septimiu E. Salcudean. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks, 2019.

[68] Jue Jiang, Yu-Chi Hu, Chia-Ju Liu, Darragh Halpenny, Matthew D. Hellmann, Joseph O. Deasy, Gig Mageras, and Harini Veeraraghavan. Multiple resolution residually connected feature streams for automatic lung tumor segmentation from CT images. IEEE Transactions on Medical Imaging, 38(1):134–144, January 2019.

[69] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation, 2021.

[70] Yuyuan Liu, Choubo Ding, Yu Tian, Guansong Pang, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation, 2023.

[71] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition, 2022.

[72] Nazir Nayal, Mısra Yavuz, João F. Henriques, and Fatma Güney. Rba: Segmenting unknown regions rejected by all, 2023.

[73] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation, 2020.

[74] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes, 2021.

[75] Yu Tian, Yuyuan Liu, Guansong Pang, Fengbei Liu, Yuanhong Chen, and Gustavo Carneiro. Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes, 2022.



doi:10.6342/NTU202403549

[76] Silvio Galesso, Max Argus, and Thomas Brox. Far away in the deep space: Dense nearest-neighbor-based out-of-distribution detection, 2023.

[77] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[78] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008.



doi:10.6342/NTU202403549