MOUNTAIN-PLAINS CONSORTIUM

MPC 24-559 | G. Jia and W. Chen

VISIBLE & THERMAL IMAGING AND DEEP LEARNING BASED APPROACH FOR AUTOMATED ROBUST DETECTION OF POTHOLES TO PRIORITIZE HIGHWAY MAINTENANCE

MOUNTAIN-PLAINS CONSORTIUM REGION 8

A University Transportation Center sponsored by the U.S. Department of Transportation serving the Mountain-Plains Region. Consortium members:

Colorado State University
North Dakota State University
South Dakota State University
University of Colorado Denver
University of Denver
University of Utah
Utah State University
University of Wyoming

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
MPC-620		

4. Title and Subtitle	5. Report Date
Visible & Thermal Imaging and Deep Learning Based Approach for Automated Robust Detection of Potholes to Prioritize Highway Maintenance	September 2024
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Gaofeng Jia Wei-Hsiang Chen	MPC 24-559

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Colorado State University Department of Civil and Environmental Engineering Fort Collins, CO 80523	11. Contract or Grant No.

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Mountain-Plains Consortium North Dakota State University PO Box 6050, Fargo, ND 58108	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Supported by a grant from the US DOT, University Transportation Centers Program

16. Abstract

Potholes are a significant pavement distress compromising safety and causing costly damage. They result from pavement deterioration due to aging, weather, and traffic overloads, with the Mountain Plains region particularly affected due to freeze/thaw cycles. Timely identification and repair of potholes are critical for effective highway maintenance. This research develops an automated deep learning-based pothole detection and mapping tool using the fusion of visible and thermal images. Visible images alone often fail in poor lighting or adverse weather conditions, whereas thermal images offer robust detection but lack texture details. Integrating both image types enhanced detection accuracy. We created a database of geotagged and labeled trios of visible, thermal, and fused images using a low-cost FLIR ONE thermal camera connected to a smartphone. Three machine-learning algorithms were proposed and compared: Anisotropic Diffusion Fusion (ADF) + Mask R-CNN, RTFNet, and RTFNet with Enhancement Parameters (EPs). The RTFNet method achieved the best F1-score of 93.7% in daytime and 90.9% in nighttime scenarios. A Bright-Dark detector was developed to optimize algorithm selection based on lighting conditions. Detected potholes were mapped using GPS data, and the trained algorithm was packaged into a GUI tool that can be used by highway maintenance teams.

17. Key Word	18. Distribution Statement
algorithms, data collection, detection and identification technologies, machine learning, mapping, potholes (pavements), thermal imagery	Public distribution

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	52	n/a

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Visible & Thermal Imaging and Deep Learning Based Approach for Automated Robust Detection of Potholes to Prioritize Highway Maintenance

Gaofeng Jia

Wei-Hsiang Chen

Department of Civil and Environmental Engineering
Colorado State University
Fort Collins, CO 80523

September 2024

Acknowledgement

The funds for this study were provided by the United States Department of Transportation to the Mountain-Plains Consortium (MPC).

Disclaimer

ABSTRACT

Potholes are a significant pavement distress compromising safety and causing costly damage. They result from pavement deterioration due to aging, weather, and traffic overloads, with the Mountain Plains region particularly affected due to freeze/thaw cycles. Timely identification and repair of potholes are critical for effective highway maintenance. This research develops an automated deep learning-based pothole detection and mapping tool using the fusion of visible and thermal images. Visible images alone often fail in poor lighting or adverse weather conditions, whereas thermal images offer robust detection but lack texture details. Integrating both image types enhanced detection accuracy. We created a database of geotagged and labeled trios of visible, thermal, and fused images using a low-cost FLIR ONE thermal camera connected to a smartphone. Three machine-learning algorithms were proposed and compared: Anisotropic Diffusion Fusion (ADF) + Mask R-CNN, RTFNet, and RTFNet with Enhancement Parameters (EPs). The RTFNet method achieved the best F1-score of 93.7% in daytime and 90.9% in nighttime scenarios. A Bright-Dark detector was developed to optimize algorithm selection based on lighting conditions. Detected potholes were mapped using GPS data, and the trained algorithm was packaged into a GUI tool that can be used by highway maintenance teams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

This report presents the development of a visible and thermal imaging-based deep learning approach for the automated and robust detection of potholes to prioritize highway maintenance. Potholes, a primary pavement distress, significantly compromise road safety and incur substantial repair costs. The Mountain Plains region experiences frequent potholes due to harsh weather conditions, necessitating efficient and accurate detection for timely highway maintenance.

Objectives

The main objectives of this study are: (1) To create a database of geotagged and labeled images (visible, thermal, and fused) for training pothole detection algorithms. (2) To develop deep learning algorithms for accurate and robust pothole detection using these images. (3) To compare the performance of these algorithms under various conditions to determine the benefits of integrating thermal and visible images. (4) To develop automated tools for pothole detection, mapping, and updating.

Methodology

A unique dataset of pothole images was created using a low-cost FLIR ONE thermal camera attached to a smartphone. This dataset includes geotagged and labeled trios of visible, thermal, and fused images. Three machine learning algorithms, i.e., Anisotropic Diffusion Fusion (ADF) + Mask R-CNN, RTFNet, and RTFNet with Enhancement Parameters (EPs), were proposed and compared for their effectiveness in pothole detection.

Key Findings

1. **Image fusion and detection**: The fusion of visible and thermal images significantly improved pothole detection accuracy. The RTFNet algorithm achieved the best F1-score of 93.7% in daytime scenarios and 90.9% in nighttime scenarios.

2. **Algorithm comparison**: The RTFNet outperformed the other methods in terms of precision and recall, particularly under varying lighting conditions. The incorporation of Enhancement Parameters further improved detection accuracy in low-light scenarios.

3. **Automated detection tool**: An automated pothole detection and mapping tool with a graphical user interface (GUI) was developed. This tool uses the trained algorithms to detect potholes and map their locations using GPS data from the images.

Implications for Highway Maintenance

The developed tool and algorithms enable highway maintenance teams to prioritize repairs based on accurate and timely pothole detection. This approach not only enhances road safety but also optimizes maintenance resource allocation.

Future Work

Future research will focus on further improving the detection algorithms by incorporating additional data and refining the enhancement parameters. The tool will be tested and validated in different regions to ensure its robustness and generalizability.

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Background

Roads contribute significantly to the development of the region and the growth of the population. Concrete and asphalt are the most commonly used materials for road paving nowadays. Potholes are primary pavement distress that can compromise safety and cause expensive damage claims (see Figure 1.1). Potholes are the results of deterioration of pavements due to aging, weather and traffic overloads and are common problems across the United States. The dim light at night or puddles after rain caused potholes to be hard to discover. According to a study by American Automobile Association (AAA) in 2016, U.S. motorists suffer repair costs of three billion dollars annually from damage caused by potholes. Drivers, on average, spend $300 for vehicle damage from potholes (AAA 2016).

Figure 1.1 Samples of potholes

Figure 1.2 shows an example of the number of potholes reported and repaired in Seattle. The number of potholes increase because of the storm passing every winter (Bergerson 2022). Potholes are even more common in the Mountain Plains region due to the snow and freeze/thaw effect. Thus, identifying and repairing potholes is one critical aspect of highway maintenance because it will form larger potholes from traffic and heavy vehicles such as trucks and buses if potholes aren't repaired in time.

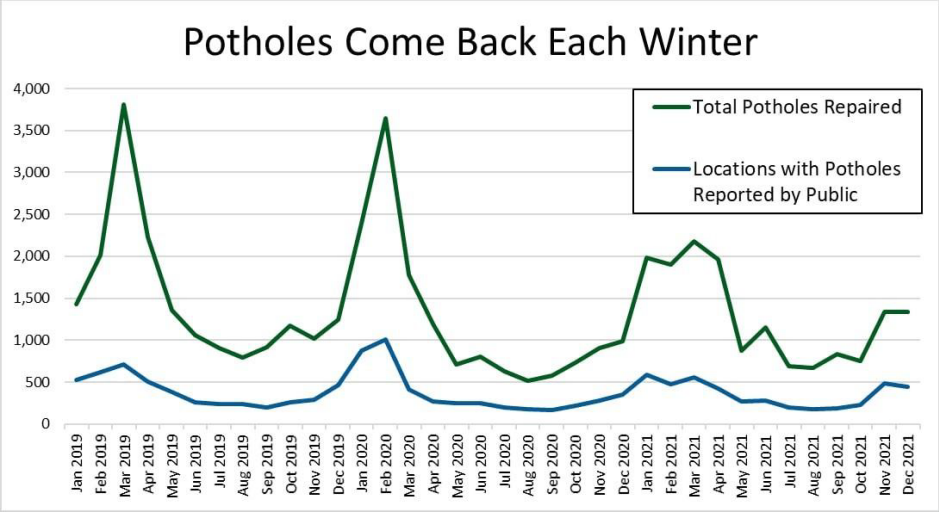

Figure 1.2 The number of new pothole locations reported by the public per week in Seattle (Bergerson 2022)

According to a report from the U.S. Department of Transportation, there are more than 820,519 miles of roads in the United States (USDT 2022). It is time- and cost-consuming to track every road condition by only using manpower. Therefore, accurate, robust, and fast detection of potholes is critical to enabling timely and cost-effective pavement maintenance.

Currently, there are some studies that have used deep learning algorithms for pothole detection. For example, Maeda et al. (2018) used deep neural networks with smartphone images to do road damage detection and classification. Arjapure & Kalbande (2021) developed deep learning model for pothole detection and area computation. However, there are some challenges that still need to be overcome. Most of the developed models so far focused on using visible (RGB) images for the detection. However, when there is insufficient lighting or low contrast with surroundings (e.g., if is dark or in poor weather conditions such as cloudy, rain, fog), the detection based on only visible images may not perform well. Despite the fact that thermal images contain fewer texture details than visible images, they can provide the temperature difference between potholes and surrounding pavement. Recently, Bhatia et al. (2022) developed a deep learning model for pothole detection using thermal images, showing the promise of using thermal images to improve pothole detection. The fusion of both visible and thermal images potentially integrates features from both. However, there is still a lot of research that needs to be done to fully explore and leverage the advantages of integrating information from both visible and thermal images to improve pothole detection. To train machine learning algorithms for pothole detection, training data/images are needed. However, so far there is no existing dataset that contains labeled visible and thermal images of potholes. Besides, the use of image fusion algorithms for the purpose of pothole detection has not been investigated. Also, tools are needed to facilitate the automated pothole detection and mapping and use by stakeholders.

1.2 Research Objectives

The goal of this research is to develop visible and thermal imaging and a deep learning-based approach for automated and robust detection of potholes to enable timely and cost-effective maintenance of highways. The following major objectives are designed to meet this goal:

1. Create a unique and valuable database of geotagged and labeled trios of visible, thermal, and fused images for training pothole detection algorithms.

2. Develop deep learning algorithms for automated and robust pothole detection based on visible, thermal, and fused images.

3. Test the hypothesis that the incorporation of thermal and fused images could lead to more accurate and robust pothole detection by comparing the detection performances for different cases.

4. Develop automated tools for pothole detection, pothole mapping and updating.

1.3 Research Method

To address the above goals, this research proposes integration of both visible and thermal images captured by a visible and thermal dual camera and the use of deep learning to enable robust, accurate and automated detection of potholes to help prioritize highway maintenance. An overview of the proposed research is shown in Figure 1.3. Instead of relying only on visible images, both visible and thermal images, and the fused images with salient features from both visible and thermal images, will be used to improve the accuracy and robustness of pothole detection.

First, a unique database of geotagged and labeled trios of visible, thermal, and fused images will be created for training pothole detection algorithms. This will be achieved by using visible and thermal FLIR one camera mounted on cars to take pictures of the same road surfaces. The images will include pavements with and without potholes. To include images under different lighting conditions and weather conditions in the image database, images will be collected for the same road segments during different times of the day and also under different weather conditions. Based on the collected visible and thermal images, image fusion techniques will be used to extract features from these images to establish corresponding fused images (Hou et al. 2021). The collected images will be geotagged using GPS information. To standardize the images, some preprocessing (e.g., cropping, resizing) will be applied to the images. Then, using tools, such as LabelMe, these images will be manually labeled as with potholes or not, and for those with potholes, the potholes will be annotated. The annotation will be applied to all three types of images. Out of all the images, three sets will be created, including a training set, validation set, and testing set, with each set including corresponding visible, thermal, and fused images for positive cases (i.e., with potholes) and corresponding annotations.

Secondly, based on the labeled images, deep learning algorithms (Mask R-CNN and RTFNet) will be trained to classify the images as with or without potholes and also for those with potholes further identify the pothole through detection and segmentation. Data augmentation (including flip, rotation, and shifting) will first be applied to increase the size of the training sets. To address the requirement of large, annotated image datasets by deep convolution neural network, transfer learning will be used, where deep convolution neural network pre-trained on other existing large-scale image datasets will be fine-tuned through the collected images (Gopalakrishnan et al. 2017). This way the required number of labeled images can be reduced. Different deep learning algorithms will be investigated. The performance of the trained candidate deep neural networks will be compared in terms of accuracy and efficiency. The impacts of additionally incorporating thermal and fused images for pothole detection and segmentation on the pothole detection performance will be investigated. The results will provide guidance on how to best make use of images from different sensors.

Third, develop automated tools for pothole detection, pothole mapping and updating. The established deep neural network models with the best performance will be used for automated pothole detection when new images are collected and need to be processed. The location and picture of the potholes will be shown on maps using the GPS information for the images. Functionality of the tool, in terms of detection of the pothole and marking the potholes by their location, will be provided to facilitate prioritization of pavement maintenance.

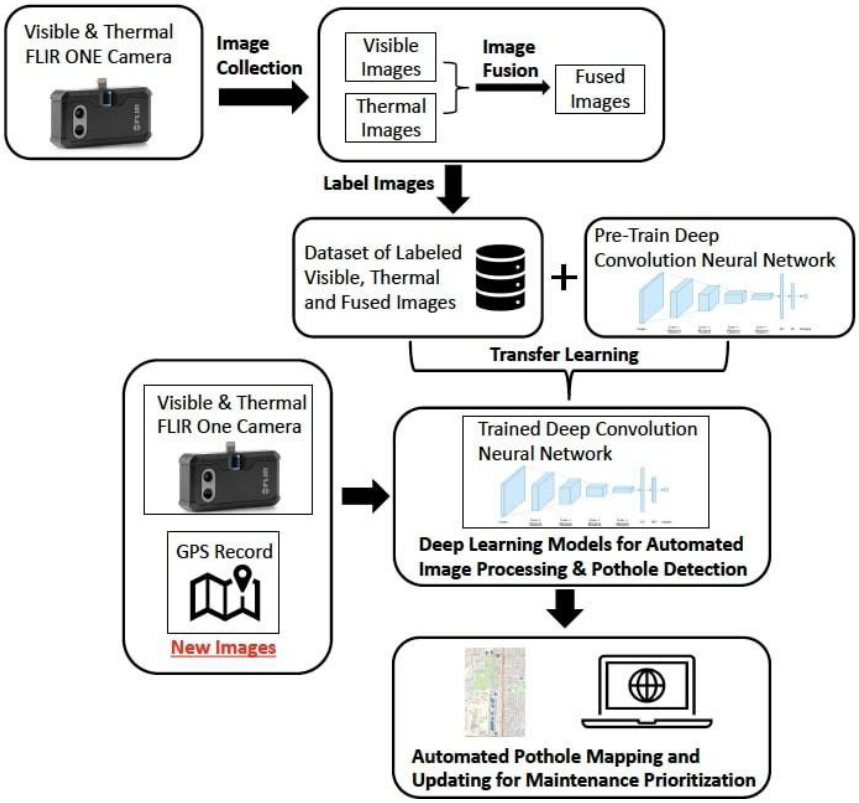

Figure 1.3 Overview of the proposed research

1.4 Organization of Report

This report is organized as follow: In Chapter 2, deep learning algorithms are introduced for pothole detection. Chapter 3 proposes three methods for pothole detection with the fusion of visible and thermal images. Chapter 4 introduces the procedure of establishing the proposed dataset. Chapter 5 explains the results of proposed methods. In chapter 6, the development of the automated pothole detection tool will be provided. Finally, conclusions and future directions are provided in Chapter 7.

2. POTHOLE DETECTION WITH MACHINE LEARNING

This chapter provides an overview of some commonly used machine learning (ML) algorithms for object detection and pothole detection.

2.1 Overview of Convolutional Neural Network (CNN)

Machine learning algorithm for object detection is typically divided into two categories: Convolutional Neural Networks (CNN) and Region-based Convolutional Neural Networks (R-CNN). A typical CNN is mainly used for image classification. However, it cannot tell the position of the detection object in the image. R-CNN is based on CNN combined with region proposal while using the Selective Search (SS) method to extract the features of candidate regions, thereby achieving the goal of target detection and regional positioning. This section provides a brief overview of CNN and R-CNN.

2.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is the simplest and most widely used deep learning method in image detection and classification (Alzubaidi et al. 2021). Figure 2.1 shows the procedure of CNN. CNN is composed of three parts: convolutional layers, pooling layers, and fully-connected layers. The convolutional layer can be considered as a filter that can extract image features and then use the pooling layer to reduce image dimension. It can not only increase the training speed but also avoid overfitting. The result is eventually generated by the fully connected layer. Different from simply using full images for computation, CNN has the ability to find characteristics of candidate images and focus on analyzing interested areas, which can enormously improve computing efficiency and accuracy. Convolutional neural networks have been used in many scenarios in recent years, such as image classification, voice recognition and face recognition (Albawi et al. 2017).

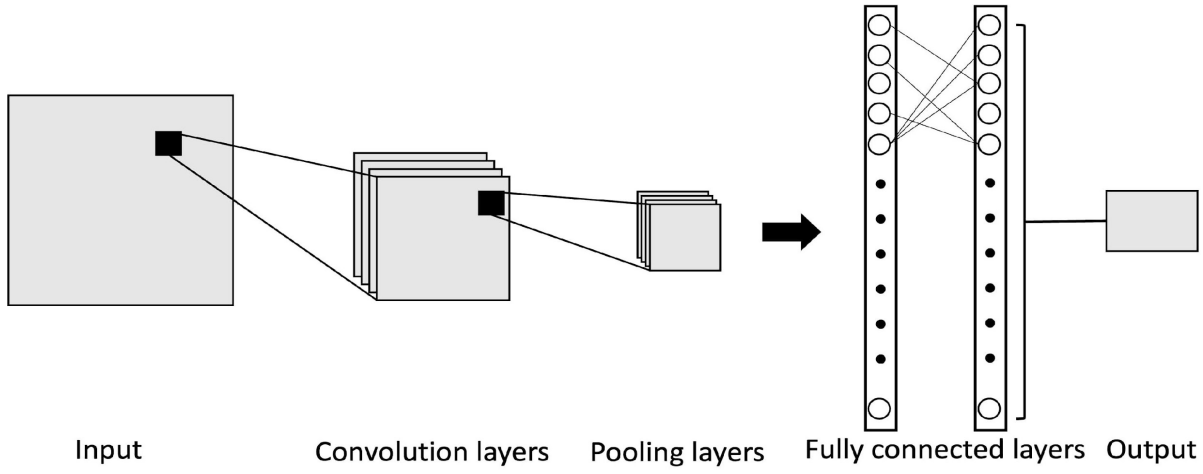

Figure 2.1 The procedure of the Convolutional Neural Network (CNN)

2.1.2 Region-based Convolutional Neural Networks (R-CNN)

The Region-based Convolutional Neural Network (R-CNN) was proposed by Girshick, et al. (2014). Different from CNN, R-CNN first uses the Selective Search method to generate thousands of region proposals and substitutes those regions into the original convolutional neural network after crop and warp instead of substituting each region proposal respectively (Girshick, et al. 2014). Note the Selective Search is an algorithm to locate possible objects with boxes in different scales. Refer to the Selective Search paper (Uijlings, et al. 2013) for more details.

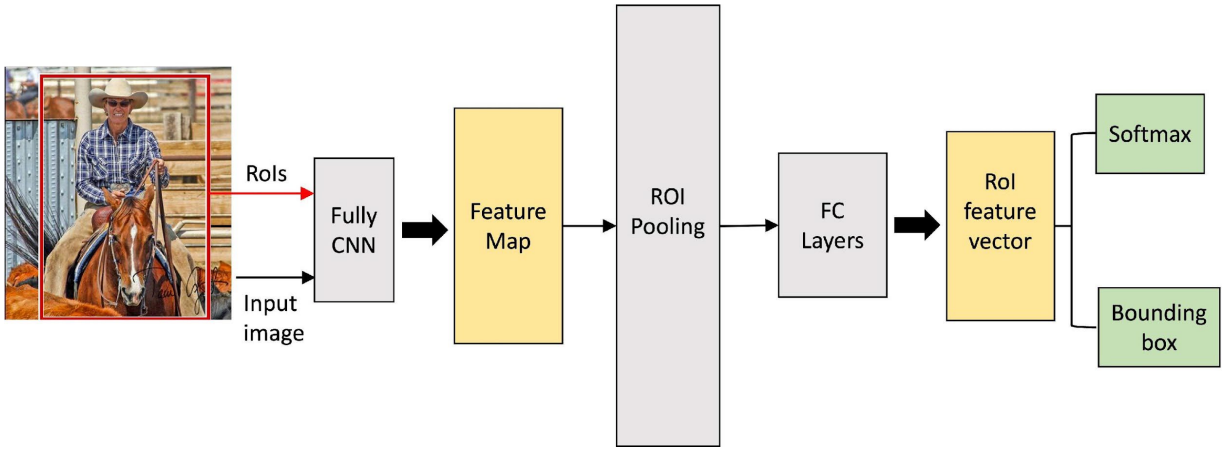

Figure 2.2 The architecture of Fast R-CNN

However, proposing thousands of regions by only R-CNN and bringing them into the CNN operation at once are time-consuming processes. Hence, the Fast R-CNN method was proposed by Girshick (2015) to address shortcomings of R-CNN. Figure 2.2 shows the image-processing procedure of the Fast R-CNN. The main difference from the R-CNN is the Fast R-CNN inputs entire images into CNN, extracts feature maps as regions of interest (RoIs) and maps the corresponding region proposal to output feature maps. Then, RoI pooling is used to convert the RoI area into a feature map with fixed and integer edges to facilitate the segmentation of the region proposal. At the end of Fast R-CNN, the Softmax loss will be applied for classifying and calculating the probability and outputting the corresponding bounding boxes (Girshick 2015).

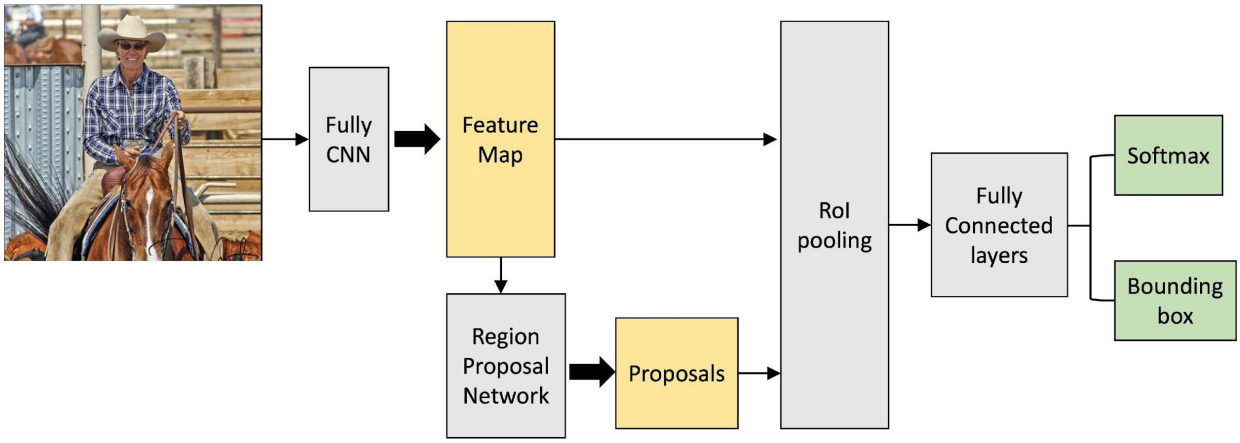

Figure 2.3 The architecture of Faster R-CNN

Though the Fast R-CNN includes the concept of RoI, both R-CNN and Fast R-CNN are still using a Selective Search method to extract region proposals before passing to the convolution layers, which still spends too much time on image processing. Therefore, new object detection algorithms, Faster R-CNN, were proposed by Ren, et al. (2015). Figure 2.3 shows the architecture of the Faster R-CNN. Instead of randomly creating RoI regions before convolution layers, Faster R-CNN extracts RoI regions directly from feature maps by generating different sizes of anchor boxes through the Region Proposal Network (RPN), which is designed for predicting object bounds and objectness scores at each position simultaneously. Ultimately, similar to Fast R-CNN, the RoI pooling layers are applied for reshaping the predicted region proposals. It helps the network classify objects and predict offset values of the bounding box in the Fully Connected layers. In Faster R-CNN, the speed of generating RoI is accelerated because of replacement of the Selective Search by RPN.

2.2 Mask Regions with Convolutional Neural Network (Mask R-CNN)

The results from Faster R-CNN are only predictions with boundary boxed. There is no object mask created from the previous R-CNN we introduced earlier. Therefore, the Mask Regions with Convolutional Neural Network (Mask R-CNN) was proposed by He, et al. (2017). Figure 2.4 shows the architecture of Mask R-CNN. As can be seen, the architecture of Mask R-CNN is similar to that of the Faster R-CNN but with an additional branch for instance segmentation. The image features after RoI Align will go through the additional fully convolutional network for instance segmentation. In this section, we discuss the two major parts that differ from the Faster R-CNN: Feature Pyramid Network (FPN) and RoI Align.

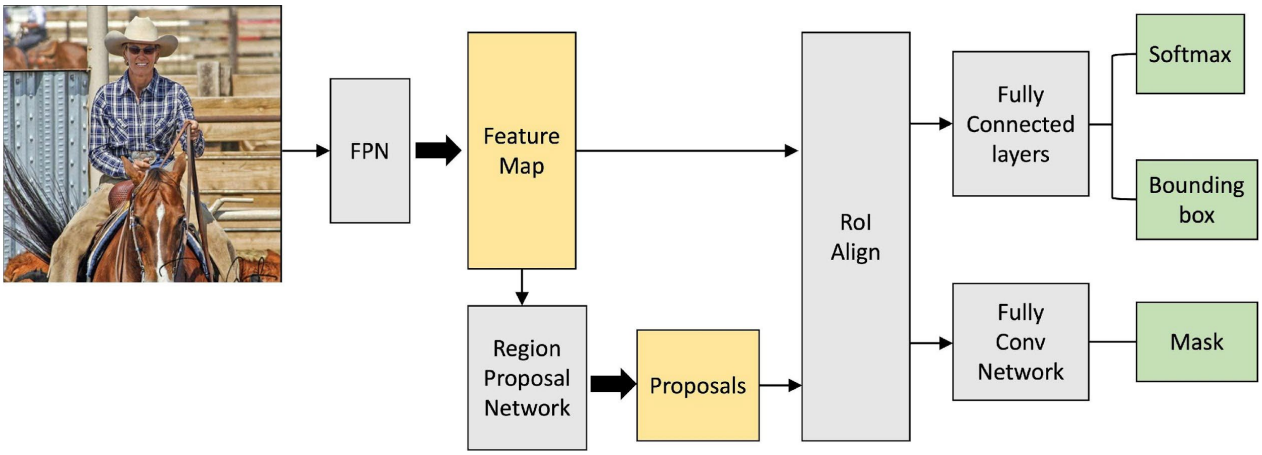

Figure 2.4 The architecture of Mask R-CNN

2.2.1 Feature Pyramid Network (FPN)

The regular feature pyramid uses only the last layer of feature maps for object prediction. It is widely used in image recognition models, such as ImageNet and most of the CNN structural models. However, its shortcoming is that it is hard to detect small objects due to the low resolution of the feature map in the last layer, resulting in predictions with low accuracy. Hence, to make complete usage of the feature maps from CNN output and to preserve or enhance each feature in the pyramid, the Feature Pyramid Network (FPN) was proposed (Lin et al. 2017)), which became the main feature extractor in Mask R-CNN. Figure 2.5 shows how FPN processes feature maps. FPN first upsamples in the top-down pathway to coarse feature map. Then, the FPN leverages the information of small objects passed from top to bottom to increase the feature resolution for small objects. The FPN fuses the feature maps from the bottom layer to the higher layer in the fully convolutional method, so the features in each stage can be thoroughly extracted.

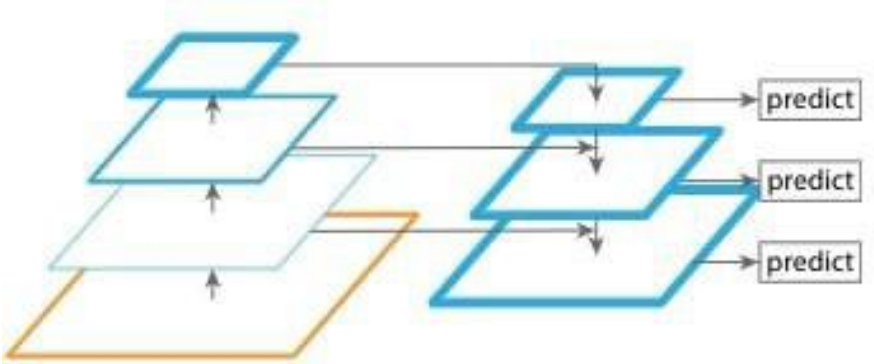

Figure 2.5 The data flow of Feature Pyramid Network. Feature maps are indicated by blue outlines and thicker outlines denote semantically stronger features (Lin et al. 2017)

2.2.2 RoI Align

The role of RoI pooling is to pool the corresponding region into a fixed-size bin in the feature map according to the position coordinates of the preselected box for subsequent classification and regression. In the process of RoI pooling, the boundary of the bin will be converted into an integer, so the position of the prediction will deviate from the actual classification, which is called "Misalignment" (He et al. 2017).

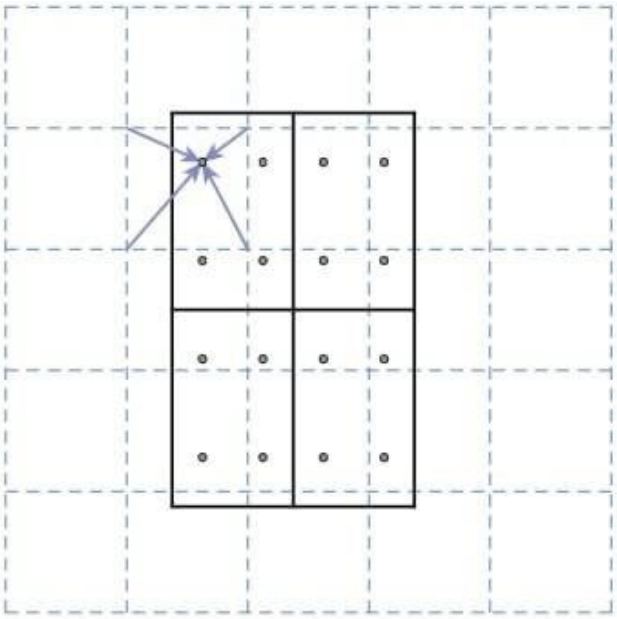

Figure 2.6 RoIAlign samples the float coordinate (He et al. 2017)

To keep floating numbers instead of converting them into integers to solve the problem of deviation from RoI pooling, He, et al. (2017) proposed RoI Align, which is a method of regional feature aggregation proposed in the Mask R-CNN. It has improved accuracy of the detection model by replacing the RoI pooling in the previous R-CNN model. Shown as Fig. 2.6, the solid lines represent RoI while dashed lines show a feature map. Four dots are sampling points in each bin. Instead of using quantization, RoI Align uses Bilinear Interpolation to calculate the value of four sampling points from the feature map. Finally, the result of RoI Align can be obtained by max pooling four sampling points in each bin.

2.3 Pothole Detection Using Deep Machine Learning

In the context of pothole detection, with recent advances in deep learning, several methods have been proposed and used for pothole detection and localization. Kulkarni, et al. (2014) proposed a pothole detection system with a neural network technique using the Accelerometer Sensor of Android smartphone and the GPS information for detecting and plotting the pothole. They named this system "Encog" which can obtain an accuracy of 95%. Song, et al. (2018) proposed a CNN method with the utilization of Inception V3 and Transfer Learning to detect potholes. The model can correctly recognize all instances of pothole. Maeda, et al. (2018) leveraged a deep learning model to train on a self-developed dataset, which was divided into nine

categories according to different types of road damage. The dataset has been published as Road Damage Dataset (RDD- 2018), which contains road damage images from several different countries. With the use of the RDD-2018, Arya, et al. (2021) evaluated 16 deep neural network models trained with different combinations of datasets from different countries. This research showed the road damage detection model for a country can be trained with the dataset developed with the local data mixed with data from another country. Majidifard, et al. (2020) implemented two object detection models YOLO-v2 and Faster R-CNN for detecting potholes. They used the road damage datasets with a top-down view and a wide view of the pavement segment to train the deep learning network, with the aim of automated pavement rating. Arjapure & Kalbande (2021) implemented Mask R- CNN to detect and predict potholes and calculated their area. They achieved an accuracy of 90% for predicting the area of pothole. Fan, et al. (2019) presented a robust pothole detection algorithm based on stereo vision with the 3D road database. The algorithm utilized the difference of disparity maps between the modeled and the actual to detect potholes. By using this technique, they achieved a successful detection accuracy of 98.7% and overall pixel-level accuracy of 99.6%. Ahmed (2021) compared the performances of 10 CNN models for pothole detection including YOLOv5, YOLOR and Faster R-CNN. They achieved the highest precision of 91.9% from Faster R-CNN with the backbone of ResNet50 while the fastest prediction model from the Small YOLOv5.

The above-mentioned works all use visible images for pothole detection. Recently, the self-built convolutional neural model for pothole detection based on thermal images had been established by Bhatia, et al. (2022). This is the first application of thermal images for pothole detection and they achieved the best accuracy of 97.08%. Gupta et al. (2020) also used thermal images for pothole detection. By utilizing deep neural networks and bounding boxes, they proposed a novel method of pothole localization from thermal images. Their model is based on a modification of the ResNet50-RetinaNet model. Overall, they achieved a precision of 91.15%. Another model for pothole detection with thermal images was proposed by (Sathya & Saleena 2022), combining CNN and aquilla optimization (AO) algorithm. The model overcame limitations of CNN model and reduced the processing time of detection. The precision and recall obtained from this model are 96.6% and 97.2%, respectively.

These literature reviews show deep learning algorithms have been gaining popularity in pothole detection using either visible images or thermal images with a good amount of accuracy and performance. However, there are some challenges that must be overcome. Most of the developed models have so far focused on using visible (RGB) images for the detection. However, when there is insufficient lighting or low contrast with surroundings (e.g., if it is dark or in poor weather conditions such as cloudy, rain, fog), the detection based on only visible images may not perform well. On the other hand, relying only on thermal images may potentially lead to low accuracy. While the fusion of both visible and thermal images shows promises, there is still a lot of research that must be done to fully explore and leverage the advantages of integrating information from both visible and thermal images to improve pothole detection. This research tries to combine information from visible and thermal images by image fusion algorithms to develop a pothole detection tool suitable for both daytime and nighttime conditions.

3. PROPOSED APPROACH FOR POTHOLE DETECTION USING VISIBLE & THERMAL IMAGES

3.1 Introduction

The aim of this research is to develop machine learning algorithms for accurate and robust pothole detection using both visible and thermal images through image fusion. However, from the previous chapter, we know the Mask R-CNN network uses visible images and does not include the image fusion algorithm. For this reason, we need to first fuse images independently before going through the Mask R-CNN process. Apart from adding a separate fusion algorithm, the development of the data-fused semantic segmentation model is also growing (e.g., MFNet (Ha, et al. 2017), FuseNet (Hazirbas, et al. 2017)). Essentially, it combines the fusion and segmentation processes to enable an end-to-end model structure. This section will first introduce an algorithm for image fusion known as Anisotropic Diffusion Fusion (ADF) to perform before the Mask R-CNN. This section will then introduce the RGB-Thermal Fusion Network (RTFNet), which is an end-to-end data-fused semantic segmentation model based on the fusion of visible and thermal images. In the end, we proposed three methods with different combinations of algorithms for pothole detection. Their performances will be evaluated later in Chapter 5.

3.2 Image Fusion Algorithms

3.2.1 Anisotropic Diffusion Fusion (ADF)

Anisotropic Diffusion is a method that reduces noise and smooths the given images by using a partial differential equation (PDE) without reducing the presence of image features, such as edges and lines (Perona & Malik 1990). The equations of Anisotropic Diffusion of a grayscale image I_t is given as:

$$I_{t+1}(x,y) = I_t + c_t(x,y) \cdot \nabla I_t \tag{3.1}$$

$$c_t(x,y) = g(|\nabla I_t(x,y)|) = \frac{1}{1+(\nabla I/K)^2} \tag{3.2}$$

where ∇ is the gradient operator with respect to the space variables. $c_t(x, y)$ is the diffusion coefficient, which can simply be known as the rate of diffusion. K is a constant value to control the sensitivity to the boundaries. $c_t(x, y)$ equals to 1 when $\nabla I_t = 0$. It can be also considered as there is an edge or boundary. Therefore, I_t will be diffused in Eq. (3.1); otherwise, the features will be kept in the original image while $c_t(x,y)$ equals to 0.

Similar to the Gaussian blur, Anisotropic Diffusion has a diffusion coefficient. The difference is the coefficient in Gaussian blur is fixed and it will blur boundaries during the diffusion. In the Perona & Malik (1990) model, the diffusion coefficient is determined according to the boundary detection, so the edges in the input image can be preserved after diffusion.

Based on the development of Anisotropic Diffusion, the Anisotropic Diffusion Fusion (ADF) method has been proposed for fusing visible and thermal images to get a new edge preserving image (Bavirisetti & Dhuli 2015). Figure 3.1 illustrates the process of the ADF method. We usde the ADF method to extract the information of visible and thermal images into detail layers and base layers and then fused both images respectively. Images fuse with Karhunen-Loève Transform (KL-transform) in the detail layers fusion section, which can highlight the differences by eliminating the correlation between features of images. Images have been fused together with respective weights in the base layers fusion section. In the final part, detail layers and base layers are fused with a simple combination.

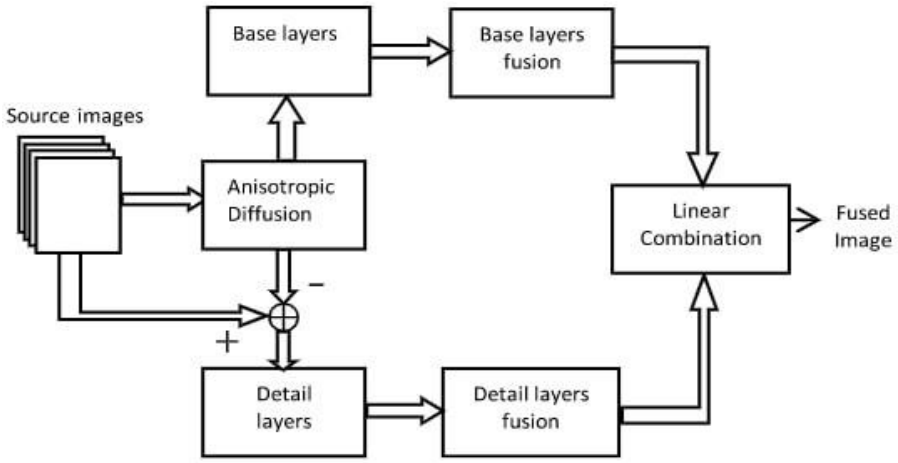

Figure 3.1 Illustration of Anisotropic Diffusion Fusion (Bavirisetti & Dhuli 2015)

3.2.2 RGB-Thermal Fusion Network (RTFNet)

RGB-Thermal Fusion Network (RTFNet) is a deep learning algorithm for semantic segmentation in urban scenes with the fusion of visible and thermal images (Sun et al. 2019). Similar to SegNet and MFNet, RTFNet adopted the encoder-decoder architecture, which can let the fusion and training process become an end-to-end process. The architecture of RTFNet is shown in Figure 3.2. The blue and yellow sections are the encoder and decoder, respectively. In the encoder part, both RGB and thermal images are encoded respectively, and feature maps are extracted from both images. Different from other image fusion networks, RTFNet applies ResNet to be the backbone of the feature extractor. However, the author removed the average pooling and the fully connected layers of ResNet not only to prevent excessive loss from the feature map but to reduce the input size to simplify computation. To reduce the resolution to simplify the extraction of the feature maps, a max pooling layer and four residual layers are sequentially employed as encoder layers following the initial block. In each encoder layer, feature maps extracted from the thermal image are fused into RGB features through element-wise summation and taken as input for the decoder.

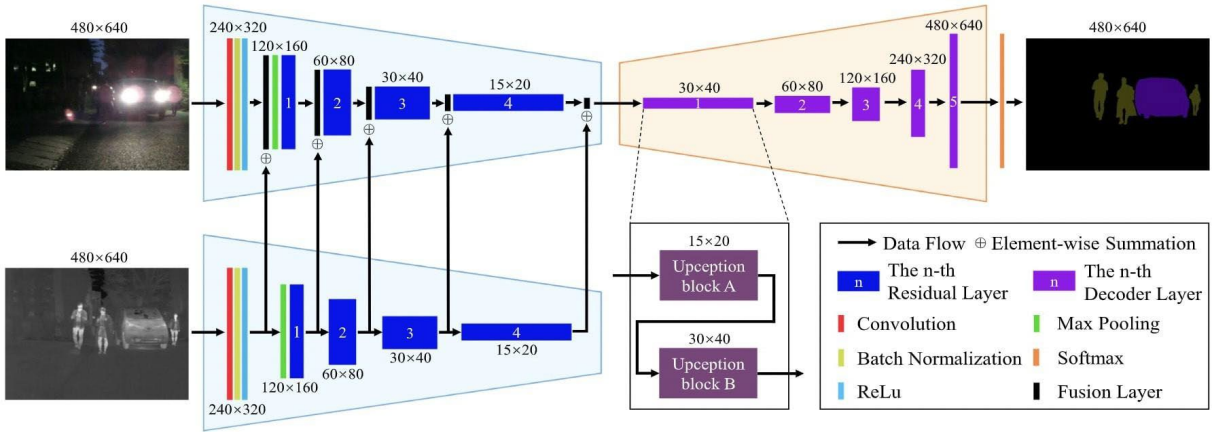

Figure 3.2 Illustration of the architecture of RGB-Thermal Fusion Network (RTFNet) (Sun et al. 2019)

The decoder section is used to recover the size of the images to make the predictions. Two Upception blocks are applied for adding up feature maps. There are five decoder layers employed in the decoder section, each decoder layer contains Upception block A and Upception block B, which can be considered as a block for add-up the input and the feature map element-wisely. In the final part of RTFNet, the Softmax layer is added for results predictions.

3.2.3 RTFNet with Enhancement Parameters (EPs)

The advantage of the RTFNet model is its capability to extract the features from visible and thermal images and fuse that information in a deep neural network. However, the fused weight of the RGB and thermal images in the original RTFNet structure are both equal to 1. In some circumstances, the thermal or RGB image may be more informative than the other, and fusing images with the same weight under different cases may affect the detection results. For example, thermal images would be more informative for pothole features in the nighttime scenario. Giving higher weight to more informative images would benefit the detection. To adjust the level of importance of information in thermal images, we add five new parameters in every encoder layer in the RTFNet. We termed these parameters Enhancement Parameters (EPs) because they enhance features in thermal images. Figure 3.3 displays the information extracted from thermal images that will multiply by each EP to enhance the thermal feature maps respectively before fusing those into visible image feature maps. The RTFNet with enhancement parameters (EPs) we proposed here will later be used in the proposed methods for pothole detection.

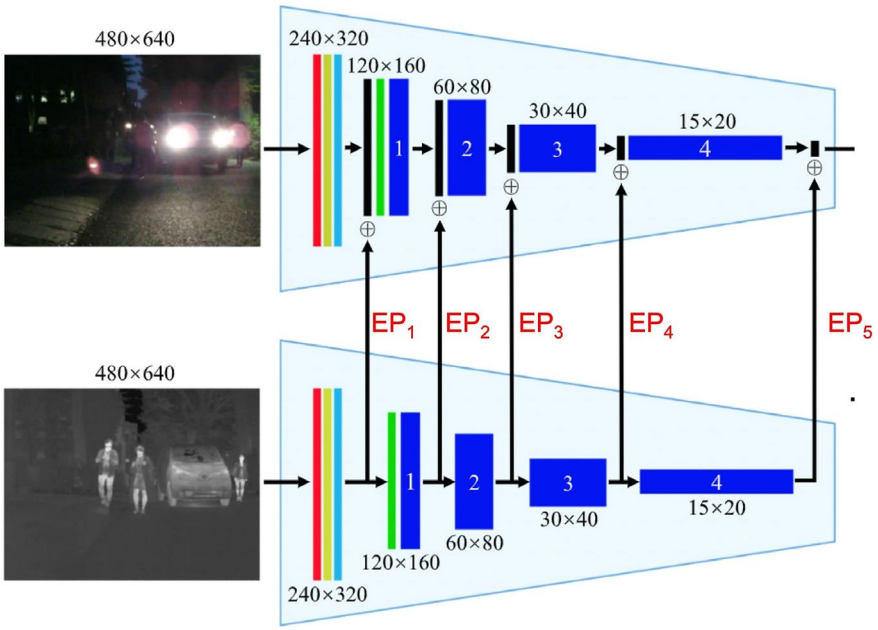

Figure 3.3 Adding Enhancement Parameters (EP) into the encoder of the RTFNet

3.3 Proposed Methods for Pothole Detection with Images Fusion

This research uses the Mask R-CNN and the RTFNet for pothole detection and segmentation. Three methods are proposed for accurate and robust pothole detection by building on and modifying the Mask R-CNN and the RTFNet. Figure 3.4 shows three proposed methods. The first method is the combination of the anisotropic diffusion fusion and the Mask R-CNN; the second method directly uses the RTFNet proposed by Sun, et al. (2019) for pothole detection; for the third method, we add a Bright-Dark detector we have developed into the RTFNet to determine which input must be enhanced for the thermal image by EPs. In this section, we will discuss in detail each method and the process of establishing the corresponding network.

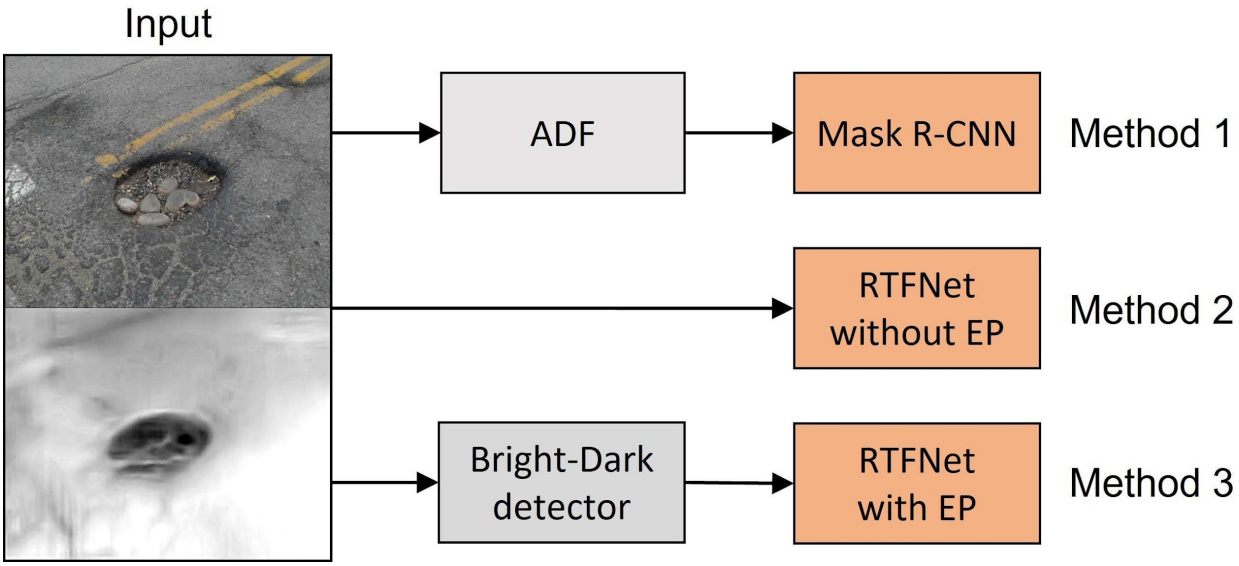

Figure 3.4 Three proposed methods for image fusion and pothole detection

3.3.1 Method 1: ADF + Mask R-CNN

Mask R-CNN is an instance segmentation framework for object classification and detection. However, image fusion is not included in this model. For this reason, we decided to fuse our inputs (i.e., visible and thermal images) with Anisotropic Diffusion Fusion (ADF) as a data preprocessing step. ADF is used because the characteristic of this fusion method is to emphasize and maximize the features from visible and thermal images. Details of ADF have been introduced in the previous section. After fusing the images, the visible and thermal images become a 3-channel RGB fused image. Then we utilize these processed images as input to the Mask R-CNN model for pothole detection.

3.3.2 Method 2: RTFNet

Different from Method 1, the RTFNet is already an end-to-end data-fuse network for semantic segmentation. The feature maps of visible and thermal images are fused by the element-wise summation and the semantic segmentation is used for pothole detection. Details of RTFNet have been introduced in the previous section.

3.3.3 Method 3: Modified RTFNet with Bright-Dark Detector

In Method 3, we added an enhancement parameter in each encoder layer in the RTFNet encoder to increase the ratio or weight of the information in thermal feature maps. More details on the enhancement can be found in the previous section. It is expected that for dark images, the enhancement of thermal features could be beneficial, while for bright images, there is less or no need to use the EPs to enhance the thermal features. To determine whether the EPs needed to be used in thermal feature maps, the brightness of the visible image was used as a reference. Hence, we developed a Bright-Dark detector to help determine the lightning condition of images. In the Bright-Dark detector, the RGB image was transformed into a grayscale image. Pixels in the image were converted to numbers from 0 to 225. We defined that pixels with numbers below 40 would be considered dark pixels. Last, we calculated the percentage of dark pixels in the whole image. If the percentage of dark pixels is higher than 0.75, we regarded this image as a dark image, otherwise as a bright image. In the dark image case, the corresponding thermal feature maps will multiply by the corresponding EP in each encoder layer in the RTFNet to strengthen information of thermal images. In the bright image case, the EPs were set as 1, which essentially means there was no enhancement.

4. DATA COLLECTION, PRE-PROCESSING, AND AUGMENTATION

4.1 Introduction

In this chapter, we will introduce the procedure for establishing the annotated fused image dataset, including the data collection, pre-processing, annotation, and augmentation. Figure 4.1 shows the workflow, and the details are presented in the following sections. The established dataset will be later used to train and compare the proposed pothole detection algorithms.

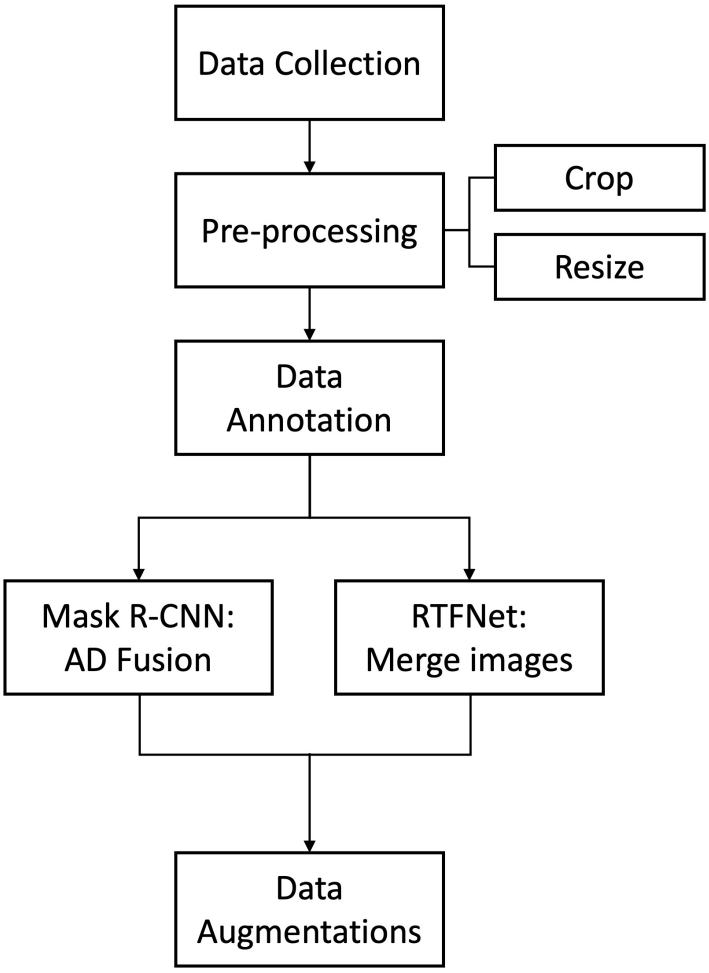

Figure 4.1 Workflow to establish the annotated fused image dataset

4.2 Data Collection

In this research, we used the FLIR ONE Pro LT thermal camera to collect our data. The FLIR one camera has the capability to capture both visible and thermal images with the same angle, which helps avoid the process of aligning both images. The FLIR camera can connect to a smartphone, and the captured images will automatically synchronize to the cloud drive. The visible and thermal images are merged by multi-spectral dynamic (MSX) technology through the FLIR ONE mobile app.

Figure 4.2 FLIR thermal camera setup

The FLIR ONE camera was connected to a smartphone attached to the rear windshield by using the suction cup and smartphone holder (shown in Figure 4.2). Instead of attaching to the front windshield, attaching it to the rear windshield can prevent the camera from being blocked by the front of the car. To make sure both visible and thermal images were obtained well, we attached the smartphone outside the car to avoid the windshield blocking the conduction of thermal infrared. When the proposed algorithm was deployed, we expect this setup would enable efficient collection of images for large number of roads and over large distances as well.

To establish a comprehensive dataset that includes different circumstances/conditions, the dataset had been collected in three conditions: daytime, cloudy, and nighttime. The samples of visible and thermal images in three conditions are shown in Figure 4.3. This dataset consisted of 224 daytime images, 222 nighttime images and 232 cloudy images.

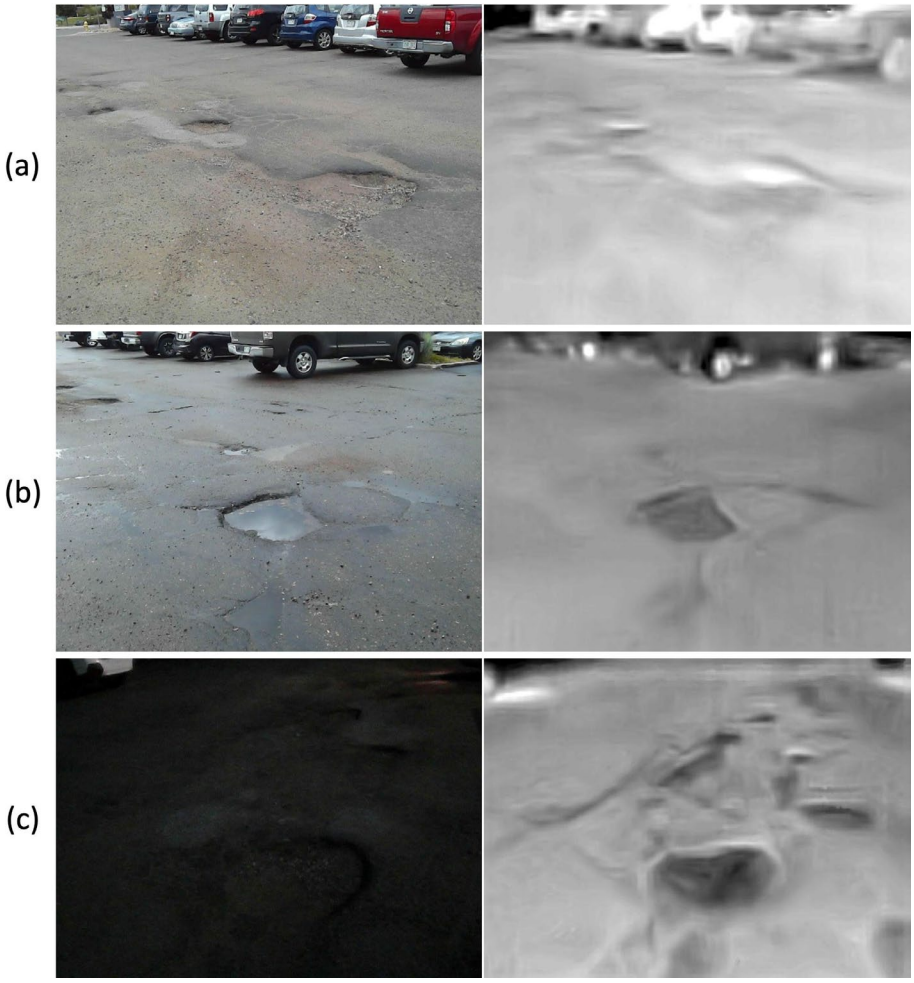

Figure 4.3 Samples of visible images (left column) and thermal images (right column) in three different conditions. (a) daytime, (b) cloudy, (c) nighttime

4.3 Data Pre-processing and Annotation

The resolution of visible images and thermal images captured from FLIR one camera was 1440x1080 and 640x480, respectively. The area of thermal image is the center part of visible image. An illustration of overlapping both images is shown in Fig 4.4. Therefore, we first needed to crop the pixels around the visible image to match the size of the thermal image. Second, the images' size must be a multiple of 64 to satisfy the limitation of dataset in the Mask R-CNN model. Therefore, we resized visible images and thermal images to 640x448 with the least impact on the image scale.

Figure 4.4 Thermal image overlay on visible image to illustrate differences between both areas

Because we used supervised machine learning as our network, the images had to be annotated manually to relate potholes with the ground truth. We did the image annotation and the JSON format by using LabelMe, which is an open-source software available on GitHub. For this research, a pothole is the only class that has been marked. Figure 4.5 shows two annotated sample images, and both one pothole and muti-potholes in a single image are acceptable. Instead of labeling by a rectangular box, we used polygons to enable the dataset to distinguish features between potholes and backgrounds properly.

Figure 4.5 Visible images annotation using LabelMe, (a) single pothole, (b) multi-potholes

4.4 Data Fuse and Merge

To ensure the visible and thermal images are transformed in the same way during augmentations, we needed process the images differently for the three methods. In Method 1, we implemented ADF to fuse images before training the Mask R-CNN. Figure 4.6 shows the workflow of the data processing in the first method. The characteristic features of visible and thermal images, such as edges, are emphasized by the ADF, and the image became a 3-channel fused image; 3-channel images from ADF will be the training data for the Mask R-CNN.

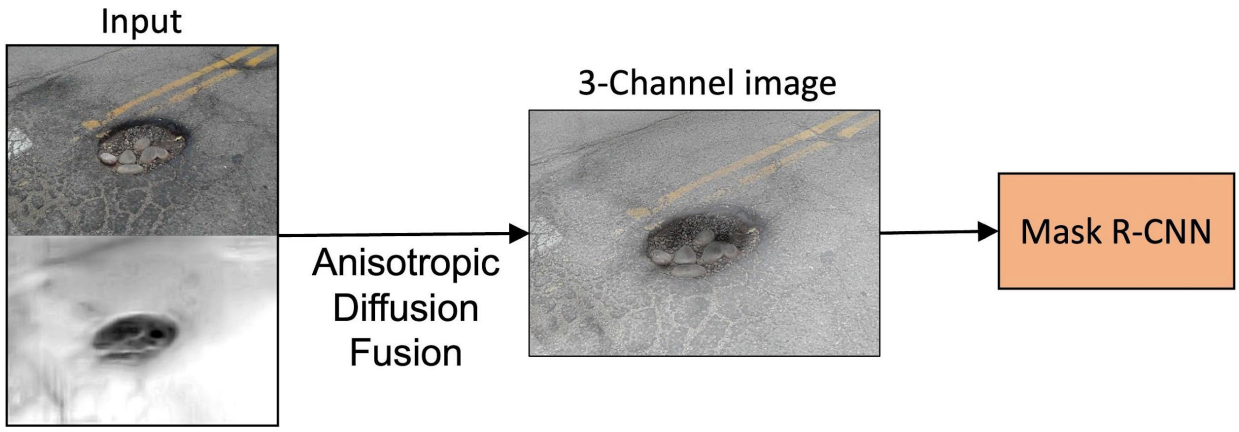

Figure 4.6 The dataset fusion process for Method 1

Figure 4.7 shows the data processing for the RTFNet. A 3-channel visible image and a 1-channel grayscale thermal image were merged into a 4-channel image. The first three layers of a 4-channel image are from a visible image while the last layer is from a thermal image. This dataset could be used in our proposed Method 2 and Method 3.

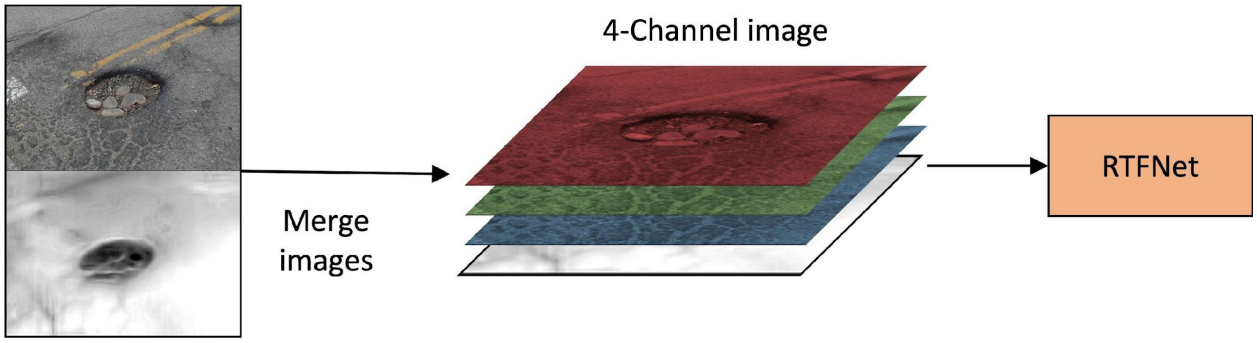

Figure 4.7 The dataset fusion process for methods 2 and 3

4.5 Data Augmentation

The original dataset a total of 678 images in the daytime, nighttime and cloudy conditions. However, the dataset was not sufficient to get the best results in model training because the limited size could lead to over-fitting. Thus, data augmentations through image transformations were implemented to increase the size of our dataset before training the ML models. The transformations included the combination of augmentations of left-right flip, up-down flip, random rotation with a probability of 0.5, and random shifting of 0.5. Examples of images augmented and used in the training model are shown in Figure 4.8. After the data augmentation, we increased our dataset to 1,200 daytime images, 1,200 cloudy images, and 1,160 nighttime images. Table 4.1 shows the numbers of images in the collected dataset before and after augmentations in the daytime, cloudy and nighttime scenarios.

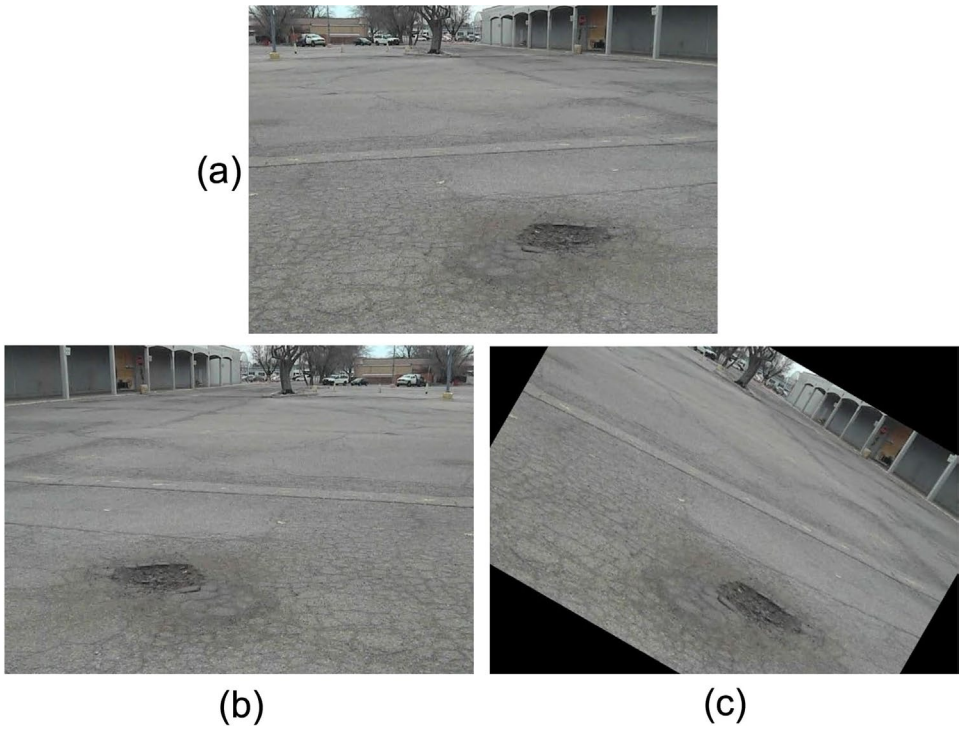

Figure 4.8 The samples of data augmentations. (a) original image, (b) left-right flip, (c) rotation

Table 4.1 The numbers of images in three different conditions before and after image augmentation

	Daytime	Cloudy	Nighttime	Total
Before augmentation	224	222	232	678
fter augmentation	1,200	1,200	1,160	3,560

5. TRAINING, VALIDATION, AND COMPARISON OF PROPOSED POTHOLE DETECTION METHODS

5.1 Algorithm Environment Setup

In this experiment, TensorFlow and PyTorch were used as the main frameworks in the Mask R-CNN and RTFNet, respectively. All experiments ran on the same equipment with the following hardware and software configurations:

1. **CPU**: AMD Ryzen 7 1700X (8-core)
2. **GPU**: NVIDIA GeForce RTX 3060 (12GB)
3. **Compiled language**:
    - Mask-RCNN: Python 3.6 with TensorFlow 1.15.0 and Keras 2.4.3
    - RTFNet: Python 3.8 with PyTorch 1.11.0, CUDA 11.3.0 and cuDNN 7.0 libraries

5.2 Performance Evaluation Metrics

5.2.1 Introduction

Too evaluate the performances of machine learning algorithms for classification, the Confusion Matrix is commonly used to present the situation between the true state and model prediction. Table 5.1 shows four different combinations of actual and predicted values in the Confusion Matrix.

Table 5.1 Confusion Matrix

		Predicted Values	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

In the Confusion Matrix, TP (True Positive) represents actual positive samples predicted as positive samples, FP (False Positive) represents actual negative samples predicted as positive samples, FN (False Negative) represents actual positive samples predicted as negative samples, and TN (True Negative) represents actual negative samples predicted as negative samples. In our research, the predicted samples were thousands of region proposals generated by Selective Search. The overlap between actual object area and the generated region proposal was the standard for obtaining each part of the Confusion Matrix, which is known as Intersection over Union (IoU). Figure 5.1 illustrates the idea of IoU, which is a ratio that specifies the amount of overlap between the ground truth and predicted bounding box defined as Eq. (5.1). In our research, the red box represents the actual pothole we labeled, while the blue box is one of the region proposals generated by the Region Proposal Network (RPN). It commonly measures the ground truth with an IoU threshold of 0.5. If the IoU is larger than the threshold, the detection would be considered as TP, otherwise as FP.

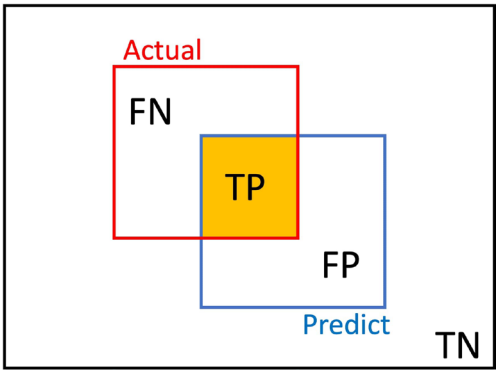

Figure 5.1 Illustration of Intersection over Union (IoU)

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{TP}{FN+TP+FP} \tag{5.1}$$

5.2.2 Performance Evaluation Metrics

Precision and recall are two commonly used metrics to evaluate the performance of machine learning algorithms. They are defined with the following equations:

$$Precision\sim(P) = \frac{TP}{TP+FP} \tag{5.2}$$

$$Recall\sim(R) = \frac{TP}{TP+FN} \tag{5.3}$$

The precision is the ratio of true positives to all predicted positives. In our research, the precision for each image is a percentage of the region proposals that we properly identified the potholes out of all region proposals in the image. The recall is a measurement of the model's ability to accurately identify the true positives. Take our research for example, for all the images with potholes in our dataset, the recall represents the percentage correctly identified as having potholes. Thus, the higher the recall, the higher the probability of predicting the actual pothole.

In general, the purpose of training a model is to maximize both precision and recall. However, it is difficult to compare the performances of models when the precision and recall conflict. For example, we cannot tell which model is the best if the precision of model A is higher than that of model B while the recall of model B is higher than that of model A. Therefore, the idea of the F1-score is to provide a sensitive indicator regarded as the harmonic mean of precision and recall. On the other hand, the F1-score can also be used as a composite metric to determine the performance of the model. A higher F1 score represents a better performance of the model. Eq. (5.4) is the definition of F1-score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.4}$$

5.3 Model Training and Testing

5.3.1 Training Details

We conducted the training, validation, and testing with our collected dataset. We randomly divided the dataset into a training set, validation set, and testing set with a ratio of 7:2:1. Therefore, the training dataset included 2,492 images, while the validation dataset included 712 images. Table 5.2 shows the number of images in each of the three datasets.

Table 5.2 Numbers of images in training, validation, and testing dataset

	Daytime	Cloudy	Nighttime	Total	%
Training	840	840	812	2,492	70
Validation	240	240	232	712	20
Testing	120	120	116	356	10
Total	1,200	1,200	1,160	3,560	100

In Method 1, we trained the Mask R-CNN model with a batch size of 4. The initial learning rate was set to 0.001 with a learning rate decay of 0.0001. In Method 2, we trained the RTFNet with a batch size of 2 (maximum batch size within the memory of our GPU). The starting learning rate was set to 0.01 with a learning rate decay of 0.0005, which is referring to the original RTFNet. In Method 3, we reduced the learning rate to 0.005 with a learning rate decay of 0.0005 to decrease the magnitude of the convergence to avoid gradient explosion. In all methods, the training datasets were shuffled before each epoch. Last, the training of the models stops when the loss has converged.

5.3.2 Transfer Learning

Transfer learning was utilized in all models in this research to improve the training speed and performance. Using this method, the pre-trained weights from the previous tasks could be reused in related works, and those weights provided a good starting point for training the machine learning models.

In the Mask R-CNN, we used the pre-trained weights based on the COCO dataset. The COCO dataset was created with the goal of image recognition. It included 80 categories (person, cars, etc) with over 200,000 images of the total 330,000 images labeled. In the RTFNet, we referred to the transfer learning method from Sun, et al. (2019). Likewise, we also used transfer learning in the RTFNet. The pre-trained weight we used to train the model is the weight of ResNet provided by PyTorch, which refer to the Sun et al. (2019) paper.

5.4 Detection Results and Comparison

5.4.1 Overall Results

First, we trained the model using full dataset (including daytime, cloudy, and nighttime), and tested the performance of this model in each condition. Table 5.3 shows the precision, recall, and F1-score of this model performed in daytime, cloudy and nighttime conditions. We tested the Mask-RCNN model only trained with visible images to compare the performances with/without thermal image fusion. We named it RGB in the table to represent that this model is trained by using only visible images of the dataset. ADF in this table represents the results from our proposed Method 1 (Mask R-CNN+ADF), while RTFNet represents the results from proposed Method 2.

Table 5.3 Performance comparison of three types of models in three scenarios (%)

	Daytime			Cloudy			Nighttime		
	RGB	ADF	RTFNet	RGB	ADF	RTFNet	RGB	ADF	RTFNet
Precision (P)	95.0	75.4	97.3	81.4	72.7	96.8	51.7	88.8	96.5
Recall (R)	92.5	71.7	94.3	75.5	70.0	90.5	85.0	88.3	63.2
F1-score	93.7	73.5	95.8	78.3	71.3	93.6	64.3	88.5	76.3

For daytime images, we achieved better performances from RGB and RTFNet than those from ADF. It is mainly because the visible images in the daytime already had enough information for accurate pothole detection. On the other hand, for ADF the redundant fusion of visible and thermal images could blur and increase noises of the pothole features. For RTFNet, the model is originally designed end-to-end for object detection, even though image fusion happened for daytime images, the RTFNet could still find the optimal results.

For cloudy images, the performances from RGB and ADF are worse than those from RTFNet. It is because puddles caused by non-potholes could be mistaken as potholes, and also emphasize the importance of non-pothole boundaries in the RGB and ADF, thereby affecting the prediction.

For nighttime images, thermal images provided more information than visible images. The ADF can enhance features from thermal images more than visible images because only thermal images provide information about pothole edges in the nighttime. However, because of the equal fusion weights of visible and thermal images in Method 2, the fusion of the less informative visible images led to low recall, which means the potholes in nighttime were hard to recognize. Therefore, the ADF performed the best in the nighttime with the highest F1-score because the ADF method can preserve the pothole edge information of thermal images.

Figure 5.2 shows sample results for RGB, ADF, and RTFNet in different scenarios. Column 1 shows images in daytime condition. The result shows the potholes could not be detected completely using ADF. In addition, the masks covered more fully of the potholes in RTFNet than others. Column 2 shows a sample in cloudy condition. Closer potholes could be predicted in all models. However, only RTFNet could identify potholes far from the camera (e.g., those located close to edges of the image) and be masked completely. Column 3 displays the samples in the nighttime condition. There are actually two potholes in this sample. The potholes can be detected in both ADF and RTFNet, while the performance of ADF is more accurate in this sample.

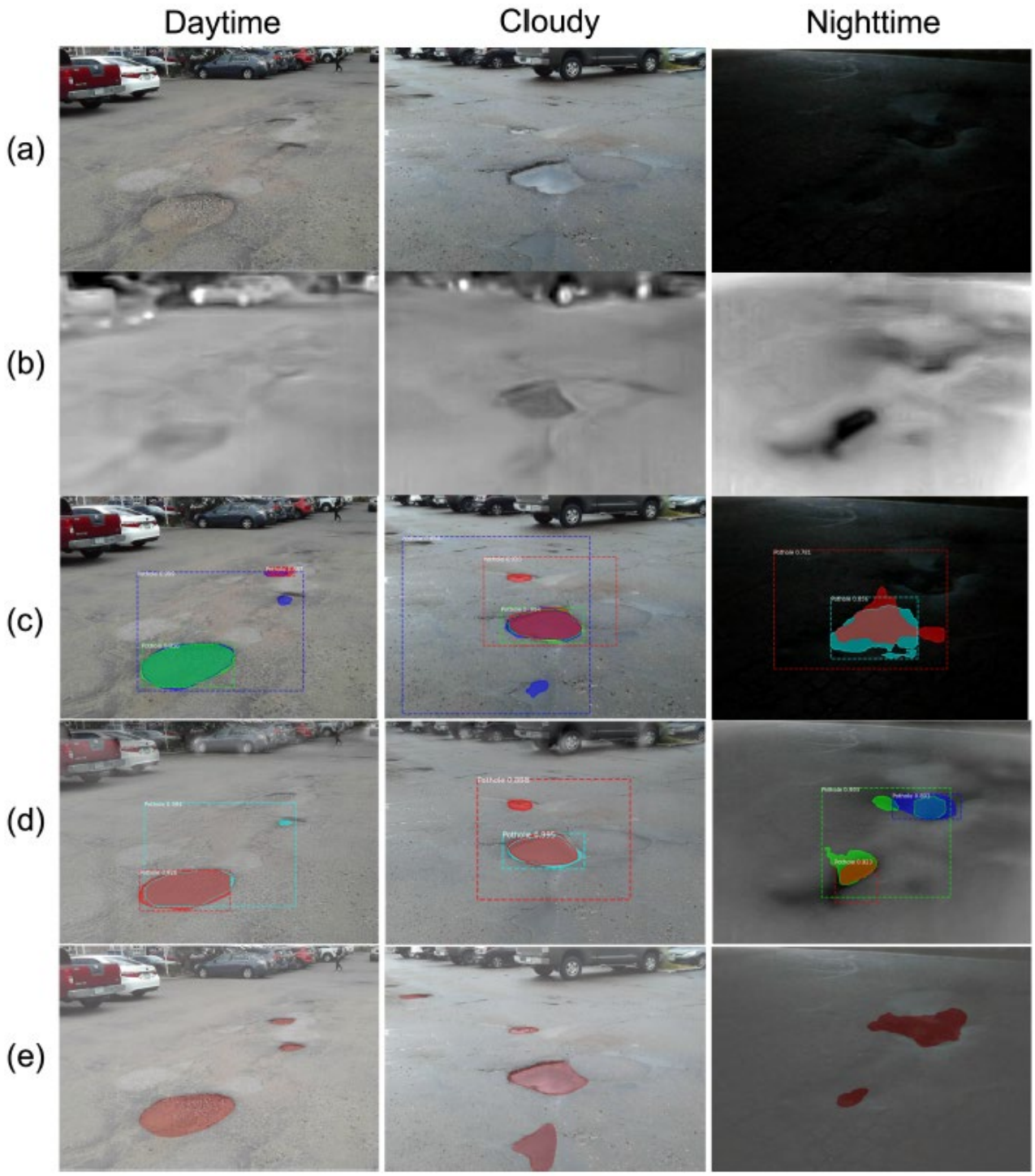

Figure 5.2 (a) Visible images, (b) Thermal images, (c) RGB, (d) ADF, (e) RTFNet

5.4.2 RTFNet + Constant EPs

According to the results above, we found RTFNet was more suitable for relatively bright scenarios than other models. The reason is the weights for visible and thermal images in the RTFNet are the same, so the importance of the thermal features could not be highlighted in the nighttime. Therefore, a major problem of the RTFNet is how to determine the best enhancement parameters (EP) to optimally emphasize the features in the thermal images. To evaluate the impact of different levels of enhancement on performance of the RTFNet model, we tested different values for the EPs. In this case, for an already trained RTFNet, we directly set the EPs for all the encoder layers in RTFNet as the same constant, while keeping other weights in the RTFNet unchanged. Then we used the model to do predictions. So, in this case, there was no training or retraining involved. Note we only tested for the nighttime condition.

Table 5.4 Impact of different level of enhancement (i.e., different EP values) on the performances of RTFNet under nighttime condition (%). Note that the RTFNet is trained under EP=1

	EP = 1	EP = 1.5	EP = 2	EP = 2.2
Precision (P)	96.8	90.2	80.2	77.3
Recall (R)	66.7	71.4	77.5	76.5
F1-score	78.9	79.7	78.8	76.9

Table 5.4 shows the testing results. The F1-score performs the best when EPs = 1.5. As EPs increase beyond 1.5, the F1-score was getting worse. Figure 5.3 displays the sample results with different EPs. When EPs=1, some area of the pothole can be predicted. The prediction of the pothole became clear and complete when increasing EPs up to 1.5. However, the thermal image was over-emphasized when continuously increasing EPs. The edge of predictions became rougher, and the locations of pothole became inaccurate as well .

Therefore, these initial investigations showed the values of EPs in the RTFNet could affect the prediction performance, and it is important to select optimal EPs in the RTFNet to optimally enhance the thermal features to improve the performance of the RTFNet in pothole detection.

5.4.3 RTFNet Trained with Constant EPs

In the previous case, the weights used in the RTFNet model were essentially trained for EPs=1 (i.e., without any enhancement). The performance variations for different EPs may be because we did not use the corresponding optimal weights for different EPs. Therefore, to gain more insights on how the EPs affect the performance of the RTFNet model, we considered another case. In this case, we first set the EP values, and then trained the model to find the corresponding optimal weights for other parameters in the RTFNet model and used the established model for prediction. Here, we trained the RTFNet with EPs equal to the constant of 1, 1.2, and 1.5 to test which EPs were the best for the nighttime scenario. In addition, we trained two RTFNet models. One was only changing the EP in the first encoder layer while keeping EPs at other layers as 1. The other was setting the EPs in all encoder layers as the same constant (i.e., 1, 1.2, 1.5). The reason for this was to observe whether the latter would over-enhance the thermal images. During the training process, we used Bright-Dark detector to classify the bright and dark images. If the detector determined the training data as the dark images (e.g., nighttime image), then constant EPs would be involved; otherwise, all EPs would be set to 1.

Table 5.5 shows the results. The overall performances of changing EPs in all layers are better than only changing EP in the first layer. When looking at the case of EP=1.5, the improvement in recall is particularly significant. Figure 5.4 shows some sample pothole detection results. In visible images, the light dims from left column to right column. The predictions in case (b) are more accurate than in case (a). The dimmer the light,

the more significant the difference in prediction completeness between (a) and (b), which can more clearly show that the recall of case (b) is higher than that of case (a).

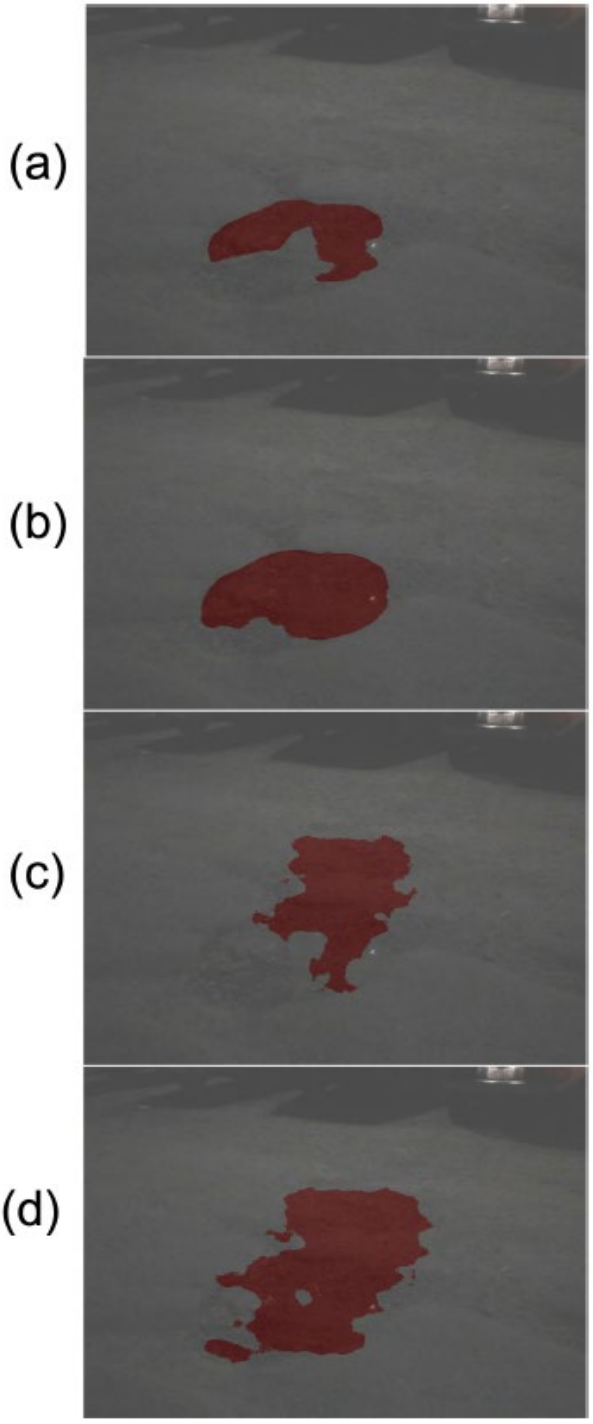

Figure 5.3 Sample results in nighttime for RTFNet with different EPs (a) EPs=1, (b) EPs=1.5, (c) EPs=2, (d) EPs=2.2

Table 5.5 Impact of different levels of enhancement (i.e., different EP values) on the performances of RTFNet under nighttime condition (%). Note that the RTFNet is trained under corresponding EPs

	First layer			All layers		
	EP = 1	EP = 1.2	EP = 1.5	EP = 1	EP = 1.2	EP = 1.5
Precision (P)	96.3	95.8	95.7	96.3	92.7	83.3
Recall (R)	71.2	70.8	63.1	71.2	76.5	79.9
F1-score	81.9	81.5	76.1	81.9	83.8	81.6

Figure 5.4 Sample results in nighttime for RTFNet, shown for the EP=1.2. (a) EP=1.2 in the first layer, (b) EP=1.2 in all layers

5.4.4 RTFNet Trained with Variable EPs for Each Layer

Based on the results from previous sections, if the pothole detection only focuses on the nighttime scenario, the following insights can be obtained:

1. Adding EPs to enhance the thermal feature for nighttime images can positively affect the prediction of potholes.
2. Training the RTFNet model with their corresponding EPs can increase the recall, which means it is easier to detect the presence of potholes.
3. It seems that the optimal value for EPs is somewhere between 1 to 1.5 for considering EPs in all encoder layers in the RTFNet.

To find the optimal EPs, we changed EPs from constants to trainable variables for each encoder layer. That is, we include the EPs as parameters/weights that will be trained together with other weights in the model. Figure 5.5 shows the graph of convergence process of each EP. The subscript number represents the encoder layer. We initially set the start point at 1.5 because the previous results showed the best range of EPs might be from 1 to 1.5. However, it seems all five EPs are still converging when they reached 1.3 after initial training, so we reset the start point at 1.3 to run more steps to eventually establish the converged values. The Bright-Dark detector was used the same way as it was used when training the model with constant EPs, that is, EPs would return to 1 when the Bright-Night detector regarded the training data as the bright image, otherwise, the EPs would continuously keep training until convergence.

We also trained a model containing trainable EP using only the nighttime dataset trying to train the best model for the nighttime scenario. The convergence trend of EPs is shown in Figure 5.6. The final EPs trained with the full dataset and nighttime dataset are listed in Table 5.6. Overall, the EP values are roughly around 1.1 with some variations between the different layers.

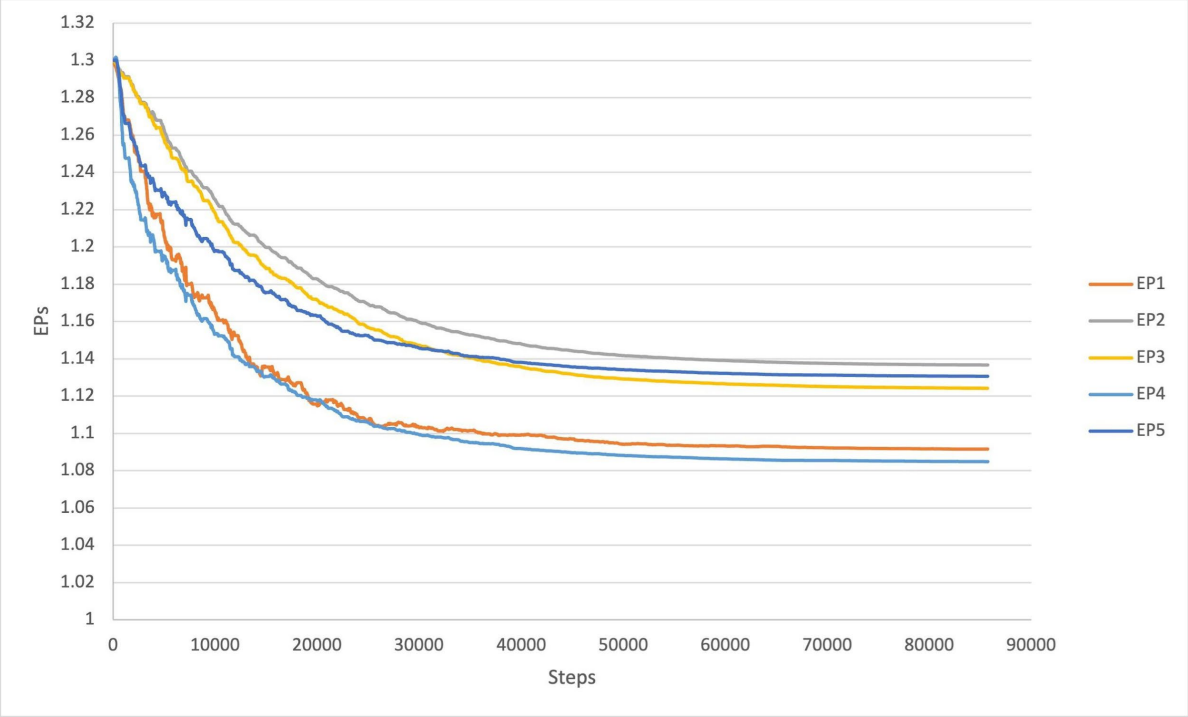

Figure 5.5 Convergence of EP values trained with the full dataset

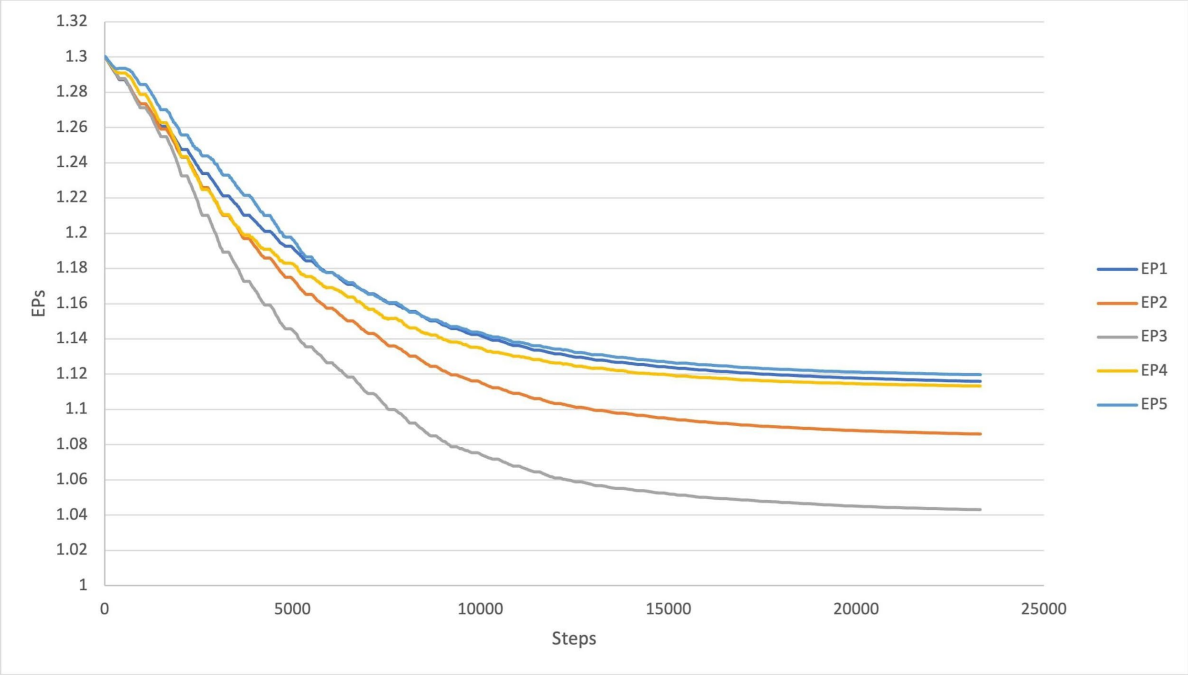

Figure 5.6 Convergence of EP values trained with the nighttime dataset

Table 5.6 The results of trainable EPs

-	EP_1	EP_2	EP_3	EP_4	EP_5
Train with full dataset	1.092	1.137	1.124	1.085	1.131
Train with nighttime dataset	1.116	1.086	1.043	1.113	1.119

Table 5.7 shows the performances of RTFNet trained with variable EPs. There were two cases: one trained with the full dataset, and the other trained with only nighttime dataset. In the case of training with full dataset, the model achieved a better balance of precision and recall than those models with constant EPs in three scenarios. In the case of training with only the nighttime dataset, the model achieved the highest precision of 94.7% and recall of 87.4% in the nighttime scenario. However, this model is not suitable for use in the daytime and cloudy conditions.

Figure 5.7 shows the comparison of EPs being constants 1.2 in all encoder layers and EPs being trainable variables. The thermal images are more informative than the visible images in these samples. We found potholes in these samples could be completely predicted and fully masked when EPs were trainable variables. In comparison, some potholes cannot be detected and marked out when EPs are the constants of 1.2.

Table 5.7 Performances of RTFNet trained with variable EPs (%)

	Trained with full dataset			Trained with nighttime dataset		
	Daytime	Cloudy	Nighttime	Daytime	Cloudy	Nighttime
Precision (P)	90.0	87.8	83.0	91.6	60.4	94.7
Recall (R)	84.0	79.2	84.0	56.7	50.9	87.4
F1-score	86.9	83.3	83.5	70.1	55.3	90.9

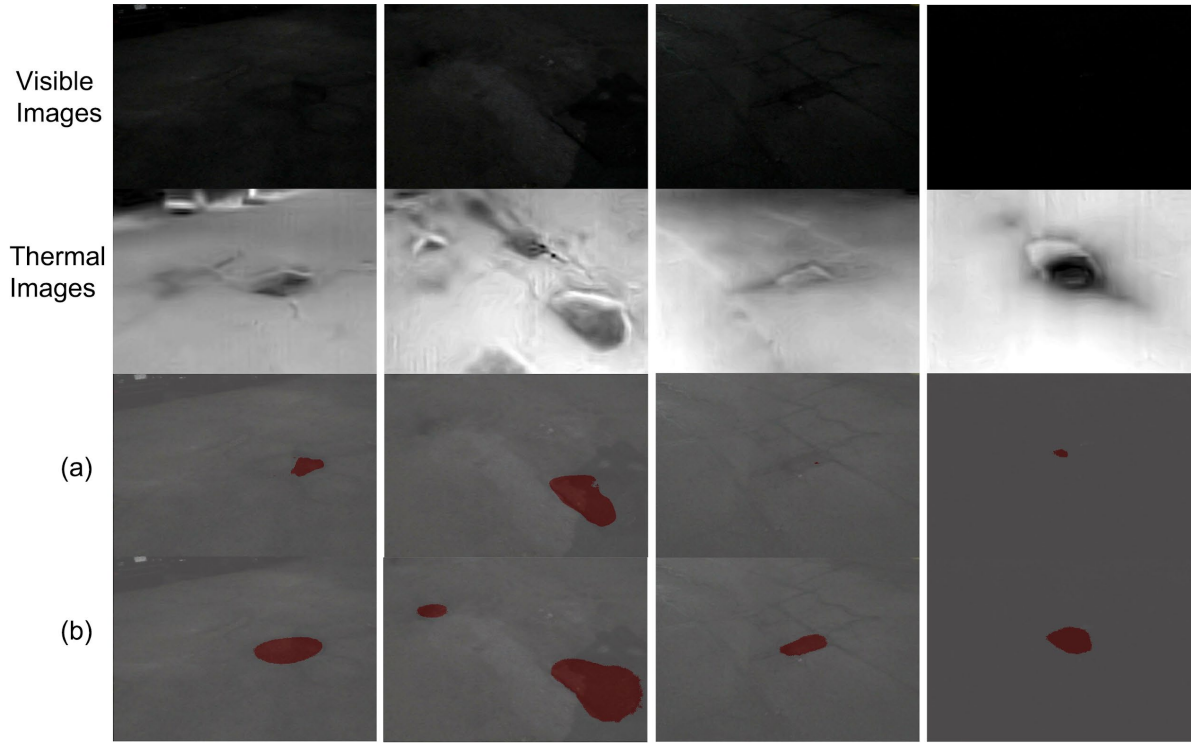

Figure 5.7 Sample results in the nighttime condition for RTFNet trained with 2 cases, (a) EPs=1.2 trained with full dataset in all layers, (b) EPs trained/optimized with nighttime dataset in all layers

5.5 Summary of Results

We experimented with three proposed methods and compared their performance in different scenarios. Overall, the RTFNet takes advantage of the end-to-end process of image fusion and object segmentation so that the model can optimize based on the information from visible and thermal images. Compare to Method 1, the ADF and Mask R-CNN is two independent algorithms so the modification and application of the subsequent model will not be as good as RTFNet. Besides, we used a Bright-Dark detector to split the dataset into bright and dark images based on the lighting condition — this can help RTFNet find the optimal weight in every lighting condition for pothole detection. It is worth noting we used trainable EPs to enhance thermal features in dark scenario, which was proved to significantly improve the model's performance at nighttime, according to experiments.

In summary, we decided to use the RTFNet as the model of the pothole detection tool. We also added the Bright-Dark detector to classify input images as a basis for using EPs. We will introduce the tool in detail in the next chapter.

6. DEVELOP AUTOMATED TOOLS FOR POTHOLE DETECTION AND MAPPING

6.1 Introduction

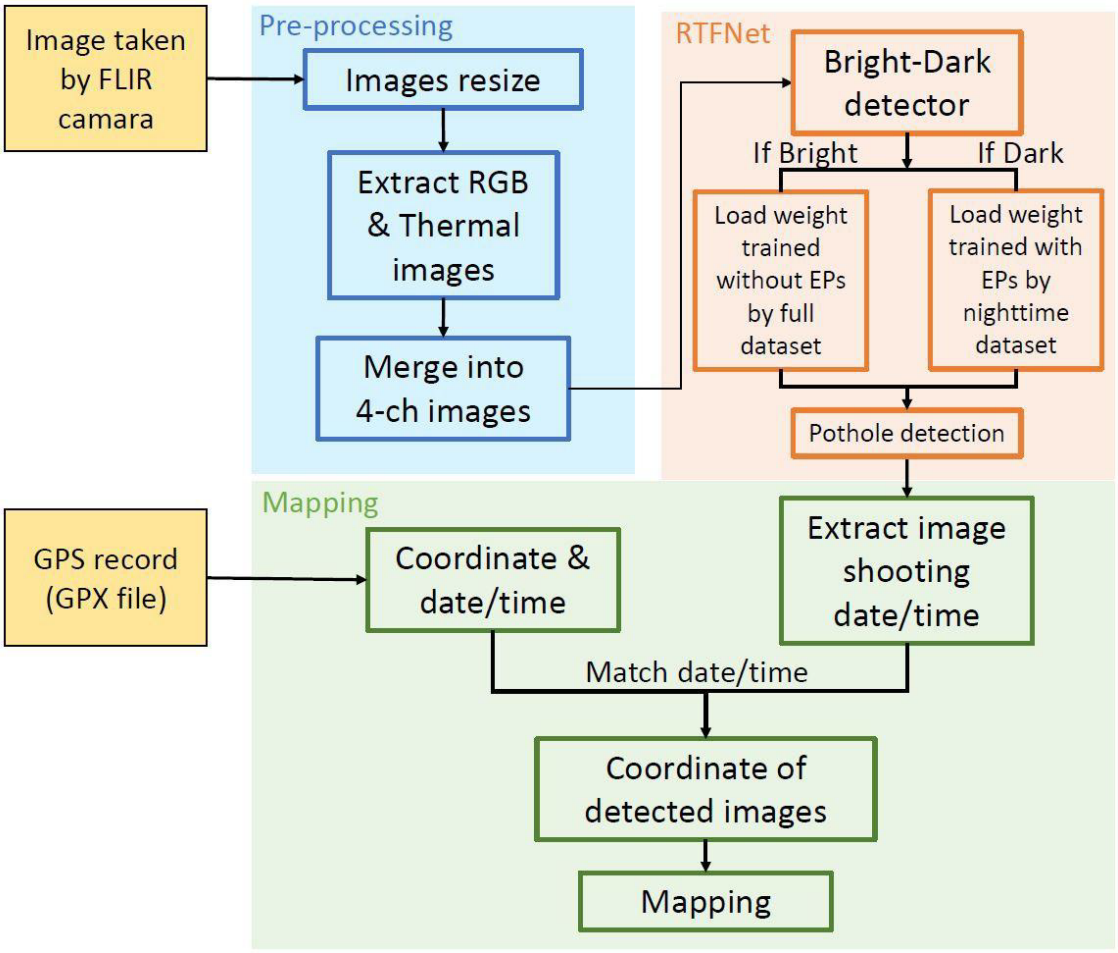

Figure 6.1 The procedure of the developed pothole detection and mapping tool

After obtaining the training models in the previous chapter, we developed a pothole detection and mapping tool with a graphical user interface (GUI) based on the RTFNet model to automatically and robustly detect potholes by fusing visible and thermal images to prioritize road maintenance. Figure 6.1 shows the procedure of the developed pothole detection tool. The procedure is divided into three main parts: Pre-processing, Pothole detection using RTFNet model, and Mapping. This chapter will discuss in detail the process of the detection and mapping of potholes with the developed tool.

6.2 Inputting Data

Figure 6.2 shows the graphical user interface of the developed tool. The inputs are the folder/path of the candidate images and the file path of the GPX file created from the GPS collector. For input images, they were directly captured by the FLIR Visible-Thermal camera with visible and thermal sensor data embedded in

the JPG metadata. As for the GPS, because the FLIR camera doesn't have the ability to update the coordinates of images while shooting in different locations, the GPS collector, which can record the coordinates, dates and times, must be turned on when collecting pothole images.

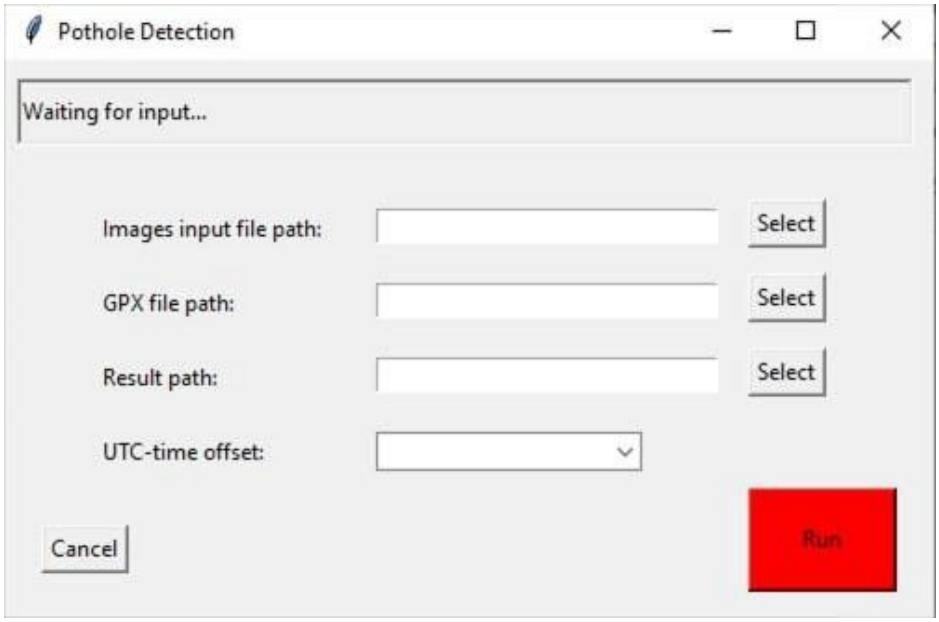

Figure 6.2 The graphical user interface of the developed pothole detection tool

A GPX file is a GPS data file saved in the GPS Exchange format, containing position data in longitude and latitude and time data. GPX file is an open standard used by many GPS programs, so it is acceptable to use, regardless of the type of third-party GPS collector. It is noteworthy that the times in the GPX file are in Universal Coordinated Time (UTC), which unified the standard of the recording of time. Therefore, the date and time in the GPX file need to be converted to the corresponding location of input images. For example, the time has to be minus seven hours to correspond to the Denver time zone, where our data is collected.

After inputting the folder path of images, the file path of the GPX file, the path you want to store the detecting and mapping results, and adjusting the UTC-time offset based on the location of the input collection, then one can simply click the "Run" button to start the detection process.

6.3 Pre-processing Process

When the display bar shows "Processing" on the top of the interface, it means the detection has started. In the pre-processing part, we standardized the image pre-processing process used for establishing the training dataset. The size of images will be resized to 640x448 to correspond to the default size of the training dataset. Because the input images from the FLIR camera are JPG files with thermal sensor data embedded, we needed to extract thermal information and divide it into 3-channel visible and 1-channel thermal images by the FLIR Image Extractor. The names of the visible and thermal images were renamed according to the date and time they were taken plus the UTC-time offset, so the shooting time of the images could be easily matched with the corresponding coordinates in the mapping part. In the final step of pre-processing process, the extracted visible and thermal images were merged into 4-channel images as the input to the RTFNet model.

6.4 RTFNet Detection

Before inputting 4-channel candidate images into the RTFNet, the Bright-Dark detector is added to distinguish bright or dark conditions and divide the images into two groups. The differentiation of bright and dark images is according to the brightness of the visible images. The detection of bright and dark images were separated because the best weight (and hence the best model) for each condition is different. First, the RTFNet processes bright images with the weight trained by Method 2 because the F1-score of this weight was around 95% for daytime pothole detection. This process would be skipped if there is no input detected as bright images. Next, the RTFNet processed the dark images with the weight established from training with only the nighttime dataset in Method 3. The reason is that the F1-score of this weight was around 90% and was the best performance compared to other methods. The trained Enhanced Parameters (EPs) was involved in five encoder layers of the RTFNet to emphasize the feature of thermal images. The values of EPs in each layer were the converged ones shown in Table 5.6. Similarly, this process would be skipped automatically if there is no dark image. After the detection process by the RTFNet, only images with detected potholes were masked and saved as the detection results.

6.5 Mapping

The FLIR mobile app has the function of recording the image's shooting time and coordinates. However, we found it can only record the coordinate of the first image. In other words, the app cannot update the coordinate when the camera is moving. Thus, we used the "Gaia GPS," a third-party mobile app that simultaneously records the times and coordinates when using the FLIR camera. Other GPS recorders that have the ability to export the GPX file can also be used. Because we renamed the input images by their own shooting time in the pre-processing phase, we could easily match the detected results with their shooting time from the GPX file and match them with the corresponding coordinates.

We used the Folium python package to map the detected potholes. The output is an HTML file, the standard markup language for documents designed to be displayed in a web browser, stored in the result path that had been inputted at the beginning. Figure 6.3 displays the mapping results with ± 16 ft GPS accuracy. The GPS accuracy depended on the performance of the GPS recorder. The higher the refresh rate was, the higher the resolution will be. The locations of the detected potholes were marked with blue pins. Moreover, the detected pothole image showed up when clicking on the pin.

6.6 Applications and Limitations

This tool can be used to facilitate pothole detection and pothole mapping for highway maintenance. The candidate images are directly obtained from the FLIR ONE camera without any additional operations. For the GPX file, it can be generated from any GPS recorder because it is a typical format for recording the coordinate and date/time information. Besides, this tool not only can detect potholes in bright conditions but also in insufficient lighting conditions. Being able to detect potholes even in conditions with insufficient lighting can be very useful for pavement maintenance. For example, because of the traffic during the daytime, many times the pavement maintenance is carried out during the nighttime. This tool has the ability to detect potholes in the dark scenario, so it can also work in the nighttime or in general low lighting conditions to facilitate road maintenance.

When implementing the tool, there are several aspects to keep in mind. First, depending on how the camera is mounted and the camera angle, some of the potholes may be difficult to detect. For example, potholes captured close to the edge of the images may show up as small potholes with a distorted shape because of the view angle. We believe that this can be solved by collecting more data and doing the data augmentation with perspective transformation to consider more pothole shapes and types and account for the view angle. Second, because of the fixed frame rate of the FLIR ONE camera (five images per second for our data collection), some road segments may be missed (e.g., no images collected) if driving too fast, while driving too slow may lead to multiple images for the same road segment or pothole (which may lead to overestimation of the number of potholes, since the same pothole may show up in multiple images). We think this problem can be solved by accounting for the relationship between the vehicle speed and camera frame rate to achieve the best coverage of road surface and proper identification of the correct number of potholes.

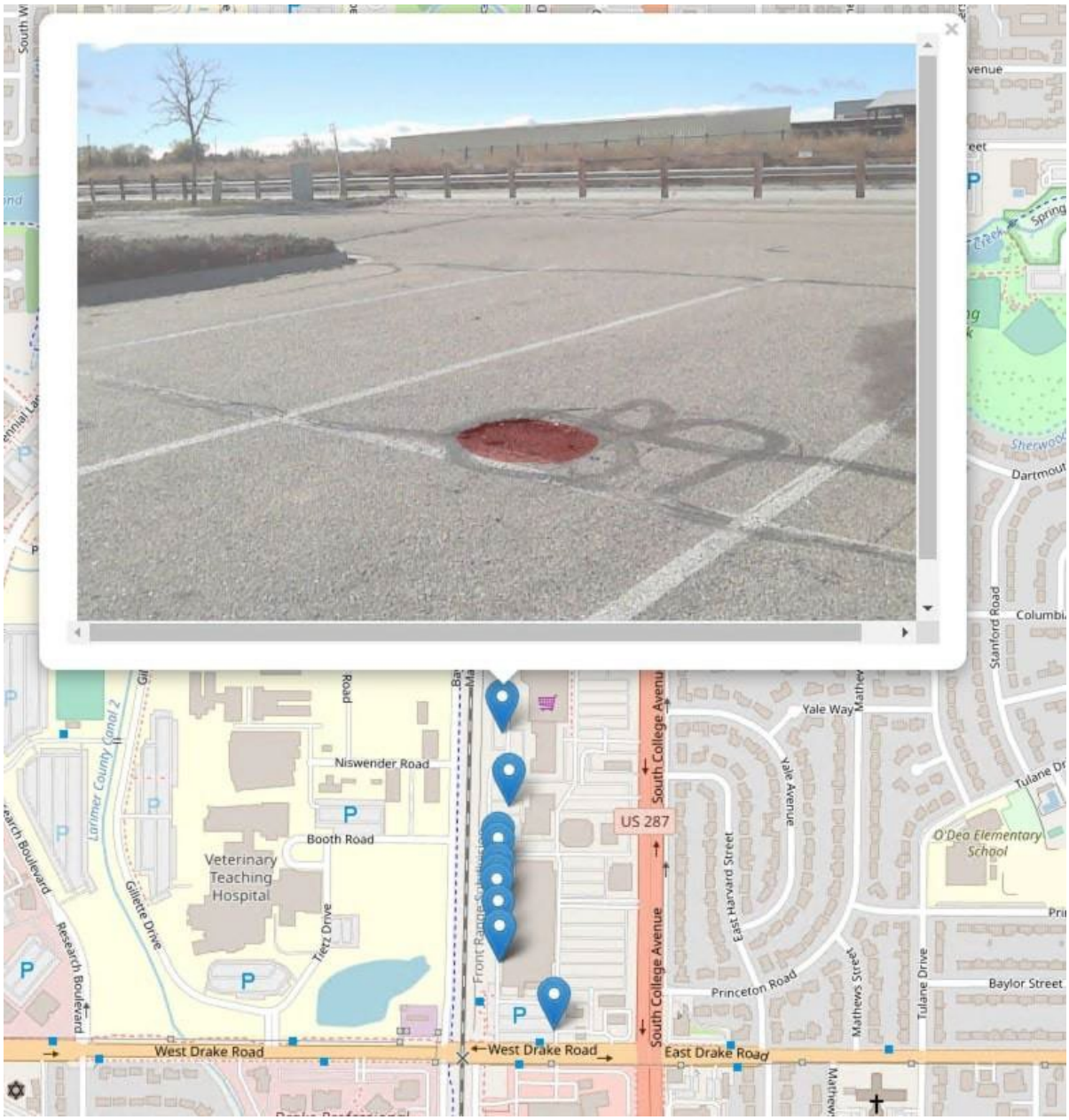

Figure 6.3 Sample result of mapping the detected potholes

7. CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Summary

In this report, the overall objective was to utilize machine learning algorithms for the automated and robust detection of potholes and develop a pothole detection and mapping tool that can be used to help prioritize highway maintenance.

Chapter 2 provided an overview of some of the commonly used machine learning (ML) algorithms for objective detection and pothole detection. First, convolutional neural networks (CNN) and region-based convolutional neural networks (R-CNN) are reviewed. Aspects related to the structure of CNN and R-CNN, the evolution of R-CNN models, and the critical elements in the Mask R-CNN. In addition, this chapter provided literature reviews on the use of machine learning algorithms in pothole detection. Through the descriptions, we highlighted improvements from fast R-CNN to Faster R-CNN and the additional primary function to differentiate the faster R-CNN and Mask R-CNN. The Mask R-CNN algorithm was chosen in this research for pothole detection and segmentation.

Chapter 3 presented the three methods we proposed for pothole detection by fusing visible and thermal images. First, we introduced the structure and principle of Anisotropic Diffusion Fusion (ADF), which is an image fusion method for emphasizing the feature maps (e.g., edges, lines). We fused our dataset with ADF before running the Mask R-CNN model, and we regarded this as our proposed Method 1. Moreover, we introduced the RGB-Thermal Fusion Network (RTFNet), which is an end-to-end fusion and segmentation encoder-decoder model. Utilization of the RTFNet is our proposed Method 2. In Method 3, we explicitly introduced the enhancement parameters (EPs) in the encoder of the RTFNet to enhance the thermal feature maps. Meanwhile, we developed a Bright-Dark detector to distinguish whether the input images are in the daytime or nighttime. Then, depending on it is daytime or nighttime, the corresponding RTFNet will be used for pothole detection.

Chapter 4 introduced the procedures for establishing the annotated fused image dataset. The procedures included data collection, data pre-processing, data annotations, and data augmentations. The image fusion processes would vary, depending on the requirements of the three methods we proposed in Chapter 3.

Chapter 5 presented the training processes and compared the performances of the three proposed methods. The dataset was split into training, validation, and testing datasets. We trained models for each method where transfer learning was used to speed up the training process and improve the training efficiency. We evaluated the performance of each model with the F1-score and selected the one with better performance as the backbone of the pothole detection and mapping tool we developed.

Chapter 6 introduced the function, procedure, and usage of the developed pothole detection and mapping tool. Based on performance investigations in Chapter 5, the RTFNet and RTFNet with EPs were selected as the backbone models of this tool. A graphical user interface (GUI) for the tool was also developed. The GUI made the tool easier and more intuitive to use. The output from the tool was a map marked with the locations of detected potholes. Also, the photo with masked potholes shows when clicking the location icon on the map. The map could help maintenance team to visualize where the potholes are and to manage/plan the repair of the potholes.

7.2 Conclusion

Through this research, a unique dataset consisting of trios of visible, thermal and fused images are established. Three methods are proposed for pothole detection using image fusion. Through comparing the performances of the three proposed methods, comprehensive insights were obtained on the impact of thermal images on pothole detection. Overall, it was found that fusing thermal images can help improve the pothole detection performance of the ML algorithms, especially in the nighttime scenario. We were able to get the F1-score value of 93% in the daytime scenario with the use of the Mask R-CNN and the highest F1-score value of 88.5% in the nighttime with the Mask R-CNN + ADF model. However, we got the best performance of the F1-score value of 95.8% and 93.6% with the RTFNet in the daytime and cloudy, respectively. In addition, we introduced five new variables, Enhancement Parameters (EPs), to the five encoder layers in the RTFNet to emphasize/enhance the features of thermal images, especially in the nighttime. Eventually, we got the best F1-score value of 90.9% by the modified RTFNet when focused on only the nighttime images.

An automated pothole detection and mapping tool with graphical user interface was developed. The inputs to the tool were candidate images, which can be directly imported from the FLIR One visible and thermal camera, and the GPX file, which can store the time and GPS records. In addition, the Bright-Dark detector was developed and integrated in the overall algorithm. The Bright-Dark detector can recognize the daytime or nighttime scenarios of the candidate images. The daytime images were predicted with the original RTFNet, while the nighttime images were predicted with the RTFNet with EPs. The output is a map with locations of the detected potholes. When clicking on the pin on the map, the pothole image with mask will show up.

7.3 Future Directions

In the development of pothole detection by visible and thermal image fusion and the mapping tool, there are some key recommendations that can be considered for future research work.

1. Due to the lack of potholes on Fort Collins' roads and highways, some pothole images in our proposed dataset were collected from the parking lot. The difference of environment conditions between highway images and parking lot images may impact the detection accuracy. For example, traffic light and signs appear on highways while parking lots do not. Because the tool we developed is to help highway maintenance, the pothole data can be collected only on highways in the future to focus on the highway scenario.

2. In this research, we collected daytime, cloudy, and nighttime data for model training. However, the temperature difference between potholes and road surfaces may vary depending on the weather/season. Future research may look into the influence of thermal image fusion on model prediction accuracy at varying temperature differences.

3. The computation of the pothole area had been done in the work by Arjapure & Kalbande (2021). The method they used was to calculate the error or deviation of the number of pixels between the predicted pothole area and the actual pothole area. This method can be combined with our developed pothole detection and mapping tool to obtain more detailed pothole information (e.g., size, shape, area, depth, etc.) for highway maintenance.

8. REFERENCES

AAA (2016). Pothole damage costs drivers $3 billion annually nationwide. URL: https://news. aaa-calif.com/news/pothole-damage-costs-drivers-3-billion-annually-nationwide.

Ahmed, K. R. (2021). Smart pothole detection using deep learning based on dilated convolution. *Sensors*, *21*, 8406.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)* (pp. 1–6). IEEE.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, 1–74.

Arjapure, S., & Kalbande, D. (2021). Deep learning model for pothole detection and area computation. In *2021 International Conference on Communication Information and Computing Technology (ICCICT)* (pp. 1–6). IEEE.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2021). Deep learning-based road damage detection and classification for multiple countries. *Automation in Construction*, *132*, 103935.

Bavirisetti, D. P., & Dhuli, R. (2015). Fusion of infrared and visible sensor images based on anisotropic diffusion and karhunen-loeve transform. *IEEE Sensors Journal*, *16*, 203–209.

Bergerson, E. (2022). Sdot blog. URL: https://sdotblog.seattle.gov/2022/01/14/ we-filled-15000-potholes-in-2021-but-winter-storms-are-wreaking-havoc-on-our-roads/.

Bhatia, Y., Rai, R., Gupta, V., Aggarwal, N., & Akula, A. (2022). Convolutional neural networks based potholes detection using thermal imaging. *Journal of King Saud University-Computer and Information Sciences*, *34*(3), 578-588.

Fan, R., Ozgunalp, U., Hosking, B., Liu, M., & Pitas, I. (2019). Pothole detection based on disparity transformation and road surface modeling. *IEEE Transactions on Image Processing*, *29*, 897–908.

Girshick, R. (2015). Fast r-cnn. URL: https://arxiv.org/abs/1504.08083. doi:10.48550/ARXIV. 1504.08083.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, *157*, 322–330.

Gupta, S., Sharma, P., Sharma, D., Gupta, V., & Sambyal, N. (2020). Detection and localization of potholes in thermal images using deep neural networks. *Multimedia tools and applications*, *79*, 26265–26284.

Ha, Q., Watanabe, K., Karasawa, T., Ushiku, Y., & Harada, T. (2017). Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5108–5115). IEEE.

Hazirbas, C., Ma, L., Domokos, C., & Cremers, D. (2017). Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision* (pp. 213–228). Springer.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. URL: https://arxiv.org/abs/1703.06870. doi:10.48550/ARXIV.1703.06870.

Hou, J., Zhang, D., Wu, W., Ma, J., & Zhou, H. (2021). A generative adversarial network for infrared and visible image fusion based on semantic segmentation. *Entropy*, *23*, 376.

Kulkarni, A., Mhalgi, N., Gurnani, S., & Giri, N. (2014). Pothole detection system using machine learning on android. *International Journal of Emerging Technology and Advanced Engineering*, *4*, 360–364.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 1127–1141.

Majidifard, H., Jin, P., Adu-Gyamfi, Y., & Buttlar, W. G. (2020). Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. *Transportation Research Record*, *2674*, 328–339.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 629–639. doi:10.1109/34. 56205.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. URL: https://arxiv.org/abs/1506.01497. doi:10.48550/ARXIV.1506.01497.

Sathya, R., & Saleena, B. (2022). Cnn-mao: Convolutional neural network-based modified aquilla optimization algorithm for pothole identification from thermal images. *Signal, Image and Video Processing*, (pp. 1–9).

Song, H., Baek, K., & Byun, Y. (2018). Pothole detection using machine learning. *Advanced Science and Technology*, (pp. 151–155).

Sun, Y., Zuo, W., & Liu, M. (2019). Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes. *IEEE Robotics and Automation Letters*, *4*, 2576–2583.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*, 154–171. URL: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013.

USDT (2022). Road condition. URL: https://www.bts.gov/road-condition.