University of Rhode Island

DigitalCommons@URI

Open Access Dissertations

2019

COMPUTER VISION BASED METHOD FOR AUTOMATIC REBAR DETECTION IN GROUND PENETRATING RADAR DATA

Pouria Asadi

University of Rhode Island, pasadi2008@gmail.com

Follow this and additional works at: https://digitalcommons.uri.edu/oa_diss

Recommended Citation

Asadi, Pouria, "COMPUTER VISION BASED METHOD FOR AUTOMATIC REBAR DETECTION IN GROUND PENETRATING RADAR DATA" (2019). *Open Access Dissertations*. Paper 1111.
https://digitalcommons.uri.edu/oa_diss/1111

This Dissertation is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

COMPUTER VISION BASED METHOD FOR AUTOMATIC
REBAR DETECTION IN GROUND PENETRATING RADAR DATA

BY

POURIA ASADI

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

CIVIL AND ENVIRONMENTAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2019

DOCTOR OF PHILOSOPHY DISSERTATION

OF

POURIA ASADI

APPROVED:

Dissertation Committee:

Major Professor Mayrai Gindy

George Tsiatis

Marco Alvarez

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2019

ABSTRACT

Statistics by the Federal Highway Administration (FHWA)'s report indicates that 11% of the bridges in the United States are rated as "structurally deficient" and over 30% of existing bridges have exceeded their 50-year design life, meaning that condition assessment and repair programs will require substantial budget in the near future. Current observation-based bridge inspection techniques consist largely of time consuming and subjective measures for quantifying deterioration of bridges. During bridge inspection, simplistic methods for assessing deterioration in concrete bridge decks are only capable of detecting deterioration in its moderate to severe stages. To provide a more thorough assessment of deterioration in concrete bridge decks, advanced technologies should be incorporated into bridge inspection. Some advanced nondestructive testing methods such as Ground Penetrating Radar (GPR), are being implemented that provide sub-surface information. GPR has been successfully used in a wide range of applications. Using advanced technologies like ground penetrating radar (GPR), deterioration hidden from the naked eye or missed using traditional methods, like Chain Dragging and Hammering Sounding, can be more accurately detected.

Automatic rebar detection in GPR data is the basic step in an automatic system for GPR-based condition evaluation of bridge decks. Achieving real-time performance on Acorn Reduced Instruction Set Computing Machine (ARM) based platforms for onsite applications still remains a challenge. Development an accurate and cost-effective system for real-time onsite rebar detection in GPR images is goal of this study. The authors proposed a novel computer vision-based method for automatic detection of rebars in

complex GPR images in highly deteriorated concrete bridge decks. Extensive experiment performed to develop a reference for selecting a deep learning-based detection architecture that provides the right accuracy, speed, and memory usage balance for real-time detection of rebars on the latest version of ARM-based platforms. A deep learning-based detector is presented that can be deployed on the latest version of ARM-based platforms. State of the art results is obtained on GPRDETNet detection task by implementing rebar detection model using Faster R-CNN with ResNet 101 CNN backbone.

ACKNOWLEDGEMENT

This work would not have been possible without the financial support of University of Rhode Island, Department of Civil and Environmental Engineering, and Rhode Island Department of Transportation (RIDOT). I am grateful to all of those with whom I have had the pleasure to work during this project. Each of the members of my Dissertation Committee has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general. I would especially like to thank Dr. Mayrai Gindy, my advisor. As my teacher and mentor, she has taught me more than I could ever give her credit for here. She has shown me, by her example, what a good scientist (and person) should be. I would like to express my gratitude to my committee members, Prof. George Tsiatis and Prof. Marco Alvarez. I would like to thank Kelly Wharton for providing the GPR scans and conducting all the field work.

Nobody has been more important to me in the pursuit of this research than the members of my family. I would like to thank my mother, father, and brothers; whose love and guidance are with me in whatever I pursue.

Pouria Asadi

December 12th, 2019

PREFACE

Ground penetrating radar (GPR) has been increasingly used to locate and assess the condition of reinforcing steel in concrete bridge decks. Rhode Island Department of Transportation (RIDOT) recently invested in a GPR system from Geophysical Survey Systems, Inc. that has been used to scan several concrete bridges around the State. The technology is proving to be a useful tool to assess subsurface condition of bridges. The objective of this research is to develop a method for automatic detection of bridge deck rebars from GPR images. This dissertation is prepared in Manuscript format.

In chapter 1, The authors proposed a novel machine learning based processing for automatic interpretation and quantification of concrete bridge deck GPR B-scan images. The proposed method is based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration in concrete bridge decks. For the first time, the authors introduced a dataset of 4,000 B-scan images cropped from real bridge deck GPR field data, named DECKGPRH1.0.

In chapter 2, Rebar detection in bridge deck Ground Penetrating Radar (GPR) B-scans using Convolutional Neural Networks (CNN) for on-site applications is addressed. The authors investigated accuracy/frame rate trade-off of modern deep learning detection methods for automatic rebar detection in GPR B-scans on Raspberry Pi 3 Model B+.

In chapter 3, The authors proposed a novel computer vision-based method for automatic detection of rebars in complex GPR images in highly deteriorated concrete

bridge decks. The proposed detection model consists of a fine-tuned Histogram of Oriented Gradients feature descriptor, a Multi-Layer Perceptron for classification, and a post processing algorithm for eliminating false detections and labeling rebar in Region of Interest.

In chapter 4, extensive experiment performed to develop a reference for selecting a deep learning-based detection architecture that provides the right accuracy, speed, and memory usage balance for real-time detection of rebars on the latest version of ARM-based platforms, named: Raspberry Pi 4 Model B. Various ways to trade accuracy for speed and memory usage in convolutional neural network (CNN)-based detection models is investigated. A unified implementation of the Faster R-CNN and SSD meta-architecture-based models is implemented to evaluate the accuracy, speed, and memory usage trade-off by using various CNN backbones and varying other training parameters. A deep learning-based detector is presented that can be deployed on the latest version of ARM-based platforms.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iv
PREFACE	v
TABLE OF CONTENTS	vii
LIST OF FIGURES	viii
LIST OF TABLES	xii
MANUSCRIPTS	
CHAPTER 1: A MACHINE LEARNING BASED APPROACH FOR AUTOMATIC REBAR DETECTION AND DETERIORATION MAPPING IN BRIDGE DECK GROUND PENETRATING RADAR DATA	1
CHAPTER 2: MODERN CONVOLUTIONAL NEURAL NETWORKS FOR REBAR DETECTION IN BRIDGE DECK GPR B-SCANS ON MOBILE AND EMBEDDED SYSTEMS	31
CHAPTER 3: A COMPUTER VISION BASED REBAR DETECTION CHAIN FOR AUTOMATIC PROCESSING OF CONCRETE BRIDGE DECK GPR DATA	60
CHAPTER 4: REAL-TIME ONSITE PROCESSING OF GPR DATA ON ARM- BASED MOBILE DEVICES USING MODERN CONVOLUTIONAL NEURAL NETWORKS	100
APPENDIX: LITERATURE REVIEW	127

LIST OF FIGURES

CHAPTER 1

Figure 1 The flowchart of the proposed method for concrete bridge deck GPR data processing and quantification.	6
Figure 2 EM wave penetration depth and wavelength for concrete.	8
Figure 3 GSSI SIR System-3000 and a 1.6GHz antenna mounted on a cart GPR system mounted on a cart used for data collection in this study.	9
Figure 4 Sample images in DECKGPRH dataset (a) Sample positive images; (b) sample normalized images.	10
Figure 5 An overview of histogram computation chain.....	11
Figure 6 Visualization of HOG features of a GPR image.....	11
Figure 7 The structure of a typical cascade classifier.	13
Figure 8 Typical Recall-Precision diagram.....	14
Figure 9 Automatic processing of B-scan GPR images: (a) Concatenating location of rebars in each two-dimensional cross-section GPR image to determine spatial location of rebars in bridge deck; (b) Comparing color intensity of each rebar with a threshold for labeling deteriorated rebars; and (c) Deterioration mapping.	15
Figure 10 Pseudocode for filtering algorithm.	17
Figure 11 Non-reflection hyperbola pattern due to overlapping of reflection hyperbola.....	20
Figure 12 Effect of Normalization Method on Performance of the Classifier.	21

Figure 13 Precision-Recall plot for different parameters: (a) Boosting algorithm; (b) Window size; (c) Number of training stages; and (d) Number of orientation bins.....	22
Figure 14 Graphical user interface of the proposed program for automatic processing of bridge deck GPR data: (a) Detection of rebars in B-scan GPR images and (b) Visualization of deterioration	26
CHAPTER 2	
Figure 1 GPRDETNet dataset: (a) number of instances in each image; and (b) size of instances comparing to image size	45
Figure 2 Accuracy of detection model (overall mAP on COCO and GPRDETNet) vs. accuracy of feature extractors.....	47
Figure 3 Accuracy stratified by size of instance, meta-architecture and feature extractor on GPRDETNet dataset	48
Figure 4 Memory (Mb) usage and detection speed (FPS) for each model on GPU..	50
Figure 5 Memory (Mb) usage and detection speed (FPS) for each model on ARM based platform	51
Figure 6 Thermal performance of the ARM platform: (a) Raspberry Pi 3 B+ while running detection task; (b) thermal image while idling; (c) thermal image 30seconds after running detection task; and (d) thermal image after 30min.	52
CHAPTER 3	
Figure 1 The flowchart of the proposed detection model.	64
Figure 2 GPR frequency and antenna: (a) tradeoff between spatial resolution and penetration depth of EM wave; (b) 1.6GHz antenna mounted on a cart [12]....	65

Figure 3 URIGPR dataset: (a) Sample images; (b) Sample normalized images; and (c) Non-reflection hyperbola pattern due to overlapping of reflection hyperbola.	66
Figure 4 Utilizing an image pyramid (multi-scale representation) and sliding a window approach allows to find objects in GPR images at different scales.....	71
Figure 5 Pseudocode for the proposed filtering algorithm.....	73
Figure 6 ROC curve for different parameters: (a) cell size; (b) block size; (c) block overlap; and (d) number of orientation bins.....	76
Figure 7 Visualization of HOG features: (a) unsigned histograms; and (b) signed histograms.	80
Figure 8 The effect of MLP model architecture on performance of HOG based classifier.....	81
Figure 9 ROC curve for different parameters: (a) windows size; (b) cell size; (c) number of neighbors and radius; and (d) histogram normalization.....	82
Figure 10 LBP feature descriptors: (a) The circular neighborhoods; and (b) an example of computing the LBP representation from the original input image ..	86
Figure 11 The effect of MLP architecture on performance of LBP based MLP classifier.....	87
Figure 12 (a) F-measure comparison: (a) Hand-crafted feature descriptors and convolutional neural network on URIGPRv1.0 dataset; (b) Effect of number of training samples on performance of the classifier.	88
Figure 13 Output of detection system: (a) before applying the filter; and (b) after applying the filter.....	90

Figure 14 URICAB program's user interface: (a) Automatic detection of rebars and (b) Visualization of deterioration.....	92
CHAPTER 4	
Figure 1 Diagrams of the Faster RCNN and SSD detection meta-architectures.....	108
Figure 2 Accuracy vs time on GPU platform, with colors and marker symbols indicating feature extractor backbone and meta-architecture, respectively.	113
Figure 3 Accuracy vs time on ARM-based platform, with colors and marker symbols indicating feature extractor backbone and meta-architecture, respectively.	114
Figure 4 Accuracy of detector on COCO and DETDATASET vs accuracy of feature extractor (as measured by top-1 accuracy on ImageNet-CLS).	115
Figure 5 Accuracy stratified by object size, meta-architecture and CNN backbone.	116
Figure 6 Effect increasing number of box proposals on accuracy (mAP).....	117
Figure 7 Memory (Mb) usage and detection speed (FPS) for each model on ARM platform.....	118
Figure 8 Running time (milliseconds) for each model on GPU and ARM platform	119
Figure 9 Thermal throttling benchmark of ARM-based platform while running the detection model for 15 minutes.	121

LIST OF TABLES

CHAPTER 2

Table 1 Comparison of CPU, GPU, and ARM Platforms	35
Table 2 Properties of the Feature Extractors Implemented in this Study.....	42
Table 3 Convolutional-based Detection Models that Use One of the Meta- architectures that is Implemented in This Study	43
Table 4 Overall mAP and AR Values.....	49

CHAPTER 3

Table 1 The effect of various HOG computation parameters on performance of the classifier	77
Table 2 The effect of various LBP computation parameters on performance of LBP based classifier.....	83
Table 3 Comparison of the performance of the proposed rebar detection chain in this paper with GSSI RADAN v.7 program	91

CHAPTER 4

Table 1 Comparison of GPU and ARM-based Platforms for rebar detection.....	103
Table 2 Properties of the Feature Extractor Backbones Implemented in this Study	110

Chapter 1

“A Machine Learning Based Approach for Automatic Rebar Detection and Deterioration Mapping in Bridge Deck Ground Penetrating Radar Data”

BY

Pouria Asadi¹; Mayrai Gindy²; Marco Alvarez³

is published in KSCE Journal of Civil Engineering

¹ PhD Candidate, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: asadi@uri.edu

² Associate Professor, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: mayraig@uri.edu

³ Assistant Professor, Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881. Email: malvarez@cs.uri.edu

ABSTRACT

Ground penetrating radar (GPR) is a non-destructive method (NDT) for subsurface object identification. Interpretation of GPR data is often done manually by an engineer, which is a time-intensive task and requires moderate to significant level of training. The authors proposed a novel machine learning based processing for automatic interpretation and quantification of concrete bridge deck GPR B-scan images. The proposed method is based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration in concrete bridge decks. For the first time, the authors introduced a dataset of 4,000 B-scan images cropped from real bridge deck GPR field data, named DECKGPRH1.0. The proposed method is tested on bridge deck GPR data collected from three bridges with different NBI ratings. The results presented indicate that by implementing a ML based classifier and a fine-tuned filter, the proposed approach provides a robust solution for automatic quantification GPR field data.

Keywords: Ground Penetrating Radar, GPR, rebar detection, deterioration mapping, HOG, AdaBoost

1 INTRODUCTION

Statistics by the Federal Highway Administration (FHWA)'s report indicates that 11% of the bridges in the United States are rated as "structurally deficient" and over 30% of existing bridges have exceeded their 50-year design life, meaning that condition assessment and repair/rehabilitation programs will require substantial budget in the near future. Current observation-based bridge inspection techniques consist largely of time consuming and subjective measures for quantifying deterioration of bridges. Some advanced nondestructive testing methods (NDT), such as Ground Penetrating Radar (GPR), are being implemented that provide sub-surface information without drilling, coring, or digging by producing a continuous image of subsurface features. GPR has been successfully used in a wide range of applications, including void localization in concrete (Trela et al., 2015), land mine detection (Torrione et al., 2014), underground utility tracing and mapping (Jaw and Hashim, 2011), optimization and assessment of railway ballast (Shao et al., 2011), and condition assessment (Wang et al., 2011). Although in GPR method data collection is fast and efficient but interpretation of GPR data is time-consuming and sensitive to operator decision to provide reliable results. To address these issues the authors proposed a Machine Learning (ML) based detection algorithm for detecting GPR reflection hyperbolas in concrete bridge deck. The proposed method especially developed for complex GPR field data in highly deteriorated concrete bridge decks. The authors proposed a novel machine learning based processing for automatic interpretation and quantification of concrete bridge deck GPR B-scan images. The proposed method is based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration in concrete bridge decks. For

the first time, the authors introduced a dataset of 4,000 B-scan images cropped from real bridge deck GPR field data, named DECKGPRH1.0 (Asadi and Gindy, 2019).

An extensive literature exists regarding automatic identification of buried objects using GPR data, but here just a few relevant and newer works are mentioned. Bouzerdoum et al. (2016) focused on removing background clutter in GPR scans. They proposed a method to suppress the background clutter to improve target detection (Bouzerdoum et al., 2016). Qiao et al. (2015) proposed a general hyperbola pattern detection system based on a novel method called multi-resolution monogenic signal analysis for processing GPR B-scans (Qiao et al., 2015). Pasolli et al. (2008) developed a method for identification of buried objects by localizing hyperbolic patterns using Genetic Algorithm for data classification which is relatively computationally expensive method (Pasolli et al., 2008). Cui et al. (2010) developed feature recognition algorithm based on center-surround difference detecting and implemented fuzzy logic approach to classify extracted hyperbolic signatures which was sensitive to GPR system settings (Cui et al., 2010).

The authors proposed a new ML based detection especially developed for processing low-contrast and complex GPR data in highly deteriorated bridges. The proposed ML-based detection method is consisting of image processing and supervised machine learning data classification for processing GPR B-scan images and a novel data filtering algorithm and spatial pattern analysis for post-processing the results. This paper shows combination of Histogram of Oriented Gradients (HOG) feature descriptors, coupled with off-the-shelf supervised ML-based binary data classifier, called AdaBoost, and a filter designed and tuned for this problem provides a robust and reliable system for automatic rebar detection in bridge deck GPR field data. A dataset of 4,000 reflection hyperbolas cropped from bridge

deck B-scan images, named DECKGPRHv1.0 is introduced in this study, for first time. It allows researchers to develop and evaluate performance of novel approaches in identification and labeling of objects in GPR field data.

Overall, the contribution includes:

- Novel approach for rebar detection/localization in concrete bridge deck GPR field data is proposed
- Optimum GPR antenna frequency for scanning concrete bridge decks based on trade-off between special resolution and penetration depth is identified
- Optimized HOG feature computation and AdaBoost ML classifier parameters for maximizing F-Measure in classification task are identified
- A dataset containing 4,000 reflection hyperbolas cropped from real bridge deck GPR field data, named “DECKGPRHv1.0”, is introduced for first time in this paper
- A computer program is developed for processing and post-processing bridge deck GPR data:
 - Supervised machine learning based localization of rebars in bridge deck
 - Three-dimensional rebar mapping which defines spatial location of each rebar
 - 3D deterioration mapping based on color intensity at location of rebars
 - Quantification of bridge deck deterioration based on spatial pattern analysis

The authors discussed previous studies on object detection in GPR data in §1. Components of the proposed method for processing and quantification of bridge deck GPR

data is described in §2. A detailed description of implementation of the proposed method and performance study is provided in §3. The main conclusions are summarized in §4.

2 METHODS

This section gives an overview of the proposed method for automatic quantification of concrete bridge deck GPR data, which is summarized in Figure 1. The proposed method consists of two main components: (i) Offline training of supervised ML based binary image classifier and (ii) Online processing and quantification of concrete bridge deck GPR B-scan images.

Figure 1 The flowchart of the proposed method for concrete bridge deck GPR data processing and quantification.

2.1 Offline Training of Image Classifier

This section gives an overview of the proposed machine learning based image binary classification chain. In offline training phase effect of HOG computation and classifier parameters on performance of binary image classifier on DECKGPRH dataset was studied. The output of offline training is a trained binary classification model which will be used to create a hyperbola pattern (rebar) detector in GPR B-scan images for determining location of rebar, (X, Y), in each two-dimensional cross-section GPR images.

2.1.1 Data Collection

A GPR system consists of a few components that emit an electromagnetic (EM) wave into the sub-surface and receives the response to detect subsurface object. When the wave passes into the new medium with different electromagnetic properties a part of the electromagnetic wave is reflected back to the receiving antenna. The depth and shape of the reflecting interface as well as information about the permittivity of materials on either side of that surface can be determined. Two factors need to be considered in selecting GPR antenna: “Penetration depth” which indicates of how deep EM radiation can penetrate in concrete and “Spatial resolution” which is the ability of the antenna to see two closely spaced objects separately located in a specific medium. Penetration depth for dielectric materials with low conductivity is approximated as defined in Equation 1. Spatial resolution is determined by wavelength which is a function of frequency and velocity as defined in Equation 2.

$$dp = (1/\pi(\epsilon' \tan \delta) f) \times \sqrt{\epsilon'/\mu_0} \quad (1)$$

$$\lambda = v/f = c/(f \sqrt{\epsilon'/\epsilon_0}) \quad (2)$$

Where, dp is penetration depth in concrete, μ_0 is permeability in vacuum, $\tan \delta$ is loss tangent, λ is wavelength, v is velocity of wave inside medium, c is speed of light in vacuum, ϵ_0 is permittivity in vacuum, f is wave frequency, and ϵ' is dielectric constant.

In GPR system, as the wavelength decreases inside the medium, spatial resolution increases; however, this limits penetration depth. Penetration depth and wavelength of EM wave in concrete over the frequency range are calculated based on experimentally obtained values (Rhim and Buyukozturk, 1998) of dielectric constant and loss tangent of concrete

with uniaxial strength of 21 MPa at 28 days in two moisture condition: (i) Saturated: Concrete specimen contains moisture inside; and (ii) Air dried: Concrete specimen exposed to room temperature for 28 days. Penetration depth and wavelength for concrete plotted in Figure 2.

Figure 2 EM wave penetration depth and wavelength for concrete.

According to AASHTO LRFD Bridge Design Specifications minimum clear spacing between parallel bars in a layer is 3.8 cm and the depth of a concrete deck should not be less than 17.5 cm (AASHTO, 2012). According to Figure 2 at 1.6 GHz, the penetration depth for air dried moisture condition is 25 cm and spatial resolution for saturated moisture condition is 8 cm. Since minimum clear spacing between bars specified in the code and in concrete bridge deck can be less than this value a higher resolution is preferred but since increase of resolution reduces penetration depth using an antenna with a higher frequency does not provide useful information for this problem. The authors recommend a 1.6 GHz antenna for data collection in concrete bridge decks.

Figure 3 GSSI SIR System-3000 and a 1.6GHz antenna mounted on a cart GPR system mounted on a cart used for data collection in this study.

The authors created a dataset of 4,000 reflection hyperbolas, as shown in Figure 4(a), cropped from real GPR field data, named DECKGPRHv1.0. The data is recorded by scanning concrete deck bridges with a GSSI SIR System3000 GPR and a 1.6 GHz antenna mounted on a cart. Hyperbola-free images is cropped from the CIFAR-10 dataset (Krizhevsky and Hinton, 2009).

2.1.2 Data Normalization

As shown in Figure 4, due to deterioration in bridge deck, many collected GPR images have poor contrast. In order to bring the image into a range that is more familiar or normal to the senses, the intensity values of the image need to be normalized. A common image normalization method is to make the data have a Gaussian form with zero mean and unit variance using the following formula (Patro and Sahu, 2015):

$$I_{norm} = (I - I_{mean})/I_{std} \quad (3)$$

where I is the raw input image, I_{norm} is the normalized image, I_{mean} is the mean of image intensities, and I_{std} is the standard deviation. Figure 4(b) shows sample normalized images in the DECKGPRH dataset.

Figure 4 Sample images in DECKGPRH dataset (a) Sample positive images; (b) sample normalized images.

2.1.3 Histogram of Oriented Gradients

Histogram of Oriented gradients (HOG) is a feature description method. HOG have shown to perform well in the field of human detection (Dalal and Triggs, 2005). HOG feature descriptors provide a concise representation of image fundamentally based on gradient vectors. HOG feature descriptors can be created based on the basis on the gradient magnitude F and orientation θ at each pixel which can be calculated as defined in Equation 4 and Equation 5, respectively:

$$F = \|\nabla f_{(x,y)}\| = \sqrt{\left(\frac{\partial f_{(x,y)}}{\partial x}\right)^2 + \left(\frac{\partial f_{(x,y)}}{\partial y}\right)^2} \quad (4)$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right) \quad (5)$$

Considering n_β orientation angle bins, computed gradient vector for $n \times n$ *pixel* cells will vote to the appropriate orientation bin according to the orientation angle of each pixel. Contribution of each pixel to the histogram is given by the magnitude of the gradient vector F at the pixel. The gradient histogram is a form of data compression, by encapsulating the gradient vector of all pixels in a cell into a histogram the gradient components are reduced

down to an array of n_β values, which are the sum of magnitudes of each orientation bin (Figure 5).

Figure 5 An overview of histogram computation chain.

To increase robustness of feature descriptors cells are grouped into overlapped blocks and normalize the block. Each block consists of m^2 cells. Two horizontally or vertically consecutive blocks overlapped by a multiple of cell size called “block stride”, block stride size is C pixel. Concentrate all cell histograms in a block creates a $m^2 \times n_\beta \times 1$ element vector representing a block. The last step is to concentrate the vector of normalized cell histograms from all overlapped blocks in an image window. Figure 6 shows HOG features computed for a GPR image.

Figure 6 Visualization of HOG features of a GPR image.

2.1.4 AdaBoost Classifier

The authors utilized a cascade classifier based on Adaptive Boosting (AdaBoost) ML algorithm for data classification in this study. Boosting is an effective supervised machine learning algorithm in classification methodology (Freund and Schapire, 1996). The idea is creating a strong classifier by combining a set of weak classifiers that are not powerful enough to classify data alone because they perform only slightly better than a random classifier. The most widely used version of the boosting algorithm is Discrete Adaptive Boosting (Discrete AdaBoost). Considering a training set $\mathcal{S} = \langle (x_1, y_1), \dots, (x_N, y_N) \rangle$ consist of N labeled instances where x_i is a learner (typically in form of a feature vector) belongs to space $\mathcal{X}$ which labeled with y_i belongs to a finite label space $\mathcal{Y}$. For a binary classification $\mathcal{Y} = \{-1, 1\}$. On each round $t = 1, \dots, T$, a distribution D_t over the N training examples is computed, and a weak learning algorithm is implemented to construct a weak classifier $h_t: \mathcal{X} \rightarrow \{-1, 1\}$ appropriate for the distribution D_t . Classification error rate for a weak classifier, ε_t , for a binary classifier should be less than 0.5. The strong classifier $F_t(x)$ is weighted linear combination of weak classifiers and output of final classifier $H_t(x)$ for a binary classifier is $\text{sign}[F_t(x)]$.

Based on the original discrete adaptive boosting scheme, there are several boosting algorithms that improve performance of the original algorithm. Unlike Discrete AdaBoost which uses weak classifiers with outputs belong to a discrete set of classes, Real AdaBoost works with real-valued output weak classifiers. Real AdaBoost computes the probability that a given weak learner belongs to a class set to perform optimization with respect to $h_t(x)$.

In order to detect a target in an image it's necessary to examine all possible windows for all positions and scales. The idea is based on the fact that within any image an

overwhelming majority of windows are non-targets. The key insight is that based on observations a simple classifier constructed with very few features is needed to reject non-target windows and keep potential target windows (Viola and Jones, 2001). As shown in Figure 7, the cascade classifier attempts to reject as many non-target windows as possible at the earliest stage possible with minimum computational cost and simpler classifiers made with very few features.

Figure 7 The structure of a typical cascade classifier.

Each stage of cascade classifier works as a filter that rejects non-target and allow potential targets pass to the next stage with minimum possible number of used weak classifiers. A cascade classifier increases the efficiency of target detection system by reducing the computational costs and time spent on classifying non-target windows. Inputs to the cascade classifier algorithm are desired false alarm rate, f , detection rate, d , of classifier for each stage and target false alarm rate, f_{Target} of the entire cascade classifier. In a loop, each stage of the cascade is trained with number of weak classifiers used being increased until the conditions of the loop, f, d , are satisfied. Then false alarm rate of cascade classifier should be compared with desired f_{Target} .

For training and evaluation of cascade classifier, available data in dataset is randomly divided into two parts for the training and testing the performance of the cascade classifiers. A set of cascade classifiers with different parameters is created to study effect of different parameters on performance of rebar detection system in GPR images. To quantify detector performance Precision-Recall curves are used. The precision-recall plot is a model-wide measure for evaluating binary classifiers (Blum, 1992). The quality of cascades is assessed by drawing a $\text{Precision} = TP / TP + FP$ versus $\text{Recall} = TP / TP + FN$ curve for each cascade. In binary classification problems precision is the number of correctly detected positive objects divided by total number of detected objects. Recall is the number of correctly detected positive objects divided by total number of positive objects that should be accepted based on the ground truth. Figure 8 represents precision and recall diagram.

Figure 8 Typical Recall-Precision diagram.

In diagram above, TP, FP and FN are correctly detected positive (True Positive), incorrectly detected positive (False Positive) and incorrectly detected negative (False

Negative), respectively. The goal is to have a classifier be at the upper right corner, a perfect classifier provides only TP with no FP and FN in output. Furthermore, area under curve for each of the Precision-Recall curves (PRAUC) corresponding to each classifier is estimated. The higher the PRAUC is, the better the classifier is.

2.2 Online GPR Data Processing

This section gives an overview of the proposed bridge deck GPR data processing chain, including identification of rebar location and properties in GPR scans and then performing analysis for quantification and visualizing bridge deck GPR data.

2.2.1 Rebar Properties Identification

Rebar properties identification includes: (1) Determining coordinates of rebars in each GPR profile; (2) Removing false positive detections (outliers) and non-top layer rebars; (3) Extracting color intensity at the location of rebar; (4) Creating a three-dimensional map of rebars in deck with corresponding color intensity, (X, Y, Z, I) .

Based on the obtained binary classification model obtained from previous section, a sliding a window-based detector is implemented to determine exactly “where” in a GPR image a rebar resides. According to the distance of two-dimensional cross-section scans, a three-dimensional rebar map is created which defines spatial location of each rebar in bridge deck, as shown in Figure 9(a).

Figure 9 Automatic processing of B-scan GPR images: (a) Concatenating location of rebars in each two-dimensional cross-section GPR image to determine spatial

location of rebars in bridge deck; (b) Comparing color intensity of each rebar with a threshold for labeling deteriorated rebars; and (c) Deterioration mapping.

Rebar corrosion is among the most important deterioration mechanisms those are of highest concerns to bridge engineers (Gucunski and National Research Council, 2013) and top rebar layers have the highest potential for corrosion due to penetration of moisture through cracks. So, obtaining precise information about top layer rebars is necessary for developing an automated system for condition assessment of bridge decks based on GPR data. To remove false positive and non-top layer rebar detections a data filtering algorithm is developed. Fine steel rebar mesh is a complete reflector of radar energy in GPR system, so a tight mesh disguise targets behind the top layer of rebars then have a higher density of detections close to the surface. It brings the idea of implementing an adaptive polynomial filter to remove false target detections. The authors proposed an algorithm for removing outliers and non-top rebar ROIs. Output of the proposed algorithms is coordinates of top layer rebars and intensity of image at the area where the rebar is located. Pseudocode for the proposed filtering algorithm is shown in Figure 10.

Input: $\mathcal{S} = \langle (x_1, y_1, f_1), \dots, (x_N, y_N, f_N) \rangle$

Initialize: $d = 2, n = 3$

Iterate: *FOR* $j = 1$ *to* 3

$$d = 0.75 * d$$

$$[P_0, P_1, P_2, \dots, P_n] = \text{Polynomial}(\mathcal{S}, n)$$

$$P(y_N) : P(x_N) \Rightarrow P_0 + P_1 x_N^1 + \dots + P_n x_N^n$$

$$D = d \times (\max(x) - \min(x)/N)$$

```

FOR  $k = 1$  to  $N$ 
    IF  $|P(y_k) - y_k| < D$  THEN
         $\mathcal{S} = \mathcal{S} - \{(x_k, y_k)\}$ 
    END FOR
END FOR

Output:  $\mathcal{S}$ 

```

Figure 10 Pseudocode for filtering algorithm.

2.2.2 Data Quantification and Visualization

A bridge deck GPR scan provides a huge amount of data in form subsurface profiles at a fixed interval spacing between each GPR profile (Two-dimensional cross-section GPR images). Deterioration mapping is performed by comparing signal magnitude or color intensity at location of rebar with a threshold. For quantification of deterioration in bridge deck spatial quadrant analysis is implemented based on the provisions of Standard Test Method for Evaluating Asphalt-Covered Concrete Bridge Decks Using Ground Penetrating Radar, ASTM D6087 (ASTM, 2008). According to ASTM D6087, the amplitudes of the reinforcing reflections along each pass provide a gradational scale. The lower the reflection amplitude, the higher the likelihood of deterioration. The spatial location of scans containing reflection amplitude less than 6 to 8 dB below the maximum reflection amplitudes recorded typically correspond to deterioration detected using other information, such as results from other deterioration assessment techniques.

The authors implemented two parameters for quantification of GPR data and characterizing deterioration condition of bridge deck based on Spatial Pattern Analysis. Special Pattern Analysis includes a variety of techniques for describing the evaluation of

the pattern, or distribution, of a set of points on a surface. Quadrat Analyses is a method in field of Spatial Pattern Analysis for studying the spatial arrangement of point locations through describing a procedure of sampling and recording point-based data within a grid of square cells (Thomas, 1977). Since information about rebars in concrete deck can be described as grid, it is possible to analyze grids where the contents of grid cells are regarded as counts of point objects (rebars), P_{Total} on bridge deck surface. The event distribution has then been coded as number of deteriorated rebars, P_{Events} , in the grid cells. Simple statistics is computed to determine mean and variance. The program describes bridge condition based on GPR data using the following equations:

$$Deterioration\ intensity = P_{Events}/P_{Total} \quad (6)$$

$$Dispersion\ index = Variance/Mean \quad (7)$$

Deterioration intensity range is 0 to 1. Deterioration intensity for bridge deck with no deterioration is 0. For a uniform distribution of deteriorated regions on bridge deck, the variance is zero. Therefore, a variance-mean ratio close to 0 is expected. For a clustered distribution of deteriorated regions on bridge deck, the variance is relatively large. Therefore, a variance-mean ratio above 1 is expected.

The program provides the spatial location of rebars (X, Y, Z) and color intensity corresponding to each rebar in Text (.txt) format as output. Visual output of the program is a contour plot of deterioration on bridge deck.

3 RESULTS

The authors give details of HOG implementations and systematically study effects of the various choices in training cascade classifiers on detector performance. Then the components of the program are described.

3.1 Performance of Rebar Detector

Throughout this section the results are referred to the “default” detector which has the following properties: mono color space with no gamma correction; gradients casted into 9 histogram bins based on their orientation in a range from 0 to π ; 16X16 pixel blocks of four 8X8 pixel cells; Gaussian spatial window with sigma = 8 pixel; L2-norm block normalization; block spacing stride of 8 pixels; 64X128 detection window; 21 stages for training cascade classifier; and Discrete AdaBoost classifier. Figure 12 and Figure 13 summarize the effects of the various parameters on overall detection performance. These will be examined in detail below:

3.1.1 Windows size

The effect of three different window sizes on performance of detection system is evaluated. A 64X128 pixel window includes approximately 20 pixel right-left margin and 52px top-bottom margin around the hyperbola pattern. 48X48 pixel window includes approximately 12 pixels of margin around the target on all four sides. For 32X64 pixel window the left-right bottom margin and top-bottom margin around the target is 4 pixel and 20 pixels, respectively. It was expected a larger margin provides more information about the background and improves the detection performance but the best result is obtained with 48X48 pixel window. The reason is the non-reflection based hyperbolic patterns are developed due to overlapping of reflection hyperbola tales in a dense rebar layer in reinforced concrete deck as shown in Figure 11.

Figure 11 Non-reflection hyperbola pattern due to overlapping of reflection hyperbola.

Those non-reflection hyperbola patterns don't represent a target in B-scan GPR image but these non-reflection hyperbola patterns produce HOG feature descriptors similar to a reflection hyperbola (rebar). A non-reflection-based hyperbola provides misleading information to the classifier and decreases the overall performance of detection system. So, although bigger margins may increase the detection performance in different problems, for example the classic problem of pedestrian detection in field of machine learning and computer vision, by providing more information about the background of hyperbola patterns but in this problem a 48X48 pixel windows provides the best results. As shown in Figure 13(b) limiting the margin to 12 pixels from all four the performance increased by 3.3%.

3.1.2 Block Normalization Methods

Effect of four different normalization methods on performance of detection system is evaluated by training four cascade classifiers based on the default parameters with different block normalization methods. Let u be un-normalized descriptor vector containing all histograms in a block, $\|u\|_i$, be i^{th} -norm of u for $i = 1 \& 2$, ϵ be a small positive

infinitesimal quantity that prevents division by zero in gradient-less blocks, and $\hat{u} = u/norm(u)$ be normalized descriptor vector. The implemented normalizing schemes are: (i) $l2$ -norm (Equation 8) which is the most widely used norm scheme in almost every field of engineering and science; (II) $l2$ -hys which is Euclidean norm followed by limiting the maximum values of u to 0.2 [19]; (III) $l1$ -norm (Equation 9); and (IV) $l1$ -sqrt (Equation 10).

$$\hat{u} = u / \sqrt{\|u\|_2^2 + \varepsilon^2} \quad (8)$$

$$\hat{u} = u / \|u\|_1 + \varepsilon \quad (9)$$

$$\hat{u} = \sqrt{u / \|u\|_1 + \varepsilon} \quad (10)$$

It is observed L2-Hys, L2-norm and L1-sqrt have very similar performance. L2-hys has a silently better performance, while as shown in Figure 12, implementation of L1-norm reduces performance by 24%.

Figure 12 Effect of Normalization Method on Performance of the Classifier.

3.1.3 Classifier

The authors compared performance of three boosting machine learning algorithms to combine the prediction of each weak learner and convert weak classifiers to strong classifiers: (1) Discrete AdaBoost, (2) Real AdaBoost; and (3) Gentle AdaBoost. Three classifiers with the respective boosting algorithm and default parameters is evaluated. As shown in Figure 13(a) Real AdaBoost outperformed the two other boosting algorithms. With both Real AdaBoost and Gentle AdaBoost a better performance is obtained comparing to Discrete AdaBoost which is the default boosting algorithm implemented in cascade classifier architecture. Real AdaBoost provides a better performance (3.1%) comparing to Gentle AdaBoost. Using Real AdaBoost instead of Discrete AdaBoost increases the performance by 6% at the cost of 9% more run-time.

Figure 13 Precision-Recall plot for different parameters: (a) Boosting algorithm; (b) Window size; (c) Number of training stages; and (d) Number of orientation bins.

3.1.4 Orientation Angle Binning

The gradient vector at each pixel is computed and placed into equally divided orientation angle bins to create a histogram. The obtained results show that in case of presence of a strong gradient right on boundary of two bins, a very small change in gradient angle has a drastic effect on the histogram. To reduce this effect on performance the detection algorithm, contribution of each gradient vector linearly distributed between two closest bins for gradients located on boundary of two orientation bins. The authors studied effect of θ both in range of 0 to π (un-signed gradients) and 0 to 2π (signed gradients). According to the obtained results, using un-signed orientations perform better than signed orientation for hyperbola detection in GPR data. As shown in Figure 13(d) using unsigned orientation slightly improves performance of detector. It seems for hyperbola detection in GPR data, the wide range of background patterns and anomalies presumably makes the signs of orientation uninformative. However, including sign information may help substantially in some object recognition and labeling problems. As shown in Figure 13(d) using fine orientation binning turns out to be an important factor in performance of detector so that increasing number of orientation bins significantly improves performance up to about 9 orientation bins. Encapsulating gradient vectors into 6-bins and 3-bins reduces overall performance of detection algorithm by 6% and 18% respectively comparing to the result obtained with 9-bins.

3.1.5 Effect of Adaptive Polynomial Filtering

To increase performance of the detection system an Adaptive Polynomial Filter applied on output of cascade classifier-based rebar detector. By fine tuning the parameters of cascade classifier and then by fine tuning parameters of filter very good results is obtained

in finding top-layer rebars. To obtain best parameters of filter for this problem effect of difference parameters of filter on performance of the detection system is tested. Empirically, the best result is obtained by applying the adaptive filter three times on data. The filter applied on the results in three rounds and in each round some points is filtered. Applying the filter on the data increases the performance by 12%.

3.1.6 Number of Stages

Effect of the number of stages which a classifier should have after training is studied on performance detector. Figure 13(c) shows increasing number of training stages to 21 increases performance of detection system comparing to the result obtained with 17 stages. Increasing number of stages to 21 improves performance by 12%. Decreasing number of stages to 13 decreases performance by 18%. Increasing number of stages from 21 to 25 increases the performance by 1.3%. Although increasing number of stages to 25 drastically increases training time but since in this application the goal is gaining highest accuracy, then the authors recommend 25 stages for training cascade classifier for classification GPR data.

Overall, there are several notable findings in this study. Although it was expected a larger window size and margin provides more information about the background and increases overall performance of detector but in this problem, it was observed that because of negative effect of non-reflection-based hyperbola appears in GPR images due to overlapped reflection hyperbolas, limiting the margin size provides a significantly better performance. The reason is hyperbola patterns formed in GPR field data due to overlapping of reflection hyperbola tales. Creating a chain of cascade classifiers tuned to provide a high detection rate (and false alarm rate) and an adaptive polynomial filter to remove false

detections from output data set provides a significantly better result comparing to a detection system without an adaptive polynomial filter. Local normalization is essential for good results in low-contrast data, better results can be obtained by normalizing each element several times with more overlapped cells. It was observed that there is no significant difference in using signed or un-signed orientation angle bin in classification GPR image using AdaBoost ML algorithm. By increasing the number of positive samples in training, performance of classifier increased and also with a fixed number of negative samples, training time decreased. Increasing number of negative samples increased required time for the training cascade classifier. It was observed that the number of stages in process of training cascade classifier has a considerable effect on both training time and performance of detection system. Based on the obtained results the following properties provides the best result: Nine (9) un-signed orientation angle bins; 48X48 pixel windows size; Gaussian spatial window with $\sigma = 8$ pixel; L2-hys block normalization; block spacing stride of 8 pixels; 25 stages for training cascade classifier; and Read AdaBoost machine learning method for training the cascade classifier for binary classification.

3.2 GPR Data Processing/Post-Processing Program

The authors proposed a program for automatic processing and quantification concrete bridge deck GPR data. As shown in Figure 14, the program provides a graphical user interface for loading GPR scans into the program. The program quantifies and visualizes

the deterioration in bridge deck by implementing statistical methods on GPR data based on the provisions of ASTM D6087.

Figure 14 Graphical user interface of the proposed program for automatic processing of bridge deck GPR data: (a) Detection of rebars in B-scan GPR images and (b) Visualization of deterioration

4 CONCLUSIONS

In this study, a program for automatic processing and interpretation of concrete bridge deck GPR data was introduced to help engineers and decision makers instantly obtain precise information about bridge deck condition based on GPR data, and to improve and facilitate processing and interpretation of GPR data. This method significantly reduces the time of processing GPR data and improves computing efficiency.

It was observed that a detector algorithm based on HOG feature descriptor, Real AdaBoost machine learning algorithm and adaptive polynomial filter gives very good

result for rebar detection even in low-contrast and complex GPR field data. Performance of HOG/AdaBoost based detection system was studied on obtained GPR field data from several bridges and it was observed that performance of the detector on bridges with deterioration was very dependable. Effect of various training parameters showed that to obtain a good performance the margin size should be limited. For first time, the authors introduced a bridge deck GPR image dataset, named DECKGPRHv1.0, which is publicly available.

For future work, the research will continue to improve the accuracy and speed of automatic rebar detection algorithm. For this purpose, performance of deep learning-based image classifiers accompanied with region proposal algorithms (instead of sliding window approach) needs to be studied. The proposed GPR B-scan detection/interpretation chain will be used for developing an automated Robotic/Drone based system for condition assessment of highway bridges using non-destructive testing (NDT) methods.

REFERENCES

- Asadi, P. and Gindy, M. (2019). "DECKGPRHv1.0 Dataset." Access date: January 10, 2019 <https://github.com/PouriaAI/GPR-Detection/>.
- ASTM (2008). *ASTM D6087: Standard Test Method for Evaluating Asphalt-Covered Concrete Bridge Decks Using Ground Penetrating Radar*. ASTM International, West Conshohocken, PA.
- AASHTO (2012). *AASHTO LRFD Bridge Design Specifications: Customary US Units*, American Association of State highway and transportation officials, Washington, DC.
- Blum, A. (1992). "Neural networks in C++: an object-oriented framework for building connectionist systems." New York: Wiley, Vol. 1, No. 1, pp. 223-224.

- Bouzerdoun, A., Tivive, F. H. C., and Abeynayake, C. (2016). "Target detection in GPR data using joint low-rank and sparsity constraints." In *Compressive Sensing V: From Diverse Modalities to Big Data Analytics. International Society for Optics and Photonics*, Vol. 9857, No. 1, p. 98570A, DOI: 10.1117/12.2228345.
- Cui, Y. A., Wang, L., and Xiao, J. P. (2010). "Automatic feature recognition for GPR image processing." *World Academy of Science, Engineering and Technology*, Vol. 61, No. 1, pp. 176-179.
- Dalal, N. and Triggs, B. (2005). "Histograms of oriented gradients for human detection." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, IEEE*, Vol. 1, No. 1, pp. 886-893, DOI: 10.1109/CVPR.2005.177.
- Freund, Y. and Schapire, R. E. (1996). "Experiments with a new boosting algorithm." In *Icml*, Vol. 96, No. 1, pp. 148-156, DOI: 10.1.1.51.6252.
- Gucunski, N. and National Research Council. (2013). "Nondestructive testing to identify concrete bridge deck deterioration." *Transportation Research Board*, Vol. 1, No. 1, pp. 78-86, DOI: 10.17226/22771.
- Jaw, S. W. and Hashim, M. (2011). "Accuracy of data acquisition approaches with ground penetrating radar for subsurface utility mapping." In *2011 IEEE International RF & Microwave Conference*, pp. 40-44.
- Krizhevsky, A. and Hinton, G. (2009). "Learning multiple layers of features from tiny images." *Technical report, University of Toronto*, Vol. 1, No. 4, p. 7, DOI: 10.1.1.222.9220.
- Pasolli, E., Melgani, F., Donelli, M., Attoui, R., and De Vos, M. (2008). "Automatic

- detection and classification of buried objects in GPR images using genetic algorithms and support vector machines.” In Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, IEEE, Vol. 2, No. 1, pp. II-525, DOI: 10.1109/IGARSS.2008.4779044.*
- Patro, S. and Sahu, K. K. (2015). Normalization: A preprocessing stage, arXiv preprint arXiv:1503.06462.*
- Qiao, L., Qin, Y., Ren, X., and Wang, Q. (2015). “Identification of buried objects in GPR using amplitude modulated signals extracted from multiresolution monogenic signal analysis.” Sensors, Vol. 15, No. 12, pp. 30340-30350, DOI: 10.3390/s151229801.*
- Rhim, H. C. and Buyukozturk, O. (1998). “Electromagnetic properties of concrete at microwave frequency range.” Materials Journal, Vol. 95, No. 3, pp. 262-271, DOI: 10.3141/1574-02.*
- Shao, W., Bouzerdoun, A., Phung, S. L., Su, L., Indraratna, B., and Rujikiatkamjorn, C. (2011). “Automatic classification of ground-penetrating-radar signals for railway-ballast assessment.” IEEE Transactions on Geoscience and Remote Sensing, Vol. 49, No. 10, pp. 3961-3972, DOI: 10.1109/TGRS.2011.2128328.*
- Thomas, R. W. (1977). An introduction to quadrat analysis, Norwich, UK: Geo Abstracts.*
- Torrione, P. A., Morton, K. D., Sakaguchi, R., and Collins, L. M. (2014). “Histograms of oriented gradients for landmine detection in groundpenetrating radar data.” IEEE Transactions on Geoscience and Remote Sensing, Vol. 52, No. 3, pp. 1539-1550, DOI: 10.1109/TGRS.2013. 2252016.*
- Trela, C., Kind, T., and Schubert, M. (2015). “Detection of air voids in concrete by radar*

in transmission mode.” In Advanced Ground Penetrating Radar (IWAGPR), 2015 8th International Workshop on, IEEE, pp. 1-4, DOI: 10.1109/IWAGPR.2015.7292650.

Viola, P. and Jones, M. (2001). “Rapid object detection using a boosted cascade of simple features.” In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, IEEE, Vol. 1, No. 1, pp. I-I, DOI: 10.1109/CVPR.2001.990517.

Wang, Z. W., Zhou, M., Slabaugh, G. G., Zhai, J., and Fang, T. (2011). “Automatic detection of bridge deck condition from ground penetrating radar images.” IEEE Transactions on Automation Science and Engineering, Vol. 8, No. 3, pp. 633-640, DOI: 10.1109/TASE. 2010.2092428.

Chapter 2

“Modern Convolutional Neural Networks for Rebar Detection in Bridge Deck GPR B-Scans on Mobile and Embedded Systems”

BY

Pouria Asadi¹; Mayrai Gindy²; Marco Alvarez³; Alireza Asadi⁴

is accepted for a poster presentation at the 99th Annual Meeting of the Transportation
Research Board, Washington, D.C. and is under review for the Transportation Research

Record

¹ PhD Candidate, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: asadi@uri.edu

² Associate Professor, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: mayraig@uri.edu

³ Assistant Professor, Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881. Email: malvarez@cs.uri.edu

⁴ PhD Candidate, Department of Technical and Engineering, Azad University, Tehran, Iran. Email: alireza.asadi@azad.ac.ir

ABSTRACT

Rebar detection in bridge deck Ground Penetrating Radar (GPR) B-scans using Convolutional Neural Networks (CNN) for on-site applications is addressed. The authors investigated accuracy/frame rate trade-off of modern deep learning detection methods for automatic rebar detection in GPR B-scans on ARM platforms. The cost, portability, power consumption and Thermal Design Power (TDP) are advantages of ARM processors over parallel computing platforms, makes them a good option for wide range of applications including robotic and drone-based bridge inspection. The authors review three recent meta-architectures (R-FCN, Faster, and SSD, decoupling the choice of meta-architecture from feature extractor so that effect of various feature extractors is investigated to obtain the best option for detecting rebars in B-scans on mobile and embedded systems. State of the art results is obtained for ARM platform-based detection task on GPRDET dataset by implementing rebar detection model using SSD as meta-architecture and Inception v2 as feature extractor. The obtained results indicate that SSD meta-architecture with MobileNet as feature extractor achieves overall mAP of 46 on GPRDET dataset and remained stable during stress test on ARM platform. SSD meta-architecture with Inception v2 feature extractor outperformed MobileNet detection model by 16.6% with overall mAP of 56 and remained stable by increasing the memory allocated to the processor. A dataset of 520 bridge deck B-scans and 4,085 instances, named GPRDET, annotated for detection tasks according to Pascal VOC, is created and introduced in this paper for first time

Keywords: MobileNets, Single-Shot Detector, Rebar Detection, GPR, Mobile Devices, Embedded Systems

INTRODUCTION

Ground Penetrating Radar (GPR) as a modern Nondestructive Testing (NDT) method has been used in a wide range of applications including evaluation of deck thickness [1], detection and characterization of deterioration progression in bridge deck [2], predicting deck repair quantities [3], void localization in concrete [4], tunnel detection [5], subsurface utility mapping [6], railway-ballast assessment [7], and land mine detection [8]. Although data acquisition using GPR technology is fast, interpretation of GPR data is a labor-intensive task which highly relies on the operator's decisions to provide useful and reliable results. Detection of hyperbola patterns which represent rebars in bridge deck GPR B-Scan images is a necessary first-step of localizing, extracting, and characterizing the rebar in GPR data.

Extensive literature exists regarding processing GPR data for automatic localization of buried objects, but here just a few relevant studies is mentioned. Dinh et al. [9] developed an algorithm consisting of a convolutional network for hyperbola pattern detection. Then a CNN was trained to locate potential rebars by retaining the likely true positive, and discarding likely false positive detections. The main drawback of the proposed method is that it requires a large number of data for the training of CNN. Asadi et al. [10] proposed a method based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration and creating a 3D rebar location and deterioration map in bridge decks. The limitation of the proposed method is accuracy in highly deteriorated bridges. Qiao et al. [11] proposed a hyperbola pattern detection method based on a novel approach called multi-resolution monogenic signal analysis for processing GPR B-scan images.

Bouzerdoun et al. [7] developed a method for reducing background clutter in GPR data. The authors proposed an algorithm to reduce the background clutter to increase accuracy and reduce false positive detections. Cui et al. [12] proposed a feature identifications algorithm based on center-surround difference for detecting and implemented the Fuzzy Logic method to classify computed hyperbolic patterns. The proposed method is highly sensitive to GPR data acquisition system settings. Pasolli et al. [13] approach for detecting buried objects in GPR B-scan images is based on localizing hyperbolic patterns using Genetic Algorithm. This method is a computationally expensive algorithm and limits the application of that on mobile and embedded systems.

A lot of progress has been made in recent years on object detection task due to advances in development of parallel computing platforms which are suitable for computationally intensive optimization algorithms that are used in training CNNs. Modern object detectors based on deep convolutional neural networks, such as Region-based Fully Convolutional Networks (R-FCN) [14], Faster Region-based Convolutional Networks (R-CNN) [15], and Single Shot Multi-Box Detector (SSD) [16], are now accurate enough to be implemented in commercial products for classic detection task such (e.g., Human Pose Estimation, Face Detection) and some have been shown to be fast enough for being used on mobile and embedded system. However, it can be difficult for practitioners to decide what method is best suited for detecting rebars in GPR B-scan images on mobile devices using ARM processor. Advantages of a rebar detection architecture suitable for being used on ARM processors over Compute Unified Device Architecture (CUDA) processors are as follow: (1) significantly lower cost comparing to CUDA based platforms, and (2) lower Thermal Design Power (TDP) as shown in (Table 1). These main advantages make ARM

based platforms a suitable option for portable, robotic, and drone systems.

TABLE 1 Comparison of CPU, GPU, and ARM Platforms

Platform	PC (CPU)	PC (GPU)	Raspberry Pi (Model 3 B+)
Computation Unit	Intel Core i7-8700K	NVIDIA GTX 1080 Ti	ARM Cortex-A53
Cores/Base Freq.	6/3.60 GHz (12 Threads)	3584/1.6GHz (CUDA Cores)	4/1.4GHz
TDP	95W	280W	5W
System Memory	16GB DDR4	16GB DDR4	1GB SRAM
Storage	SSD	SSD	SD Card
Cooling	Liquid Cooling (Radiator)	Cooling Fan + Heatsink	Passive (Heatsink)
Operating System	Ubuntu 16.04	Ubuntu 16.04	Raspbian 9.6
Library	TensorFlow	TensorFlow GPU	TensorFlow
Price	\$1100	\$2000	\$35

The authors seek to investigate speed/accuracy trade-off of modern deep learning-based detection models for automatic rebar detection in GPR B-scan images in an exhaustive and fair way. To achieve this goal, the authors primarily study single-model and single-pass object detectors. These family of object detectors do not use multi-crop, ensembling, or other multi-crop methods such as horizontal flipping. This means in

investigated models a B-scan image is passed through a single network only once. For simplicity and because it is more important for practitioners, the authors focus mainly on performance of detection models than focusing on performance in training phase.

Comparing every recently developed detection system is impractical. Since many of the leading state of the art approaches have converged on a common methodology, the authors compared a large number of detection methods in a unified manner. The authors have created implementations of the R-FCN, Faster R-CNN, and SSD meta-architectures, which at a high level consist of a single CNN, trained with a mixed regression for computing bounding boxes (coordinates) and binary classification objective, and use sliding window-type predictions.

Since the obtained results from the proposed methods in the literature are based on different dataset which are not publicly released, the obtained results are not comparable. To address this problem, in this study, a dataset of 520 bridge deck B-scan images annotated according to Pascal VOC protocol [17], named GPRDETNET [18] is presented. The proposed dataset allows researchers and practitioners to develop and evaluate the performance of novel approaches in object detection task in GPR B-scan images. To summarize, the main contributions of this work are as follows:

- Extensive experiments are performed that trace the accuracy/frame rate trade-off for different detection models for rebar detection, varying meta-architecture and feature extractors.
- State of the art results is obtained for ARM platform-based detection task on GPRDETNET dataset by implementing rebar detection model using SSD as meta-architecture and Inception v2 as feature extractor.

- A computer tool is developed for Raspbian OS. The tool detects rebars and provides location of rebars in a standard format as output.
- A dataset of 520 bridge deck B-scan images and 4,085 instances, named GPRDETNet, annotated for detection tasks according to Pascal VOC is created and is introduced in this paper for first time.

METHODS

In order to perform a valid comparison of models, a detection platform in TensorFlow [19] is created for creating training pipelines, inference, and benchmarking. A unified framework facilitates the process of configuring the training parameters and swapping feature extractors. In addition, it allows for fast and easy portability to diverse platforms for benchmarking and deployment. In the following, meta-architectures, dataset statistics, various choices for model architecture, and loss function is discussed.

Meta-architectures

Deep convolutional neural networks have become the leading method for various computer vision tasks. The R-CNN method by Girshick et al. [20] is considered as one of the first modern applications of convolutional network-based detection systems. R-CNN method took the straightforward approach of cropping externally computed class-agnostic bounding box proposals out of an input image and running a neural network classifier on these proposals. Depending on performance of implemented class-agnostic proposal generation algorithm and number of generated proposals, this method can be computationally expensive. Fast R-CNN [21] improved detection speed of R-CNN method by sharing the computation load through feeding image to the network only once and using extracted features of one of the intermediate layers for cropping. In Fast R-CNN, the region

proposals are generated separately by another algorithm (Selective Search algorithm in the original paper [21]), that is computationally expensive and fairly slow, that was found to be the bottleneck of the overall model architecture. Development of Faster R-CNN [15] method is based on the idea of making generation of proposals an almost computationally cost-free step by reusing those same CNN results for region proposals instead of running a separate class-agnostic proposal generation branch in the model. In this method a single CNN is trained to perform both region proposal generation and classification tasks.

In Faster R-CNN method, there is a collection of bounding boxes overlaid on the image at different spatial locations with various scales and aspect ratios, called “anchors”. A model with two output heads is then trained to make two predictions for each anchor: (1) classification head: a discrete class prediction for each anchor, and (2) regression head: a continuous prediction of an offset by which the anchor needs to be modified to fit the ground truth bounding box. In this method the loss function sums up the cost of classification and bounding box prediction and it needs to be minimized.

If there is the best matching ground truth bonding box for each anchor a , then anchor a is labeled as a “positive anchor” and two properties are assigned to anchor a : (1) a class label $y_a \in \{1 \dots N\}$ and (2) a vector encoding of box b with respect to anchor a , called the box encoding, $\phi(b_a; a)$. If no matching ground truth bonding box is found, anchor a is labeled as a “negative anchor” and the class is labeled to $y_a = 0$.

If for the anchor a a box encoding, $f_{location}(I; a, \theta)$ and a corresponding class, $f_{class}(I; a, \theta)$ is predicted, where I is the input image and θ the model parameters, then the loss function for anchor a is defined as a weighted sum of combination of a classification loss and a location-based loss, as shown in (Equation 1):

$$\begin{aligned} \mathcal{L}(a; I, \theta) = & \alpha \cdot 1(a \text{ is positive}) \cdot l_{location}(\varphi(b_a; a)) - f_{location}(I; a, \theta) + \\ & \beta \cdot l_{class}(y_a, f_{class}(I; a, \theta)) \end{aligned} \quad (1)$$

Where α , β are model parameters balancing localization loss and classification loss, respectively. In training phase, the loss function (Equation 1) is averaged over anchors and minimized with respect to θ .

In this study, three recent meta-architectures are investigated: Faster R-CNN [15], R-FCN [14], and SSD [16]. While these methods were originally presented with a particular CNN as feature extractor (e.g., VGG16, Resnet), the authors in this study review these three methods, decoupling the choice of meta-architecture from feature extractors so that effect of various feature extractors is investigated with Faster R-CNN, R-FCN, or SSD to obtain the best option for detecting rebars in B-scan images on mobile and embedded devices.

Faster R-CNN

Faster R-CNN detection method consists of two main stages: (1) Region Proposal Network (RPN): images are processed by a feature extractor (e.g., ResNet 101), and features at some selected intermediate convolutional layers (e.g., “conv5”) are used to predict class-agnostic bounding box proposals. The loss function for this first stage takes the form of (Equation 1). (2): classification: boundary box proposals are used to crop features from the same intermediate convolutional feature map which are subsequently fed to the remainder of the feature extractor (e.g., “fc6” followed by “fc7”) in order to predict a class for each proposal. The loss function for this stage takes the form of (Equation 1).

RFCN

The region-specific stage in Faster R-CNN method must be applied several hundred times per input image. Dai et al. [14] proposed R-FCN method which unlike the Faster R-CNN, crops are taken from the last layer of features prior to prediction. This approach minimizes the amount of per region computation by pushing cropping to the last layer. The proposed model by Dai et al. [14] which is based on R-FCN meta-architecture and Resnet 101 as feature extractor achieved comparable accuracy to Faster R-CNN at faster running times in some cases. Notably, the R-FCN based model is also adapted to perform instance segmentation task [22], and won the 2016 COCO instance segmentation task competition.

SSD

SSD meta-architecture uses different activation maps (multiple-scales) for prediction of classes and bounding boxes. More specifically, SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4_3 layer. In this study, the term SSD refers to meta-architectures that use a single feed-forward convolutional network to directly predict classes and bounding box proposals without requiring a second stage per-proposal classification operation.

Experimental setup

The introduction of standard benchmarks such as ImageNet [23] and COCO [24] has made it easier to compare detection methods with respect to accuracy. However, when it comes to speed and resource usage, it's more difficult to make an apples-to-apples comparisons. In some cases, evaluation metrics are reported using slightly different training sets. In this study, an object detection platform is implemented in TensorFlow [19] which facilitates the process of swapping meta-architectures, feature

extractors, loss functions, and fine-tuning model parameters. In addition, it allows for easy portability and transferring detection models to diverse platforms for benchmarking or deployment. In the following we discuss ways to configure detection model parameters on the implemented detection platform.

CNN as feature extractor

In this study a convolutional feature extractor, also called “Backbone Network”, is applied to the input B-scan image to obtain high-level features on top of the low-level features. The choice of feature extractor is crucial as the number of parameters and types of layers directly affect resource usage, speed and accuracy of the detection system. The authors selected four CNN based feature extractors for evaluation in this study and, all have open source TensorFlow implementations and have had significant influence on the machine learning and computer vision community in the literature. **ResNet 101** [25] as winner of competitions such as COCO 2015 challenge for classification, detection, and segmentation tasks. **Inception v2** [26], which set the state of the art in the ILSVRC 2014 challenge both in classification and detection task. The Inception network is an important milestone in the development of CNN based feature extractors. Prior to introducing “Inception units”, deep networks generally just stacked convolution layers deeper and deeper to improve accuracy of the network. Implementation of Inception units made it possible to increase the depth and width of a network without increasing its computational cost. **Inception Resnet v2** [27], which combines implementation of residual links for optimization and updating the weight in the network with the computation efficiency of Inception units. **MobileNet** [28] which is designed to maximize accuracy while being mindful of the restricted resources for detection on mobile and embedded systems. To

reduce computational cost and number of parameters, “Depth wise Separable Convolutions” are deployed in MobileNet which factorize a standard convolution operation into a depth wise convolution and a 1×1 convolution. Properties of the feature extractors that are implemented in this study are provided in (Table 2). Top-1 accuracy is the classification accuracy on ImageNet dataset [23].

TABLE 2 Properties of the Feature Extractors Implemented in this Study

Backbone	Top-1 Accuracy (%)	Parameters (million)
Inception v2	73.9	10.2
ResNet 101	76.4	42.6
Inception Resnet v2	80.4	54.3
MobileNet	71.1	3.2

There are choices to be made in order to implement it within a meta-architecture. For both Faster R-CNN and R-FCN, convolutional layer which should be used for predicting region proposals must be specified. In this study, the authors used the choices laid out in the original papers. Feature extractors used in literature are compared in (Table 3).

TABLE 3 Convolutional-based Detection Models that Use One of the Meta-architectures that is Implemented in This Study

Meta-architecture + Backbone	Matching	Box Encoding $\varphi(b_a; a)$	$l_{location}(\varphi(b_a; a))$
SSD + Inception [29]	Bipartite	$[x_0, y_0, x_1, y_1]$	L_2
SSD + Inception [30]	Argmax	$[x_0, y_0, x_1, y_1]$	L_2
SSD + GoogLeNet's variation [31]	Box Center	$[x_0, y_0, \sqrt{w}, \sqrt{h}]$	L_2
Faster R-CNN + ResNet [25]	Argmax	$[x_c/w_a, y_c/h_a, \log(w), \log(h)]$	Smooth L_1
R-FCN + ResNet [14]	Argmax	$[x_c/w_a, y_c/h_a, \log(w), \log(h)]$	Smooth L_1

Loss function

Configuration of the loss function (Equation 1) impacts training stability and testing performance of the detection models. Predicting the labels (classification) and localization (regression) of instances for each bounding box requires matching bounding boxes to ground truth instances in a dataset (label and coordinates). In this study, Argmax matching with threshold values according to the original paper for each meta-architecture is implemented. Ratio for number of positive and negative bounding boxes are those recommended by the original paper for each meta-architecture. In accordance with prior studies [20, 21, 15, 16], the following function is used to encode a ground truth box with respect to its matching bounding box: $\varphi(b_a; a) = [10 \cdot x_c/w_a, 10 \cdot y_c/h_b, 5 \cdot \log(w), 5 \cdot \log(h)]$. Following prior works [21, 15, 16], to combine advantages of L_1 loss (steady gradients for large values) and L_2 loss (less oscillations during updates for small values), Smooth L_1 loss [32] is used in all

experiments.

Input size

In SSD, models are resized to a fixed shape $N \times N$ whereas in Faster R-CNN and R-FCN, models are trained on input images scaled to N pixels on the shorter edge. In this study input image size is set to $N = 300$. Notably, with all other parameters held constant, the SSD models processes fewer pixels than a Faster R-CNN or R-FCN model.

Training, fine-tuning and benchmarking

Stochastic Gradient Descent (SGD) with momentum optimization algorithm is used for Faster R-CNN and R-FCN. Since the models using input image with different size, batch size parameter is set to 1. For SSD, Root Mean Square Propagation (RMSProp) [33] algorithm is used with batch size parameter set to 32. Note that for implementation of Faster R-CNN and R-FCN models in TensorFlow, instead of using the RoI Pooling layer [15] and position-sensitive RoI Pooling layers [14] which are used in the original papers, TensorFlow's "CropAndResize" function (`TensorFlow::ops::CropAndResize`) is implemented. TensorFlow's CropAndResize function extracts crops from the input image tensor and resizes them using bilinear sampling or nearest neighbor sampling (possibly with aspect ratio change) to a common output size [19] which is similar to the differentiable cropping approach [34].

The investigated models for rebar detection task in this study are trained on the COCO dataset [24] and fine-tuned on the GPRDETNet [18] dataset which introduced in this paper. A dataset of 520 B-scan images and 4,085 instances is created and introduced in this

paper for first time, named GPRDETNet. The instances in GPRDETNet dataset are annotated for detection tasks according to Pascal VOC [24] format. Images are cropped from real GPR B-scans collected from several bridges in the US using a GPR antenna of 1.6 GHz. On average the dataset contains 7.9 instances per image, which is similar to COCO [24] dataset with 7.7 instances per image which is used for pre-training the detection models. In addition, instances in COCO are smaller than PASCAL VOC [17] which makes it similar to GPRDETNet dataset. Generally smaller objects are harder to recognize [24] and require more contextual reasoning to recognize. (Figure 1) shows distribution of size of instances in GPRDETNet. As shown in (Figure 1) there are four (4) or less instances (hyperbola-patterns) in 0.75% of B-scan images in GPRDETNet.

Figure 1 GPRDETNet dataset: (a) number of instances in each image; and (b) size of instances comparing to image size

Official COCO API is used [35] to evaluate performance of detection models, which measures mean Average Precision (mAP) averaged over Intersection over Union (IoU) thresholds in $[0.5 : 0.05 : 0.95]$, amongst other metrics. To train the models, a machine with Ubuntu 16.04 OS, 16GB RAM, Intel Core i7-8700K processor and an NVIDIA GTX 1080 Ti card is used. The benchmarking results on GPRDETNet dataset for detection task is performed on GPU and ARM based platforms.

RESULTS/DISCUSSION

The obtained results from testing different models are presented in this section. Each detection model mainly includes a choice of one of the three meta-architecture and four feature extractors. Resource usage, accuracy, and frame rate for each model is provided.

Intuitively, stronger performance on classification should be positively correlated with better detection performance on COCO and GPRDETNet. The relationship between overall mAP of different detection models and the Top-1 accuracy on ImageNet classification task is investigated. The obtained result indicates that there is a correlation between performance of classification task and detection task. As shown in (Figure 2), the correlation between performance of classification task and detection task is significant for Faster R-CNN and R-FCN while the performance of SSD based detection models both on COCO and GPRDETNet datasets are less dependent on the accuracy of the convolutional based feature extractors. It was observed that there is a strong correlation between detection performance of the models with R-FCN meta-architecture and size (number of parameters) of the feature extractors.

On COCO dataset, the highest mAP is obtained with Faster R-CNN as meta-architecture with Inception ResNet v2 feature extractor. The highest mAP on GPRDETNet dataset is obtained with R-FCN model with Inception ResNet v2 feature extractor. Unlike SSD based models, the accuracy of Faster R-CNN and R-FCN models are significantly more depended on size of feature extractor network. As it shown in (Figure 2) high capacity feature extractor like Resnet 101 (42.6 million parameters) and Inception ResNet v2 (54.3 million parameters), significantly increases accuracy of detection models in both COCO

and GPRDETN datasets.

Figure 2 Accuracy of detection model (overall mAP on COCO and GPRDETN) vs. accuracy of feature extractors

All models performed significantly better on large size instances comparing to medium size instances, except Faster R-CNN + Inception ResNet v2 model. Accuracy of SSD + ResNet 101 models are comparable for both medium size and large size instances. Detecting smaller instances is a challenging task because the activations of small instances become smaller after passing each pooling layer. So, selecting the right size for input images is very important to guarantee a good accuracy while the features extractor is small enough to be run on ARM based platforms (e.g., Raspberry Pi 3 B+) which has 1GB RAM. In addition, identification of small objects surrounded by generic clutter in the background is a challenge for detectors that rely on “objectness” and class-agnostic bounding box proposal due to the drastic increase in number of RoIs. SSD models with MobileNet and

Inception v2 outperformed both Faster R-CNN and R-FCN models both for medium and large size instances. The highest overall mAP on GPRDET dataset, which contains medium and large size instances, is obtained with SSD + Inception v2 model, as shown in (Figure 3).

Figure 3 Accuracy stratified by size of instance, meta-architecture and feature extractor on GPRDET dataset

As shown in (Figure 3) and (Table 4), SSD models with Inception v2 and MobileNet feature extractors are most accurate of the fastest models. R-FCN model with dense output Inception Resnet v2 feature extractor and SSD model with Inception v2 feature extractor attain the highest accuracy, achieving the state-of-the-art single model performance on GPRDET dataset. However, running R-FCN model with dense output Inception Resnet v2 which requires a large amount of memory allocated to the processor is not possible on the ARM based platform, Raspberry Pi 3 B+ with 1GB ram, used in this study. The mAP and AR values according to the COCO protocol for the investigated

detection models are provided in (Table 4).

TABLE 4 Overall mAP and AR Values

Feature Extractor	Meta- architecture	mAP	mAP _{.5}	mAP _{.75}	mAP _M	mAP _L	AR @1	AR @10	AR @100	AR @100 _M	AR @100 _L
MobileNet	Faster	49	93	46	41	53	11	39	56	52	68
	R-CNN										
	R-FCN	40	78	39	36	44	10	51	52	46	67
	SSD	48	91	50	45	61	12	53	59	54	73
Inception v2	Faster	44	80	57	48	67	10	38	53	49	65
	R-CNN										
	R-FCN	44	78	42	40	58	11	56	57	52	74
	SSD	56	98	61	51	71	13	61	64	60	78
ResNet 101	Faster	56	95	60	52	69	14	61	63	58	75
	R-CNN										
	R-FCN	47	86	52	40	65	14	59	62	58	69
	SSD	55	91	63	56	55	11	58	63	57	62
Inception ResNet v2	Faster	58	96	71	69	53	16	68	69	62	76
	R-CNN										
	R-FCN	68	98	74	48	71	19	69	71	68	81
	SSD	56	98	62	49	68	15	65	68	62	79

Performance and memory usage for the detection models on GPU and ARM platform are shown in (Figure 3) and (Figure 4), respectively. The obtained result indicates that R-FCN and SSD based detection models are generally faster while Faster R-CNN

tends to lead to slower but more accurate detection models.

For memory benchmarking, total memory usage is measured. (Figure 4) and (Figure 5) plot memory usage against frame rate on GPU and ARM platforms. Overall, there is a strong correlation between memory usage and frame rate. As with frame rate, MobileNet and Inception v2 are computationally cheapest in almost all models, requiring less than 1GB allocated memory.

Figure 4 Memory (Mb) usage and detection speed (FPS) for each model on GPU

The ARM based platform used in this study has 1GB of LPDDR2 memory, As shown in (Table 1), The actual amount of memory can be allocated to the processor is less than 1GB because 128MB memory is reserved for GPU. It means due to limitation of allocated memory to processor, ResNet 101 and Inception ResNet v2 detection models cannot be run on the ARM platform, Raspberry Pi 3 B+, used in this study. Notably, instability because of insufficient memory was observed while benchmarking Faster R-CNN model with Inception v2. To solve this issue allocated memory increased to 992MB by decreasing amount of allocated memory for GPU to 16MB. New generation of ARM based platforms, Raspberry Pi 4 B, is recently released. The new generation has up to 4GB of RAM. As of the time of writing the paper, there is no stable official TensorFlow wheel

for Raspberry Pi 4.

As shown in (Figure 5), all MobileNet and Inception v2 models in this study can be used on an ARM based processor with 1GB memory. As shown in (Figure 3) accuracy of SSD model with Inception v2 feature extractor is comparable to R-FCN + Inception ResNet v2 model at a significantly higher (10 times faster) frame rate.

Figure 5 Memory (Mb) usage and detection speed (FPS) for each model on ARM based platform

The Raspberry Pi 3 B+ is built from commercial chips which are qualified to different temperature ranges; the LAN9512 (with 2 USB ports) is specified by the manufacturers as being qualified from 0°C to 70°C, while the SoC is qualified from -40°C to 85°C. During a 30min run time, the temperature on any region on Raspberry Pi 3 B+ didn't exceed 61°C (ambient temperature: 21°C), as shown in Figure 6. The passive fan less cooling system (3 heatsinks) kept the system stable and no thermal throttling or decrease in frame rate was observed. A stress test performed to evaluate thermal performance of the ARM platform.

Figure 6 Thermal performance of the ARM platform: (a) Raspberry Pi 3 B+ while running detection task; (b) thermal image while idling; (c) thermal image 30seconds after running detection task; and (d) thermal image after 30min.

The test consists of running the detection model while using the LAN to send the output of the detection model, image size and location of rebars, to URICAB [10] platform. Temperature of the ARM platform measured using a Seek thermal camera. As shown in (Figure 6) the highest temperature recorded on the LAN chipset the maximum temperature while idling is 48°C. It was observed that 30 seconds after running the test the maximum temperature increased to 59°C. The temperature gradually increased for 30 seconds up to 61°C. After 1 minute, the temperature stabled until the end of the test.

CONCLUSIONS

In this study, an experimental comparison of some of the main aspects that affect the speed and accuracy of modern object detection models is performed to help practitioners choose an appropriate detection model when deploying detection models for rebar detection in GPR B-scan images on mobile and embedded system. The authors identified and presented model configurations to achieve a light weight rebar detection

models capable for being used on ARM platforms without sacrificing much accuracy GPRDETNet dataset. State of the art results is obtained for ARM platform-based detection task on GPRDETNet dataset by implementing rebar detection model using SSD as meta-architecture and Inception v2 as feature extractor. The obtained results indicate that SSD meta-architecture with MobileNet as feature extractor achieved overall mAP of 46 on GPRDETNet dataset and remained stable on stress test with default settings of the ARM platform, Raspberry Pi 3 B+. In term of accuracy, the SSD meta-architecture with Inception v2 feature extractor outperformed the MobileNet based detection model by 16.6% with overall mAP of 56 but due to insufficient allocated memory, instability was observed during stress test with default settings of the ARM platform.

New generation of ARM platforms, called Raspberry Pi 4 B, is recently released. The new generation is available in 1GB, 2GB, and 4GB configurations. As of the time of writing the paper, there is no stable official TensorFlow wheel for the new generation of ARM processors. As future work, the performance and capability of the new generation of ARM platforms in running detection models with more dense feature extractors needs to be investigated.

REFERENCE

- [1] N. Gucunski, B. Pailes, J. Kim, A. Hoda and K. Dinh, "Capture and quantification of deterioration progression in concrete bridge decks through periodical NDE surveys.," *Journal of Infrastructure Systems*, vol. 23, no. 1, p. B4016005, 2016, DOI: [https://doi.org/10.1061/\(ASCE\)IS.1943-555X.0000321](https://doi.org/10.1061/(ASCE)IS.1943-555X.0000321).

- [2] N. Gucunski, "Nondestructive testing to identify concrete bridge deck deterioration.," *Transportation Research Board*, vol. 1, no. 1, pp. 78-86, 2013, DOI: <https://doi.org/10.17226/22771>.
- [3] C. L. Barnes and J.-F. Trottier, "Effectiveness of ground penetrating radar in predicting deck repair quantities.," *Journal of infrastructure systems*, vol. 10, no. 2, pp. 69-76, 2004, DOI: [https://doi.org/10.1061/\(ASCE\)1076-0342\(2004\)10:2\(69\)](https://doi.org/10.1061/(ASCE)1076-0342(2004)10:2(69)).
- [4] T. Ch, T. Kind and M. Schubert, "Detection of air voids in concrete by radar in transmission mode," *Proc. 8th International Workshop on Advanced Ground Penetrating Rada*, p. 1-4, 2015, DOI: <https://doi.org/10.1109/IWAGPR.2015.7292650>.
- [5] L. L. Monte, D. Erricolo, S. Francesco and C. W. Michael, "Radio frequency tomography for tunnel detection.," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1128-1137, 2010, DOI: <https://doi.org/10.1109/TGRS.2009.2029341>.
- [6] J. Siow Wei and M. Hashim, "Accuracy of data acquisition approaches with ground penetrating radar for subsurface utility mapping," *RF and Microwave Conference (RFM)*, pp. 40-44, 2011, DOI: <https://doi.org/10.1109/RFM.2011.6168691>.
- [7] W. Shao, A. Bouzerdoun, S. L. Phung, B. Indraratna, C. Rujikiatkamjorn and L. Su, "Automatic classification of ground-penetrating-radar signals for railway-ballast assessment.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3961-3972, 2011, DOI: <https://doi.org/10.1109/TGRS.2011.2128328>.
- [8] P. A. Torrione, "Histograms of oriented gradients for landmine detection in ground-penetrating radar data," *IEEE Transactions on Geoscience and Remote Sensing*, pp.

- 1539-1550, 2014, DOI: <https://doi.org/10.1109/TGRS.2013.2252016>.
- [9] K. Dinh, N. Gucunski and T. H. Duong, "An algorithm for automatic localization and detection of rebars from GPR data of concrete bridge decks.," *Automation in Construction*, vol. 1, no. 89, pp. 292-298, 2018, DOI: <https://doi.org/10.1016/j.autcon.2018.02.017>.
- [10] P. Asadi, M. Gindy and M. Alvarez, "A Machine Learning Based Approach for Automatic Rebar Detection and Quantification of Deterioration in Concrete Bridge Deck Ground Penetrating Radar B-scan Images.," *KSCE Journal of Civil Engineering*, vol. 1, no. 1, pp. 1-10, 2019, DOI: <https://doi.org/10.1117/12.2228345>.
- [11] L. Qiao, Q. Yao, R. Xiaozhen and W. Qifu, "Identification of buried objects in GPR using amplitude modulated signals extracted from multiresolution monogenic signal analysis.," *Sensors*, vol. 15, no. 12, pp. 30340-30350, 2015, DOI: <https://doi.org/10.3390/s151229801>.
- [12] Y.-a. Cui, L. Wang and J.-p. Xiao, "Automatic feature recognition for GPR image processing.," *World Academy of Science, Engineering and Technology*, vol. 61, no. 1, pp. 176-179, 2010.
- [13] P. Edoardo, F. Melgani, M. Donelli, R. Attoui and M. D. V. De Vos, "Automatic detection and classification of buried objects in GPR images using genetic algorithms and support vector machines.," *Geoscience and Remote Sensing Symposium*, 2008. IGARSS 2008. IEEE International, vol. 2, no. 1, pp. II-525, 2008, DOI: <https://doi.org/10.1109/IGARSS.2008.4779044>.
- [14] J. Dai, Y. Li, K. He, and J. Sun. *R-fcn: Object detection via region-based fully*

convolutional networks.arXiv preprintarXiv:1605.06409, 2016.

- [15] S. Ren, K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*. In *Advances in neural information processing systems*, pages 91–99, 2016.
- [16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. *Ssd: Single shot multibox detector*. *European Conference on Computer Vision*, pages 21–37. Springer, 2016.
- [17] M. Everingham, L. Van Gool, C. KI Williams, J. Winn and A. Zisserman, "The pascal visual object classes (voc) challenge.," *International journal of computer vision*, vol. 88, no. 2, pp. 303-338, 2010.
- [18] P. Asadi and M. Gindy, "GPRDETNv1.0 Dataset," 24 5 2019. [Online]. Available: <https://github.com/PouriaAI/GPR-Detection.git>. [Accessed 1 4 2019].
- [19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. *Tensorflow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org, 1, 2015.
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [21] R. Girshick. *Fast r-cnn*. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.
- [22] Y. Li, H. Qi, J. Dai, X. Ji, and W. Yichen. *Translation-aware fully convolutional instance segmentation*. <https://github.com/daijifeng001/TA-FCN>, 2016.

- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 1 May 2014.
- [25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [27] . Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.
- [28] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [29] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2015.
- [30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *European Conference on Computer Vision*, pages 21–37. Springer, 2016.
- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

- [32] *P. J. Huber et al. Robust estimation of a location parameter. The Annals of Mathematical Statistics, 35(1):73–101, 1964..*
- [33] *. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4(2),2012.*
- [34] *. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. arXiv preprint arXiv:1512.04412, 2015.*
- [35] *T. Y. Lin and P. Dollar. "Ms coco api" 24 5 2019. [Online]. Available: <https://github.com/pdollar/coco>. [Accessed 1 4 2019].*
- [36] *Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer et al. "Speed/accuracy trade-offs for modern convolutional object detectors." IEEE conference on computer vision and pattern recognition, pp. 7310-7311. 2017.*
- [37] *C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567, 2015.*
- [38] *Asadi, P, & Gindy, M. (2019). "GPRDET v1.0 Dataset." <https://github.com/PouriaAI/GPR-Detection/>.*

Chapter 3

“A COMPUTER VISION BASED REBAR DETECTION CHAIN FOR AUTOMATIC PROCESSING OF CONCRETE BRIDGE DECK GPR DATA”

BY

Pouria Asadi¹; Mayrai Gindy²; Marco Alvarez³; Alireza Asadi⁴

is under review in Journal of Automation in Construction

¹ PhD Candidate, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: asadi@uri.edu

² Associate Professor, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: mayraig@uri.edu

³ Assistant Professor, Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881. Email: malvarez@cs.uri.edu

⁴ PhD Candidate, Department of Technical and Engineering, Azad University, Tehran, Iran. Email: alireza.asadi@azad.ac.ir

ABSTRACT

Manual processing of Ground Penetrating Radar (GPR) images is a very time-intensive task. The authors proposed a novel computer vision-based method for automatic detection of rebars in complex GPR images in highly deteriorated concrete bridge decks. The proposed detection model consists of a fine-tuned Histogram of Oriented Gradients feature descriptor, a Multi-Layer Perceptron for classification, and a post processing algorithm for eliminating false detections and labeling rebar in Region of Interest. State-of-art results are obtained on testing the method on real bridge deck GPR data and comparing the results with RADAN software. Overall accuracy of 89.4% is obtained on URIGPRv1.0 dataset, which is introduced in this paper. The proposed method is 54.35% more accurate comparing to the results obtained by RADAN software. The proposed classifier outperformed accuracy of a 3-layer convolutional neural network by 11.9%.

Keywords: Ground penetrating radar; Rebar detection; Automation; Image processing; Convolutional neural network; Deep learning; Bridge inspection

1 INTRODUCTION

Statistics from the Federal Highway Administration (FHWA) [1] show that 11% of the nation's bridges are rated as "structurally deficient" and over 30% of existing bridges have exceeded their 50-year design life, meaning that condition assessment and repair/rehabilitation programs will require substantial budget allocations in the near future. Common bridge inspection methods like chipping, drilling, and coring consist largely of time-consuming and subjective measures for quantifying deterioration of bridges.

Ground Penetrating Radar (GPR) has been successfully used in a wide range of applications including capture and quantification of deterioration progression in concrete bridge decks, evaluation of the deck thickness, measurement of the concrete cover, predicting deck repair quantities, void localization in concrete, underground utility tracing and mapping, land mine detection, optimization and assessment of railway ballast, and tunnel assessment [2, 3, 4, 5, 6, 7, 8, 9, 10].

Although GPR data collection is fast and efficient but interpretation of GPR data is a very time-consuming task and relies on the operator's decisions to provide accurate information and reliable results. Detecting reflection hyperbola-patterns which represent objects in GPR data is a necessary first-step in GPR data processing systems, with the purpose of localizing, extracting, and characterizing the rebar in GPR data. Extensive literature exists regarding processing GPR data for automatic localization of buried objects, but here just a few relevant and recent studies is mentioned. Dinh et al. [11] proposed an algorithm consisting of a Convolutional Neural Network (CNN) for hyperbola signatures and then implementing a CNN to locate potential rebars by retain the likely true positive,

and to discard likely false positive rebar detections. The overall accuracy of the method was found to be 99.60%. The main drawback of the proposed method is that, depending on the nature of problem a minimum 10,000 images per class is needed for the training of CNN. Asadi et al. [12] proposed a method based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration and creating a 3D deterioration map in concrete bridge decks. The value of F-measure was found to be 86.20%. The limitation of the proposed method is accuracy in highly deteriorated bridges. Qiao et al. [13] developed a general hyperbola pattern detection algorithm based on a novel method called multi-resolution monogenic signal analysis for processing GPR B-scans data. The reported F-measure for the method was 75%. Bouzerdoun et al. [9] focused on reducing background clutter in GPR data. The authors proposed a method to reduce the background clutter to improve object detection. The area under the ROC curve (AUC) is computed to measure the detection accuracy of the proposed method. AUC for large and small targets was found to be 98% and 97.9%, respectively. Cui et al. [14] proposed a feature recognition method based on center-surround difference detecting and implemented the Fuzzy Logic approach to classify computed hyperbolic signatures. The proposed method is sensitive to GPR data collection system settings. Pasolli et al. [15] approach for identification of buried objects in GPR data is based on localizing hyperbolic patterns using Genetic Algorithm for data classification which due to computationally complexity is a slow method, especially when employed with high-dimensional input data. The overall accuracy of the method was found to be 80%.

The authors proposed a dataset of 8,000 labeled GPR images cropped from bridge deck GPR images, named URIGPRv1.0 [16]. The proposed dataset allows researchers to develop and evaluate the performance of novel approaches in identification and labeling of objects in GPR B-scan images. The authors proposed a novel computer vision (CV)-based method for automatic detection of rebars in complex GPR images in highly deteriorated concrete bridge decks. The proposed method consisting of a fine-tuned Histogram of Oriented Gradients (HOG) feature descriptor, a Multi-Layer Perceptron (MLP) for data classification, a multi-scale pyramid sliding window object detector, and a post processing algorithm developed for reducing false detection rate and labeling exact location of rebar in Region of Interest (RoI). Overall, the contribution of this work includes:

- Proposing a novel detection chain yields a new state-of-the-art result in rebar detection in concrete bridge deck GPR B-scan images
- Optimizing HOG/LBPH feature computation and MLP model parameters for maximizing F-Measure in classification task
- Developing a post-processing algorithm for minimizing false RoI detections, removing non-top layer RoI, and extracting coordinates of rebar in RoI
- Introducing a GPR dataset with 8,000 hyperbola/hyperbola-free images, named URIGPRv1.0, was cropped from real concrete bridge deck GPR B-scan images.
- A computer program, named URICAB, is developed based on the proposed rebar detection chain for automating interpretation and quantification of GPR data

2 COMPONENTS OF DETECTION SYSTEM

This section gives an overview of the proposed machine learning based detection chain, which is summarized in Figure 1. In training branch effect of the HOG computation and classifier parameters on performance of binary image classifier is studied on URIGPRv1.0 dataset. The output of training branch of the proposed detection chain is a trained MLP binary image classification model. In detection branch, a multi-scale pyramid sliding window detector is developed to detect and propose RoIs in GPR images. To reduce false detection and locating “where” in a RoI a rebar resides, a post-processing algorithm is introduced.

Figure 1 The flowchart of the proposed detection model.

2.1 DATA COLLECTION

ANTENNA SETUP

GPR is a time-dependent technique that works based on principle of scattering of electromagnetic (EM) wave to locate subsurface objects. Physical properties that affect the survey are electrical conductivity and dielectric permittivity. When the EM wave passes into the new medium with different electromagnetic properties a part of the electromagnetic

wave is reflected back to the receiving antenna. The depth and shape of the reflecting interface can be determined based on physical properties of material and travel time of EM wave. Two factors need to be considered in GPR antenna selection: (1) Depth of penetration, which indicates how deep EM radiation can penetrate in concrete, and (2) Spatial resolution, which is the ability of the antenna to see two closely spaced objects separately located in a specific medium. As show in Figure 2, there is a tradeoff between spatial resolution and penetration depth: as the wavelength decreases inside the medium, spatial resolution increases; however, this limits penetration depth. According to AASHTO LRFD Bridge Design Specifications, the minimum clear spacing between parallel bars in a layer is 3.8cm and the depth of a concrete deck should not be less than 17.5cm [17]. At 1.6 GHz, the penetration depth for air dried moisture condition is 25cm and spatial resolution for saturated moisture condition is 8cm. Since minimum clear spacing between bars specified in the code and in concrete bridge deck can be less than this value, a higher resolution is preferred, but since increase of resolution reduces penetration depth then a 1.6GHz antenna (GSSI SIR System-3000) is recommended for data collection [12].

Figure 2 GPR frequency and antenna: (a) tradeoff between spatial resolution and penetration depth of EM wave; (b) 1.6GHz antenna mounted on a cart [12].

GPR Image Dataset

The authors introduced a dataset of 4,000 rebar reflection hyperbolas images; and 4,000 hyperbola-free (non-rebar) images cropped from bridge deck GPR images, named URIGPRv1.0 [16]. Images are cropped from GPR B-scan images collected from concrete deck bridges with low to significant amounts of deterioration. The hyperbola patterns appeared in the same orientation and against a wide variety of backgrounds. Every record in the dataset is cropped, labeled, and saved as a separate bitmap (BMP) image file. Figure 3 shows examples of records in URIGPR dataset. The URIGPR dataset contains both easy to detect sample images from regions with low levels of deterioration, which produces GPR images with minimal anomalies, and more challenging samples from highly deteriorated regions in bridge deck, which produces GPR images with low contrast and hard to detect reflection hyperbolas.

Figure 3 URIGPR dataset: (a) Sample images; (b) Sample normalized images; and (c) Non-reflection hyperbola pattern due to overlapping of reflection hyperbola.

2.2 Hand-Crafted Feature Descriptors

In computer vision and image processing feature detection includes methods for

computing abstractions of image information and generalizing the target (in this study reflection hyperbola patterns) in such a way that the same target produces the same (or as close as possible) feature descriptor when it appears under various conditions. In this study, performance of two hand-crafted features description methods in GPR B-scan image classification is studied: (1) Histogram of Oriented Histograms (HOG), which in the original paper pointed out its superiority in performance when compared to other popular object detection methods such as SIFT [18] and successfully applied to a number of classic computer vision problems such as Human Detection [18], Face Recognition [19], and Visual Classification of Coarse Vehicle Orientation for Autonomous Vehicles [20], and (2) Local Binary Pattern Histograms (LBP) [21], which theoretically and computationally is robust in terms of contrast level and grayscale variations. This property of LBP makes it a good candidate for detecting hyperbola patterns in grayscale GPR B-scan images with various contrast and brightness levels due to noise and deterioration. LBP feature descriptors has been widely used in field of face recognition to overcome challenges such as illumination and expression variations [22]. Unlike Convolutional Neural Networks (CNN) which requires a large number of images [29] for optimizing the weight and training the network, by implementing hand-crafted feature descriptors a large training set is not required. In addition, with hand-craft feature descriptors it's much easier to avoid descriptors driven by artifacts which is a challenge in complex GPR B-scan images in highly deteriorated concrete bridge decks.

HOG feature descriptors provide a concise representation of images fundamentally based on gradient vectors. Gradient vectors can be computed for every pixel of an image as defined in Equation 1:

$$\nabla \vec{f}_{(x,y)} = \frac{\partial f_{(x,y)}}{\partial x} \hat{i} + \frac{\partial f_{(x,y)}}{\partial y} \hat{j} \quad (1)$$

Where $\partial f / \partial x$ is the gradient in x direction and $\partial f / \partial y$ is the gradient in y direction. HOG feature descriptors are computed based on the gradient magnitude F and orientation θ at each pixel. For processing GPR B-scan images, when the background regions are darker than the target, gradient vectors will always tend to flow toward the targets, thus values of θ between 0 to 2π , may be germane to inaccuracy of the detection system. Considering n_β orientation angle bins, computed gradient vector for $n \times n$ pixel cells will align with the appropriate orientation bin. To increase illumination invariance and robustness of feature descriptors, cells are grouped into overlapped blocks then blocks are normalized. Each block consists of m^2 cells. Concentrating all cell histograms in a block creates a $m^2 \times n_\beta \times 1$ element block feature vector.

LBP feature descriptor, introduced by Ojala et al [21], provides very good results in terms of both discrimination performance and computation cost [22, 23]. The most important property of the LBP feature descriptor in real-world applications is its robustness to monotonic grayscale changes. It makes LBP feature descriptor a good candidate for detecting hyperbola patterns in grayscale GPR B-scan images with various contrast and brightness levels due to noise and deterioration. The LBP algorithm labels the pixels of an image by thresholding the n -by- n neighborhood of each pixel with the center value and considering the result as a binary number. Then the histogram of the labels can be used as a feature descriptor vector.

2.3 Multilayer Perceptron (MLP) Classifier

A Multilayer Perceptron (MLP) for classification feature descriptor of GPR images is implemented. The MLP model is a nonlinear function from a set of input variables $\{x_i\}$ to a set of output variables $\{o_k\}$ controlled by a vector w of adjustable parameters called weight vector. Fundamental elements of an MLP are (n) linear combination of the input values $x_1, x_2, \dots, x_D$ called “activation” as defined in Equation 2 [24]:

$$a_j = \sum_i^D w_{ji}^{(n)} x_i + w_{j0}^{(n)} \quad (2)$$

Where j corresponds to the number of units in each layer, and the superscript (n) indicates that the corresponding parameters are in n^{th} ‘layer’ of the network. The quantities $w_{ji}^{(n)}$, $w_{j0}^{(n)}$, and a_j are known as weights, biases, and activations, respectively. The implemented method for training of the network is Back-propagation algorithm. To improve the model’s generalization and avoid overfitting two techniques are implemented: (1) Early-stopping, and (2) Cross-validation.

Early-Stopping Method: The dataset is divided into three subsets: (i) Training (70%); (ii) Validation (15%); and Test (15%). The error on the validation subset is monitored during the training. During the initial steps of training the validation and training error normally decreases. However, when the network begins to overfit the data, the error on the validation subset begins to rise. When the validation error increases for a specified number of iterations, the training is stopped. Number of iterations is a critical parameter that is strongly problem-dependent. The authors suggest five iterations is adequate for this

problem. Increasing the number of iterations beyond five doesn't improve the performance of the model.

Cross-Validation Method: MLP classifier is trained using four-fold cross-validation which is a common approach [25]. All the training data is divided at random into four distinct subsets, trained the model using three subsets, and test the model on the remaining subset. The process of training and testing is then repeated for each of the four possible choices of the subset omitted from the training. The mean performance on the four omitted subsets is then an estimate of the generalization performance. Cross-validation technique has the advantage that allows a high proportion of the available training data being used for training, while making use of all the data points in estimating the generalization error.

2.4 Multi-scale Sliding Window Detector

A sliding window detector is developed to detect regions in a GPR image containing a reflection hyperbola pattern. The sliding window detector is a bounding box of fixed width and height that “slides” across an image (or a convolutional layer in a CNN) and extracts image patches. Sliding window classification is the dominant paradigm in object detection [26]. Due to rebar depth and size variance the reflection hyperbola patterns appear in different scales. As shown in Figure 4, a multi-scale pyramid approach is implemented to find objects in different sizes. For each of the sliding windows, feature descriptor vector of the part of image inside the windows is computed then the MLP classifier determines if the window is a RoI.

Figure 4 Utilizing an image pyramid (multi-scale representation) and sliding a window approach allows to find objects in GPR images at different scales.

Computational cost is a disadvantage of sliding window-based object detector. Increasing window and stride size makes it faster but at cost of decreased accuracy. Sliding window approach does not detect objects accurately unless sliding window and strides are small enough comparing to the input image. So, the parameters in the sliding window algorithm need to be optimized problem by problem

2.5 Outlier Filtering and Rebar Locator Algorithm

Rebar corrosion is among the most important deterioration detection mechanisms and is of highest concern to bridge engineers [4]. Top rebar layers have the highest potential for corrosion due to penetration of moisture through cracks in the covering surface, so obtaining precise information about top layer rebars has the highest priority for condition assessment of concrete bridge decks. A fine steel rebar mesh is a complete reflector of radar energy in a GPR system and acts like a metal sheet, so a tight mesh may completely disguise targets behind the top layer of rebar.

Since there is a higher density of detected hyperbola patterns close to the top layer of rebars, it brings the idea of implementing an adaptive polynomial outlier detector to remove non-reflection hyperbolas and false detections. The authors proposed a filter for removing non-top rebar reflection hyperbola RoIs. The proposed adaptive polynomial based filter detector works based on the idea that since in concrete deck GPR images the number of detected hyperbola patterns in the top layer region is more than in the other regions then the polynomial interpolation of coordinates of all detected RoIs is closer to the top layer rebars. By implementing an adaptive procedure based on computing vertical distance of each detected RoI to the polynomial interpolation of coordinates of detected RoIs and comparing the distance with a threshold, non-top layer RoIs is detected. The threshold is computed as a decaying function of average horizontal distance of detected RoIs. Then the algorithm extracts the location of rebar in each RoI based on the fact that, the location of rebar is the hyperbola peak, where the color intensity is maximum. Figure 5 shows pseudocode for the proposed outlier filtering and rebar locator.

Input: $\mathcal{S} = \langle (x_1, y_1, f_1), \dots, (x_N, y_N, f_N) \rangle$

Initialize: $d = 2, n = 3$

Iterate: *FOR* $j = 1$ *to* 3

$d = 0.75 * d$

$[P_0, P_1, P_2, \dots, P_n] = \text{Polynomial}(\mathcal{S}, n)$

$P(y_N) : P(x_N) \Rightarrow P_0 + P_1 x_N^1 + \dots + P_n x_N^n$

$D = d \times (\max(x) - \min(x)/N)$

FOR $k = 1$ *to* N

IF $|P(y_k) - y_k| < D$ *THEN*

$\mathcal{S} = \mathcal{S} - \{(x_k, y_k, f_k)\}$

END FOR

END FOR

Load (GPR Image)

FOR $k = 1$ *to* *Size* ($\mathcal{S}$)

FOR $i = x_i - \text{win}_{\text{size}}/2$ *to* $x_i + \text{win}_{\text{size}}/2$

FOR $j = y_j - (\frac{\text{win}_{\text{size}}}{2})$ *to* $y_j + (\frac{\text{win}_{\text{size}}}{2})$

If $I(x_i, y_j) > I(x_{i-1}, y_{j-1})$

Then Set $\mathcal{S}_k = (x_i, y_j)$

END FOR

END FOR

END FOR

Output: $\mathcal{S}$

Figure 5 Pseudocode for the proposed filtering algorithm.

2.6 Experimental Setup

Due to deterioration in bridge deck, many collected GPR images have poor contrast, as shown in Figure 3(a). In order to bring the image into a range that is more familiar or normal to the senses, the intensity values of the image are normalized. A very common preprocessing step in machine learning is to make the data have a Gaussian form with zero mean and unit variance using the following formula [27]:

$$I_{norm} = (I - I_{mean})/I_{std} \quad (6)$$

Where I is the raw input image, I_{norm} is the normalized image, I_{mean} is the mean of image intensities, and I_{std} is the standard deviation. Figure 3(b) shows sample normalized images in the URIGPR dataset.

The effect of various parameters in computing features descriptors is systematically studied to find the best feature descriptor computation parameters for rebar detection in GPR images. For this purpose, MLP classifiers with different combinations of feature descriptor computation parameters is trained. To determine the best MLP architecture (number of layers and neurons) several network architectures with various number of layers and neurons is trained and tested. In addition, effect of number of training samples on performance of the classifier is studied. The results obtained from the best classifier is compared with CNNs architectures: (1) Three convolution layers with a ReLU activation and followed by max-pooling layers, very similar to the architectures that Yann Leun et al. implemented for binary image classification [28]; (2) VGG16 and VGG19 which is pre-trained on ImageNet dataset [29].

As a result of pre-training, VGG model learned rich feature representations for a wide range of images in different object categories. VGG architecture is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a SoftMax classifier. The “16” and “19” stand for the number of weight layers [30]. Using the best image classifier obtained from previous steps, a multi scale pyramid sliding window detector is deployed to label RoIs in GPR images. To reduce the number of false positive detections and removing non-top layer rebars the adaptive polynomial based filter is applied on output of the RoI detector, then location of rebar is extracted in each RoI. The output of the proposed rebar detection chain is the coordinates of top-layer rebars in GPR image.

Next, performance of the proposed method is evaluated by processing bridge deck GPR data collected from three bridges with low to significant amounts of deterioration in deck. Finally, the obtained results from the proposed rebar detection chain is compared with GSSI RADAN v7 program. GSSI RADAN is a commercial software widely used for post-processing GPR data. To quantify detector performance Receiver Operating Characteristic (ROC) and F-Measure is used. In statistics, a ROC curve shows the possible tradeoff between a classifier’s true positive (TP) rate versus its false positive rate. The true positive rate is commonly referred to as “sensitivity”, and (1-false positive rate) is called “specificity” [31]. Accuracy of the classifier is measured by the area under the ROC (AUC), where for a perfect classifier $AUC = 1$ and for a random classifier $AUC = 0.5$.

3 DISCUSSION OF RESULTS

Details of the proposed detection chain is presented in this section. The effects of the various choices in different components of detection chain is systematically studied and presented.

HOG Feature Descriptor Computation

Throughout this section the obtained results are referred to “default” HOG computation parameters which has the following properties:

Figure 6 ROC curve for different parameters: (a) cell size; (b) block size; (c) block overlap; and (d) number of orientation bins.

mono color space with no gamma correction; gradients casted into 6 histogram bins based on their orientation in a range from 0 to 180 (un-signed); 24X24 pixel blocks of four 6X6 pixel cells; block overlap of 6 pixels; and 48X48 detection window. Figure 6 and Table 1 show the effects of the various parameters on overall detection performance.

Table 1 The effect of various HOG computation parameters on performance of the classifier

Parameter set	Cell(pixel)	Block(cell)	Overlap(cell)	Bins	Signed	Num. features	F-Measure
#1	4X4	4X4	1	6	No	864	0.806
#2	6X6	4X4	1	6	No	384	0.830
#3	8X8	4X4	1	6	No	96	0.814
#4	6X6	2X2	1	6	No	1176	0.783
#5	6X6	3X3	1	6	No	486	0.775
#6	6X6	4X4	2	6	No	864	0.894
#7	6X6	4X4	3	6	No	2400	0.812
#8	6X6	4X4	1	3	No	192	0.801
#9	6X6	4X4	1	9	No	576	0.801
#10	6X6	4X4	1	6	Yes	384	0.615
#11	6X6	4X4	1	12	Yes	768	0.679
#12	6X6	4X4	1	18	Yes	1152	0.576
#13	8X8	2X2	1	9	No	900	0.798

Effect of three different window sizes on performance of the classifier is studied. A 48X48 pixel window includes approximately 12 pixels of margin around the target on all four sides, a 64X32 pixel window includes approximately 4 pixels right/left margin and 20 pixels top/bottom margin around the hyperbola pattern, and a 128X48 pixel window, which is recommended for pedestrian detection [18], includes approximately 20 pixels right/left margin and 52 pixel top/bottom margin around the hyperbola pattern. The best result is obtained with the 48X48 pixel window. Changing the margin around the target from 12 pixels of margin on all four sides to 4 pixel right/left margin and 20 pixels top/bottom (64X32 detection window) decreased F-Measure by 3%. Changing the margin around the target from 12 pixels of margin on all four sides to 20 pixel right/left margin and 52 pixels top/bottom (128X64 detection window) decreased F-Measure by 5%. A 48X48 pixel window shows a better performance compared to a wider window (128X64) because of the negative effect of non-reflection hyperbola patterns in GPR image. The non-reflection hyperbolic patterns appeared in GPR images due to overlapping of reflection hyperbola tails in a dense rebar layer in reinforced concrete deck, as shown in Figure 3(c), produces misleading information to the classifier and decreases the overall performance of the classifier. So, although bigger margins may increase the detection performance in some problems by providing more information about the background, in this problem a 48X48 pixel window which limits the margin to 12 pixels on all four sides provided the best result. To find the best size for “cell”, performance of the classifier with 4X4, 6X6, and 8X8 pixel cell size, parameter set #1, #2, and #3 respectively, is evaluated and compared the result with the recommended parameters proposed by Dalal and Triggs for pedestrian detection

(parameter set #13). As shown in Figure 6(a), the best result is obtained with a 6X6 pixel cell. With a 6X6 pixel cell size, F-Measure increased by 4.01% and 2.98%, respectively.

Evaluation of the effect of different “block” sizes (parameter set #2, #4, and 5) on performance of classifier shows that a block consisting of 4X4 cells provides the best performance. As shown in Figure 6(b), by increasing the block size from 3X3 cell to 4X4 cell, F-Measure and AUC increased by 7.1% and 3.52%, respectively. By overlapping blocks, each cell appears multiple times in the final feature descriptor vector but normalized by a different group of neighboring cells. Specifically, the corner cells appear once, the other edge cells appear twice each, and the interior cells appear four times each. As shown in Figure 6(c), the highest accuracy obtained by overlapping each block with two cells (the blocks have “50%” overlap). In the proposed method by Dalal and Triggs for pedestrian detection the blocks have “50%” overlap. By increasing the overlap to three cells the F-Measure decreased by 2.17%.

To create a histogram, gradient vector is computed at each pixel and stored into equally divided orientation angle bins. The results showed that when there is a strong gradient right on the boundary of two bins, a very small change in the gradient angle has a drastic effect on the histogram. To reduce the negative effect of this effect on performance of the detection algorithm, contribution of each gradient vector located on the boundary of two orientation bins linearly is distributed between two closest bins. Effect of histogram orientation bin, θ , both in range of 0° to 180° (un-signed gradients) and 0° to 360° (signed gradients) is studied. As shown in Figure 6(d), using a fine orientation binning turns out to be an important factor in performance of the detector. The obtained results show that increasing number of orientation bins significantly improves performance up to about 6

orientation bins. Encapsulating gradient vectors into 3 bins reduced F-Measure and AUC by 3.49% and 1.41%, respectively, compared to the result obtained with 6 bins.

The obtained results showed that un-signed orientations perform significantly better than signed orientations for hyperbola detection in GPR B-Scan images because un-signed gradients produce feature vectors with higher magnitude in direction of hyperbola pattern, as shows in Figure 7, which provides more discriminative features between the hyperbola pattern and the background.

Figure 7 Visualization of HOG features: (a) unsigned histograms; and (b) signed histograms.

According to the obtained results using 12 signed (0° to 360°) bins instead of 6 un-signed (0° to 180°) bins drastically decreased F-Measure and AUC by 18.19% and 7.56%, respectively. The highest accuracy is obtained with 6 un-signed orientation bins. By increasing the number of bins to 9 un-signed bins, which is the proposed value by Dalal and Triggs for pedestrian detection, the F-Measure and AUC decreased by 3.49% and 2.84%, respectively. As shown in Table 1, the highest accuracy is achieved with the parameter set #6. The final HOG based feature vector consist of 864 elements.

There are a number of key parameters in BP algorithm need to be optimized problem by problem. First, the number of input units (based on the results from the previous section, the input layer consists of 864 units) and the number of the neurons in output layer, which for a binary classifier with a sigmoid activation function in output layer is always one neuron. The number of hidden layers and units are the two major experimental factors. The optimum number of neurons in the hidden layer is determined by trial/error procedure. Performance of MLP classifier with 1, 2, and 3 hidden layers is studied. A rule of thumb suggests that the size of the hidden layer should be somewhere between the input layer size and the output layer size [31]. Number of units in each hidden layer is selected proportional to the number of units in the input layer, “D”. As shown in Figure 8, the number of units in each hidden layer varies from $(0.125 \times D)$ to $(1.25 \times D)$.

Figure 8 The effect of MLP model architecture on performance of HOG based classifier.

The performance of the studied MLP model architectures is presented in Figure 8. The highest accuracy measures, F-Measure & AUC, is obtained with an MLP architecture consists of 2 hidden layers and 432 units for each hidden layer.

LBP Feature Descriptor Computation

Throughout this section results are referred to “default” LBP computation parameters which has the following properties:

Figure 9 ROC curve for different parameters: (a) windows size; (b) cell size; (c) number of neighbors and radius; and (d) histogram normalization.

mono color space with no gamma correction; Number of neighbors used to compute the LBP for each pixel, $N = 16$; 48X48 pixel window; four cell histograms of 12X12 pixel; L2 normalization applied to each LBP cell histogram; and radius of circular pattern, $R = 2$.

Figure 6 and Table 1 show the effects of the various parameters on overall detection performance.

A 48X48 pixel window includes 12 pixels of margin around the target on all four sides, a 48X24 pixel window includes 12 pixels right/left margin and no top/bottom margin around the hyperbola pattern, a 72X72 pixel window includes 24 pixels of margin around the target on all four sides and a 72X48 pixel window includes 12 pixels right/left margin and 24 pixels top/bottom margin around the hyperbola pattern. The best result is obtained with the 48X48 pixel window. Changing the margin around the target from 12 pixels of margin on all four sides to 12 pixels right/left margin and no top/bottom margin (48X24 pixel window) decreased F-Measure by 9.81%. Changing the margin around the target from 12 pixels of margin on all four sides to 12 pixels right/left margin and 24 pixels top/bottom (72X48 pixel window) decreased F-Measure by 7.26%. Changing the margin around the target from 12 pixels of margin on all four sides to 24 pixels of margin on all four sides (72X72 pixel window) decreased F-Measure by 3.15%.

Table 2 The effect of various LBP computation parameters on performance of LBP based classifier

Window (pixel)	Cell (pixel)		Neighbors	Radius	Normalization	Num. features	Recall Precision F-Score		
48X48	12X12	16	2	L2-norm	288	0.77	0.891	0.841	
48X24	12X12	16	2	L2-norm	144	0.692	0.808	0.745	
72X48	12X12	16	2	L2-norm	432	0.685	0.87	0.766	
72X72	12X12	16	2	L2-norm	648	0.767	0.836	0.8	

48X48	6X6	16	2	L2-norm	1152	0.689	0.786	0.734
48X48	8X8	16	2	L2-norm	648	0.743	0.866	0.8
48X48	16X16	16	2	L2-norm	162	0.764	0.825	0.793
48X48	24X24	16	2	L2-norm	72	0.784	0.779	0.781
48X48	12X12	8	1	L2-norm	160	0.777	0.81	0.793
48X48	12X12	8	2	L2-norm	160	0.791	0.791	0.791
48X48	12X12	8	3	L2-norm	160	0.709	0.778	0.742
48X48	12X12	16	3	L2-norm	288	0.774	0.857	0.814
48X48	12X12	16	2	L1-norm	160	0.764	0.843	0.801
48X48	12X12	16	2	L1-Sqrt	288	0.767	0.849	0.805
48X48	12X12	16	2	None	288	0.764	0.825	0.793

The results show a window with 1:1 aspect ratio provides the best performance for rebar detection in GPR images. One of the reasons that a 48X48 pixel window shows a better performance comparing to a wider window (72X72 pixel) is the negative effect of non-reflection hyperbola patterns in GPR image. The non-reflection hyperbolic patterns appeared in GPR images due to overlapping of reflection hyperbola tails in a dense rebar layer in reinforced concrete deck makes the marginal information dominant and produces misleading information to the classifier which leads to a lower F-measure. So, although bigger margins for LBP feature descriptors computation may increase the classification performance in some problems by providing more information about the background, in hyperbola pattern detection in GPR data a 48X48 pixel window which limits the margin to 12 pixels provides the best result.

To find the best size for “cell”, performance of the classifier with various sizes is evaluated. Selecting larger cell sizes allows to collect information over larger regions. However, increasing the cell size causes data loss (local details). Finding the best cell size for each problem is an important task. Performance of the classifier with 6X6, 8X8, 12X12, 16X16, and 24X24 pixel cell size is evaluated. As shown in Table 1, the best result is obtained with a 12X12 pixel cell. The experimental results show that either increasing the cell size to 16X16 and 24X24 or decreasing the cell size to 6X6 and 8X8 reduced the performance of classifier. Interestingly, despite creating a longer LBP feature vector through dividing the image to finer cells (decreasing the cell size to 6X6 and 8X8 pixel) a lower performance in classification is observed. By decreasing the cell size to 6X6 and 8X8 the F-Measure decreased by 11.14% and 3.15%, respectively. Table 1 shows the effect of cell size on performance of the classifier.

Number of neighbors, N , and Radius of circular pattern, R , are two important parameters used to compute the LBP for each pixel in the input image. The group of surrounding pixels (neighbors) is selected from a circularly symmetric pattern around each pixel, as shown in Figure 10(a), by increasing the number of neighbors greater detail around each pixel will be encoded. Radius of circular specifies the distance of surrounding pixels to each reference pixel so by increasing the radius detail over a larger spatial scale will be captured. According to Kambi Beli et al., $N = 8$ with $R = 1$ are commonly values and provides very good results to solve certain face recognition problems, such as illumination and expression variations.

Figure 10 LBP feature descriptors: (a) The circular neighborhoods; and (b) an example of computing the LBP representation from the original input image

LBP feature descriptors is computed based on different values for N and P . As shown in Figure 10(c) and Table 2, with $N = 8$, increasing the distance of surrounding pixels reduces performance of the classifier. Using $N = 16$ and $R = 2$ for computing LBP features descriptors instead of common values ($N = 8$ and $R = 1$) improved the F-Measure by 4%.

Contrast is a property of texture usually regarded as a very important cue for human vision system. In this problem, due to deterioration in concrete bridge deck there are hyperbola patterns with very low contrast in GPR images. As shown in Figure 10(b), LBP operator by itself totally ignores the magnitude of gray level differences. LBP based features descriptor which is a purely gray-scale invariant texture operator may waste useful information. Although this property may be problematic in some applications but in this task which due to lack of number of training images, it reduces the need of training the MLP using GPR images with various contrasts.

Effect of three methods for normalization of LBP cell histogram is studied. The implemented normalizing schemes are: (i) $L1$ -norm; (ii) $L1$ -sqrt; and (iii) $L2$ -norm which

is the most widely used norm scheme in engineering and science. The results indicate that L1-norm, L1-sqrt have very similar performance. Applying L2-norm to each LBP cell histogram improved performance of the classifier by 4.16% comparing to the classifier without normalization of LBP cell histogram.

There are a number of key parameters in BP algorithm. First, the number of input units (based on the obtained results from the previous section, MLP consists of 288 input units) and the number of the neurons in output layer, which is one neuron for a binary classifier with sigmoid activation function. The number of hidden layers and units are the two major experimental factors that needs to be studied for each specific problem. Number of neurons in the hidden layer is determined by trial/error procedure. Performance of the classifier with 1, 2, 3 and 4 hidden layers is studied. A rule of thumb suggests that the size of the hidden layer should be somewhere between the input layer size and the output layer size. Number of units in each hidden layer is selected proportional to the number of units in the input layer, “D”. The number of units in each hidden layer varies from $(0.125 \times D)$ to $(1.5 \times D)$. The performance of the studied MLP architectures is presented in Figure 11. The highest accuracy with F-Measure = 0.83 is obtained with three hidden layers and 288 units for each hidden layer.

Figure 11 The effect of MLP architecture on performance of LBP based classifier.

Hand-crafted Feature Descriptor/MLP vs. CNN Classifier

Performance of the best hand-crafted feature descriptors, which is a HOG feature descriptor vector with 864 elements, is compared with three different CNN models for image classification. Unlike HOG based classifier which is based on computing hand-crafted feature descriptors, CNN is a trainable feature classifier. This means CNN extracts the features at each layer by getting Gaussian filter responses and weighing them accordingly to the labels.

As shown Figure 12(a), the fine-tuned HOG/MLP based classifier outperforms a three-layer CNN model very similar to the architectures that Yann Leun et al. implemented for binary image classification, by 11.51%. Implementing VGG 16 architecture pre-trained on ImageNet dataset increased F-score by 8.01%, comparing to our three-layer CNN.

Figure 12 (a) F-measure comparison: (a) Hand-crafted feature descriptors and convolutional neural network on URIGPRv1.0 dataset; (b) Effect of number of training samples on performance of the classifier.

Number of Training Samples

Number of sample images for creating an MLP classifier is entirely dependent on the domain and the quality of the data samples. Rules based on the degree of freedom in an MLP have been proposed for selecting the number of training data, e.g. “The number of parameters in the MLP should be significantly less than the number of the training data” [32]. The authors approach to find the optimum number of training samples is to plot the performance of MLP against the size of the training data. Training performance and generalization behavior of MLP is investigated with a fixed MLP architecture and various number of training samples. For all cases, the training data processed with the MLP architecture $864 : 432 : 432 : 1$, which is obtained from previous section. The highest accuracy of the MLP is obtained with 8,000 training samples, as shown in Figure 12(b). So far it is observed that a fine-tuned HOG/MLP based classifier provides the highest accuracy for binary classification GPR B-scan images.

For each of the pyramid sliding windows, feature descriptor vector of the part of image inside the windows is computed then the MLP binary image classifier determines if the window a ROI or not. The developed pyramid image sliding window detector has three parameters: (i) Step size, which indicates how many pixels need to be skipped in both the x and y direction; (ii) Scale, which controls how much the image need to be resized at each scaling step; and (iii) Range, which is minimum and maximum values for scaling GPR images.

Figure 13 Output of detection system: (a) before applying the filter; and (b) after applying the filter.

Best results are achieved with the following values for these parameters: (i) Step size: 6 Pixels; (ii) Scale: 0.2; and (iii) Range: 0.8 to 1.2. For reducing the number of false ROI detections, removing non-top layer ROI and finally extracting coordinate of rebar in ROI the developed post-processing algorithm (Figure 5) is applied on to the output of ROI detector. The obtained results indicate that implementation of the proposed post-processing algorithm decreases the false positive rate and improves overall performance of the proposed rebar detection chain by 8.33%. Figure 13 shows output of the proposed rebar detection chain in concrete bridge deck GPR B-scan images.

Performance of the proposed rebar detection chain in this paper is tested on three bridges and the results is compared with output from the GSSI RADAN v.7 program. Test data set

#1, #2, and #3 is collected from three bridges with different conditions of deck. Based on the latest inspection reports test set #1, #2, and 3# collected from bridges with “5/10 (Fair)”, “8/10 (Very Good)”, and “6/10 (Satisfactory)” conditions, respectively, according to the National Bridge Inventory (NBI) condition ratings. As show in Table 3, filtering false positive detections increased F-Measure of the proposed detection chain by 5.8%, 11.8%, and 8% for test set #1, #2, and #3, respectively.

Table 3 Comparison of the performance of the proposed rebar detection chain in this paper with GSSI RADAN v.7 program

	Test set #1		Test set #2		Test set #3	
	(Singleton Bridge)		(Ramp BB Bridge)		(Potowomut Bridge)	
	<u>The proposed</u>	<u>RADAN</u>	<u>The proposed</u>	<u>RADAN</u>	<u>The proposed</u>	<u>RADAN</u>
	<u>Method</u>		<u>Method</u>		<u>Method</u>	
Precision	0.818	0.714	0.931	0.64	0.933	0.567
Recall	0.545	0.357	0.844	0.485	0.667	0.405
F-Measure	0.655	0.476	0.885	0.551	0.778	0.472

The performance of the proposed method is 37.61%, 60.62%, and 64.83% more accurate than the result obtained by GSSI RADAN for test set #1, #2, and #3, respectively.

A computer program is developed based on the proposed rebar detection chain for automating interpretation and quantification of concrete bridge deck using GPR data, named “URICAB”. As shown in Figure 14, the program provides a graphical user interface for importing GPR B-scan images into the program. The program quantifies and visualizes the deterioration in bridge deck.

Figure 14 URICAB program's user interface: (a) Automatic detection of rebars and (b) Visualization of deterioration.

4 CONCLUSIONS

In this work, a Computer Vision-based rebar detection chain for automatic processing of concrete bridge deck GPR images is proposed. It is observed that the proposed rebar detection chain consisting of a fined-tuned HOG/MLP based binary image classifier which is trained on URIGPR dataset and applying a post-processing algorithm provides very good results in automatic processing of bridge deck GPR images and outperforms GSSI RADAN program. The performance of the proposed rebar detection chain is 54.35% more accurate than the result obtained by GSSI RADAN in deteriorated bridge decks. Performance of

HOG feature descriptors based MLP classifier which is a hand-crafted features descriptor computation method and CNNs based model is compared in this study. The obtained experimental results indicate that for classification of gray-scale GPR B-scan images a HOG/MLP classifier outperforms all studied CNN models on URIGPR dataset. Some False Negative detections of hyperbola pattern in highly deteriorated regions are observed in the output of the proposed model which may be improved by using more images from highly deteriorated regions in training stage. The output of the proposed detection chain is the coordinates of rebar and intensity of GPR image at location of rebar. This information can be used as an input for a computer vision based and system for condition assessment of bridge decks.

Future studies can be conducted in two direction: (i) Evaluating performance of very deep CNN architectures for image classification on URIGPR dataset; and (ii) Studying capability of other approaches for detecting objects in GPR data including utilizing Region-based Convolution Neural Networks (Faster R-CNN) and Single Shot Multi-Box Detector for rebar detection in GPR data. The authors planning to perform further studies to investigate performance of the proposed rebar detection chain on bituminous wearing surface bridge decks.

5 REFERENCES

- [1] ASCE, "Report Card for America's Infrastructure," 2013. [Online]. Available: <http://www.infrastructurereportcard.org/>. [Accessed 1 4 2019].
- [2] Martino, N., Maser, K., Birken, R. and Wang, M., "Quantifying bridge deck corrosion using ground penetrating radar". *Research in Nondestructive Evaluation*, vol. 27, no. 2,

pp. 112-124, 2016, DOI: <https://doi.org/10.1080/09349847.2015.1067342>.

[3] N. Gucunski, B. Pailes, J. Kim, A. Hoda and K. Dinh, "Capture and quantification of deterioration progression in concrete bridge decks through periodical NDE surveys". *Journal of Infrastructure Systems*, vol. 23, no. 1, pp. B4016005, 2016, DOI: [https://doi.org/10.1061/\(ASCE\)IS.1943-555X.0000321](https://doi.org/10.1061/(ASCE)IS.1943-555X.0000321).

[4] N. Gucunski, "Nondestructive testing to identify concrete bridge deck deterioration". *Transportation Research Board*, vol. 1, no. 1, pp. 78-86, 2013, DOI: <https://doi.org/10.17226/22771>.

[5] C. L. Barnes and J.-F. Trottier, "Effectiveness of ground penetrating radar in predicting deck repair quantities". *Journal of infrastructure systems*, vol. 10, no. 2, pp. 69-76, 2004, DOI: [https://doi.org/10.1061/\(ASCE\)1076-0342\(2004\)10:2\(69\)](https://doi.org/10.1061/(ASCE)1076-0342(2004)10:2(69)).

[6] T. Ch, T. Kind and M. Schubert, "Detection of air voids in concrete by radar in transmission mode". *Proceedings of the 8th International Workshop on Advanced Ground Penetrating Radar*, pp. 1-4, 2015, DOI: <https://doi.org/10.1109/IWAGPR.2015.7292650>.

[7] J. Siow Wei and M. Hashim, "Accuracy of data acquisition approaches with ground penetrating radar for subsurface utility mapping". *RF and Microwave Conference (RFM)*, pp. 40-44, 2011, DOI: <https://doi.org/10.1109/RFM.2011.6168691>.

[8] P. A. Torrione, "Histograms of oriented gradients for landmine detection in ground-penetrating radar data". *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1539-1550, 2014, DOI: <https://doi.org/10.1109/TGRS.2013.2252016>.

[9] A. Bouzerdoun, F. H. C. Tivive and C. Abeynayake, "GPR Target Detection by Joint Sparse and Low-Rank Matrix Decomposition" *IEEE Transactions on Geoscience and*

Remote Sensing, vol. 57, no. 5, pp. 2583-2595, 2019, DOI: <https://doi.org/10.1109/TGRS.2018.2875102>.

[10] L. L. Monte, D. Erricolo, S. Francesco and C. W. Michael, "Radio frequency tomography for tunnel detection". *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1128-1137, 2010, DOI: <https://doi.org/10.1109/TGRS.2009.2029341>.

[11] K. Dinh, N. Gucunski and T. H. Duong, "An algorithm for automatic localization and detection of rebars from GPR data of concrete bridge decks". *Automation in Construction*, vol. 1, no. 89, pp. 292-298, 2018, DOI: <https://doi.org/10.1016/j.autcon.2018.02.017>.

[12] P. Asadi, M. Gindy and M. Alvarez, "A Machine Learning Based Approach for Automatic Rebar Detection and Quantification of Deterioration in Concrete Bridge Deck Ground Penetrating Radar B-scan Images". *KSCE Journal of Civil Engineering*, vol. 23, no. 6, pp. 2618–2627, 2019, DOI: <https://doi.org/10.1117/12.2228345>.

[13] L. Qiao, Q. Yao, R. Xiaozhen and W. Qifu, "Identification of buried objects in GPR using amplitude modulated signals extracted from multiresolution monogenic signal analysis" *Sensors*, vol. 15, no. 12, pp. 30340-30350, 2015, DOI: <https://doi.org/10.3390/s151229801>.

[14] Y.-a. Cui, L. Wang and J.-p. Xiao, "Automatic feature recognition for GPR image processing" *World Academy of Science, Engineering and Technology*, vol. 61, no. 1, pp. 176-179, 2010.

[15] P. Edoardo, F. Melgani, M. Donelli, R. Attoui and M. D. V. De Vos, "Automatic detection and classification of buried objects in GPR images using genetic algorithms and support vector machines" *IEEE Geoscience and Remote Sensing Symposium*, vol. 2, no. 1,

pp. II-525, 2008, DOI: <https://doi.org/10.1109/IGARSS.2008.4779044>.

[16] P. Asadi and M. Gindy, "URIGPRv1.0 Dataset" 2019. [Online]. Available: <https://github.com/PouriaAI/GPR-Detection.git>. [Accessed 14 2019].

[17] AASHTO, "LRFD Bridge Design Specifications: Customary US Units, Washington, D.C.: American Association of State Highway and Transportation Officials", 2012, ISBN: 9781613447529 1613447523.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection". *IEEE Computer Vision and Pattern Recognition*, vol. 1, no. 1, pp. 886-893, 2005, DOI: <https://doi.org/10.1109/CVPR.2005.177>.

[19] O. Deniz, G. Bueno, J. Salido and F. De la Torre, "Face recognition using histograms of oriented gradients". *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1598-1603, 2011, DOI: <https://doi.org/10.1016/j.patrec.2011.01.004>.

[20] P. E. Rybski, D. Huber, M. Daniel D. and R. Hoffman, "Visual classification of coarse vehicle orientation using histogram of oriented gradients features". *IEEE Intelligent vehicles symposium*, vol. 1, no. 1, pp. 921-928, 2010, DOI: <https://doi.org/10.1109/IVS.2010.5547996>.

[21] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". *IEEE Transactions on pattern analysis and machine intelligence*, vol. 7, no. 1, pp. 971-987, 2002, DOI: <https://doi.org/10.1109/TPAMI.2002.1017623>.

[22] T. Ahonen, H. Abdenour and M. Pietikäinen, "Face recognition with local binary patterns". *European Conference on Computer Vision*, vol. 1, no. 12, pp. 469-481, 2004,

DOI: https://doi.org/10.1007/978-3-540-24670-1_36.

[23] B. Kambi, I. Laure and C. Guo, "Enhancing Face Identification Using Local Binary Patterns and K-Nearest Neighbors". *Journal of Imaging*, vol. 3, no. 17, p. 37, 2017, DOI: <https://doi.org/10.3390/jimaging3030037>.

[24] C. M. Bishop, "Pattern recognition and machine learning", New York: springer, 2006, ISBN: 978-0-387-31073-2.

[25] R. Daniěl, L. M. Collins and J. M. Malof, "On Choosing Training and Testing Data for Supervised Algorithms in Ground Penetrating Radar Data for Buried Threat Detection". *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 497-507, 2017, DOI: <https://doi.org/10.1109/TGRS.2017.2750920>.

[26] Dollar, P., Wojek, C., Schiele, B. and Perona, P., "Pedestrian detection: An evaluation of the state of the art". *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743-761, 2017, <https://doi.org/10.1109/TPAMI.2011.155>.

[27] S. Patro and K. K. S. Sahu, "Normalization: A preprocessing stage". *arXiv preprint*, vol. 1, no. 1, pp. 1503.06462, 2015, DOI: <https://doi.org/10.17148/IARJSET.2015.2305>.

[28] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition.", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, DOI: <https://doi.org/10.1109/5.726791>.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database". *IEEE Computer Vision and Pattern Recognition*, vol. 1, no. 1, pp. 248-255, 2009, DOI: <https://doi.org/10.1109/CVPR.2009.5206848>.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale

image recognition". IEEE Asian Conference on Pattern Recognition, vol. 1, no. 1, pp. 1409-1556, 2014, DOI: <https://doi.org/10.1109/ACPR.2015.7486599>.

[31] A. Blum, "Neural networks in C++: an object-oriented framework for building connectionist systems". New York: John Wiley & Sons, Inc., 1992, ISBN: 13: 978-0471538479.

[32] E. B. Baum and D. Haussler, "What size net gives valid generalization?". *Neural Computation*, vol. 1, no. 1, pp. 81-90, 1989, DOI: <https://doi.org/10.1162/neco.1989.1.1.151>.

Chapter 4

“Real-Time Onsite Processing of GPR Data on ARM-Based Mobile Devices Using Modern Convolutional Neural Networks”

BY

Pouria Asadi¹; Mayrai Gindy²; Marco Alvarez³

Will be submitted to Journal of Automation in Construction

¹ PhD Candidate, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: asadi@uri.edu

² Associate Professor, Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI 02881. Email: mayraig@uri.edu

³ Assistant Professor, Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881. Email: malvarez@cs.uri.edu

ABSTRACT

Real time automatic detection of rebars in concrete bridge deck ground Penetrating Radar (GPR) data is addressed. Automatic rebar detection in Ground Penetrating Radar (GPR) data is the basic step in an automatic system for GPR-based condition evaluation of highway bridges. Achieving real-time performance on ARM-based platforms for onsite applications still remains a challenge. Development a cost-effective, light-weight, and energy-efficient system using an embedded ARM-based platform for real-time onsite rebar detection in ground penetrating radar images is goal of this paper. The goal of this study is to serve as a reference for selecting a deep learning-based detection architecture that provides the right accuracy, speed, and memory usage balance for real-time detection of rebars on the latest version of ARM-based platforms, named: Raspberry Pi 4 Model B. Various ways to trade accuracy for speed and memory usage in convolutional neural network (CNN)-based detection models is investigated. A unified implementation of the Faster R-CNN and SSD meta-architecture-based models is implemented to evaluate the accuracy, speed, and memory usage trade-off by using various CNN backbones and varying other training parameters. A deep learning-based detector is presented that can be deployed on the latest version of ARM-based platforms. State of the art results is obtained on GPRDETNet detection task by implementing rebar detection model using Faster R-CNN with ResNet 101 CNN backbone.

Keywords: Inception, MobileNets, Single-Shot Detector, Rebar Detection, GPR, ARM

1. INTRODUCTION

Statistics from the Federal Highway Administration (FHWA) [1] show that 11% of the nation's bridges are rated as "structurally deficient" and over 30% of existing bridges have exceeded their 50-year design life, meaning that condition assessment and repair/rehabilitation programs will require substantial budget allocations in the near future. Common bridge inspection methods like chipping, drilling, and coring consist largely of time-consuming and subjective measures for quantifying deterioration of bridges. Automatic rebar detection in Ground Penetrating Radar (GPR) data is the basic step in GPR-based condition assessment of bridges and has been studied for many years. However, achieving real-time performance on computation resource limited embedded devices for onsite applications still remains an open challenge.

Extensive literature exists regarding detection of objects in GPR data. Dinh et al. [2] proposed an algorithm consisting of a convolutional network for detection hyperbola patterns in GPR B-scan images. Then a Convolutional Neural Network (CNN) was trained to extract location of potential rebars by retaining the likely true positive, and discarding likely false positive detections. Memory usage and speed of detection was not reported. Asadi et al. [3] proposed a method based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration and creating a 3D rebar location and deterioration map in bridge decks. The proposed application is design for Window operating system and is not supported by open source operating systems. Qiao et al. [4] proposed a hyperbola pattern detection method for processing GPR B-scans based on multi-resolution monogenic signal analysis. Trade-off between accuracy, speed, and memory usage is not reported in that

study. Bouzerdoun et al. [5] proposed a method for reducing background clutter in GPR data. The authors proposed an algorithm to reduce the background clutter to increase accuracy and reduce false positive detections. Cui et al. [6] proposed a feature identifications algorithm based on center-surround difference for detecting and implemented the Fuzzy Logic method to classify computed hyperbolic patterns. The proposed method is highly sensitive to GPR data acquisition system settings and computationally expensive which limits the application of that on devices with limited memory. Pasolli et al. [7] approach for detecting buried objects in GPR B-scan images is based on localizing hyperbolic patterns using Genetic Algorithm. This method is a computationally expensive algorithm and limits the application of that on ARM-based devices.

A lot of progress has been made in recent years on CNN-based object detection and segmentation tasks due to advances in development of parallel computing platforms which are suitable for computationally intensive optimization algorithms that are used in training and deployment of CNN-based detection and segmentation models. Modern object detectors based on deep CNNs, such as Faster Region-based Convolutional Networks (R-CNN) [8] and Single Shot Multi-Box Detector (SSD) [9], are now accurate enough to be implemented in commercial applications in field of autonomous cars (e.g., traffic lane detection, traffic sign detection, etc.) and some have been shown to be fast enough for being used on mobile devices. However, it can be difficult for practitioners to decide which model is best suited for detecting rebars in GPR B-scan images on ARM-based platforms. Advantages of a rebar detection architecture suitable for being used on ARM processors

over Compute Unified Device Architecture (CUDA) processors are as follow: (1) significantly lower cost comparing to GPU-based platforms, (2) much lower power demand, and (3) better thermal performance, as shown in Table 1. These main advantages make ARM-based platforms a suitable option for real-time onsite GPR-based evaluation of bridges using manual/on-cart, robotic, and drone systems.

Table 1 Comparison of GPU and ARM-based Platforms for rebar detection

Platform	PC (GPU)	Raspberry Pi 3 Model B+	Raspberry Pi 4 Model B
Computation	NVIDIA	Broadcom BCM2837B0	Broadcom BCM2711
Unit	GTX 1080 Ti	Quad core Cortex-A53	Quad core Cortex-A72 (ARMv8-A)
Cores/Base Freq.	3584 CUDA Cores/1.6GHz	4 Cores /1.4GHz	4 Cores/1.5GHz
Power	280W	6.4W	7.6W
System Memory	16GB DDR4	1GB LPDDR2 SDRAM	4GB LPDDR4-2400 SDRAM
Storage	SSD	microSD card	microSD card
Cooling	Cooling Fan + Heatsink	Passive (Heatsink)	Passive (Heatsink)
Operating System	Ubuntu	Raspbian	Raspbian

Library	TensorFlow	TensorFlow	TensorFlow
	GPU		
Price	\$2000	\$35	\$75

Standard method for evaluating accuracy of detection models, such as mean average precision (mAP) which widely used in the literature, does not provide memory usage and running time in details. Furthermore, they don't give a full picture of the accuracy, speed, and memory usage trade-off. For deployments of CNNs based detection models on mobile and embedded devices, other parameters like speed and memory usage are also critical.

In this study, the authors seek to provide a comprehensive picture of the accuracy, speed, and memory usage trade-off of modern CNNs-based detection models for rebar detection in GPR data.

Since test-time is more important for practitioners and for simplicity, only test-time performance is studied not how long these models take to train. It is impractical to compare every recently proposed CNN-based detection model. Since many of the recent and state of the art approaches have converged on a common methodology, it allows the authors to implement and compare a large number of CNN-based detection models in a unified manner. In this paper, the authors created implementations of the Faster R-CNN and SSD meta-architectures.

To summarize, the main contributions in this study are as follows:

- State of the art results is obtained for detection task on GPRDET detection task by implementing rebar detection model using Faster R-CNN with ResNet 101 CNN backbone.

- A flexible and unified implementation of two meta-architectures (Faster R-CNN and SSD) in Google’s TensorFlow fine-tuned on GPRDETNet is developed to perform comprehensive experiments that evaluate the accuracy, speed, and memory usage trade-off.
- The obtained result shows that using fewer box proposals for Faster R-CNN meta-architecture can significantly improve detection speed without a big loss in accuracy on GPRDETNet detection task.
- Sweet spots on the accuracy, speed and memory usage trade-off based on Accuracy vs Time on GPRDETNet detection task is presented.
- Several of the meta-architecture and CNN backbones combinations that is reported in this paper have never appeared before in literature.

2. META-ARCHITECTURES

Deep convolutional neural networks have become the leading method for various computer vision tasks. The R-CNN method by Girshick et al. [10] is considered as one of the first modern applications of CNN-based object detection systems. R-CNN method took the straightforward approach of cropping externally computed class-agnostic bounding box proposals out of an input image and running a classifier on these proposals. Depending on the performance of implemented class-agnostic proposal generation algorithm and number of generated proposals, this method can be computationally expensive. Fast R-CNN [11] improved detection speed of R-CNN method by sharing the computation load through feeding image to the network only once and using extracted features of one of the intermediate layers for cropping. In Fast R-CNN, the region proposals are generated separately by another algorithm, called, “Selective Search” algorithm, which is

computationally expensive and fairly slow, that was found to be the bottleneck of the overall model architecture. Development of Faster R-CNN [8] method is based on the idea of making generation of proposals an almost computationally cost-free step by reusing those same CNN results for region proposals instead of running a separate class-agnostic proposal generation branch in the model. In this method a single CNN is trained to perform both region proposal generation and classification tasks.

In Faster R-CNN method, there is a collection of bounding boxes overlaid on the image at different spatial locations with various scales and aspect ratios, called “anchors”. A model with two output heads is then trained to perform two predictions for each anchor: (1) classification: a discrete class prediction, and (2) regression: a continuous prediction of an offset by which the anchor needs to be modified to fit the ground truth bounding box. In this method the loss function sums up the cost of classification and bounding box prediction and it needs to be minimized. If there is the best matching ground truth bounding box for each anchor a , then anchor a is labeled as a “positive anchor” and two properties are assigned to anchor a : (1) a class label $y_a \in \{1 \dots N\}$ and (2) a vector encoding of box b with respect to anchor a , called the box encoding, $\phi(b_a; a)$. If no matching ground truth bounding box is found, anchor a is labeled as a “negative anchor” and the class is labeled to $y_a = 0$.

If for the anchor a a box encoding, $f_{location}(I; a, \theta)$ and a corresponding class, $f_{class}(I; a, \theta)$ is predicted, where I is the input image and θ the model parameters, then the loss function for anchor a is defined as a weighted sum of combination of a classification loss and a location-based loss, as shown in (Equation 1):

$$\mathcal{L}(a; I, \theta) = \alpha \cdot 1(a \text{ is positive}) \cdot l_{location}(\phi(b_a; a)) - f_{location}(I; a, \theta) +$$

$$\beta \cdot l_{class}(y_a, f_{class}(I; a, \theta)) \quad (1)$$

Where α, β are model parameters balancing localization loss and classification loss, respectively. In training phase, the loss function (Equation 1) is averaged over anchors and minimized with respect to θ .

In this study, three recent meta-architectures are investigated: Faster R-CNN [8] and SSD [9]. While these methods were originally presented with a particular CNN backbone, the authors in this study review these two methods, decoupling the choice of meta-architecture from CNN backbone so that effect of various feature extractors is investigated with Faster R-CNN or SSD to obtain the best option for detecting rebars in B-scan images on ARM-based platform.

Faster R-CNN

Faster R-CNN detection method consists of two main stages: (1) Region Proposal Network (RPN): input images are processed by a CNN backbone, and features at selected intermediate layers are used to predict class-agnostic bounding box proposals. The loss function for this first stage takes the form of (Equation 1). (2): classification: boundary box proposals are used to crop features from the same feature map which are subsequently fed to the remainder of the CNN backbone in order to predict a class for each proposal. The loss function for this stage takes the form of (Equation 1).

SSD

SSD meta-architecture uses different activation maps (multiple-scales) for prediction of classes and bounding boxes. More specifically, SSD uses VGG16 [9] to extract feature maps. Then it detects objects using the Conv4_3 layer. In this study, the term SSD refers to meta-architectures that use a single feed-forward convolutional network

to directly predict classes and bounding box proposals without requiring a second stage per-proposal classification operation.

Figure 1 Diagrams of the Faster RCNN and SSD detection meta-architectures [19].

3- EXPERIMENTAL SETUP

The benchmarks such as ImageNet [12] and COCO [13] made comparing performance of detection models with respect to accuracy easier. However, when it comes to study performance of detection models with respect to accuracy, speed, and memory usage, which is necessary for deployment of these models on resource limited platforms such as ARM-based platforms, as shown in Table 1, it's more difficult to make a point-by-point comparison. To facilitate the process of benchmarking, including studying performance of different feature extractors, meta-architectures, and fine-tuning detection model on GPRDETNet for rebar detection, an object detection platform is implemented in Google's TensorFlow [14]. Having a unified framework also improve the portability and simplifies the process of transferring detection models to ARM-based platform for benchmarking. In the following, methods to setup detection model parameters is presented.

CNN Backbone

A CNN backbone is applied to the input B-scan image to obtain high-level features on top of the low-level features. The choice of backbone is crucial as the number of

parameters, operations, and types of layers directly affects performance of detection model. In this study, four CNN backbones are selected for feature extraction. All CNN backbones evaluated in this study have open source TensorFlow implementations and have had significant influence on the deep learning-based object detection community in the literature.

MobileNet [15] architecture is designed based on the idea of using “Depth wise Separable Convolutions”, which consist of a depth wise and a pointwise convolution after one another. It factorizes a standard convolution operation into a depth wise convolution and a 1×1 convolution. The building block of MobileNet architecture is an inverted residual structure where the input and output of the residual block are thin bottleneck layers. *Inception v2* [16], which set the state of the art in the ILSVRC 2014 challenge both in classification and detection task. The Inception network is an important milestone in the development of CNN based feature extractors. Prior to introducing “Inception units”, deep networks generally just stacked convolution layers deeper and deeper to improve accuracy of the network. Implementation of Inception units made it possible to increase the depth and width of a network without increasing its computational cost. *ResNet 101* [17] as winner of competitions such as COCO 2015 challenge for classification, detection, and segmentation tasks. *Inception Resnet v2* [18], which combines implementation of residual links for optimization and updating the weight in the network with the computation efficiency of Inception units.

Table 2 Properties of the Feature Extractor Backbones Implemented in this Study

Backbone	Top-1 Accuracy (%)	Parameters (million)
Inception v2	73.9	10.2
ResNet 101	76.4	42.6
Inception Resnet v2	80.4	54.3
MobileNet	71.1	3.2

Box Proposals

For Faster R-CNN meta-architecture, the number of box proposal per input image to be sent to the classification head of detector needs to be specified as a training parameter. Typically, the number of predictions is set to 300 in both meta-architectures [19]. Sending fewer box proposals to the classification head of detector is a method for reducing computation potentially at the risk of reducing accuracy (F-measure) by decreasing recall value. In order to evaluate trade-off between accuracy and computation cost (detection time), number of proposals is set in range of 20 to 300 and the performance of the detector is reported.

Loss Function

Configuration of the loss function (Equation 1) impacts training stability and testing performance of the detection models. Predicting the labels (classification) and localization (regression) of instances for each bounding box requires matching bounding boxes to ground truth instances in a dataset (label and coordinates). In this study, Argmax matching with threshold values according to the original paper for each meta-architecture is implemented. Ratio for number of positive and negative bounding boxes are those recommended by the original paper for each meta-architecture. In accordance with prior

studies [10, 11, 8, 9], the following function is used to encode a ground truth box with respect to its matching bounding box: $\varphi(b_a; a) = [10 \cdot x_c/w_a, 10 \cdot y_c/h_b, 5 \cdot \log(w), 5 \cdot \log(h)]$. Following prior works [11, 8, 9], to combine advantages of L_1 loss (steady gradients for large values) and L_2 loss (less oscillations during updates for small values), Smooth L_1 loss [20] is used in all experiments.

Training and Fine-tuning

Stochastic Gradient Descent (SGD) with momentum optimization algorithm [21] is used for Faster R-CNN. Since the models using input image with different size, batch size parameter is set to 1. For SSD, Root Mean Square Propagation (RMSProp) algorithm is used with batch size parameter set to 32. Note that for implementation of Faster R-CNN model in TensorFlow, instead of using the RoI Pooling layer [14] and position-sensitive RoI Pooling layers [22] which are used in the original papers, TensorFlow’s “CropAndResize” function (TensorFlow :: ops :: CropAndResize) is implemented which extracts crops from the input image tensor and resizes them using bilinear sampling or nearest neighbor sampling (possibly with aspect ratio change) to a common output size,

The investigated models for rebar detection task in this study are trained on the COCO dataset [13] and fine-tuned on the GPRDETNet [23] dataset. GPRDETNet dataset contains 520 bridge deck B-scan images and 4,085 instances. The instances in GPRDETNet dataset are annotated for detection tasks according to Pascal VOC [24] protocol. Images are cropped from real GPR B-scans collected from several bridges in the US using a GPR antenna of 1.6 GHz. On average the dataset contains 7.9 instances per image, which is similar to COCO dataset with 7.7 instances per image [13] which is used for pre-training

the detection models. In addition, instances in COCO are smaller than PASCAL VOC which makes it similar to GPRDET dataset. Generally smaller objects are harder to recognize [19] and require more contextual reasoning to recognize.

Official COCO API is used [25] to evaluate performance of detection models, which measures mean Average Precision (mAP) averaged over Intersection over Union (IoU) thresholds in $[0.5 : 0.05 : 0.95]$, amongst other metrics. To train the models, a machine with Ubuntu 16.04 OS, 16GB RAM, Intel Core i7-8700K processor and three NVIDIA GTX 1080 Ti card is used.

Performance Benchmarking

For benchmarking the models, a machine with Ubuntu 16.04 OS, 16GB RAM, Intel Core i7-8700K processor and an NVIDIA GTX 1080 Ti and an ARM-based platform with a Quad core Cortex-A72 processor and 4GB LPDDR4-2400, as shown in Table 1, is used. Timings are reported for a batch size of one. The average time for 10 images is reported. The TensorFlow Profiler (TFProf) tool [14] is used to measure the total memory demand of the models during test; The average memory usage for 10 images is reported.

4. RESULTS AND DISCUSSION

The obtained data from experiments is analyzed in this section. The data is collected by training and benchmarking object detectors, sweeping over model configurations as described in Section 3. Each such model configuration includes a choice of meta-architecture, feature extractor, stride (for Resnet and Inception Resnet) and number of proposals (for Faster R-CNN). For each such model configuration, timings on GPU and ARM platform and memory demand is measured as described below.

Accuracy, Speed, and Memory Usage Tradeoff

Figure 2 shows the mAP of each of the model configurations, with colors and symbol representing CNN backbone and meta-architecture, respectively. On GPU platform, running time per input image ranges from 10 to 400 ms. The obtained result shows that models with SSD meta-architecture are faster while Faster R-CNN models tends to lead to slower but more accurate models, although speed of Faster R-CNN models can be increased by limiting the number of regions proposed, as it is discussed below.

Figure 2 Accuracy vs time on GPU platform, with colors and marker symbols indicating feature extractor backbone and meta-architecture, respectively.

Most Accurate: Faster R-CNN with dense output Inception Resnet v2 model provides the best possible accuracy on GPRDET on GPU. However, this model is slower than other models. Due to memory demand of this model, the ARM-based model is not capable to run this model (). On ARM-based platform, Faster R-CNN with ResNet 101

CNN backbone, as shown in Figure 3, provides the highest accuracy, achieving, the state-of-the-art model accuracy. The overall mAP numbers for models are shown in Figure 3.

Figure 3 Accuracy vs time on ARM-based platform, with colors and marker symbols indicating feature extractor backbone and meta-architecture, respectively.

Fastest: It was observed that SSD model with MobileNet and Inception v2 backbones for feature extraction are the fastest models. The obtained result shows that all SSD models can be run on the ARM-based platform used in this study. *Sweet Spot:* Faster R-CNN with Resnet 101 and 50 proposals: Faster R-CNN w/Resnet models can attain similar speeds if the number of proposals is limited to 50.

The effect of CNN backbone

Intuitively, stronger performance on classification should be positively correlated with higher accuracy on both COCO and GPTDET. The correlation between accuracy (overall mAP) of different models and the Top-1 accuracy in ImageNet classification

challenge is obtained by the pretrained feature extractor used to initialize parameters on detection each model. As shown in Figure 4, there is a correlation between classification and detection performance. However, this correlation appears to only be significant for Faster R-CNN models on COCO. The performance of Faster RCNN and SSD models on DETDATASET appears to be less reliant on its feature extractor's classification accuracy.

Figure 4 Accuracy of detector on COCO and DETDATASET vs accuracy of feature extractor (as measured by top-1 accuracy on ImageNet-CLS).

The effect of instance size

The accuracy (overall mAP) for different models on different sizes of objects is shown in Figure 5. The obtained result shows that accuracy of all methods is significantly higher on large instances. It was observed that even though SSD models typically have poor performance on smaller objects, but they are competitive with Faster RCNN on larger objects.

All models performed significantly better on large size instances comparing to

medium size instances, except Faster R-CNN meta-architecture model with Inception ResNet v2 CNN backbone. Accuracy of SSD meta-architecture model with ResNet 101 CNN backbone are close for both medium size and large size instances. Detecting smaller instances is a challenging task because the activations of small instances become smaller after passing each pooling layer. So, selecting the right size for input images is very important to guarantee a good accuracy while the features extractor is small enough to be run on ARM-based platforms which has up to 4GB RAM. In addition, identification of small objects surrounded by generic clutter in the background is a challenge for detectors that rely on “objectness” and class-agnostic bounding box proposal due to the drastic increase in number of RoIs. Faster R-CNN models with all four CNN backbone studied in this paper, outperformed both SSD models. The highest overall mAP on GPRDET dataset, which contains medium and large size instances, is obtained with Faster RCNN meta-architecture with ResNet 101 model, as shown in Figure 5.

Figure 5 Accuracy stratified by object size, meta-architecture and CNN backbone.

The effect of the number of proposals

The number of proposals generated by the region proposal network (RPN) is one of the adjustable parameters for Faster R-CNN. The suggested number of proposals by the authors in the original paper [8] is 300, however, the obtained result in this study shows the number of computed proposals per input image can be significantly reduced without harming accuracy (overall mAP) for detecting rebars in GPR B-scan images. As shown in Figure 6, in feature extractor backbones where the “box classifier” portion of Faster R-CNN is detection models with computationally expensive (more parameters) CNN backbones, reducing number of box proposals can lead to significant computational savings and increasing speed of detector.

Figure 6 Effect increasing number of box proposals on accuracy (mAP)

Figure 6 shows trade-off for Faster RCNN models for different CNN backbones. The obtained result show that Inception Resnet v2 backbone, which has 55.4% mAP with 300 proposals can still have surprisingly high accuracy (47.2% mAP) with only 20

proposals. The authors, recommend 50 proposals as the sweet spot for rebar detection on GPRDETN, which provides 93% of the accuracy of using 300 proposals while reducing running time by a factor of 3. While the computational savings are most pronounced for Inception Resnet v2, similar tradeoffs hold for all CNN backbones is observed.

Memory analysis:

For memory benchmarking, total memory usage is measured. The latest generation of ARM-based platform, used in this study, Raspberry Pi 4 Model B, has 4GB of LPDDR4 memory, As shown in (Table 1). The actual amount of memory can be allocated to the processor is less than 4GB because 128MB memory is reserved for GPU. It means due to limitation of allocated memory to processor, Faster RCNN + Inception ResNet v2 detection models cannot be run on this platform.

Figure 7 Memory (Mb) usage and detection speed (FPS) for each model on ARM platform

Figure 7 plots memory usage against running time on GPU and ARM-based platform. Overall, it was observed high correlation with running time with larger and more powerful feature extractors requiring much more memory.

As with speed, MobileNet is the cheapest, requiring less than 1Gb (total) memory in almost all settings which allows models with this CNN backbone being run on both the latest and older generation (3rd generation) of ARM-based computational platforms.

As shown in Figure 8, all models with models with SSD meta-architecture in this study can be used on the ARM-based processor with 4GB memory. As shown in Figure 5, accuracy of SSD model with ResNet 101 feature extractor is comparable to ResNet 101 model with SSD meta-architecture at a significantly higher (5 times faster) speed.

Figure 8 Running time (milliseconds) for each model on GPU and ARM platform

Thermal and Power Draw:

The ARM-based platform used for benchmarking in this study needs a 5-volt, 3-amp power supply. It draws 3.4 and 7.6 watts at idle and under load, respectively.

In this benchmark the Raspberry Pi 4 is subjected to a 15-minute run of Faster R-CNN + ResNet 101 detection model, and the temperature and clock speed measured once every second (900 data points is collected) using the SoC's internal sensors. this test took place in an ambient temperature of nearly 27°C.

The ARM-based platform in this study has a passive cooling system (3 heatsinks). Top of the processor and the areas of the board near the processor experience temperature of 81°C after running the detection model for 15 minutes. As with any modern processors, if the processor and the board reach a temperature threshold, the processor will throttle down to protect itself from harm.

While running the model for 15 minutes, the processor hit 81 degrees and began throttling down from 1.5 to 1 GHz after 5 minutes. However, the system kept bringing itself back to the full 1.5 GHz when it dipped down to around 80 degrees, but then it would get warm again and go down to 1 GHz, which results in reducing speed for the detection model.

The obtained result shows to achieve a sustained performance under load, an active cooling system is necessary. This benchmark clearly demonstrates additional cooling is going to be a must-have to maintain top performance for workloads including sustained processor activity over the five minutes mark.

Figure 9 Thermal throttling benchmark of ARM-based platform while running the detection model for 15 minutes.

the test consists of running the detection model while using the LAN to send the output of the detection model, image size and location of rebars, to URICAB [3].

5. CONCLUSIONS

In this paper, the authors performed an experimental comparison of some of the main aspects that influence the accuracy, speed, and memory usage of modern CNN-based object detectors for onsite detection of rebars on concrete bridge decks using low-cost ARM-based computation platforms. The obtained result will help practitioners choose an appropriate method when deploying object

detection in the real world based on the computation capacity and limited hardware properties of ARM-based platforms. The experimental result show that for Faster RCNN meta-architecture-based detection models, reducing number of box proposals significantly

improves detection speed (FPS) without sacrificing much accuracy.

State of the art results is obtained for ARM-platform based detection task on GPRDETNet dataset by implementing rebar detection model using Faster RCNN as meta-architecture and ResNet 101 as feature extractor. The obtained results indicate that SSD meta-architecture with MobileNet as feature extractor achieved provides the fastest detection speed with overall mAP of 46 on GPRDETNet.

6. REFERENCE

- [1] ASCE, "Report Card for America's Infrastructure," 2013. [Online]. Available: <http://www.infrastructurereportcard.org/>.
- [2] K. Dinh, N. Gucunski and T. H. Duong, "An algorithm for automatic localization and detection of rebars from GPR data of concrete bridge decks.," *Automation in Construction*, vol. 1, no. 89, pp. 292-298, 2018, DOI: <https://doi.org/10.1016/j.autcon.2018.02.017>.
- [3] P. Asadi, M. Gindy and M. Alvarez, "A Machine Learning Based Approach for Automatic Rebar Detection and Quantification of Deterioration in Concrete Bridge Deck Ground Penetrating Radar B-scan Images.," *KSCE Journal of Civil Engineering*, vol. 1, no. 1, pp. 1-10, 2019, DOI: <https://doi.org/10.1117/12.2228345>.
- [4] L. Qiao, Q. Yao, R. Xiaozhen and W. Qifu, "Identification of buried objects in GPR using amplitude modulated signals extracted from multiresolution monogenic signal analysis.," *Sensors*, vol. 15, no. 12, pp. 30340-30350, 2015, DOI: <https://doi.org/10.3390/s151229801>.

- [5] A. Bouzerdoum, F. H. C. Tivive and C. Abeynayake, "GPR Target Detection by Joint Sparse and Low-Rank Matrix Decomposition," *IEEE Transactions on Geoscience and Remote Sensing*, pp. pp. 2583-2595, 2019.
- [6] Y.-a. Cui, L. Wang and J.-p. Xiao, "Automatic feature recognition for GPR image processing.," *World Academy of Science, Engineering and Technology*, vol. 61, no. 1, pp. 176-179, 2010.
- [7] P. Edoardo, F. Melgani, M. Donelli, R. Attoui and M. D. V. De Vos, "Automatic detection and classification of buried objects in GPR images using genetic algorithms and support vector machines.," *Geoscience and Remote Sensing Symposium*, 2008. IGARSS 2008. IEEE International, vol. 2, no. 1, pp. II-525, 2008, DOI: <https://doi.org/10.1109/IGARSS.2008.4779044>.
- [8] S. Ren, K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2016.*
- [9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. *Ssd: Single shot multibox detector. European Conference on Computer Vision, pages 21–37. Springer, 2016.*
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.*
- [11] R. Girshick. *Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.*

- [12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 1 May 2014.
- [14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. *Tensorflow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org, 1, 2015.
- [15] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [18] . Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.
- [19] Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer et al. "Speed/accuracy trade-offs for modern convolutional object detectors." *IEEE conference on computer vision and pattern recognition*, pp. 7310-7311. 2017.

- [20] *P. J. Huber et al. Robust estimation of a location parameter. The Annals of Mathematical Statistics, 35(1):73–101, 1964..*
- [21] *. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Dividethe gradient by a running average of its recent magnitude.COURSERA: Neural Networks for Machine Learning, 4(2),2012.*
- [22] *. Dai, K. He, and J. Sun. Instance-aware semantic seg-mentation via multi-task network cascades.arXiv preprintarXiv:1512.04412, 2015.*
- [23] *P. Asadi and M. Gindy, "GPRDETNv1.0 Dataset," 24 5 2019. [Online]. Available: <https://github.com/PouriaAI/GPR-Detection.git>. [Accessed 1 4 2019].*
- [24] *M. Everingham, L. Van Gool, C. KI Williams, J. Winn and A. Zisserman, "The pascal visual object classes (voc) challenge.," International journal of computer vision, vol. 88, no. 2, pp. 303-338, 2010.*
- [25] *T. Y. Lin and P. Dollar. "Ms coco api" 24 5 2019. [Online]. Available: <https://github.com/pdollar/coco>. [Accessed 1 4 2019].*

Appendix: Literature review

Application of ground penetrating radar (GPR) technology in the condition assessment of concrete bridge decks has been well recognized. Detecting rebars is a necessary first-step in condition assessment of bridge decks based on Ground Penetrating Radar (GPR) data. Locating rebars in GPR data is often done manually by an engineer, which is a time-intensive task and requires moderate to significant level of training. Extensive literature exists regarding processing GPR data using machine learning and artificial neural networks (ANN) techniques for automatic detection of objects in GPR data.

Costamagna et al. [1] presented a neural procedure for the analysis of GPR data. The method works by adapting the input image to the search of some objects' patterns, that are successively identified by means of a recognition step using a back-propagation optimization. The results on actual data showing buried pipe signatures present the same level of accuracy than the analyses performed by a trained human operator. Gamba et al. [2] presented a processing chain for the spatial analysis of pipes in GPR data. The processing of GPR data is performed by a suitably trained simple ANN-based detector after some pre-processing steps aiming toward the enhancement of the buried objects' patterns. The algorithm has been tested on real GPR images of buried pipes and compared with ground-truth (Data labeled by trained human operators), and satisfactory accuracy was obtained. Moreover, the effectiveness and advantages to exploiting some sort of "spatial diversity" by combining the analysis of data simultaneously extracted by different GPR antennas was discussed.

Gader et al. [3] presented a complex recognition system for detecting land mines. The proposed system is evolving from basic research into a practical fielded system. Some components of the proposed system have been field tested with excellent results, whereas other components have achieved such results in the laboratory tests. Information fusion algorithms are central to the excellent results obtained. Multiple-detection algorithms are applied to field GPR data. The authors combined the output of the object detection algorithms using the fuzzy logic and Sugeno and Choquet fuzzy integrals to obtain the best results.

Odhiambo et al. [4] presented an application of a fuzzy-neural network (F-NN) classifier for classification of soil profile using GPR data. A model was developed for classification of soil profile strips along a traverse based on common signature similarities that can relate to physical features of the soil such as depth, texture and structure of the horizons; and relative arrangement of the horizons. The obtained result shows that the proposed model is able to classify the collected GPR data into zones that corresponded with those obtained by visual inspections that are performed by a trained human operator.

Shihab and Nuaimy [5] presented an automatic object-detection method based on unsupervised learning-based ANN classifier along with image processing techniques to extract useful patterns representing objects in and filtering noise and clutter. The proposed classifier is capable of labeling regions of targets in GPR data. This classifier was applied to pipe and land mine GPR data sets and it achieved rapid and accurate results.

Shaw et al. [6] presented an ANN model to automate and facilitate the post-processing of GPR data. The GPR data is reduced to a simplified data set by using an edge detection algorithm and detection task was performed by using a multi-layer perceptron (MLP)

network with a single hidden layer containing 8 nodes to detect objects based on the output of the edge detector algorithm. The process of training, validation, and testing of the model were carried out making use of an emulsion analogue tank, simulating the properties of concrete, and using real concrete specimens. The obtained results showed that the use of an MLP-based model could be quite effective in automating detection of embedded steel reinforcing bars from a GPR survey.

Moysey et al. [7] presented a method to estimate radar facies probabilities from GPR data based on ANN techniques, yielding stochastic facies-based models that honor the large-scale architecture of the sub-surface. The obtained results on synthetic GPR images showed that the proposed model was able to correctly identify radar facies with an accuracy of ~90%. Manual interpretation of a set of 450 MHz GPR data resulted in the identification of four radar facies. Of these, a neural network was able to identify two facies with an accuracy of ~80% and one with an accuracy of 44%. The neural network was not able to identify the fourth facies, likely due to the choice of defining facies characteristics.

Gilmore et al. [8] presented an ANN model for automatic detection of unexploded ordnance (UXO) and landmines based on classification of particular features in GPR data. These features are the so-called “invariant moments” of a GPR data. The detection results for both metal and dielectric targets buried in a sandbox were reported.

Caorsi and Cevini [9] used ANNs to reconstruct the geometric and dielectric characteristics of buried cylinders. The ANN architecture was designed to work with input data extracted from the transient electric fields scattered by the objects (buried cylinders). To this aim, a simulation of a typical GPR setting is performed and different sets of data was evaluated. The authors studied various ANN models, and results have been reported compared. To

evaluate the "robustness" of the proposed approach, the models has been tested against noisy data.

Yang and Bose [10] applied ANNs for detecting landmines from data generated by different kinds of sensors. Real-valued ANNs have been used for detecting land mines from scattering parameters measured by GPR after disregarding phase information. The authors presented results using complex-valued ANNs, capable of phase-sensitive detection followed by classification task. A two-layer hybrid ANN structure incorporating both supervised and un-supervised learning approach was proposed to detect various types of landmines.

Lee et al. [11] presented a new method for maximizing the area under the receiver operating characteristic (ROC), called AUC. A common approach to training ANN models in a supervised learning setting is to minimize the mean-square error between the output for each sample during training phase and some desired output. In the context of landmine detection and discrimination, although the performance of an algorithm is correlated with the minimize the mean-square error, it is ultimately evaluated by using ROC curves. In general, the larger the area under the ROC curve, the better model. Desirable properties of the proposed algorithm were derived and discussed by the authors. A hypothesis test is used to compare the proposed algorithm to an existing algorithm.

Travassos et al. [12] presented a method for detecting and characterizing inclusions in concrete structures by inverting GPR data. In the proposed approach, the data was pre-processed using the principal component analysis (PCA) and then used to train an ANN model. The GPR data consists of 1200 time steps. Using PCA algorithm, the data was compressed to 286 dimensions. This dimensional reduction makes the ANN training easier

and faster. The ANN were trained to find the buried inclusions characteristics. The obtained results showed that the expected maximum error was kept under 1%, which is a satisfactory result.

Navneet and Manisha [13] focused to simplify the processing and interpretation of the hyperbolic patterns appeared in GPR data and estimate the position of the objects using ANNs and curve fitting algorithms. The authors presented an efficient dynamic runtime buried object detection algorithm for real-time identification of buried Improvised Explosive Devices (IEDs) and buried fusing mechanisms in GPR data and reported the results.

Qiao et al. [14] applied a novel method called the Multiresolution Monogenic Signal Analysis (MMSA) for detecting metal objects in GPR data. The proposed method consists of four steps. First the image is decomposed by the MMSA to extract the amplitude component of the GPR images. The amplitude component enhances the object reflection and suppresses the direct wave and reflective wave to a large extent. Then the authors used the region of interest extraction method to locate the genuine object reflections by calculating the normalized variance of the amplitude component. To find the pick of the hyperbola pattern, a Hough transform is used in the region of interest. Finally, the horizontal and vertical coordinates of the object were extracted.

Szymczyk and Szymczyk [15] proposed a new method based-on a new representation of GPR signals by polynomials approximation. The coefficients of the polynomial (the feature vector) are neural network inputs for automatic classification of a special kind of geologic structure, a sinkhole. The obtained results showed that the classifier can effectively distinguish sinkholes from other geologic structures in GPR data

Sakaguchi and Morton [16] focused on convolutional neural networks (CNNs) for object detection on GPR data. The benefit of using a CNN is that features extracted from the data are a learned parameter of the model. However, the implementation of a CNN must be done carefully for each application as network parameters can cause performance to vary widely. The authors presented results from using CNNs for object detection in GPR data and discusses proper parameter settings and other considerations.

Bralich et al. [17] applied transfer learning method for training a CNN based detector to overcome the problem of lack of GPR data. The authors trained two CNN on large datasets (Cifar10 and a dataset of high-resolution aerial imagery for detecting solar photovoltaic arrays) for feature extraction and training parameters of models. The authors performed experiments on a large collection of GPR data. The obtained results showed that these approaches improve the performance of CNNs for buried target detection in GPR data.

Zhao and Al-Qadi [18] implemented regularized deconvolution to analyze simulated GPR signals to increase their range resolution. The authors evaluated effect of applying four types of regularization methods, including Tikhonov regularization and total variation, on noisy GPR signals, then performance was evaluated in terms of accuracy in estimating distance of close impulses. The L-curve method was used to choose the appropriate regularization parameter. The total variation regularization method and zeroth-order Tikhonov regularization outperform first-order and second-order Tikhonov regularization in terms of average asphalt layer thickness estimation error and the standard deviation of the error. The proposed method was evaluated with the GPR field data. The reported results show that the algorithm based on regularization is a simple and effective approach to

increase the GPR signal range resolution with presence of noise in the case of thin asphalt overlay thickness prediction.

Sakaguchi et al. [19] implemented convolutional neural networks (CNNs) in order to jointly learn features across two sensor modalities and fuse the information in order to distinguish between object and background. This joint optimization is possible by modifying the traditional CNNs configuration to extract data from multiple sources. The filters generated by the proposed approach creates a learned feature extraction method that is optimized to provide the best discrimination performance when fused. The authors presented the results of applying CNNs and compared these results to the use of fusion performed with a linear classifier.

Noreen and Khan [20] presented a machine learning approach to detect hyperbolic patterns using a support vector machine (SVM) with the histogram of oriented gradient features (HOG). For this purpose. The reported results showed that HOG feature-based classifier achieve a high detection rate of 0.758 with a low false positive rate of 0.394. The authors evaluated the proposed model is tested on both real GPR field data and synthetic GPR data. Synthetic GPR data is created on an open source software gprMax.

Khalaf et al. [21] presented a new feature for detecting landmines at various depths. The proposed approach can be described mathematically by applying Prony's method, to calculate the complex resonance frequencies (CNR), which are considered as suitable features to discriminate different objects. The authors evaluated performance of different classification methods: artificial neural network (ANN), K-Nearest Neighbors (KNN), Multi-Class Support Vector Machine (MC-SVM) and Decision Tree (DT).

Ajithkumar et al. [22] studied effectiveness of five different classifiers namely: Hidden Markov Model, Support Vector Machine, Artificial Neural Network, Gradient Boosted Decision Tree and Adaptive Boosted Decision trees for addressing robotic landmine detection problem. Two GPR based datasets have been used both of which are open source and contain data for foliage and dry, desert type soils respectively. Based on the obtained results, a selection table has been designed which allows the practitioners to select the classifier that is most likely to give the best performance with respect to a preferred metric.

Dou et al. [23] proposed a method of automatically recognizing and fitting hyperbola patterns from GPR data which is computationally suitable for real-time processing of data. After pre-processing of the input GPR images, a novel thresholding method is applied to separate the region containing targets from background. The authors applied a column-connection clustering (C3) algorithm to separate the regions of interest from each other. Subsequently, a machine learning based classifier was applied to identify hyperbolic signatures from outputs of the C3 algorithm, and a hyperbola is fitted to each such pattern with an orthogonal-distance hyperbola fitting algorithm. The proposed method successfully identified and fit hyperbolic signatures with intersections with others, hyperbolic signatures with distortions, and incomplete hyperbolic signatures with one leg fully or largely missed.

Xu et al. [24] focused on the problem of segmenting echogram radar data collected from the polar ice sheets, which is challenging because segmentation boundaries are often very weak and there is a high degree of noise. The authors proposed a multi-task spatiotemporal ANN that combines 3-dimensional CNNs and Recurrent Neural Networks (RNNs) to estimate ice surface boundaries from sequences of tomographic radar images. The authors

show that the proposed method outperforms the state-of-the-art on this problem by avoiding the need for hand-tuned parameters, extracting multiple surfaces simultaneously, requiring less non-visual metadata, and being faster.

Asadi et al. [25] proposed a method based on combination of image processing, machine learning (ML) data classification, data filtering, and spatial pattern analysis for quantification of deterioration and creating a 3D deterioration map in concrete bridge decks. The value of F-measure was found to be 86.20%.

Dinh et al. [26] proposed an algorithm consisting of a Convolutional Neural Network (CNN) for hyperbola signatures and then implementing a CNN to locate potential rebars by retain the likely true positive, and to discard likely false positive rebar detections. The overall accuracy of the method was found to be 99.60% on the GPR data used in this study.

REFERENCE

- [1] Costamagna, E., P. Gamba, and S. Lossani. "A neural network approach to the interpretation of ground penetrating radar data." In *IGARSS'98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No. 98CH36174)*, vol. 1, pp. 412-414. IEEE, 1998.
- [2] Gamba, Paolo, and Simone Lossani. "Neural detection of pipe signatures in ground penetrating radar images." *IEEE Transactions on Geoscience and Remote Sensing* 38, no. 2 (2000): 790-797.
- [3] Gader, Paul D., James M. Keller, and Bruce N. Nelson. "Recognition technology for the detection of buried land mines." *IEEE transactions on fuzzy systems* 9, no. 1 (2001): 31-43.
- [4] Odhiambo, Lameck O., Robert S. Freeland, Ronald E. Yoder, and J. Wesley Hines.

"Application of fuzzy-neural network in classification of soils using ground-penetrating radar imagery." In 2002 ASAE Annual Meeting, p. 1. American Society of Agricultural and Biological Engineers, 2002.

[5] Shihab, S., W. Al-Nuaimy, and A. Eriksen. *"Image processing and neural network techniques for automatic detection and interpretation of ground penetrating radar data." WSEAS advances in circuits, systems and signal processing (2002): 360-363.*

[6] Shaw, M. R., S. G. Millard, T. C. K. Molyneaux, M. J. Taylor, and J. H. Bungey. *"Location of steel reinforcement in concrete using ground penetrating radar and neural networks." NDT & E International 38, no. 3 (2005): 203-212.*

[7] Moysey, S., J. Caers, R. Knight, and R. M. Allen-King. *"Stochastic estimation of facies using ground penetrating radar data." Stochastic Environmental Research and Risk Assessment 17, no. 5 (2003): 306-318.*

[8] Gilmore, Colin, Scott Peters, Joe LoVetri, and Dean McNeill. *"GPR target detection using a neural network classifier of image moments as invariant features." In 2004 10th International Symposium on Antenna Technology and Applied Electromagnetics and URSI Conference, pp. 1-4. IEEE, 2004.*

[9] Caorsi, Salvatore, and Gaia Cevini. *"An electromagnetic approach based on neural networks for the GPR investigation of buried cylinders." IEEE Geoscience and Remote Sensing Letters 2, no. 1 (2005): 3-7.*

[10] Yang, Chih-Chung, and N. K. Bose. *"Landmine detection and classification with complex-valued hybrid neural network using scattering parameters dataset." IEEE transactions on neural networks 16, no. 3 (2005): 743-753.*

[11] Lee, Wen-Hsiung, Paul D. Gader, and Joseph N. Wilson. *"Optimizing the area under*

a receiver operating characteristic curve with application to landmine detection." IEEE Transactions on Geoscience and Remote Sensing 45, no. 2 (2007): 389-397.

[12] Travassos, X. Lucas, D. A. G. Vieira, Nathan Ida, Christian Vollaire, and Alain Nicolas. "Characterization of inclusions in a nonhomogeneous GPR problem by artificial neural networks." *IEEE Transactions on Magnetics* 44, no. 6 (2008): 1630-1633.

[13] Singh, Navneet P., and Manisha J. Nene. "Buried object detection and analysis of GPR images: Using neural network and curve fitting." In *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*, pp. 1-6. IEEE, 2013.\

[14] Qiao, Lihong, Yao Qin, Xiaozhen Ren, and Qifu Wang. "Identification of buried objects in GPR using amplitude modulated signals extracted from multiresolution monogenic signal analysis." *Sensors* 15, no. 12 (2015): 30340-30350.

[15] Szymczyk, Piotr, Sylwia Tomecka-Suchoń, and Magdalena Szymczyk. "Neural networks as a tool for georadar data processing." *International Journal of Applied Mathematics and Computer Science* 25, no. 4 (2015): 955-960.

[16] Sakaguchi, Rayn T., Kenneth D. Morton Jr, Leslie M. Collins, and Peter A. Torriane. "Recognizing subsurface target responses in ground penetrating radar data using convolutional neural networks." In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, vol. 9454, p. 94541A. International Society for Optics and Photonics, 2015.

[17] Bralich, John, Daniël Reichman, Leslie M. Collins, and Jordan M. Malof. "Improving convolutional neural networks for buried target detection in ground penetrating radar using transfer learning via pretraining." In *Detection and Sensing of Mines, Explosive*

Objects, and Obscured Targets XXII, vol. 10182, p. 101820X. International Society for Optics and Photonics, 2017.

[18] Zhao, Shan, and Imad L. Al-Qadi. "Development of regularization methods on simulated ground-penetrating radar signals to predict thin asphalt overlay thickness." *Signal Processing* 132 (2017): 261-271.

[19] Sakaguchi, Rayn, Miles Crosskey, David Chen, Brett Walenz, and Kenneth Morton Jr. "Convolutional neural network-based sensor fusion for forward looking ground penetrating radar." In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXI*, vol. 9823, p. 98231J. International Society for Optics and Photonics, 2016.

[20] Noreen, T., and Umar S. Khan. "Using pattern recognition with HOG to automatically detect reflection hyperbolas in ground penetrating radar data." In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1-6. IEEE, 2017.

[21] Khalaf, Mohammad W., Fatma M. El-Hefnawi, and Hany M. Harb. "Identification and classification of reflected landmine signals based on complex resonance frequency in dispersive media." In *2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*, pp. 223-231. IEEE, 2017.

[22] Ajithkumar, Nitin, P. Aswathi, and Rao R. Bhavani. "Identification of an effective learning approach to landmine detection." In *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, pp. 1-5. IEEE, 2017.

[23] Dou, Qingxu, Lijun Wei, Derek R. Magee, and Anthony G. Cohn. "Real-time

hyperbola recognition and fitting in GPR data." IEEE Transactions on Geoscience and Remote Sensing 55, no. 1 (2016): 51-62.

[24] Xu, Mingze, Chenyou Fan, John D. Paden, Geoffrey C. Fox, and David J. Crandall. "Multi-Task Spatiotemporal Neural Networks for Structured Surface Reconstruction." In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1273-1282. IEEE, 2018.

[25] Asadi, Pouria, Mayrai Gindy, and Marco Alvarez. "A Machine Learning Based Approach for Automatic Rebar Detection and Quantification of Deterioration in Concrete Bridge Deck Ground Penetrating Radar B-scan Images." *KSCE Journal of Civil Engineering* 23, no. 6 (2019): 2618-2627.

[26] Dinh, Kien, Nenad Gucunski, and Trung H. Duong. "An algorithm for automatic localization and detection of rebars from GPR data of concrete bridge decks." *Automation in Construction* 89 (2018): 292-298.