6-26-2024 2:30 PM

Flood Hazard and Vulnerability Mapping using Deep Learning and Earth Observation Data

Nafiseh Ghasemian Sorboni, *Western University*

Supervisor: Dr. Jinfei Wang, *The University of Western Ontario*
Co-Supervisor: Dr. Mohammad Reza Najafi, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Geography and Environment

Abstract

Urban flood risk assessment is critical for safeguarding lives and infrastructure amid frequent floods. Recent advances in Earth Observation (EO) data enable the creation of flood risk maps with enhanced spatial and temporal resolutions. While traditional Machine Learning (ML) algorithms lack optimal feature selection, Deep Learning (DL) algorithms excel in extracting complex patterns from EO data. This dissertation delves into the estimation of flood hazard and vulnerability within urban environments through DL algorithms and EO data. Several innovative methodologies were proposed: 1) A Convolutional Siamese Network (CSN) was devised for urban flood mapping using SAR satellite imagery. This method employed two parallel Convolutional Neural Networks (CNNs), processing pre-event and co-event images, respectively. By measuring feature space similarity, pixels were classified as flood or background using Contrastive Loss, Weighted Double Margin Contrastive Loss (WDMCL), and Triplet Loss functions. Testing with VGG16 and ResNet50 architectures yielded Precision, Recall, and F1 Score values of 0.75, 0.6, and 0.67, respectively, for the SEN12-FLOOD dataset, which notably improved upon integrating DEM into input features. 2) First Floor Height (FFH) estimation based on vertical measurements from Google Street View (GSV) images facilitated the creation of a flood vulnerability map for Toronto's Lower Don River region. Utilizing the proposed vulnerability index, derived from water depth minus FFH, buildings were categorized by vulnerability levels. Additionally, First Floor Elevation was estimated using FFH and Lowest Adjacent Grade (LAG) heights, achieving RMSE and Bias values of 81 cm and −50 cm for the Greater Toronto Area (GTA) and 95 cm and −20 cm for Virginia. 3) A Dense Attention Network (DAN) CNN architecture was proposed for building footprint extraction from LiDAR and RapidEye images, demonstrating an improved F1 Score (0.71) over DL models like U-net (0.42) and ResUnet (0.49). 4) A fusion method combining CNN classifications from GSV, LiDAR, and Orthophoto data for building land-use type classification achieved superior accuracy indices compared to a previous CNN-based study, with an Overall Accuracy of 75%. These methodologies represent significant advancements in utilizing DL algorithms and EO data for urban flood risk assessment, promising enhanced accuracy and efficiency in mapping and mitigation efforts.

Keywords

Flood risk mapping, remote sensing, deep learning, synthetic aperture radar, Google Street View, Lidar, multispectral remote sensing, flood extent mapping, flood vulnerability mapping, first floor height, building footprint extraction, building land-use type mapping, convolutional Siamese network, YOLOv5, dense attention network, fusion, ranking classes based on F1 Score

Summary for Lay Audience

This dissertation explored the use of advanced technologies in assessing and mitigating urban flood risks, a common and significant natural disaster. Traditional flood risk assessments involve creating maps, and recent advances in Earth Observation (EO) data have enabled the use of Deep Learning (DL) algorithms for more accurate analysis. The study presented several innovative approaches: 1) Flood Mapping with Convolutional Siamese Network (CSN): A novel CSN was proposed for urban flood mapping using Synthetic Aperture Radar (SAR) satellite images. By employing DL algorithms, specifically CSN with VGG16 and ResNet50 architectures, the study achieved promising results, enhancing flood and background accuracy indices. 2) Estimating First Floor Height (FFH) for vulnerability assessment: The research utilized Google Street View (GSV) images and building information to estimate FFH. A vulnerability map for the Lower Don region was produced, considering FFH and water depth. The approach demonstrated accuracy improvements compared with a previous method. 3) Building Footprint Extraction with Dense Attention Network (DAN): A DL model called DAN was proposed for accurate building footprint extraction using LiDAR and RapidEye images. Compared to other models, DAN outperformed with an impressive F1 Score of 0.71. 4) Building Land-Use Type Classification using a Fusion Method: A fusion method was introduced to combine DL classifications derived from various data sources (GSV, LiDAR, Orthophoto) for building land-use type classification. This ranking-based fusion method showed higher accuracy indices compared to a traditional Fuzzy Fusion method. The research contributes valuable insights and methodologies for enhancing flood risk assessment using cutting-edge technologies, ultimately promoting better urban planning and disaster preparedness.

Co-Authorship Statement

The following dissertation has been prepared in accordance with the integrated-article format established by the Faculty of Graduate Studies at the University of Western Ontario in London, Ontario, Canada. The author of this work is Nafiseh Ghasemian Sorboni, who completed it under the guidance of Drs. Jinfei Wang and Mohammad Reza Najafi. Chapters 2, 3, and 4 have been published, and Chapter 5 has been submitted as a co-authored peer-reviewed Journal paper. The author contributed to the methodology, experiments, discussions, and drafting of all chapters. Dr. Wang provided invaluable research insights for Chapter 4 and assisted with data preparation for Chapters 2 and 4, and also helped with hardware and software. Dr. Najafi contributed to the research concepts of Chapters 3 and 5 and provided assistance with data preparation for Chapter 3. Both Dr. Wang and Dr. Najafi contributed to the revision, editing, proofreading, and financial support of all chapters. Finally, all authors approved the final version and agreed to be wholly responsible for the work's integrity and accuracy.

Chapter 2 was published as: **Ghasemian Sorboni, N.**, Wang, J., & Najafi, M. R. (2024). Urban flood mapping using Sentinel-1 and RADARSAT Constellation Mission image and convolutional Siamese network. *Natural Hazards*, 1-32.

Chapter 3 was published as: **Sorboni, N. G.**, Wang, J., & Najafi, M. R. (2024). Automated first floor height estimation for flood vulnerability analysis using deep learning and Google Street View. *Journal of Flood Risk Management*, e12975.

Chapter 4 was published as: **Ghasemian, N.**, Wang, J., & Reza Najafi, M. (2022). Building detection using a dense attention network from LiDAR and image data. *Geomatica*, *75*(4), 209-236.

Chapter 5 was published as a peer-reviewed Journal paper as: **Ghasemian Sorboni, N.**, Wang, J., & Najafi, M. R. (2024). Fusion of Google Street View, LiDAR, and Orthophoto Classifications Using Ranking Classes Based on F1 Score for Building Land-Use Type Detection. *Remote Sensing*, *16*(11), 2011.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Jinfei Wang and Dr. Mohammad Reza Najaf, for their invaluable guidance, encouragement, and unwavering support throughout my research journey. Their expertise and dedication have been instrumental in shaping my academic growth, and their insights have greatly enriched my work. I am truly fortunate to have had the opportunity to work under their mentorship.

I extend my heartfelt thanks to my committee members, Dr. Ying Zhang from Natural Resources Canada (NRCan) and Dr. Katsu Goda from the Department of Earth Sciences, for their insightful feedback, constructive criticism, and support. Additionally, I am grateful to Dan Clayton from the Toronto and Region Conservation Authority (TRCA) for his support with data provision. Their input has been vital in refining my research and ensuring its academic rigor. I am also thankful for the funding support from the Multi-Hazard Risk and Resilience Interdisciplinary Development Initiative (IDI) grant from the University of Western Ontario and the Natural Sciences and Engineering Research Council of Canada (NSERC). Furthermore, I appreciate my labmates at the Geographic Information Technology (GITA) Lab and the Hydroclimate Extremes and Climate Change Lab (HydroClimEx) for their camaraderie, assistance, and the stimulating discussions that have enriched my research experience.

I would also like to acknowledge my PhD examiners, Dr. Yun Zhang, Dr. Mohammed Zaki, Dr. Jinhyung Lee, and Dr. James Voogt, for their time, effort, and valuable feedback during the examination process. Their expertise and critical evaluations have helped me to further enhance the quality of my research, and their suggestions have provided me with new perspectives to consider in my future work. Additionally, I extend my gratitude to the Department of Geography and Environment staff, Lori Johnson and Angelica Lucaci, for their administrative support and assistance throughout my studies.

Finally, I am deeply grateful to my parents for their unwavering emotional and financial support throughout the past years. Their belief in me and their sacrifices have been the

foundation of my academic journey. Their encouragement has been a constant source of strength and motivation, and I owe my achievements to their enduring love and support.

Table of Contents

List of Tables

List of Figures

List of Acronyms, Glossary

Adam: Adaptive Moment Estimation. An optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

API: Application Programming Interface. A set of rules or protocols that let software applications communicate with each other to exchange data, features and functionality.

AUC: Area under Curve. A classification metric measuring the entire two-dimensional area underneath the entire ROC curve from (0, 0) to (1, 1).

BE: Building Extent. The building façade boundary in GSV image.

BN: Batch Normalization. A technique used to improve the training of deep neural networks. It is used to normalize the inputs of each layer in such a way that they have a mean output activation of zero and a standard deviation of one.

BW: Basement Window. A window located in the basement of a building, typically positioned to allow natural light to enter the below-ground level.

CF: Confidence Factor. The degree of membership to a class.

CFRS: Complement of Fuzzy Rank Sum. The Complement of Fuzzy Rank Sum is a concept related to fuzzy rank sum analysis. In fuzzy logic and statistics, the complement typically refers to the opposite or negation of a given condition or operation. Therefore, the Complement of Fuzzy Rank Sum could involve considering the elements or outcomes not covered by the Fuzzy Rank Sum, providing a broader perspective on the analyzed data or decision-making process.

CNN: Convolutional Neural Networks. A deep learning neural network designed for processing structured arrays of data such as images.

Conv2D: 2D Convolution. The convolution operation performed on two-dimensional arrays.

CSN: Convolutional Siamese Network. A type of neural network architecture that consists of two or more identical sub-networks. The parameters of both sub-networks are updated in a mirrored way, and this technique is used to identify similarities between inputs by comparing their feature vectors. These networks can be applied for change detection.

DAN: Dense Attention Network. A convolutional neural network with one or multiple dense attention blocks embedded in its layers. Dense attention blocks constitute of a cascade of batch normalization, 2D convolution, drop out, and average pooling layers.

DEM: Digital Elevation Model. A representation of the bare ground (bare earth) topographic surface of the Earth excluding trees, buildings, and any other surface objects.

DL: Deep Learning. A type of machine learning that employs multi-layered neural networks that simulate the intricate decision-making capabilities of the human brain.

DSM: Digital Surface Model. The process of representing a topographic surface using a regular array of continuous elevation values. The surface includes natural features like vegetation and man-made features and all measurements are referenced to a common datum.

DTM: Digital Terrain Model. A DTM augments a DEM by including linear features of the bare-earth terrain, such as ridges and break lines.

EC: Elevation Certificate. An optional tool utilized by the national flood insurance Program to assess the first floor height of a building and its surrounding elevation.

EO: Earth Observation. The process of collecting information about the Earth's physical, chemical, and biological systems using remote sensing technologies.

FCL: Fully Connected Layer. A layer in a neural network where each neuron is connected to every neuron in the previous layer, allowing for complex relationships and patterns to be learned.

FD: Front Door. The main entrance or access point to a building or structure.

FDS: Final Decision Score. The ultimate score or result obtained from a decision-making process.

FEMA: Federal Emergency Management Agency. A United States government agency responsible for coordinating the federal government's response to natural and man-made disasters.

FFH: First Floor Height. The building first floor height relative to the Lowest LAG.

FN: False Negative. In binary classification, an outcome where the model incorrectly predicts the negative class when the true class is positive.

FOV: Field Of View. The extent of the observable world that can be seen at any given moment, often used in the context of cameras, sensors, or visual perception.

FP: False Positive. In binary classification, an outcome where the model incorrectly predicts the positive class when the true class is negative.

FR: First Return. In LiDAR, the first reflection of the laser beam from the surveyed surface.

FRS: Fuzzy Rank Sum. A method or algorithm that involves fuzzy logic to determine a rank sum, often used in decision-making processes.

GRD: Gridded Data. Data organized in a grid or matrix structure, often used in spatial data analysis or modeling.

GRNN: Gated Residual Refinement Network. An end-to-end trainable neural network designed specifically for building extraction. It combines the strengths of feature learning and pixel-level labeling capabilities from convolutional neural networks.

GSV: Google Street View. A technology featured in Google Maps and Google Earth that provides interactive panoramas from positions along many streets in the world.

GTA: Greater Toronto Area. A metropolitan area in Canada that includes the city of Toronto and its surrounding regions.

HH: Horizontal-Horizontal. A single polarization system that transmits and receives the horizontal polarization.

HV: Horizontal-Vertical. A dual polarization system that transmits horizontal polarization and receives vertical polarization.

InSAR: Interferometry SAR. Interferometric Synthetic Aperture Radar, a technique used in remote sensing to measure ground deformation.

IoU: Intersection over Union. A metric used to evaluate the accuracy of object detection algorithms by measuring the overlap between predicted and true bounding boxes.

IQR: Interquartile. A measure of statistical dispersion, representing the range between the first quartile and the third quartile of a dataset.

LAG: Lowest Adjacent Grade. The lowest point of the ground level immediately next to a building.

LiDAR: Light Detection and Ranging. A method for determining ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver.

LL: Lower Left. Typically used in the context of images or maps, indicating the lower-left corner.

LOS: Line of Sight. The invisible line extending from the camera to the object it is looking at it.

LR: Lower Right (Chapter 3) or Last Return (Chapter 5). Either refers to the lower-right corner of an image (Chapter 3) or the last reflection of the laser beam from the surveyed surface (Chapter 5).

Mixed r/c: mixed residential/commercial. A zoning designation indicating an area or property is intended for a mix of residential and commercial use.

ML: Machine Learning. A subset of artificial intelligence that focuses on the development of algorithms and statistical models to enable computers to perform tasks without explicit programming.

MS: Multi Spectral. Refers to the use of multiple bands or wavelengths in remote sensing to capture information about the surface characteristics of an area.

MSE: Mean Square Error. A measure of the average squared difference between predicted and actual values, commonly used as a loss function in regression problems.

nDSM: normalized Digital Surface Model. A digital model representing the difference between the earth's surface and the top of natural or man-made features, normalized to a specific scale.

NDVI: Normalized Difference Vegetation Index. An index used in remote sensing to measure the health and density of vegetation based on the difference between near-infrared and red light reflected by plants.

NMS: Non Maxima Suppression. A technique used in object detection to filter out redundant bounding boxes and retain only the most confident predictions.

OA: Overall Accuracy. A metric used to evaluate the overall correctness of a classification model by measuring the ratio of correctly predicted instances to the total instances.

OD: Object Detector. A system or algorithm designed to identify and locate objects within an image or video.

OSM: Open Street Map. A collaborative mapping project that creates a free, editable map of the world, often used in GIS applications.

PolSAR: Polarimetry SAR. A radar imaging technique that uses polarized electromagnetic waves to gather information about the surface characteristics of objects.

ReLU: Rectified Linear Unit. An activation function commonly used in neural networks that returns the input for positive values and zero for negative values.

ResNet50: Residual Network with 50 layers. A specific architecture of a deep neural network with 50 layers, known for its use of residual connections to improve training and performance.

RCM: RADARSAT Constellation Mission. A Canadian satellite mission for Earth observation using synthetic aperture radar.

RCNN: Region-based Convolutional Neural Network. A type of deep learning architecture used for object detection and image segmentation.

RMSE: Root Mean Square Error. Similar to MSE but takes the square root, providing a measure of the average magnitude of errors without considering their direction.

ROC: Receiver Operative Characteristic. A graph showing the performance of a classification model at all classification thresholds.

RS: Remote Sensing. The process of detecting and monitoring the physical characteristics of an area by measuring its reflected and emitted radiation at a distance (typically from satellite or aircraft).

SAR: Synthetic Aperture Radar. A type of active data collection where a sensor produces its own energy and then records the amount of that energy reflected back after interacting with the Earth.

SLC: Single Look Complex. A processed SAR data format representing a single look at the terrain.

SGD: Stochastic Gradient Descent. An iterative optimization algorithm commonly used in machine learning to minimize the loss function during training.

SVM: Support Vector Machine. A supervised machine learning algorithm used for classification and regression analysis.

TFS: Trained From Scratch. Refers to training a machine learning model from the beginning without using pre-trained weights.

TL: Transfer Learning. A machine learning technique where a model trained on one task is adapted for a different but related task.

TRCA: Toronto Region Conservation Authority. An organization responsible for environmental conservation and management in the Toronto region.

TN: True Negative. In binary classification, an outcome where the model correctly predicts the negative class.

TP: True Positive. In binary classification, an outcome where the model correctly predicts the positive class.

UAS: Unmanned Aerial System. A comprehensive term encompassing both unmanned aerial vehicles (UAVs or drones) and their ground control systems.

UAV: Unmanned Aerial Vehicle. An aircraft without a human pilot on board, controlled either remotely or autonomously.

UL: Upper Left. Typically used in the context of images or maps, indicating the upper-left corner.

VGG16: Visual Geometry Group with 16 layers. A specific deep convolutional neural network architecture with 16 weight layers, known for its simplicity and effectiveness.

VH: Vertical-Horizontal. A dual polarization system that transmits vertical polarization and receives horizontal polarization.

VV: Vertical-Vertical. A single polarization system that transmits and receives the vertical polarization.

WDMCL: Weighted Double Margin Contrastive Loss. A loss function used in contrastive learning with weights and double margins to enhance feature representations.

YOLOv5s: You Look Only Once version 5 with small number of parameters. A specific version of the YOLO (You Only Look Once) object detection model with a smaller number of parameters for faster inference.

Chapter 1

Introduction

1.1 Background

Floods are among the most frequent natural hazards worldwide. Between 1995 and 2015, over 2.2 billion people were affected by floods constituting 53% of the people affected by all kinds of natural disasters (Mateo-Garcia et al., 2021). Climate change is expected to increase flood events frequency and intensity in many regions around the world (GebreEgziabher & Demissie, 2020). The main flood generation mechanisms can be divided into three categories based on the hydraulic processes involved in their generation, including fluvial, pluvial, and coastal flooding. A fluvial flood happens when water overtops the river banks, and usually occurs in the vicinity of rivers. A pluvial flood happens when intense sudden precipitation affects a watershed and water flow discharge exceeds the capacity of the drainage network. This exceedance can happen before the water flow reaches the drainage network or when the overflow from the drainage network happens. A combination of these two situations can also result in the pluvial flood (Bulti & Abebe, 2020). A coastal flood results from the combined effects of high tides and storm events (Muthusamy et al., 2019). Regardless of the type, floods cause damage to urban structures, such as damage to building contents and properties, as well as damage to building fabric, and can hinder access to roads and bridges. Also, they can result in soil erosion and deposition.

Urban flooding can cause significant damage to people and infrastructures. For example, the 2013 Toronto flood event was the most expensive disaster for Ontario; according to the Insurance Bureau of Canada, the damage of the insured properties exceeded $850 million. Thus, accurate flood risk analysis is essential for supporting urban planning and ecosystems and reducing costs.

Flood risk has three components, including hazard, exposure, and vulnerability. Hazard refers to the possible future natural or human-induced flood events that may affect

vulnerable and exposed elements. Exposure refers to the extent to which humans and structures may be affected in an area where flood events may occur, and vulnerability measures the tendency of exposed elements such as human beings, livelihoods, and assets to suffer adverse effects when impacted by flood events (Cardona et al., 2012). One parameter that gives an estimate of the flood hazard severity is flood extent. The lack of reliable First Floor Height (FFH) information, defined as the building first floor height relative to the Lowest Adjacent Grade (LAG), on a large geographic scale, causes significant restrictions on flood management. FFH estimation can contribute significantly to setting flood insurance premium/rate maps, and flood cost analysis (Montgomery & Kunreuther, 2018; Xia & Gong, 2024). Because of climate change, land use evolution is also a key factor influencing future flood risk. An increased built area results in a decrease in infiltration, baseflow, and lag times, and an increase in runoff volumes, peak discharge, and frequency of floods. Human activities such as urbanization and the growth of settlements and assets in flooding areas likewise contribute to the increasing flood damage. The risks due to such hazards in urban areas can significantly hinder daily activities, incur costly damages, and contribute to large-scale life losses, which is the reason why, when such risks are realized, they are often referred to as disasters (Beckers et al., 2013; Genovese, 2006; Park et al., 2021). Building land-use and type can be correlated through various factors and characteristics: 1) Zoning Regulations 2) Building Codes and Standards 3) Economic Factors 4) Infrastructure and Utilities 5) Demographics 6) Land Availability and Topography 7) Urban Planning Policies. For example, Zoning regulations typically dictate the types of buildings allowed in specific land-use zones. Residential zones may permit single-family homes, apartments, or townhouses, while commercial zones may allow for retail stores, offices, or mixed-use developments. Identifying land-use type of buildings can help in damage assessment context and vulnerability assessment. Besides, in practice, the estimation of direct damage to buildings is often achieved by applying the method of depth-damage function that connects flood depth directly to the economic value of damage (Pistrika et al., 2014). There is a different curve for each building land-use type and this valuable information can be applied for selecting the suitable depth-damage curve. This dissertation addresses pressing challenges in flood risk assessment and management by proposing innovative methodologies and techniques. Flood risk assessment requires a

comprehensive understanding of various factors, including flood extent, building characteristics, and land-use patterns. However, traditional methods for assessing flood risk may face limitations, such as the lack of reliable data on flood extent and building attributes. Therefore, the dissertation aims to fill these gaps by introducing novel methods for flood extent mapping, FFH estimation using Google Street View (GSV) images, and identification of building footprints and land-use types. These techniques are critical for enhancing flood risk analysis, enabling more accurate damage assessment, and supporting effective decision-making in flood-prone areas. By handling these challenges, the dissertation contributes to advancing the field of flood risk assessment and management, eventually improving the resilience of communities facing flood hazards.

1.2 Urban Flood Risk Analysis Using Remote Sensing Data

Remote Sensing (RS) is essential for generating crucial information related to water resources. Satellite-based maps depicting flood extents have been utilized to calibrate inundation models, whether they are based on a single flood event or multiple occurrences. The continuous and current measurements provided by satellite remote sensing techniques offer global coverage, contingent on their orbital characteristics. RS data proves valuable in estimating various aspects of flood risk (Wang & Xie, 2018). For instance, Optical and Synthetic Aperture Radar (SAR) data can be employed for mapping flood extents and conducting hazard analysis. Additionally, geospatial and EO data, including Light Detection and Ranging (LiDAR) and GSV, can contribute to assessing flood vulnerability and damage analysis.

1.2.1 SAR Satellite Images

RS data can help estimate flood-related parameters such as extent and depth. While Optical and SAR data help detect flooded areas, SAR images are of particular interest because of their data availability day and night and under different weather conditions. The first step

toward flood risk analysis is flood hazard mapping, and it can be accomplished by flood extent detection.

- **SAR intensity and Coherence**

SAR observations have been previously used in emergency response for flood mapping because of the cloud-penetrating and night-and-day capability of microwave instruments. Moreover, the possibility of using interferometric techniques adds an advantage to the choice of such data (Y. Li et al., 2019; Nico et al., 2000). In the context of SAR acquisition systems, *intensity* refers to the intensity of the radiation backscattered by the surface; that is, the fraction of the incident energy reflected directly back toward the sensor (Lillesand et al., 2015). *Coherence* is a measure of statistical similarity between two SAR images and was first applied for change detection applications in the mid-90s. It can be estimated between two images using equations (1-1) and (1-2). Equation (1-1) is called population coherence, and equation (1-2) refers to sample coherence.

$$\mu_0 = \frac{\langle x_1 x_2^* \rangle}{\sqrt{\langle |x_1|^2 \rangle \langle |x_2|^2 \rangle}} \tag{1-1}$$

$$|\hat{\mu}| \approx \frac{\sum_{n=1}^{L-1} x_{1,n} x_{2,n}^*}{\sqrt{\sum_{n=1}^{L-1} |x_{1,n}|^2 \sum_{n=1}^{L-1} |x_{2,n}|^2}} \tag{1-2}$$

In both equations, x_1 and x_2 are the reference (captured before the flood event) and event (captured during the flood event) images, respectively, and the asterisk sign means the complex conjugate of the image. L is the number of images used for coherence computation, and when the number of images increases, $L \to \infty$, the sample coherence approaches population coherence. In other words, coherence describes the degree of correlation between the two radar images, and its magnitude affects the phase measurement accuracy. In practice, several factors contribute to a reduction of coherence. One crucial

factor is temporal decorrelation, which describes changes in the scene microstructure such as changes in dark areas location (Moreira et al., 2013). Temporal decorrelation may, for example, be caused by changes in scattering behaviour because of the existence of floodwater or vegetation growth.

1.2.2 Google Street View

GSV service was founded in 2007, and at the time of writing this dissertation, it covers more than 90 countries, providing images of both streets and indoor spaces. GSV images capturing the streetscape view of the cities have gained great interest after the proliferation of GSV images, advances in Machine Learning (ML) and computer vision, and the growing computational capacity to process large numbers of images (Biljecki & Ito, 2021). These images have various applications for urban studies, such as detecting objects near the sidewalks, and as a replacement for in-person street surveys (X. Li & Ratti, 2019; Nesse & Airt, 2020). The images can be retrieved using the GSV Application Programming Interface (API). Parameters for the API include image size (640 × 640 is the maximum image size), geographic location (geographic coordinates or addresses), Field Of View (FOV) or zoom level, up or down angle of the camera relative to the street view vehicle (default is 0), and heading (the direction the camera is facing with 0 = north, 90 = east, 180 = south and 270 = west) (Nguyen et al., 2019).

1.2.3 Light Detection and Ranging

LiDAR sensors are active systems that emit a laser pulse and receive it returning from the ground. The LiDAR data level of detail depends on factors, including sensor velocity, attitude, FOV, and sampling frequency (Barnsley et al., 2003). Four attributes can be retrieved from LiDAR systems: 1- the angle of the emitted pulse, 2- the time elapsed before the return signal is received at the sensor, 3- the intensity of the return signal, and 4- the timing of first and last returns. LiDAR and its derived elevation products, such as the Digital Elevation Model (DEM), can be applied for various flood risk applications. For

example, LiDAR-derived DEM can serve as the input to the hydraulic models to visualize the interface of floodwater with the elevation of the ground surface and determine flood inundation and depth. LiDAR data can also be applied for identifying terrain characteristics such as roughness, contributing significantly to hydraulic modeling. Elevation data derived from LiDAR can also serve as auxiliary information for land use mapping and determining the surface runoff characteristics (Muhadi et al., 2020). The intensity of the return signal gives information about the ground object's spectral reflectance properties and the wavelength of the laser pulse. The double return properties of LiDAR systems help differentiate solid objects (buildings) from permeable objects (vegetation canopies). While buildings' return pulses are mostly from the surface, canopies produce a more complex return signal because of the LiDAR pulse penetration through the canopy. LiDAR data can be used to detect the buildings' morphology, such as roof type and height (Meng et al., 2012). This information can be used to detect building types, for example, to differentiate houses from apartments and office buildings and eventually find their corresponding depth-damage curves. The accuracy of this information depends on the LiDAR point sample density and the buildings' size.

1.2.4 Multi-Spectral Satellite Image

Multi-spectral (MS) images provide information from the ground surface in discrete spectral bands from visible to short wavelength infrared portions of the electromagnetic spectrum. Early MS sensors provided around 10 bands while newer generation hyperspectral images, provide spectral imagery covering the same range but with hundreds of spectral bands. Although hyperspectral images are newer, in many applications, such as urban studies, MS images are preferred because of the trade-off between spectral and spatial resolution. A hyperspectral image's spatial resolution is usually coarser than an MS image because it covers thin portions of the electromagnetic spectrum, degrading its signal-to-noise ratio and spatial resolution (Lim et al., 2024).

1.2.5 Orthophoto

Since the early days of aerial photography, aerial remote sensing has emerged as a form of data collection in urban areas (Alderton & Elias, 2020). An orthophoto is a geometrically

corrected aerial image after tilt, camera perspective, and terrain topography errors removal. The measurements on an orthophoto are comparable to real-world sizes and can produce meaningful information on a building's geometric features, such as perimeter, area, and length-to-width ratio. Most aerial sensors provide images in RGB and Infrared bands, and their spatial resolution is below 1m, making them suitable for urban studies.

1.3 Deep Learning for Urban Flood Risk Analysis

In recent years, ML has been coupled with RS data and geospatial analyses and applied in numerous studies related to flood risk analysis. ML utilizes computer algorithms to evaluate information and produce predictions by being trained with a custom dataset. Several ML algorithms, including support vector machines (Opella & Hernandez, 2019), artificial neural networks (Q. Li et al., 2013), and random forest classification (Farhadi & Najafzadeh, 2021) have been applied for flood risk analysis. Traditional ML methods necessitate the prior application of specific feature engineering to raw data for effective processing. In contrast, Deep Learning (DL) can autonomously uncover the requisite representations essential for detection or classification directly from raw data. DL techniques operate as representation-learning methods, featuring multiple tiers of representation. These representations are derived by combining simple non-linear modules, each tasked with transforming the representation at a given level into a representation at a more advanced and abstract level. Consequently, the model can discern concealed patterns within the data, leading to enhanced performance over time (Bentivoglio et al., 2022). DL algorithms are of great interest because of their ability to learn complicated patterns and delicate changes in the images. They can be used for various applications such as classification, object detection, object localization, and segmentation. Classification and object localization are two areas in which the DL algorithm will be used for flood risk analysis in this dissertation.

1.3.1 Convolutional Neural Networks for flood mapping using remote sensing data

○ Convolutional Neural Networks

Convolutional Neural Networks (CNN) are artificial intelligence systems usually used for image classification, object detection, image segmentation, and Natural Language Processing. Figure 1-1 shows the different components of a CNN. A CNN is typically composed of four types of layers: 1- Convolution, 2- Pooling, 3- Activation Function, and 4- Fully Connected Layers (Taye, 2023). Convolution layers are usually embedded in the DL model layers to extract feature maps. Pooling layers are used for discarding redundant information and abstracting the feature maps. Activation functions control the output value of the learning elements (neurons) and can be in linear and non-linear formats. However, their primary usage is making the neurons learn non-linear patterns. Finally, Fully Connected Layers constitute a few layers with multiple neurons connected with all the neurons in their preceding layers. These layers are usually embedded in the last layers of a CNN.

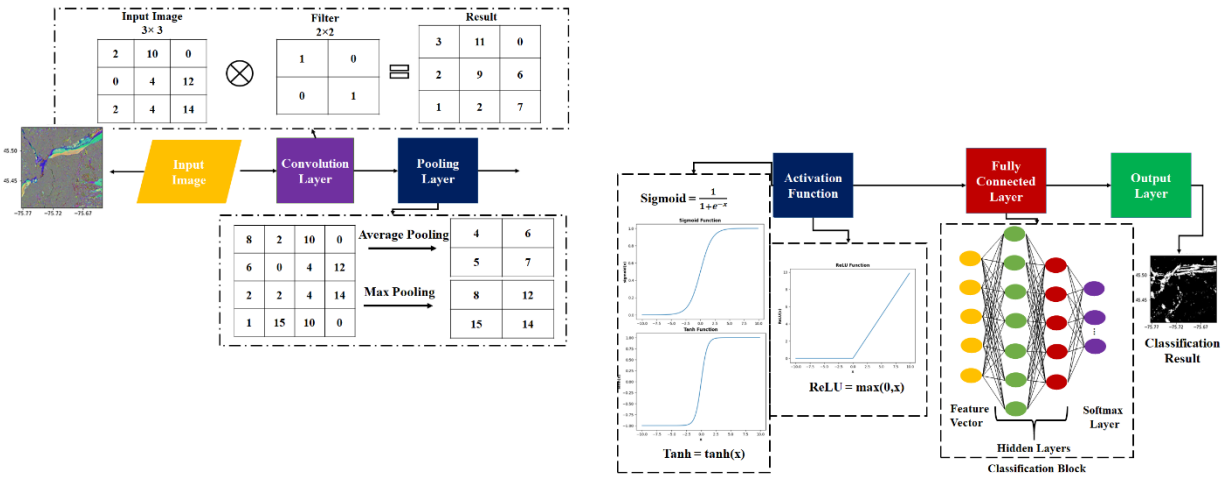


Figure 1-1: CNN for image classification; components of a CNN include Convolution Layer, Pooling Layer, Activation Function, and Fully Connected layer.

- **Convolution Layer**

Convolution Layers consist of filters (kernels) with fixed width and height. Each kernel element is assigned a weight that is a random number in the first training iteration. The weights are adjusted during training to achieve the final sub-optimal feature map. Figure 1-2 shows the convolution operation output after convolving a 2×2 filter on a 3×3 image.

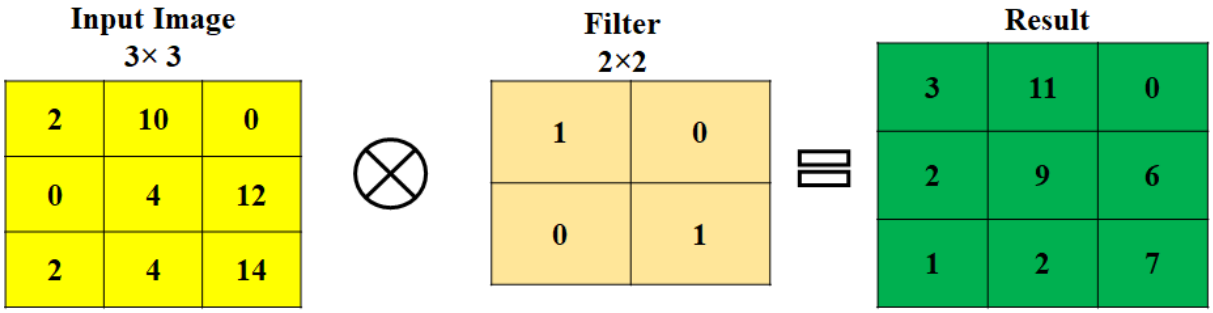


Figure 1-2: Convolution operation with filter (kernel) size 2×2 for a 3×3 image;⊗ shows the convolution sign

- **Pooling**

Pooling is used for dimension reduction in CNNs and is equivalent to reducing the spatial resolution. There are different types of pooling methods, such as max pooling, average pooling, and global average pooling. Figure 1-3 shows these operations separately. Pooling layers help CNN to discard redundant information.

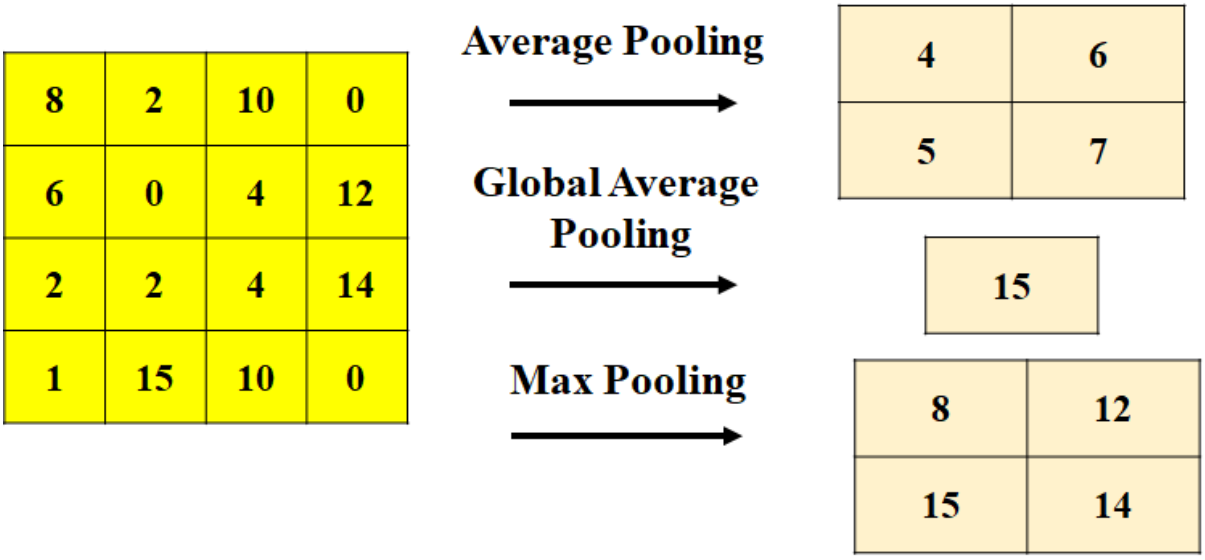


Figure 1-3: Pooling operations; Average and Max Pooling were conducted with kernel size 2×2

- **Activation Function**

This layer adds non-linearity to the network. The function input is calculated using the weighted sum of input neurons and bias. The activation function decides whether to fire a neuron or not. These layers are usually used after the convolutional and fully connected layers. Four types of activation functions commonly used in a CNN are as follows:

- Sigmoid:

  This activation function accepts real numbers as input and outputs values between 0 and 1. Equation (1-3) shows the function formula.

$$\text{Sigmoid} = \frac{1}{1+e^{-x}} \qquad (1\text{-}3)$$

- Tanh:

  This activation function is similar to Sigmoid but the output range is between -1 and 1. Equation (1-4) presents the function formula.

$$\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (1\text{-}4)$$

- Rectified Linear Unit (ReLU):

  ReLU function is the most commonly used activation function in CNNs. It transforms the negative values to zero and acts as an identity function on other values. Equation (1-5) shows the function formula.

  $$ReLU(x) = \max(0, x) \quad (1\text{-}5)$$

- Leaky-ReLU

  The ReLU function returns a zero for negative values that can cause neurons to fail to learn after the ReLU enters the negative part. This problem resulted in the introduction Leaky-ReLU function, which has a small slope to the negative input. Equation (1-6) presents the Leaky-ReLU formula. In this Equation, *k* refers to the slope for negative values, which is between 0 and 1 (Xu et al., 2020).

$$Leaky - ReLU(x) = \max(kx, x) = \begin{cases} x \; if \; x \geq 0 \\ kx \; if \; x \leq 0 \end{cases} \quad k \in [0,1] \quad (1\text{-}6)$$

- **Fully Connected Layer (FCL)**

These layers have the same architecture as the multi-layer perceptron neural networks and are usually used for classification. The output from convolution and pooling layers is the input for FCLs. Training these layers takes up lots of time because the neuron in the preceding layer is connected to all the neurons in the previous layer, making the number of

trainable parameters significant. Figure 1-4 shows the architecture of a fully connected layer applied for the classification.

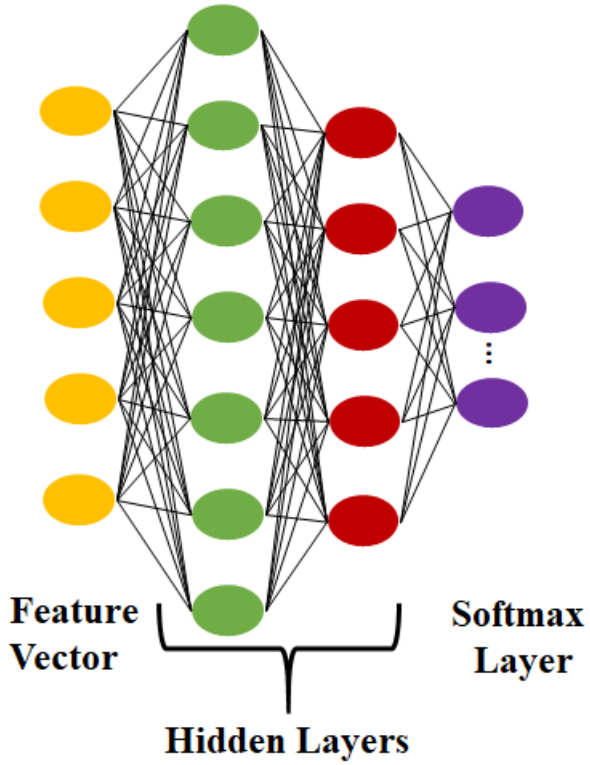


Figure 1-4: Fully Connected Layers in a CNN; these layers include an input layer (feature vector), hidden layers, and a Softmax layer used for classification.Softmax is a mathematical function that converts a vector of real numbers into a vector of probabilities, which sum up to 1. It's often used in machine learning and neural networks, particularly in multi-class classification tasks.

1.3.2 Flood mapping using Change Detection

Flood mapping using remote sensing data can be defined as a change detection problem in which flood pixels are considered as change pixels, and no change pixels are labeled as dry pixels. Among deep learning techniques, Convolutional Siamese Networks (CSN) have been widely used for change detection studies because their internal structure makes them capable of measuring similarity between the reference and event images (Chen et al., 2020; Daudt et al., 2018; de Gélis et al., 2023; Zhang et al., 2023). CSN includes two parallel CNNs, one accepting the pre-flood, and the other accepting the flood image. Then, based on the similarity between the extracted feature vectors from reference and event images, the pixels are labeled as flood or no flood. In this dissertation, the CSN is used to detect flooded areas in bi-temporal SAR images, one captured before and the other during the flood event.

1.4 Deep Learning for Object Localization in Google Street View images

Object localization aims at locating and classifying different objects in the image and creating a bounding box around each object. The uncertainty for the bounding box locations is usually reported with confidence of existence. Besides, the image coordinates of the upper left and lower right corners of the bounding boxes are included in the final results (Zhao et al., 2018).

Deep Learning-based Object Localization includes the following elements:

1. CNNs: CNNs use the convolution, pooling, and activation functions to extract patterns within the image.
2. Regression Analysis: Object localization often involves regression analysis to predict the coordinates of the upper left and lower right corners of bounding boxes around the objects. This can be done using deep learning-based regression models.
3. Intersection over Union (IoU): IoU is a fundamental metric used to assess object localization accuracy. This metric calculates the overlap between the predicted and the ground truth bounding boxes around the object.

4. Loss Functions: Loss functions, such as mean squared error (MSE) are used to quantify the difference between the predicted and ground truth bounding boxes. These functions help optimize the deep-learning model weights and produce accurate and precise bounding boxes.
5. Gradient Descent and Backpropagation: These optimization techniques are necessary for training deep learning models. They involve iteratively updating the deep learning model's weights to minimize the loss function, hence improving the accuracy of the bounding box coordinates.
6. Non-Maximum Suppression (NMS): NMS is a technique that operates based on the principle of selecting the box with the highest confidence score while discarding other overlapping boxes that probably represent the same object.
7. Matrix Operations and Linear Algebra: The principles of the mathematical operations conducted in the CNNs and deep-learning-based regression are based on matrix operations and linear algebra. Understanding these basic concepts helps us understand how these models label and locate objects in the image.

Using the above-mentioned elements, DL-based object detection algorithms can be used for various applications in computer vision. While DL-based object localization methods can be used in all kinds of images, they were used for object localization in GSV images in this dissertation.

1.5 Convolutional Neural Networks for building land-use type detection

In terms of land-use types, buildings can be categorized into apartments, houses, industrial, institutional, mixed residential/commercial, office buildings, and retail. CNNs can classify buildings' footprints into the previously mentioned building land-use type classes. The predictor part of existing DL model architectures, such as Visual Geometry Group with 16 layers (VGG16), Residual Network with 50 layers (ResNet50), and InceptionV3 can be

adjusted to fit the problem. Because of the large number of parameters and the lack of train data, especially for complex classes such as mixed residential/commercial buildings, using these pre-trained CNNs can be beneficial.

1.6 Accuracy Assessment

Our accuracy validations are based on five pixel-based indices, Overall accuracy (OA), Precision, Recall, and F1 Score and Intersection over Union (IoU). These measures have been widely used in object detection studies (Rahman and Wang, 2016; Ghasemian and Shahhosseni, 2020) and can be reported in both ratio and percentage format. In this study, the values have been reported in a ratio format. The range for all of the parameters mentioned above is between 0-1, and their best value is one. Calculation of these measures requires computing: 1- True Positive (TP), which represents the number of correctly classified building pixels. 2- True Negative (TN), the number of correctly detected background pixels. 3- False Positive (FP), i.e., the number of building pixels identified incorrectly. 4- False Negative (FN), which is defined as the number of pixels classified incorrectly as background.

OA represents the total accuracy of the object detection, and is calculated as follows:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{1-7}$$

Precision, also known as positive predictive rate, measures the ratio of correctly detected pixels in a specific class to the total pixels classified in that class. This quantity is computed as:

$$Precision = \frac{TP}{TP + FP} \tag{1-8}$$

Recall, also known as sensitivity, quantifies the ratio of truly classified pixels out of the total number of pixels in ground truth data labelled as a pre-determined class. Its value

represents the strength of the object detection algorithm in remembering the attributes of the class. This quantity is calculated as follows:

$$recall = \frac{TP}{TP + FN} \tag{1-9}$$

F1 Score is a classification accuracy measure and can be considered as the harmonic mean of the precision and recall values. It gives an overall performance of the object detection algorithm referring to both classification map and ground truth. The higher this index, the better the object detection result would be:

$$F1\ Score = \frac{2PR}{P + R} \tag{1-10}$$

In equation (1-10), P and R denote precision and recall values, respectively (Maltezos et al., 2017).

IoU measures the similarity between the object detection result and ground truth and is defined as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{1-11}$$

1.7 Research Gaps, Objectives, and Questions

Most previous SAR-based flood mapping techniques have focused on using intensity data for urban flood mapping, neglecting coherency, another valuable source of information extractable from SAR images. Further, previous DL-based flood mapping works have mainly taken advantage of optical images, overlooking the simultaneous use of CSN and SAR data for urban flood mapping. Most ML-based FFH estimation methods require training data, and no automatic FFH estimation method is solely based on image

measurements. Algorithms using deep learning for building land-use type classification have focused on residential buildings, neglecting other building types, such as office and mixed residential/commercial buildings. There is no fully automatic deep-learning-based workflow for building footprint detection and differentiating different building land-use types.

The main objective of this dissertation was to estimate two risk components, hazard, and vulnerability, using DL algorithms. Three sub-objectives were defined to achieve this task:

1- The first sub-objective was to map urban flood extent (hazard analysis). To achieve this task, we used a CSN with SAR satellite images.

2- The second sub-objective was to estimate FFH (vulnerability analysis). DL-based object localization methods and GSV images were applied to accomplish this task.

3- The third sub-objective was to detect building footprints and land-use types. CNN models were used to extract building footprints and classify them into detailed building land-use types using EO data. The output of building footprint and land-use type detection impacts flood damage analysis indirectly. Flood damage analysis falls within the vulnerability analysis domain.

This dissertation explores the following research questions to achieve flood hazard and vulnerability analysis:

How can urban flood mapping accuracy using SAR satellite images be improved?

How can FFH estimation be accomplished using computer vision?

How can the building land-use type detection accuracy using EO data be improved?

How can buildings in mixed land-use types, such as mixed residential/commercial, be detected using EO data?

1.8 Study Area and Data

Chapter 2 explored three flood events: the 2019 Ontario-Quebec and 2021 Abbotsford, BC flood events in Canada, and the Leverkusen, Germany flood event, which was one of the severe floods that hit Europe in the summer of 2021, causing significant damage to people and infrastructures. Also, two publicly available flood datasets, Sen1Floods11 and SEN12-FLOOD, were used in Chapter 2 to explore flood mapping on flood events distributed all over the globe. Figure 1-5 shows the spatial distribution of the Sen1Floods11 dataset (Bonafilia et al., 2020). SEN12-FLOOD data includes the time series of Sentinel-1 and Sentinel-2 data, and most image scenes were from Southeastern Africa, but other samples were from West African, Iranian, and Australian locations (Rambour et al., 2020). Sentinel-2 images in SEN12FLOOD data were not used in this dissertation because Chapter 2 was focused on SAR flood mapping.

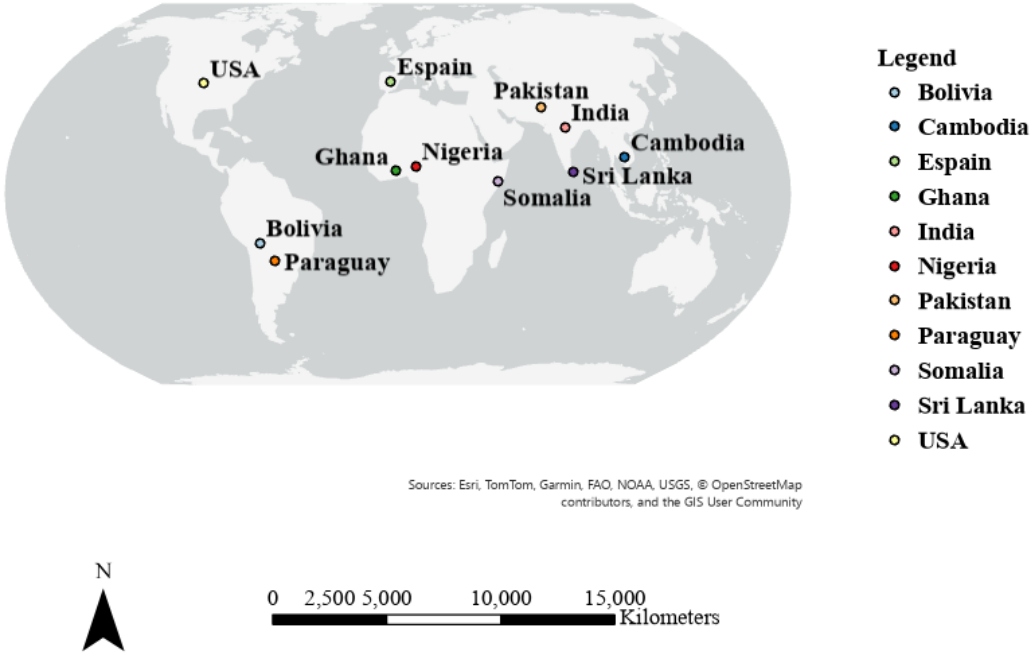


Figure 1-5: Sen1Floods11 data distribution

Flood damage assessment is conducted using depth-damage curves. These curves relate absolute or relative damage to the water depth. Each curve is usually developed separately for each building based on their land-use type. Hence, information on building land-use type is essential for accurate damage assessment. Insurance companies across North America have started documenting FFH in Elevation Certificates (EC) to set the premiums and protect people's lives and properties after flood events. Chapter 3 explores FFH estimation for the Greater Toronto Area (GTA) and Hampton Roads, Virginia, US using a DL-based object detection algorithm and GSV images.

Before detecting the building land-use type, the buildings' footprint extraction using DL algorithms is necessary. Then, the buildings' land-use type can be detected for each building footprint. The building footprint extraction, covered in Chapter 4, was explored for Toronto using a Dense Attention Network (DAN). The proposed algorithm was also tested on the Massachusetts Building Dataset. Chapter 5 presents building land-use type detection for three case studies. The case studies were selected because of the high density of buildings of different types in these regions. The first case study included five cities in the GTA, including Toronto, Markham, Vaughan, Richmond Hill, and Peel Region. The second and third case studies included parts of Vancouver City, BC, and Fort Worth City, Texas. The location of these case studies are shown in Figure 1-6.

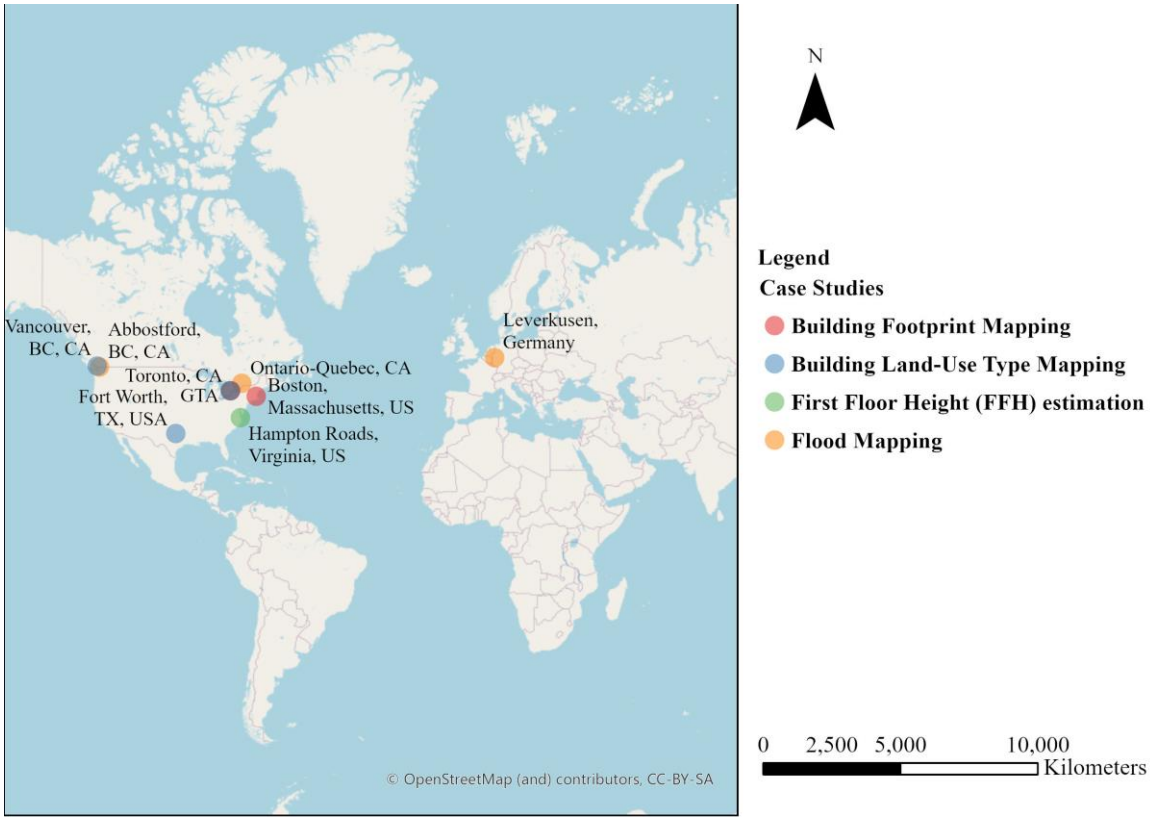


Figure 1-6: Case studies used in this dissertation. Please note that the Greater Toronto Area (GTA) was a case study for two Chapters, including Building Land-Use Type Mapping and First Floor Height (FFH) estimation. Toronto City was a case study for Chapter 4, Building Footprint Mapping.

The selection of case studies from various cities was necessitated by the constraints of data availability.

1.9 Structure of the Dissertation

This dissertation is presented in an integrated article format and contains six chapters. Chapter 1 gives background information on the following chapters' topic, and introduces

case studies, research questions, and objectives. Chapter 2 proposes a method developed based on a CNN-based change detection algorithm, CSN, for SAR flood mapping in urban and agricultural areas. Also, it discusses the effect of loss function and adding DEM on flood mapping results. Chapter 3 introduces an automatic method based on a DL-based objection detection algorithm, YOLOv5s, to estimate FFH using GSV images. Chapter 4 presents a CNN classifier based on skip connection and dense attention block concepts to detect building footprints using MS and LiDAR data. Chapter 5 introduces a fusion method, Ranking Classes Based on F1 Score, combining CNN classifiers trained on GSV, LiDAR, and Orthophoto for building land-use type detection. Chapter 6 contains the conclusion and presents future research directions. Figure 1-7 shows the overall relationships among chapters 2, 3, 4, and 5.

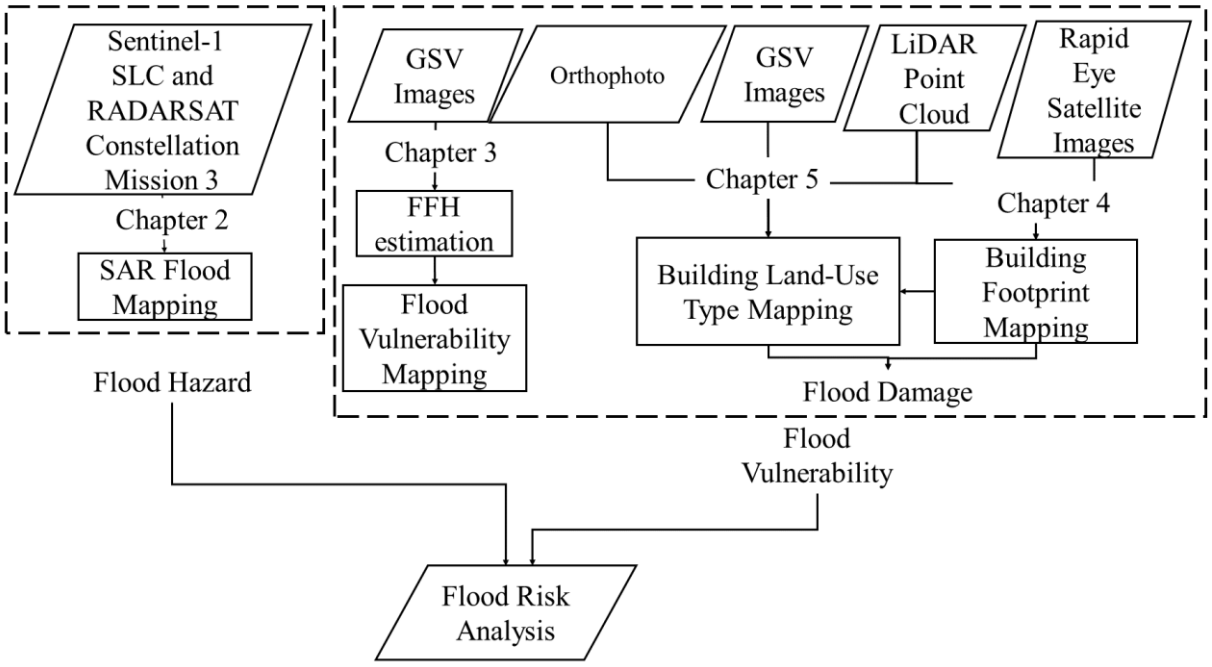


Figure 1-7: Chapter topics and their relationships with flood risk analysis

References

Alderton, D., & Elias, S. (2020). *Encyclopedia of Geology* (2nd Edition). https://shop.elsevier.com/books/encyclopedia-of-geology/elias/978-0-08-102908-4

Barnsley, M. J., Steel, A. M., & Barr, S. L. (2003). Determining urban land use through an analysis of the spatial composition of buildings identified in LIDAR and multispectral image data. *Remotely Sensed Cities*, 47–82.

Beckers, A., Dewals, B., Erpicum, S., Dujardin, S., Detrembleur, S., Teller, J., Pirotton, M., & Archambeau, P. (2013). Contribution of land use changes to future flood damage along the river Meuse in the Walloon region. *Natural Hazards and Earth System Sciences*, *13*(9), 2301–2318.

Bentivoglio, R., Isufi, E., Jonkman, S. N., & Taormina, R. (2022). Deep learning methods for flood mapping: A review of existing applications and future research directions. *Hydrology and Earth System Sciences*, *26*(16), 4345–4378.

Biljecki, F., & Ito, K. (2021). Street view imagery in urban analytics and GIS: A review. *Landscape and Urban Planning*, *215*, 104217.

Bonafilia, D., Tellman, B., Anderson, T., & Issenberg, E. (2020). Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 210–211.

Bulti, D. T., & Abebe, B. G. (2020). A review of flood modeling methods for urban pluvial flood application. *Modeling Earth Systems and Environment*, *6*, 1293–1302.

Cardona, O. D., Van Aalst, M. K., Birkmann, J., Fordham, M., Mc Gregor, G., Rosa, P., Pulwarty, R. S., Schipper, E. L. F., Sinh, B. T., Décamps, H., & others. (2012). Determinants of risk: Exposure and vulnerability. In *Managing the risks of extreme events and disasters to advance climate change adaptation: Special report of the intergovernmental panel on climate change* (pp. 65–108). Cambridge University Press.

Chen, J., Yuan, Z., Peng, J., Chen, L., Huang, H., Zhu, J., Liu, Y., & Li, H. (2020). DASNet: Dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 1194–1206.

Daudt, R. C., Le Saux, B., & Boulch, A. (2018). Fully convolutional siamese networks for change detection. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 4063–4067.

de Gélis, I., Lefèvre, S., & Corpetti, T. (2023). Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *197*, 274–291.

Farhadi, H., & Najafzadeh, M. (2021). Flood risk mapping by remote sensing data and random forest technique. *Water*, *13*(21), 3115.

GebreEgziabher, M., & Demissie, Y. (2020). Modeling urban flood inundation and recession impacted by manholes. *Water*, *12*(4), 1160.

Genovese, E. (2006). A methodological approach to land use-based flood damage assessment in urban areas: Prague case study. *Technical EUR Reports, EUR, 22497*.

Ghasemian, N. and Shahhosseini, R., 2020. Hyperspectral multiple-change detection framework based on sparse representation and support vector data description algorithms. *Journal of Applied Remote Sensing*, *14*(1), p.014523.

Li, Q., Jiang, X., & Liu, D. (2013). Analysis and modelling of flood risk assessment using information diffusion and artificial neural network. *Water Sa*, *39*(5), 643–648.

Li, X., & Ratti, C. (2019). Using Google street view for street-level urban form analysis, a case study in Cambridge, Massachusetts. *The Mathematics of Urban Morphology*, 457–470.

Li, Y., Martinis, S., Wieland, M., Schlaffer, S., & Natsuaki, R. (2019). Urban flood mapping using SAR intensity and interferometric coherence via Bayesian network fusion. *Remote Sensing*, *11*(19), 2231.

Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). *Remote sensing and image interpretation*. John Wiley & Sons.

Lim, S. L., Sreevalsan-Nair, J., & Daya Sagar, B. (2024). Multispectral data mining: A focus on remote sensing satellite images. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *14*(2), e1522.

Maltezos, E., Doulamis, N., Doulamis, A. and Ioannidis, C., 2017. Deep convolutional neural networks for building extraction from orthoimages and dense image matching point clouds. *Journal of Applied Remote Sensing*, *11*(4), p.042620.

Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., Baydin, A. G., & Backes, D. (2021). Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports*, *11*(1), 7249.

Meng, X., Currit, N., Wang, L., & Yang, X. (2012). Detect residential buildings from lidar and aerial photographs through object-oriented land-use classification. *Photogrammetric Engineering & Remote Sensing*, *78*(1), 35–44.

Montgomery, M., & Kunreuther, H. (2018). Pricing storm surge risks in Florida: Implications for determining flood insurance premiums and evaluating mitigation measures. *Risk Analysis*, *38*(11), 2275–2299.

Moreira, A., Prats-Iraola, P., Younis, M., Krieger, G., Hajnsek, I., & Papathanassiou, K. P. (2013). A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, *1*(1), 6–43.

Muhadi, N. A., Abdullah, A. F., Bejo, S. K., Mahadi, M. R., & Mijic, A. (2020). The use of LiDAR-derived DEM in flood applications: A review. *Remote Sensing*, *12*(14), 2308.

Muthusamy, M., Rivas Casado, M., Salmoral, G., Irvine, T., & Leinster, P. (2019). A remote sensing based integrated approach to quantify the impact of fluvial and pluvial flooding in an urban catchment. *Remote Sensing*, *11*(5), 577.

Nesse, K., & Airt, L. (2020). Google Street View as a replacement for in-person street surveys: Meta-analysis of findings from evaluations. *Journal of Urban Planning and Development*, *146*(2), 04020013.

Nguyen, Q. C., Khanna, S., Dwivedi, P., Huang, D., Huang, Y., Tasdizen, T., Brunisholz, K. D., Li, F., Gorman, W., Nguyen, T. T., & others. (2019). Using Google Street View to examine associations between built environment characteristics and US health outcomes. *Preventive Medicine Reports*, *14*, 100859.

Nico, G., Pappalepore, M., Pasquariello, G., Refice, A., & Samarelli, S. (2000). Comparison of SAR amplitude vs. Coherence flood detection methods-a GIS application. *International Journal of Remote Sensing*, *21*(8), 1619–1631.

Opella, J. M. A., & Hernandez, A. A. (2019). Developing a flood risk assessment using support vector machine and convolutional neural network: A conceptual framework. *2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA)*, 260–265.

Rahman, M.A. and Wang, Y., 2016, December. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing* (pp. 234-244). Springer, Cham.

Park, K., Choi, S.-H., & Yu, I. (2021). Risk type analysis of building on urban flood damage. *Water*, *13*(18), 2505.

Pistrika, A., Tsakiris, G., & Nalbantis, I. (2014). Flood depth-damage functions for built environment. *Environmental Processes*, *1*, 553–572.

Rambour, C., Audebert, N., Koeniguer, E., Le Saux, B., Crucianu, M., & Datcu, M. (2020). Flood detection in time series of optical and sar images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *43*(B2), 1343–1346.

Taye, M. M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, *11*(3), 52.

Wang, X., & Xie, H. (2018). A review on applications of remote sensing and geographic information systems (GIS) in water resources and flood risk management. *Water*, *10*(5), 608.

Xia, J., & Gong, J. (2024). Computer vision based first floor elevation estimation from mobile LiDAR data. *Automation in Construction*, *159*, 105258.

Xu, J., Li, Z., Du, B., Zhang, M., & Liu, J. (2020). Reluplex made more practical: Leaky ReLU. *2020 IEEE Symposium on Computers and Communications (ISCC)*, 1–7.

Zhang, R., Zhang, H., Ning, X., Huang, X., Wang, J., & Cui, W. (2023). Global-aware siamese network for change detection on remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *199*, 61–72.

Zhao, Z.-Q., Zheng, P., Xu, S., & Wu, X. (2018). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*, 3212–3232.

Chapter 2

Evaluation of urban flood mapping using Sentinel-1 and RADARSAT Constellation Mission image and Convolutional Siamese Network

2.1 Introduction

Urban flood mapping is challenging due to the complicated structures in cities, such as buildings, sidewalks, road culverts, and utility holes. Two types of satellite products are available for producing flood maps using remote sensing data, optical and SAR images. Optical images are not always available during a flood event because they are affected by dense cloud cover. SAR sensors, however, can capture images from flood-affected areas at longer electromagnetic wavelengths making it possible for the SAR signal to penetrate the cloud cover. Besides, since the SAR sensor is active, it is not dependent on sunlight. So, SAR images are available in all weather conditions and during the day and night. This characteristic makes SAR data suitable for flood mapping.

Generally, three specific features can be extracted from a SAR image, intensity, polarimetry decompositions, and InSAR coherence. Intensity is a measure of the reflective strength of an object, polarimetry decomposition gives the polarimetric discriminators that can be used for classification and image interpretation, and interferometric coherence (correlation) is a measure of the accuracy of the determined radar signal, and its value decreases by temporal changes. While intensity reflects the electromagnetic characteristics of the radar backscattering, it is affected by the speckle noise. Flood mapping based solely on intensity data might result in erroneous flood maps because of the complex structures in urban areas. Vertical structures, like buildings, can enhance the double bounce scattering effect, which can be further intensified when the floodwater covers the bottom of the tree. Coherence data are complementary in flood mapping studies using the SAR dataset (Pulvirenti et al., 2021; Olthof and Svacina, 2020), and the coherency maps show high values in urban areas because of the stability of urban structures during short time intervals. When producing a flood map, a coherency map can complement the intensity data and improve flood mapping accuracy (Zhang et al., 2021). Besides, steady targets such as

buildings make InSAR coherence data useful for urban areas with limited vegetated regions. When vegetation cover is limited, any decrease in coherence values in an InSAR time series can be translated into a flood event. Also, speckle noise is reduced when producing coherency images because the noise is averaged when integrating the two SAR images. Because of the dynamic behaviour of the vegetated areas (due to growth), it is difficult to attribute coherence change to the vegetated areas or a flood. Sometimes this problem is addressed using SAR images with a short revisit time, less than five days, like COSMO-SkyMed, but such datasets are not accessible quickly, especially for flood hazard management studies in which time plays a vital role (Pierdicca et al., 2018). Another limitation when using SAR data for flood extent mapping in urban areas is the shadowing effect. The shadowing effect in a SAR image happens when the SAR signal does not reach some regions because higher objects create an obstacle between the SAR antenna and the area (Bouvet et al., 2018). The shadowed areas on the image are overlooked when performing flood extent mapping using SAR data.

Flood extent mapping techniques can be categorized into four groups based on the theories applied: 1- Hydrologic/Hydraulic modelling; 2- Multi-Criteria Decision Analysis (MCDA); 3- Machine Learning; 4- Hybrid methods. Hydrologic models can simulate runoff values during a flood event, and Hydraulic models provide information on flow movement and inundation depth in areas near a river network. Multi-criteria techniques assign a weight to each flood indicator, such as topographic, hydrologic, climatic, and anthropogenic parameters, to produce a final flood risk map. Machine learning approaches, aka Artificial Intelligence techniques, use training data to discriminate between flooded and dry areas based on geospatial input features. Hybrid techniques use a combination of previously mentioned methods to model flood events, such as integrating hydraulic modelling and the Analytical Hierarchical Process (AHP) technique to produce a flood risk map (Nguyen et al., 2021).

Deep Learning, aka deep structured learning, is a machine learning technique based on artificial neural networks with representation learning. Although ML algorithms such as neural networks, random forest, and support vector machines have proven promising methods for flood mapping, DL methods, especially CNNs, have shown higher capability

than the previous ML methods to extract features at different scales such as edges and objects (Muñoz et al., 2021). Li et al. (2019a) introduced a CNN to produce a flood map in Houston, USA, during Hurricane Harvey in August 2017 based on TerraSAR-X intensity and coherence data. This study focused on fluvial flooding, and its efficiency for coastal or pluvial flooding was not examined. Some DL-based segmentation models such as Unet, Unet++, and DeepLabV3 have been proposed in the literature for flood mapping, and they have achieved promising results on both optic and SAR images. Wang et al. (2022), proposed a DL model based on Unet for flood water extraction in Poyang Lake in China using Sentinel-1 SAR images. Jaisakthi et al. (2021), proposed a modified Unet algorithm for flood detection using Planet Scope images and reported an overall accuracy of about 70% on validation data. The flood masks in this work were not compared with any ground truth dataset. Konapala et al. (2021), used Unet for flood inundation mapping using Sen1Floods11 data, including Sentinel-1 and Sentinel-2 images from 11 flood events around the globe. After adding elevation data to the input, the flood median F1 score improved from 0.62 to 0.73 compared with using only Sentinel-1 bands. Mateo-Garcia et al. (2021) used the WorldFloods dataset and CNN for flood segmentation and compared the method with linear and thresholding methods. A high recall rate above 94% was achieved for the flood. However, SAR data was not examined for flood mapping. Mayer et al. (2021) used Unet and Sentinel-1 data to map surface water in Cambodia in Southeast Asia. Accuracies above 80% were achieved. Although high accuracy indices were obtained for surface water, the method was not tested in urban areas. Chen et al. (2022) proposed a Siamese Network based on Unet for building change detection in very high-resolution remote sensing images and reported promising accuracies after comparison with ground truth data. Their method was not tested for flood-induced changes in satellite images.

Convolution Siamese Network (CSN) is one type of DL algorithm that has been applied for change detection (Yang et al., 2021; Wang et al., 2020; Chen et al., 2020). This method highlights changed areas using a bi-temporal remote sensing dataset. CSNs use two parallel CNN in their internal architecture and are used in change detection problems. In CSN, one CNN focuses on the pre-event image and the other works on the co-event image. In this way, CSN is more applicable for change detection problems (here flood mapping) than the usual CNN network. Some recent studies have used Siamese Networks for remote sensing

change detection applications. For example, Jiang et al. (2021) applied a Siamese Network called S3N. This network used Visual Geometry Group (VGG) as subnetworks and was applied to detect changes in various types of remote sensing data, including panchromatic, MS, SAR, PolSAR and NDVI images. The problem of high computational cost and lack of training data was addressed by applying the transfer learning strategy. They concluded that their proposed architecture is more computationally efficient than state-of-the-art techniques while giving comparable results to the existing methods. Wang et al. (2021) presented a fully CSN trained on Focal Contrastive Loss (FCL) to address the imbalanced data problem by focusing on the samples with fewer train data. Zhang et al. (2022) proposed a Siamese Residual Multi-kernel Pooling module (SRMP) to improve the high-level change information extraction from optical images. A feature difference module was also proposed to extract low-level features and help the model generate more accurate details. In another work, a Siamese Segmentation Network, SiHDNet, was proposed for building change detection. The proposed method was based on deep, high-resolution differential feature interaction. The difference map was created through a special fusion module to obtain sufficient and effective change information. The final binary change map was acquired through the improved spatial pyramid pooling module (Liang et al., 2022). Yang et al. (2021) proposed a new change detection algorithm based on the Siamese Network, MRA-SNet, for building, road, and land cover change detection in optical remote sensing images. The UNet network was used as the backbone architecture, and the bi-temporal images were imported separately to the encoder. The ordinary convolution blocks were replaced with Multi-Res blocks to extract spatial and spectral features of different scales in remote sensing images. Hänsch et al. (2022) developed the Spacenet 8 dataset from high-resolution optical satellite images and used the Convolutional Siamese Network to detect flooded roads and buildings. The mean overall IOU of 0.66 was reported in the best case.

These studies however were all based on optical image data for change detection, and they did not address the challenges associated with the SAR change detection problem. Recently, Siamese Networks have been used for flood mapping studies. Zhang et al. (2022) proposed a domain adaptation-based multi-source change detection method for heterogeneous remote sensing images. The Landsat-8 image was used as a pre-event, and

the Sentinel-1A image was used as the co-event for flood mapping of the 2017 California event. The area studied for flood mapping in this work covered agricultural lands, not dense urban areas. An urban flood mapping approach using SAR satellite image based on a change detection framework with CNN was proposed in this Chapter to address the mentioned research gaps.

2.2 Study Area and Data

Canada has experienced more frequent and intense flooding in recent years, such as the 2019 Ontario-Quebec and 2021 B.C flood events. Sentinel-1 (launch in 2014) satellite imagery provides an opportunity to study these floods. In addition, flood detection for the Leverkusen city in Germany, which was severely impacted during the 2021 European flood event, was investigated to show the generalization of the proposed CSN to other areas and because of the availability of the ground truth data. Figure 2-1 shows the location of these case studies on the map.

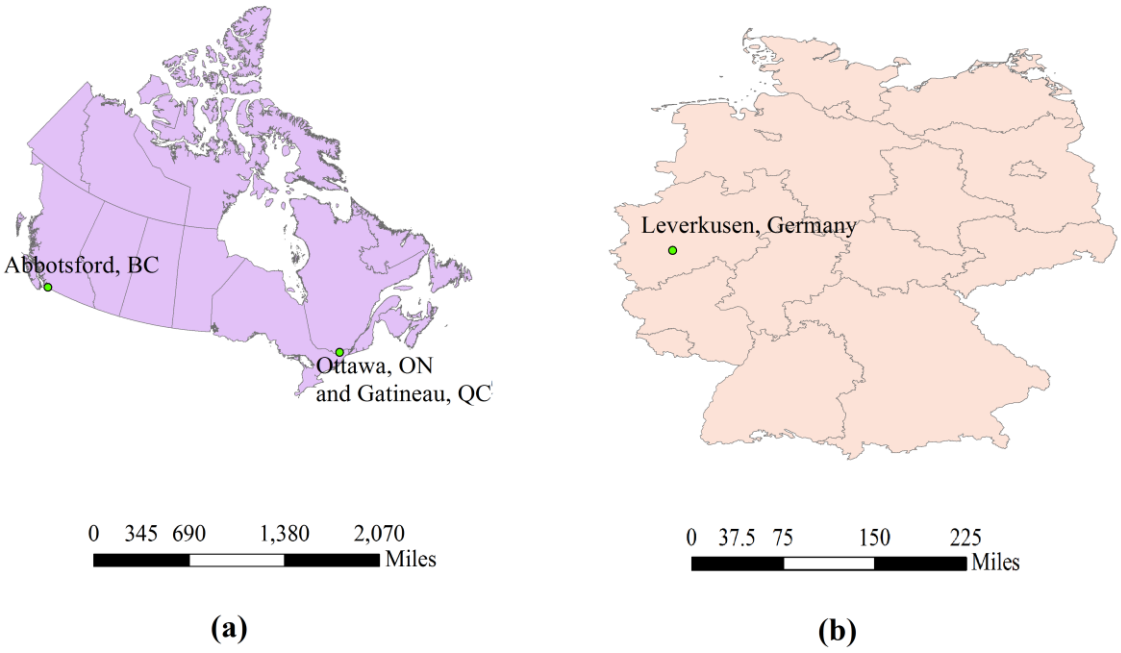


Figure 2-1: Location of case studies; a) Ontario-Quebec and BC case studies. b) Germany case study.

2.2.1 2019 Ontario and Quebec Flood Event

Heavy rainfall from mid-April until mid-May and snowfall accumulation 50% greater than expected caused flooding in eastern Ontario and southern Quebec. This event was among the top ten natural disasters of the year and was even more severe than the flood event in 2017. The Ottawa River peak height went beyond the values recorded in 2017, about 30 cm. Ottawa and Gatineau were among the affected municipalities. These cities and nearby regions experienced a severe flood causing 111 homes to evacuate, 923 people injured, and insured losses from this event cost about $201 million across Ontario and Quebec (Olthof and Svacina, 2020).

Figure 2-2 shows the extent of the study area, which includes parts of Ottawa and Gatineau cities.

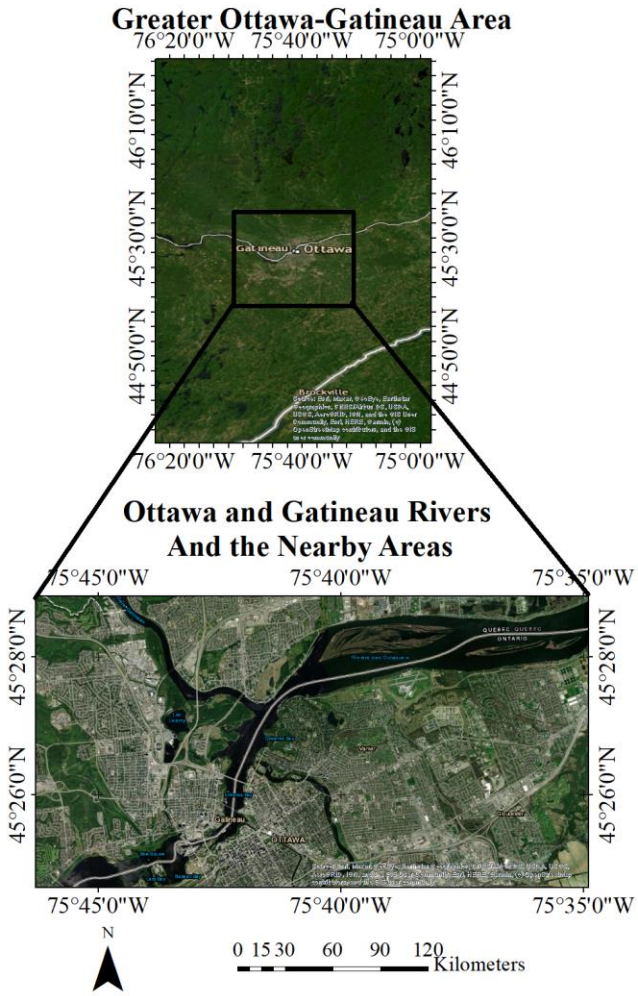


Figure 2-2: Study area for the 2019 Ontario and Quebec flood event

2.2.2 2021 British Columbia Flood Event

The 2021 Pacific Northwest floods include a series of floods that influenced British Columbia, Canada, and neighbouring Washington state in the United States. Heavy rains caused flooding in parts of southern British Columbia and the northwestern United States, starting from November 14 until December 17. In December, the Insurance Bureau of Canada reported that the flooding was the costliest natural disaster in British Columbia's history, costing at least 450 million CAD in insured damage. The natural disaster provoked

an emergency state for British Columbia, and at least five people were killed, and ten others were hospitalized during the event. The Nooksack River which flows north of Bellingham in Washington State was flooded. The floodwater ended up in the Sumas River, and the water flowed northeast, crossing the border into Abbotsford. Figure 2-3 shows Abbotsford city and the selected region on the map.

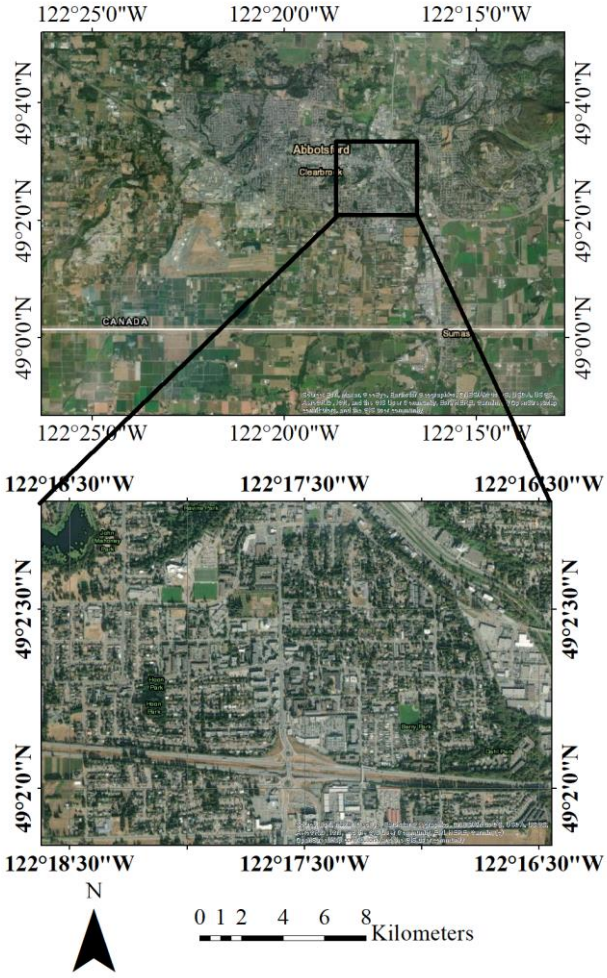


Figure 2-3: Abbotsford, BC

2.2.3 2021 Germany Flood Event

In July 2021, several countries in Europe experienced severe floods. Some of these floods caused severe impacts on lives and properties. The floods started in the United Kingdom and later affected several river basins across Europe, including Germany. The states of Rhineland-Palatinate and North Rhine-Westphalia were particularly hard hit, causing 196 death tolls. Further downstream of the Rhine river, the heaviest rainfall ever measured over 24 hours caused flooding in cities including Cologne and Hagen, while in Leverkusen, 400 people had to be evacuated from a hospital. Figure 2-4 shows the Leverkusen city location and the Rhine river on the map.

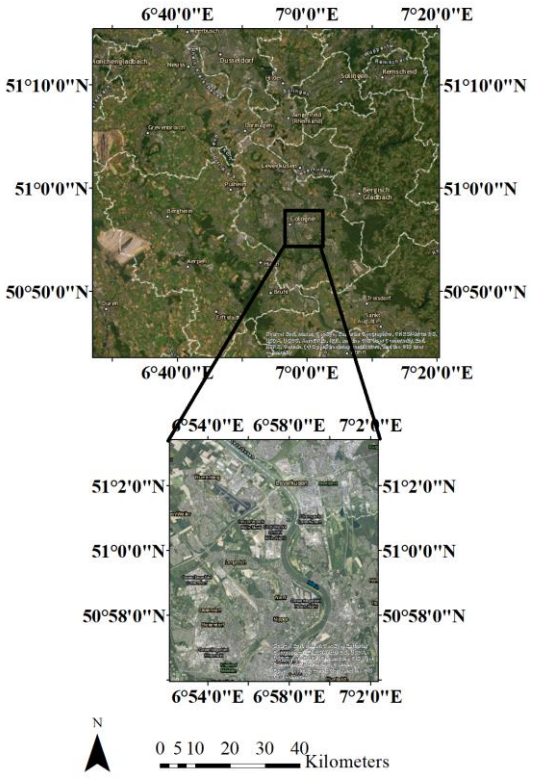


Figure 2-4: Leverkusen City, Germany

2.2.4 Input Data For Flood Mapping Using CSN

The dataset used for flood extent mapping was presented in Table 2-1.

Table 2-1: Dataset used for flood events

Case study/event	Data	Description
2019 Ontario and Quebec flood, CA	Sentinel-1 Level-1A GRD	Data type: intensity SAR image
		Resolution: 10 m
		Imaging mode: IW
	Sentinel-1 Level-1A SLC	Data type: interferometry SAR image
		Resolution: 10 m
		Imaging mode: IW
	SRTM DEM	Resolution: 30m
2021 BC flood, CA	RADARSAT Constellation Mission (RCM)	Data type: SAR image
		Resolution: 5m
		Imaging mode: Stripmap
	Gridded CDED format DEM	Resolution: 25m
2021 Leverkusen flood, Germany	Sentinel-1 Level-1A/B GRD	Data type: intensity SAR image

		Resolution: 10m
		Imaging mode: IW
	Sentinel-1 Level-1A/B SLC	Data type: interferometry SAR image
		Resolution: 10m
		Imaging mode: IW
	SRTM DEM	Resolution: 30m

Both intensity and coherence data were used in this study. Sentinel-1 GRD datasets were used for producing intensity data. One image before the flood event and the other during the flood was selected. The raw pixel values were converted to the radar backscatter coefficient (σ°) using the calibration toolbox in SNAP software. Also, the Sentinel-1 SLC dataset was used to produce the coherency feature maps for both the pre and co-event flood images. The coherency maps between two dates were computed in the SNAP software for both VV and VH images using the procedure shown in Figure 2-5.

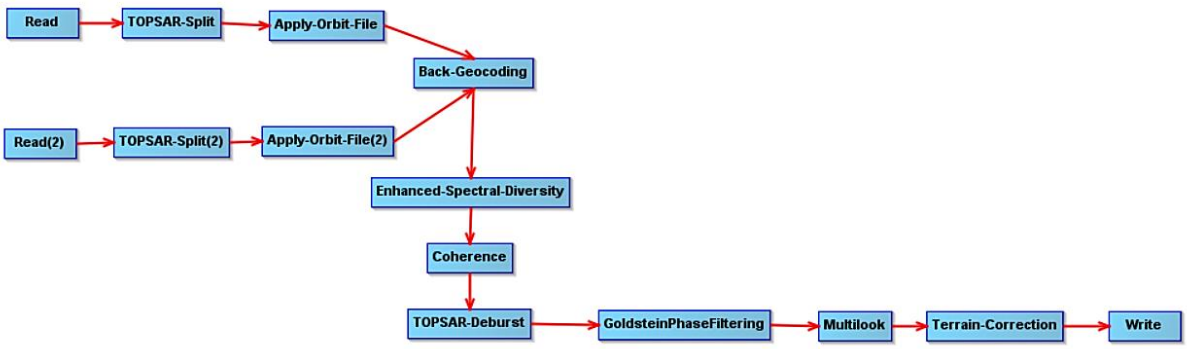
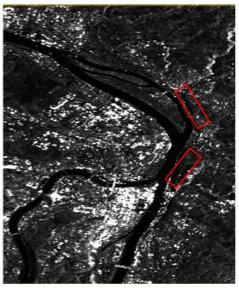


Figure 2-5: The procedure used in SNAP software for producing the coherency map

Pre and post-event images were selected temporally as close as possible to the event image acquisition date to reduce unwanted changes such as vegetation growth and anthropogenic activities. For the 2019 Ontario and Quebec case study, one coherency map was computed between the 27th Mar and the 8th Apr (two dates before the flood event). Another was computed between the 14th of May and the 8th Apr (one date during and another before the flood event). For Abbotsford city, three high-resolution dual-polarized RCM intensity images were available for the flood event. Two Radarsat Constellation Mission 3 (RCM 3) data were available, one from the flood event on 18th Nov and the other after the flood event on 30th Nov. Also, one RCM2 data was available during the flood event on 19th Nov that was used as the co-event test image. There were no RCM SLC data available for the area during the flood event. For the Leverkusen region (Germany), Sentinel-1A/B intensity and coherency data were used for flood mapping. The pre-event intensity data were captured on the 7th (S1B) and 10th (S1A) of July. The pre-event coherency maps were extracted from 24th Jun and 6th Jul, and two sets of co-event counterparts were computed between the 18th and 6th (S1A) July and 19th and 7th July (S1B). Figure 2-6 shows the original SAR images for this flood event. The input features used for flood mapping in the three case studies were shown in Table 2-2.

(a): Pre-event Sentinel-1 Intensity Image (VV) - Leverkusen, Germany



(b): Co-event Sentinel-1 Intensity Image (VV) - Leverkusen, Germany

Figure 2-6: Original Sentinel-1 intensity images for Leverkusen, Germany; (a): Pre-event intensity image (VV); (b): Co-event intensity image (VV); the highlighted areas show the regions where flood reduced backscattering values.

Table 2-2: Input dataset

Data type	Date(s) (Ottawa, ON and Gatineau, QC)		Date(s) (Abbotsford, BC)		Date(s) (Leverkusen, Germany)	
Intensity	Pre-event	Co-event	Post-event	Co-event	Pre-event	Co-event
$\sigma_{VV}^{0}\,(db), \sigma_{VH}^{0}\,(db)$	08/04/2019 (train/test)	25/04/2019 (train) 07/05/2019 (test)	-	-	2021/07/10 (train) 2021/07/07 (test)	2021/07/18 (train) 2021/07/16 (test)
$\sigma_{HH}^{0}\,(db), \sigma_{HV}^{0}\,(db)$	-	-	2019/11/30 (train/test)	2019/11/18 (train) 2019/11/19 (test)	-	-
Coherency	Pre-event	Co-event	Post-event	Co-event	Pre-event	Co-event
γ_{VV}, γ_{VH}	2019/03/27-2019/04/08 (train/test)	2019/05/02-2019/04/08 (train) 2019/05/14-2019/04/08 (test)	-	-	2021/07/06-2021/06/24 (train/test)	2021/07/18-2021/07/06 (train) 2021/07/19-2021/07/07 (test)

Two publicly available flood image datasets, Sen1Floods11 and SEN12-FLOOD, were also included in the experiments to examine the generalization ability of the proposed CSN. Sen1Floods11 consists of 11 flood events, including, events in Bolivia, Cambodia, Ghana, India, Nigeria, Pakistan, Paraguay, Somalia, Spain, Sri Lanka, and the USA (Bonafilia et al., 2020). The images in this dataset are 512×512 Sentinel-1 images with VV and VH polarizations and 10m spatial resolution and were accessed via the Google Cloud storage bucket. The labels for the dataset are in two forms, hand-labeled and labels achieved using Otsu's thresholding (Otsu, 1979). The experiments in this work are based on the hand-labeled dataset.

The SEN12-FLOOD dataset consists of co-registered Senetinel-1 and Sentinel-2 image time series for flood detection. The data is freely available via the Radiant MLHUB website. Because the primary concern of this research work was SAR flood detection, just the SAR images were imported to the CSN. The SAR images consisted of images with a size of about 620×550 and 10m spatial resolution in VV and VH polarizations. The data were from flood events in West Africa, Iran, and Australia, and included pre-flood and post-flood images in urban and rural areas captured from December 2018 to May 2019 (Rambour et al., 2020; Aparna and Sudha, 2022). It is worth mentioning that Sen1Floods11 and SEN12-FLOOD use different labeling formats. While Sen1Floods11 was labeled pixel by pixel, SEN12-FLOOD was labeled image by image. It means that the images, in which flood was present, were tagged as flooded.

2.3 Methodology

2.3.1 Flood Mapping Using CSN Based On a Change Detection Framework

In this study, flood mapping was conducted based on a deep learning-based change detection technique called Convolutional Siamese Network (CSN). Siamese networks generally consist of two CNN networks running parallel and having the same parameters, identical in number and value. One CNN network is applied on the pre-event image and the other one on the co-event image. Because of the high heterogeneity in the image scene,

the input image bands need to be divided into more homogenous regions or image segments with size *(H, W, N)*, which denote the height, width, and the number of input bands for each input image segment (*image patch* hereafter), respectively. The size of the image patch depends on the CNN input layer configuration. In this study, the input layer was designed to accept image patches with a size of 32×32. The patch size was set based on experimental experience, a very small patch size reduces the information content and a very large patch size increases the processing time. So, it is a balance between the information content imported to the network and processing time. Input image bands are segmented into regions with sizes (32, 32, *N*) to be compatible with the input layer, and the central pixel label in each image patch is used as the target label. The proposed CSN has three blocks (subnetworks), the first block is the input layer, second is the feature extraction, and the third is the prediction block. Figure 2-7 shows how the three blocks have been embedded into the network, and Figures 2-8 - 2-10 show the graphical abstracts for each block.

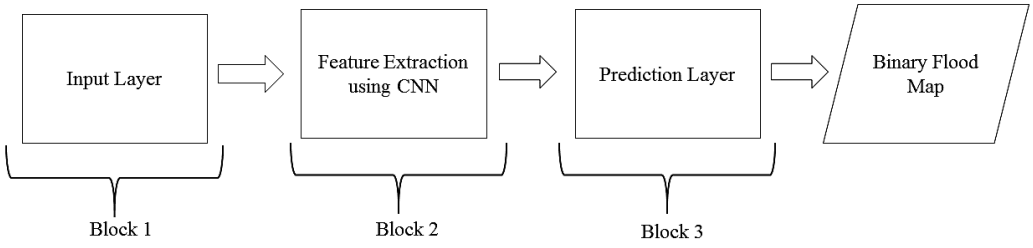


Figure 2-7: Different blocks in Flood Map generation using CSN

Block 1 and 2: Input layer and feature extraction using Resnet50

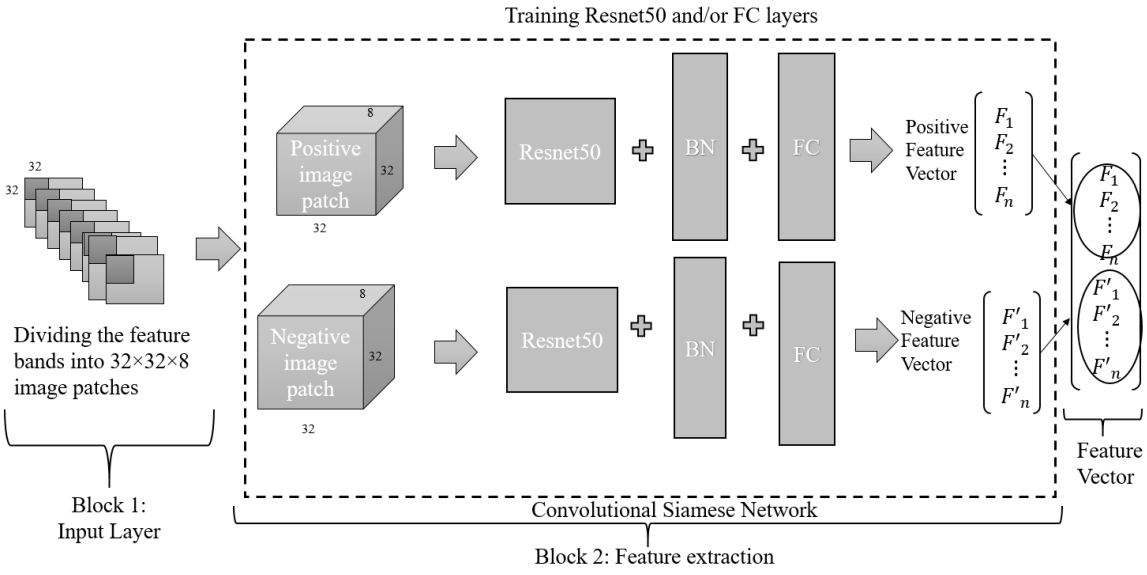


Figure 2-8: Graphical abstract for blocks 1 and 2

Block 3: Prediction layer (training phase); assigning Flood and No Flood labels

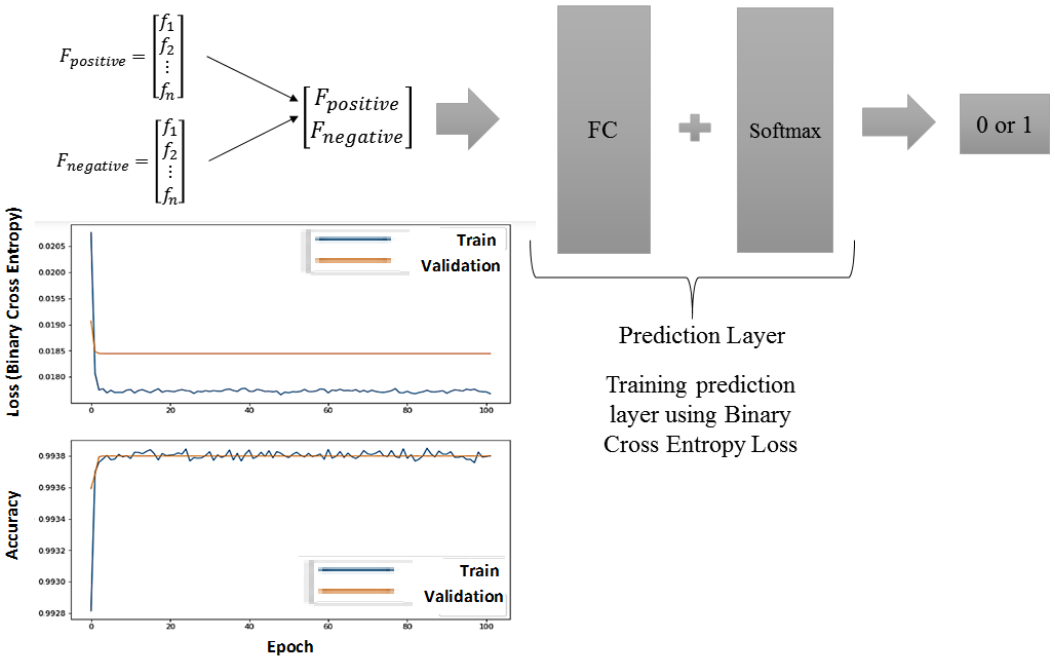


Figure 2-9: Graphical abstract for block 3 (training phase)

Block 4: Prediction layer (test phase); assigning Flood and No Flood labels

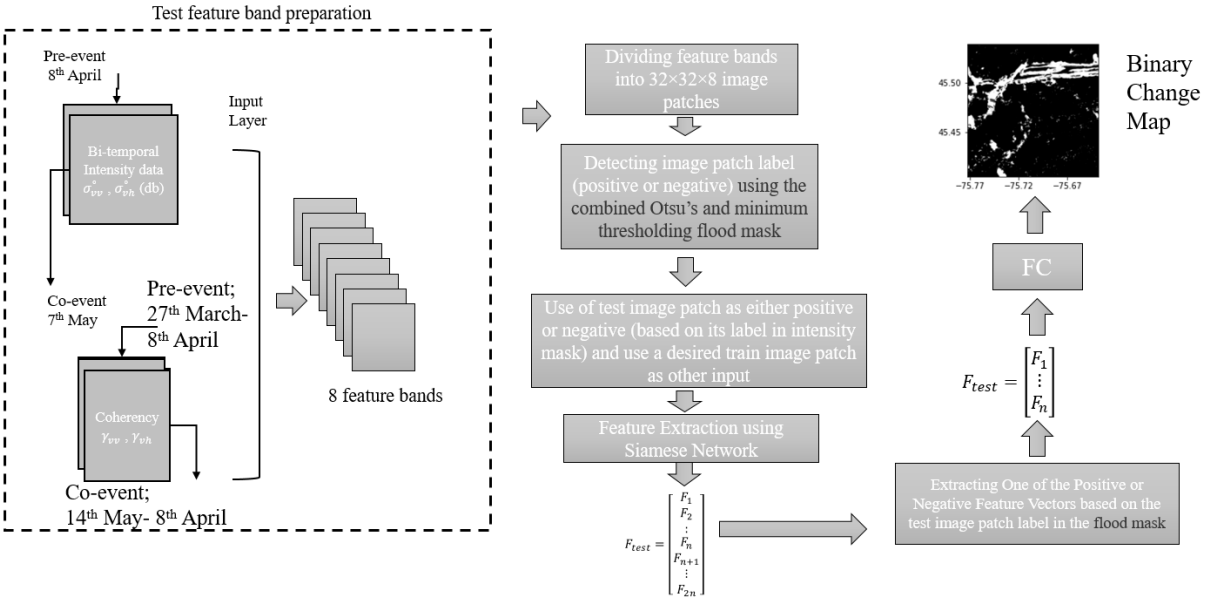


Figure 2-10: Graphical abstract for block 4 (test phase); the dates are related to the 2019 Ontario and Quebec flood event

The configuration of train data depends on the loss function applied in the feature extraction phase. For example, if the Triplet Loss function is used, a triplet set of image patches *[anchor, positive, negative]* is imported into the network. In the case of a Contrastive Loss function, the input training set will be a duplet set of image patches, *[positive, negative]*, and the anchor patches will be removed from the training data. The 2D image patch is converted into a 1D feature vector using the convolution and pooling layers in the feature extraction block. The user defines the feature vector size, which is called the embedding dimension. Convolution layers extract the 2D feature maps, and pooling layers make the extracted feature maps more abstract and extract the gist of information from the feature maps. Some CNN networks have been designed, by computer vision experts, and trained on big RGB image databases, such as *ImageNet*. These pre-trained CNN networks such as VGG16, VGG19 (Simonyan and Zisserman, 2016), ResNet50 (He et al., 2016), and inception_v3 (Szegedy et al., 2016) are available via the TensorFlow library in Python. These model parameters can be kept fixed and used for other

classification problems. Although the use of pre-trained models is valuable for flood mapping studies to create a map in the shortest possible time, because the input data was a SAR image with different textural information than the RGB image, both pre-training and training from scratch (when all the network parameters are trained from the very beginning) strategies were tested to evaluate which works better for the SAR data. The ResNet50 and VGG16 were used as the backbone CNN architecture because they have been previously found effective for image feature extraction. Although the applied CNN architectures here were ResNet50 and VGG16, the methodology is not limited to a specific architecture and can be generalized to other pre-trained CNN models. The imported train data via the input layer is used to adjust the feature extraction block parameters (CNN parameters). The CNN parameters are adjusted based on the loss function values, i.e., if the loss value is high, the parameters are changed to reduce the loss value.

2.3.2 Train Data Preparation

When using DL algorithms for flood mapping, train and test/ground truth data should not overlap for reliable accuracy assessment. In this chapter, a k-fold cross-validation procedure was applied to train and test the DL models, specifically for each of the three flood events and for each publically available dataset. In the k-fold approach, the data, D, is partitioned into k non-overlapping subsets, D_1, D_2, ..., and D_k of equal size. The model is trained k times, and at each iteration, t, the dataset D is used for training without the subset D_t, considered as test data. In the first iteration, the first subset is used as the test, while all the other subsets are considered as the train data. In the next iteration, a different subset is selected as the test while the rest of the dataset is considered as the train data, and the model is retrained. The number of folds, k, is usually set with trial and error in spatial applications (Ghorbanzadeh et al., 2018). For DL models, a validation dataset is also needed, so 20% of the train data in each fold was set as the validation. Before training the DL models in each iteration, data augmentation was applied on train, validation, and test data to increase the number of each set. The data augmentation method applied was horizontal flipping (Chen and Fan, 2021; Deng et al., 2020; Lalitha and Latha, 2022; Shawky et al., 2020; Yu et al., 2017). Please note that firstly, the data was split into folds, then data augmentation was applied to each set separately to make sure the train, validation,

and test data did not overlap. In this work, the number of folds was set to 5, and at each fold, about 400,000, 100,000, and 200,000 image patches, with size 32×32, were considered for train, validation, and test, respectively. These numbers were considered for each flood event, including 2019 Ontario and Quebec, 2021 BC, and 2021 Germany, separately. Figure 2-11 shows the flood masks used as input data for the case studies.

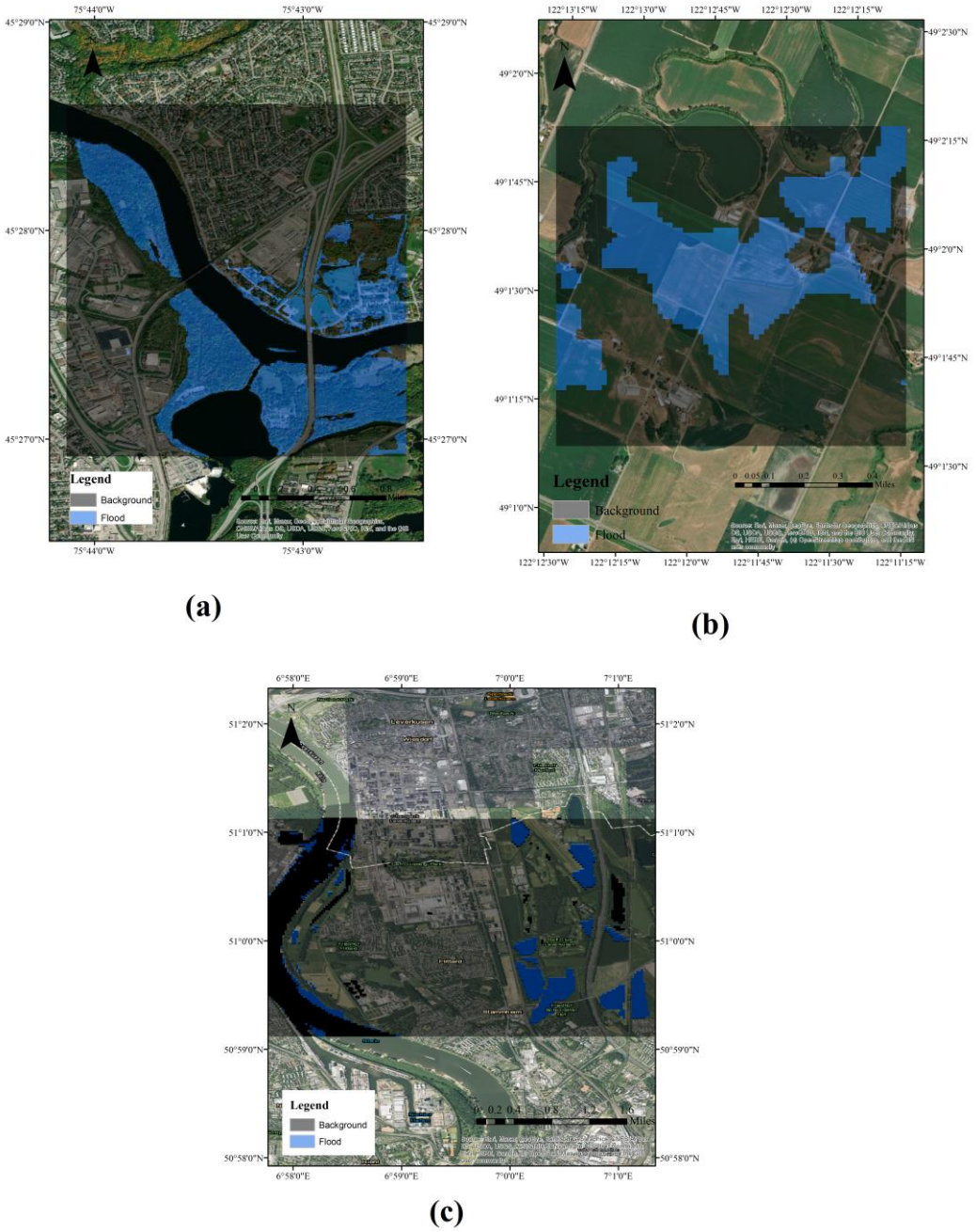


Figure 2-11: Input flood masks; the data was split to train and test using k-fold cross validation ; a) Ottawa and Gatineau area; b) Abbotsford, BC (agricultural area); c) Leverkusen, Germany; The blue and black shades were used for showing samples for the flood and background areas, respectively.

The Sen1Floods11 dataset does not include pre-event images. Sentinel-1 pre-event images were automatically downloaded for each region using Google Earth Engine. The nearest dry month to the event image was selected as the pre-event month. For a dry month with more than one Sentinel-1 image available, the closest date to the flood image was considered a pre-event image date. The dataset were partitioned into five folds, and about 310, 194, and 106 images in each fold were considered for train, validation, and testing (labeling is per pixel). The SEN12-FLOOD dataset was also divided into five folds in the same way, and about 357, 93, and 200 images were selected as non-flood samples for train, validation, and testing, respectively. Besides, about 175, 37, and 91 images were considered flood samples for train, validation, and testing, respectively (labeling is per image). We note that while image patch IDs are different for train, test, and validation, and images do not overlap completely, there might be some image patches in train, validation, and test from the same flood event. Randomly splitting data into folds without considering the flood event may increase the spatial autocorrelation between train and test data and result in model overfitting. One solution is to withhold one flood event as the test and train and validate on the others, switching to the next flood event in the next fold.

2.3.3 Training CSN

Two scenarios were tested for training the backbone CNN models in CSN. In scenario one, the CNN networks were Trained From Scratch (TFS), and all the parameters were trained from the beginning. In the second scenario, Transfer Learning (TL) was applied. TL means that the CNN model parameters are kept fixed based on their values obtained by training on popular RGB image databases like *ImageNet*, and only the newly added fully connected layers (the topmost layers) are trained. The TL method was tested for the 2019 Ontario and Quebec flood events and compared with the TFS method. For the other case studies, only

TFS was used because of the higher accuracy result it achieved for the 2019 flood event case study.

Popular backbone architectures, such as ResNet50 and VGG16 mainly accept three input feature bands because they have been trained on three-band RGB images. One problem when using these networks for remote sensing images is that the number of feature bands (N) is usually higher than three. A PCA transformation was applied to the train data, and the number of PCA components was set to three to make N (number of input feature bands) the same as the backbone networks.

A separate CSN was trained for each flood event, including 2019 Ontario and Quebec, 2021 BC, and 2021 Germany flood events because the datasets applied to Ontario-Quebec and BC flood events were from different satellites with different polarizations (VV and VH for Sentinel-1 and HH and HV for RCM) and with different spatial resolutions (10m for Sentinel-1 and 5m for RCM). While Sentinel-1 image was used for the Ontario -Quebec, and Germany flood events, separate models were trained to develop a specific CSN for each location and increase the flood mapping accuracy. For publicly available datasets, Sen1Floods11 and SEN12-Flood, a separate CSN was trained because the two datasets use different labeling techniques. While Sen1Floods11 labels each pixel as to flood/non-flood pixel, SEN12-Flood labels the whole image/scene to flood/non-flood. For training the CSN, the network parameters, including the number of epochs, batch size, learning rate, and optimizer function, need to be set. Another critical parameter that needs to be set, especially for CSN, is the feature space vector dimension, aka embedding dimension. Table 2-3 shows the assigned values to these parameters.

Table 2-3: CSN Parameters

Number of epochs	200
Batch size	50
Embedding dimension	256
Optimizer function	Adam
Initial learning rate	10^{-4}
Decay rate	0.8
Decay steps	10^{4}

The CSN was implemented using the Tensorflow library in Python. In Tensorflow, it is possible to set an early stopping condition to prevent overfitting. The condition is set so that the training will be stopped if the validation loss does not change during 50 epochs.

When training a deep learning model, it is often helpful to lower the learning rate as the training progresses. One of the strategies available in Tensorflow for reducing the learning rate is exponential decay in which the initial value of the learning rate is reduced exponentially. The parameters for exponential decay include the initial learning rate, decay rate and decay step. These parameters were presented in Table 2-3.

2.4 Accuracy Assessment

The accuracy assessment in this paper was achieved using three metrics (Chen et al., 2022; Jiang et al., 2021; Konapala et al., 2021), including precision, recall, and F1 score. These metrics have been described in Table 2-4. In this Table, TP (True Positive), refers to the number of correctly classified flood pixels, TN (True Negative) is the number of correctly detected background pixels, FP (False Positive), refers to the number of wrongly identified

flood pixels, and FN (False Negative), is defined as the number of pixels classified incorrectly as background.

Table 2-4: Accuracy metrics

Metric	Definition
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1 score	$\frac{2 \times precision \times \ recall}{precision + recall}$

2.5 Results

This section presents the experiments conducted for each case study, including the 2019 Ontario and Quebec, 2021 Abbotsford, and 2021 Leverkusen flood events. The flood mapping results and the accuracies after adding DEM have been included for each case study.

2.5.1 Flood Maps

This section presents the flood contingency maps and flood maps for the 2019 Ontario and Quebec, 2021 BC, and 2021 Germany flood events. For the first and third flood events, the original flood mask was in the vector format and was converted into the raster format for compatibility with the produced flood mask. Based on the obtained results, CSN overestimated flood in urban areas. This result was in agreement with previous study by Tanim et al. (2022) when using Sentinel-1 data for flood extent mapping. Because of the medium resolution (10m) of Sentinel-1 images, there is a high chance of mixed flood and non-flood pixels that might cause overestimation.

Figure 2-12 shows the contingency map created using the proposed CSN for the Ottawa and Gatineau area overlaid onto the Government of Quebec flood mask. As mentioned above, results show that the proposed CSN overestimated the flood area, especially in the residential parts.

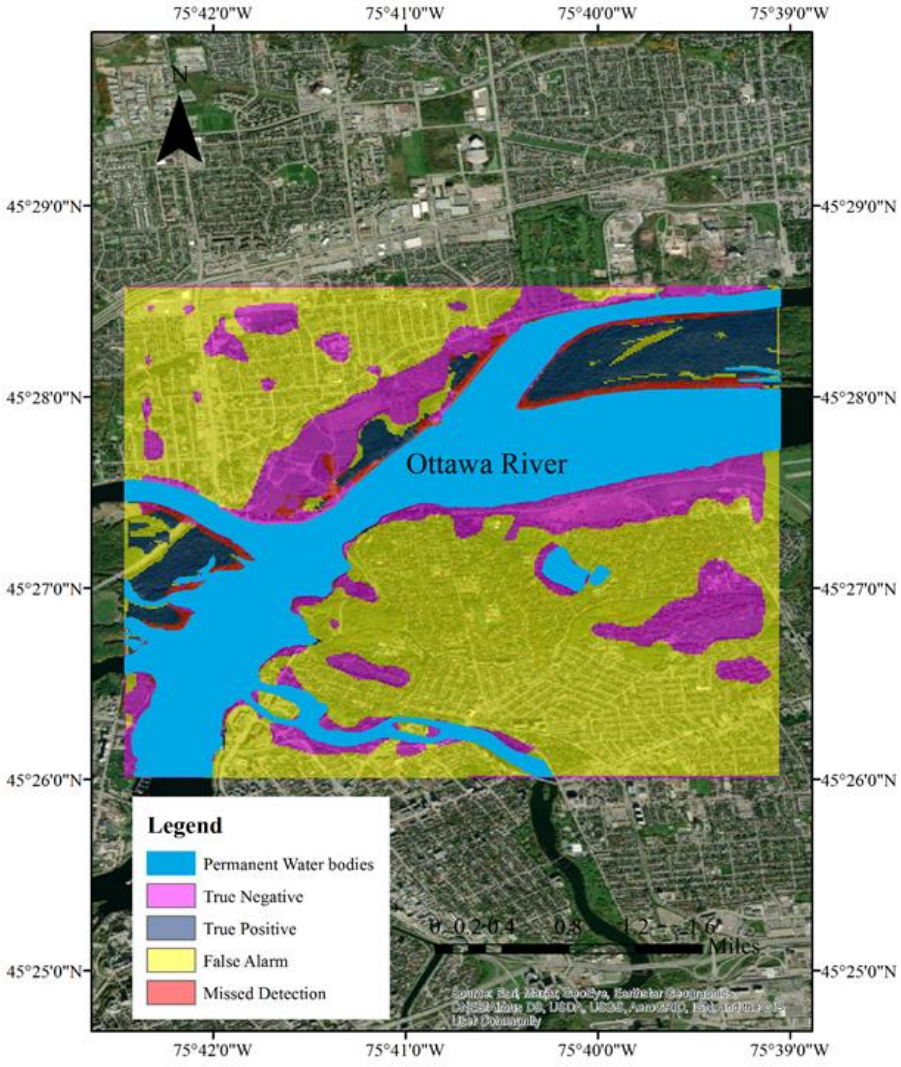


Figure 2-12: Contingency map created using the proposed CSN for the 2019 Ontario and Quebec flood event overlaid onto the ArcMap base map ; the magenta shows True Negative areas (correctly detected background pixels) detected by the proposed CSN, and the dark yellow shade areas are False Alarms (erroneously detected flood pixels)

Figure 2-13 shows the flood map created using the proposed CSN for the Abbotsford area. Based on the figure, it is evident that the proposed CSN algorithm detected some fragmented flood areas across the city in the roads and residential parts.

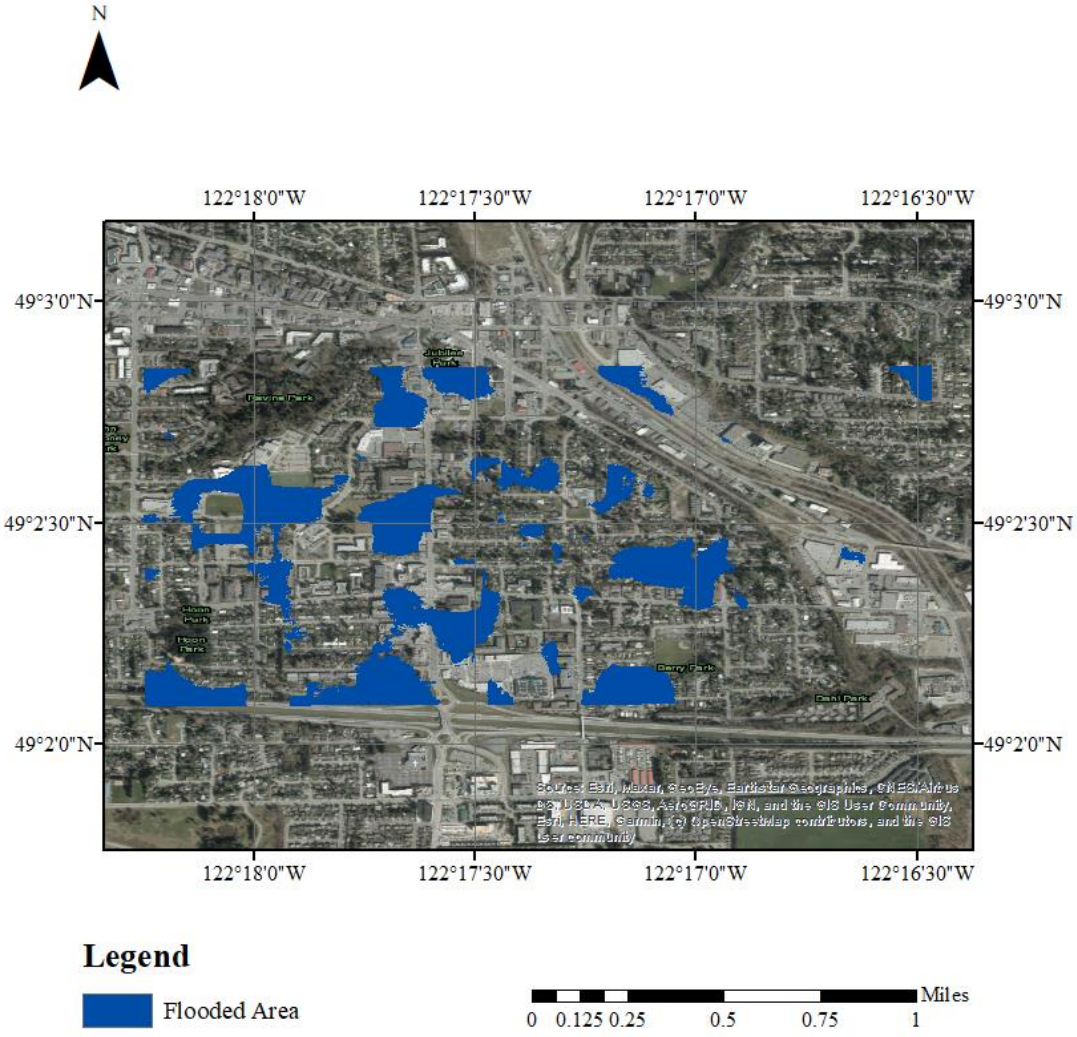


Figure 2-13: Flood map created using the proposed CSN for 2021 Abbotsford flood event

Figure 2-14 shows the flood map created using the proposed CSN for the Leverkusen city overlaid onto the rasterized ground truth data. The original ground truth data was accessed

via the Copernicus Emergency Management Website (European Union, 1995–2022, 2021). Similar to the results obtained for the Ottawa area, the flood map produced using the proposed CSN shows overestimation compared with the reference data, and some granule noisy flood patterns can be seen on the map. In terms of capturing the permanent water bodies, the Rhine river width mapped using the proposed method was thinner than the river width in the reference data.

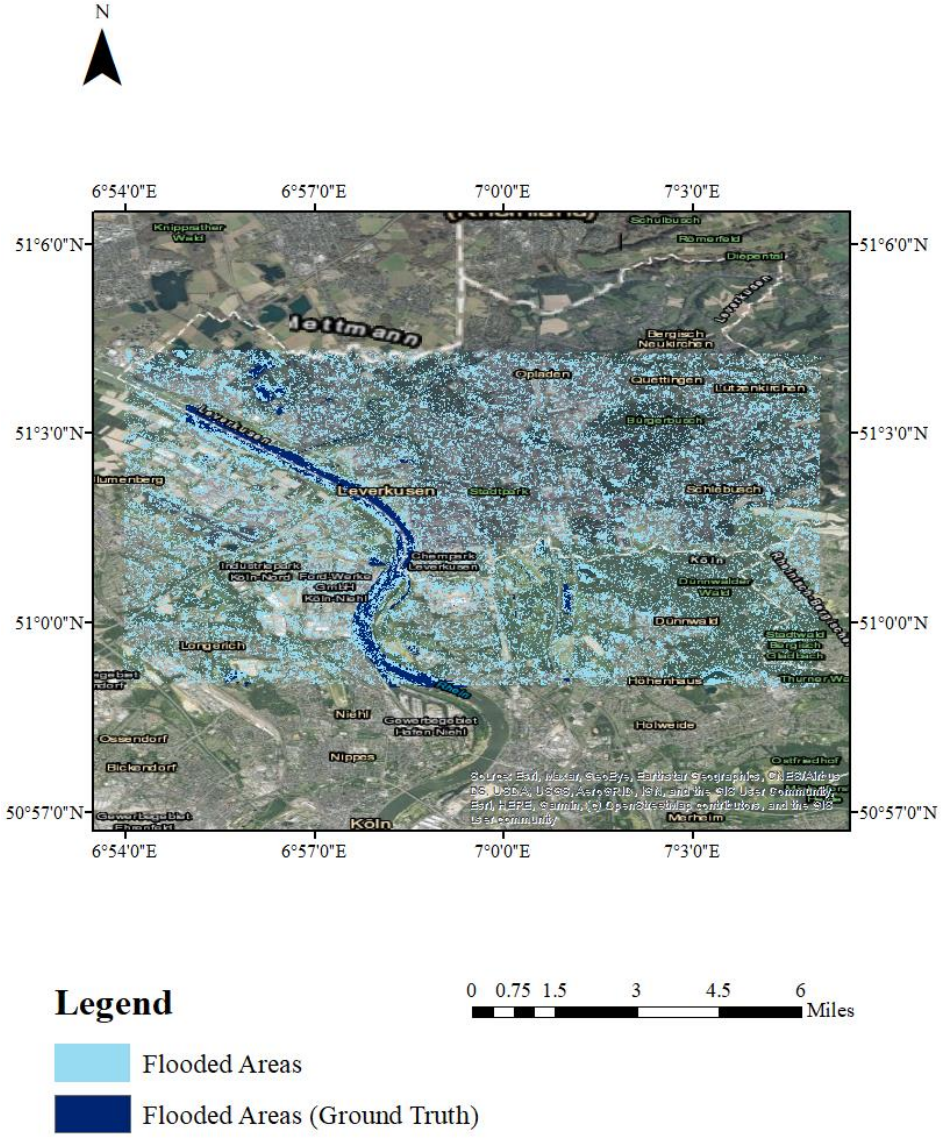


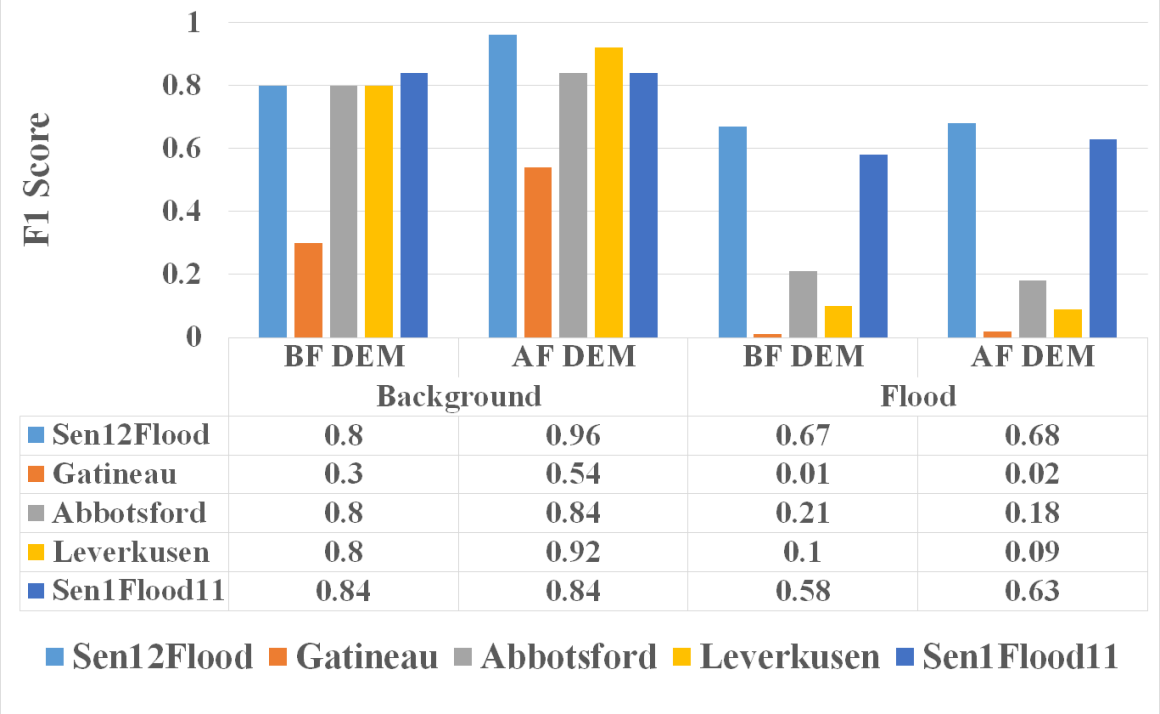
Figure 2-14: Flood map created using the proposed CSN for the 2021 Leverkusen flood event; the light blue areas are the water bodies (both permanent water bodies

and flood regions) detected by the proposed CSN, and the dark blue areas are the water bodies (both permanent water bodies and flood regions) in the ground truth data

The contingency map shown in Figure 2-12 resulted in a high false alarm rate because of the shadowing effect of both SAR sensors and high-rise buildings. Comparing flood maps in Figures 2-13 and 2-14, Figure 2-14 using Sentinel-1 achieved noisier results compared with 2-13, which used RCM with a spatial resolution of 5m, two times higher than Sentinel-1, with a spatial resolution of 10m.

2.5.2 Adding DEM Data for Flood Mapping

Figure 2-15 shows the F1 score values for three case studies and the SEN12-FLOOD and Sen1Floods11 datasets for the background and flood classes before (BF) and after (AF), adding DEM data to the SAR dataset. It is evident from the bar chart that the background F1 score improved in most cases. The background F1 score improved by 0.16, 0.24, 0.04, and 0.12 for SEN12-FLOOD, Gatineau, Abbotsford, and Leverkusen, respectively. For the flood class, while the F1 score dropped by 0.03 and 0.01 for Abbotsford and Leverkusen areas, it improved by 0.05, and 0.01 for the Sen1Floods11, SEN12-FLOOD, and Gatineau area, respectively.



	Background		Flood	
	BF DEM	AF DEM	BF DEM	AF DEM
Sen12Flood	0.8	0.96	0.67	0.68
Gatineau	0.3	0.54	0.01	0.02
Abbotsford	0.8	0.84	0.21	0.18
Leverkusen	0.8	0.92	0.1	0.09
Sen1Flood11	0.84	0.84	0.58	0.63

Figure 2-15: Effect of adding DEM data on the F1 score for the background and flood classes in Gatineau, Abbotsford, and Leverkusen case studies , along with Sen1Floods11 and SEN12-FLOOD datasets; BF is an acronym for Before Flood and AF is an acronym for After Flood

2.6 Discussion

This section examines the effect of changing the backbone architecture and the loss function in CSN. Besides, it compares the proposed CSN with other DL methods and explores the effect of adding DEM on flood mapping accuracy for three case studies. All the experiments were repeated on two publicly available datasets, Sen1Floods11 and SEN12-FLOOD, to test the generalization ability of the proposed CSN.

2.6.1 Comparison of Flood Maps In Terms of CSN Backbone Architecture

One of the uncertainty sources in the proposed CSN is the type of backbone architecture applied for the feature extraction. Figure 2-16 compares the accuracy indices for the three

case studies in terms of two kinds of networks used for feature extraction, including ResNet50 and VGG16. It can be inferred from the figure that the flood precision rate after changing the feature extractor from ResNet50 to VGG16 improved by 0.02, 0.02, and 0.08 for the Gatineau area, Abbotsford, and Leverkusen, respectively. In terms of recall rate, except for the Gatineau area, for the other two case studies, the index dropped by 0.19 and 0.64 after changing the backbone architecture to VGG16. Finally, the F1 score improved by 0.03 and 0.1 for the Gatineau area and Leverkusen areas but dropped by 0.01 for Abbotsford after using VGG16 as the feature extractor. Based on the obtained accuracy indices for the three case studies, it can be induced that VGG16 generally achieved higher flood accuracy than ResNet50. It is worth mentioning that there is no best feature extractor and the selection of the most suitable feature extractor for the Siamese Network depends on different factors such as the type of input data in terms of being optical, SAR, or topography data. Additionally, the selection can be affected by the case study, and a feature extractor might work for one case study but might not be suitable for the other case studies. Further, for the Abbotsford case study, the accuracy indices are more balanced between the background and flood classes than Gatineau and Leverkusen (Figure 2-16). The reason for more similar results between Gatineau and Leverkusen areas is the type of input data applied for these case studies. While for the Abbotsford area, the RCM dual-polarized intensity bands in HH and HV channels were tested, for Gatineau and Leverkusen, Sentinel-1 intensity and coherency data in VV and VH channels were applied for flood mapping.

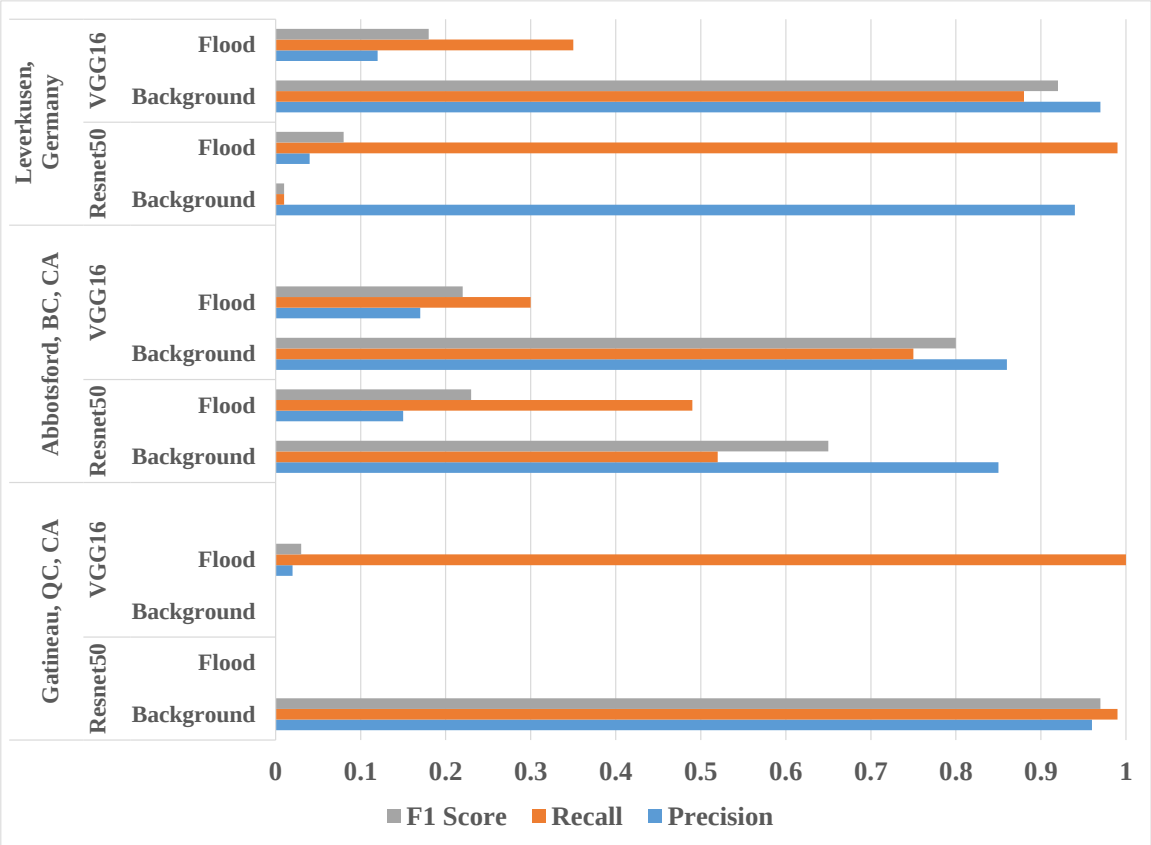
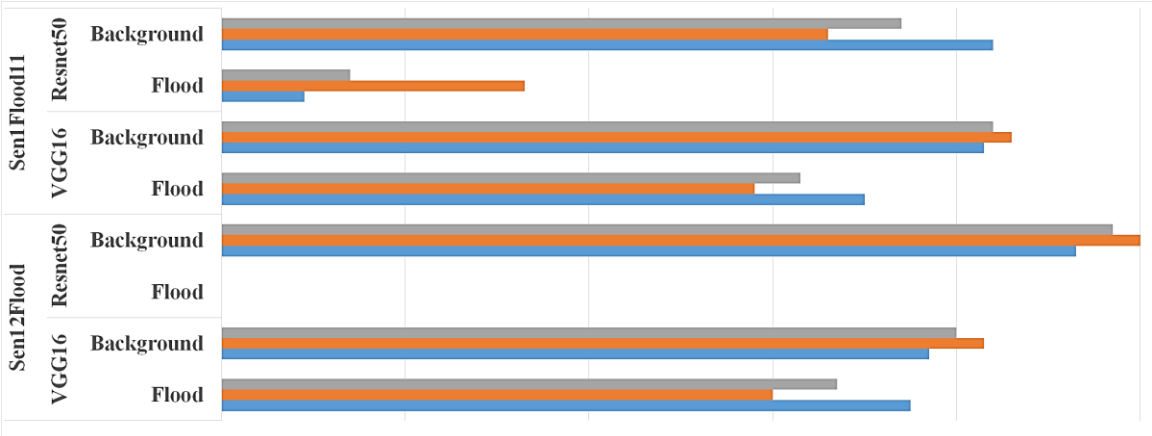


Figure 2-16: Comparison between the proposed CSN performance in terms of feature extractor for Gatineau, Abbotsford, and Leverkusen case studies

Figure 2-17 shows the flood and background accuracies for SEN12-FLOOD and Sen1Floods11 datasets using VGG16 and ResNet50 as the backbone architecture. For the SEN12-FLOOD dataset, when using ResNet50, the flood was not detected in the image scene. However, VGG16 achieved about 0.75 precision and 0.67 F1 score for the flood class. The same result was achieved for the Sen1Floods11 dataset, and VGG16 showed higher flood precision and F1 score than ResNet50. After using VGG16, flood precision increased from 0.09 to 0.2, and the F1 score increased from 0.14 to 0.25. Higher flood accuracies were obtained for SEN12-FLOOD than Sen1Floods11. The reason for this is that labeling the SEN12-FLOOD dataset is more straightforward than the Sen1Floods11 dataset. SEN12-FLOOD marks the whole image scene as flood/non-flood, but

Sen1Floods11 is a per-pixel labeling task that is more complicated because of mixed pixels and speckle noise in SAR images.



	Sen12Flood				Sen1Flood11			
	VGG16		Resnet50		VGG16		Resnet50	
	Flood	Background	Flood	Background	Flood	Background	Flood	Background
F1 Score	0.67	0.8	0	0.97	0.63	0.84	0.14	0.74
Recall	0.6	0.83	0	1	0.58	0.86	0.33	0.66
Precision	0.75	0.77	0	0.93	0.7	0.83	0.09	0.84

F1 Score Recall Precision

Figure 2-17: Comparison between the proposed CSN performance in terms of feature extractor for SEN12-FLOOD and Sen1Floods11 datasets

2.6.2 Effect of Using Different Loss Functions

Another uncertainty source in CSN is the loss function applied for training the feature extractor Network. Three loss functions, including Contrastive Loss, Weighted Double Margin Contrastive Loss (WDMCL), and Triplet Loss, were used in this study to assess the CSN sensitivity to the loss function applied.

Figure 2-18 shows the Gatineau case study's bar chart for flood and background accuracy indices. The flood recall rate improved after increasing flood sample weights. The background precision did not drop significantly after increasing flood sample weights and reducing background sample weights simultaneously, but its recall rate dropped. In other words, after decreasing background samples' contribution to the training process, the recall rate decreased. According to the bar chart, changing from Triplet Loss to WDMCL, the recall rate increased by 0.43 and by 0.47 after changing from Contrastive Loss to WDMCL. Although using WDMCL effectively increased the flood recall rate, the flood precision did not change after changing the loss function formulation. The flood precision index might be more affected by the input data type (optic, SAR or topography data) than the background and flood samples distribution.

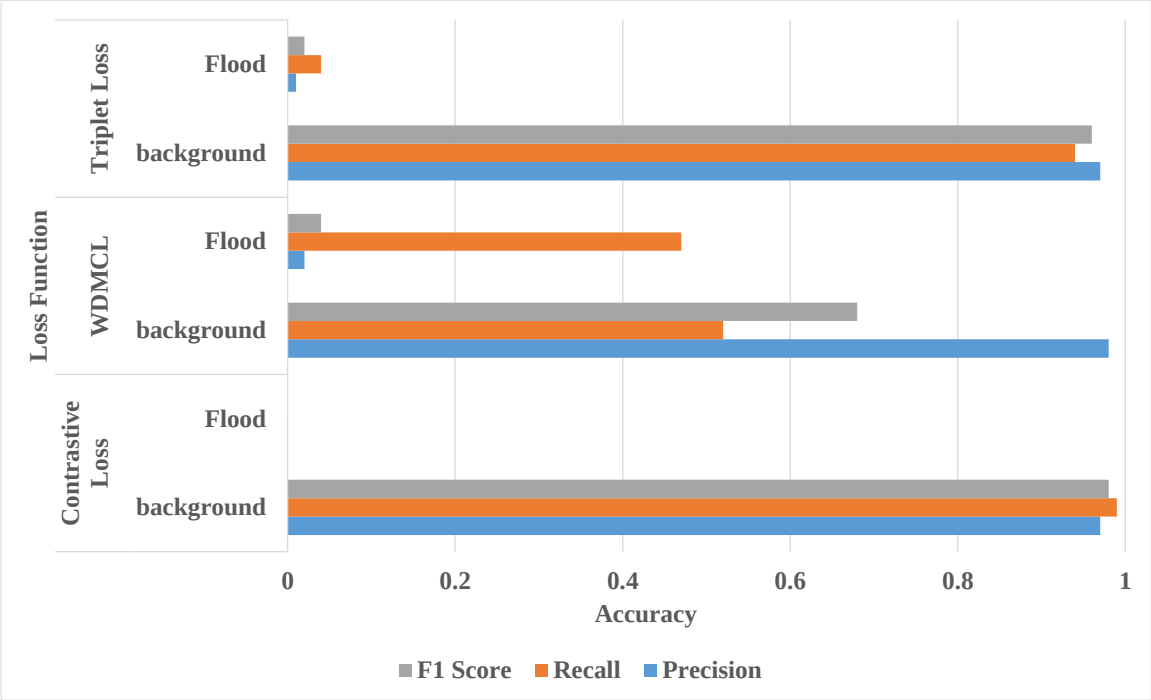


Figure 2-18: Background and flood accuracy indices for Contrastive Loss, WDMCL, and Triplet loss functions in Gatineau, QC

Figure 2-19 shows the same bar chart as figure 2-18 for Abbotsford city. Based on the figure, all three loss functions resulted in similar flood precision accuracy. Besides, the flood recall rate improved after emphasizing flood class in the WDMCL formulation. In terms of flood precision index, all three loss functions had comparable performance, achieving precision values lower than or equal to 15%, which means that in almost only 15% of the cases, the detected flood pixel had consistency with the reference data. This low precision index might be related to the limitations of using only the SAR intensity data for flood mapping. Although the complementary role of SAR coherency and polarimetry data can be inferred from the literature, the only high-resolution RCM data available for the area was the intensity, adding limitations to examining the effect of other SAR products, such as polarimetry and coherency, for flood extent mapping.

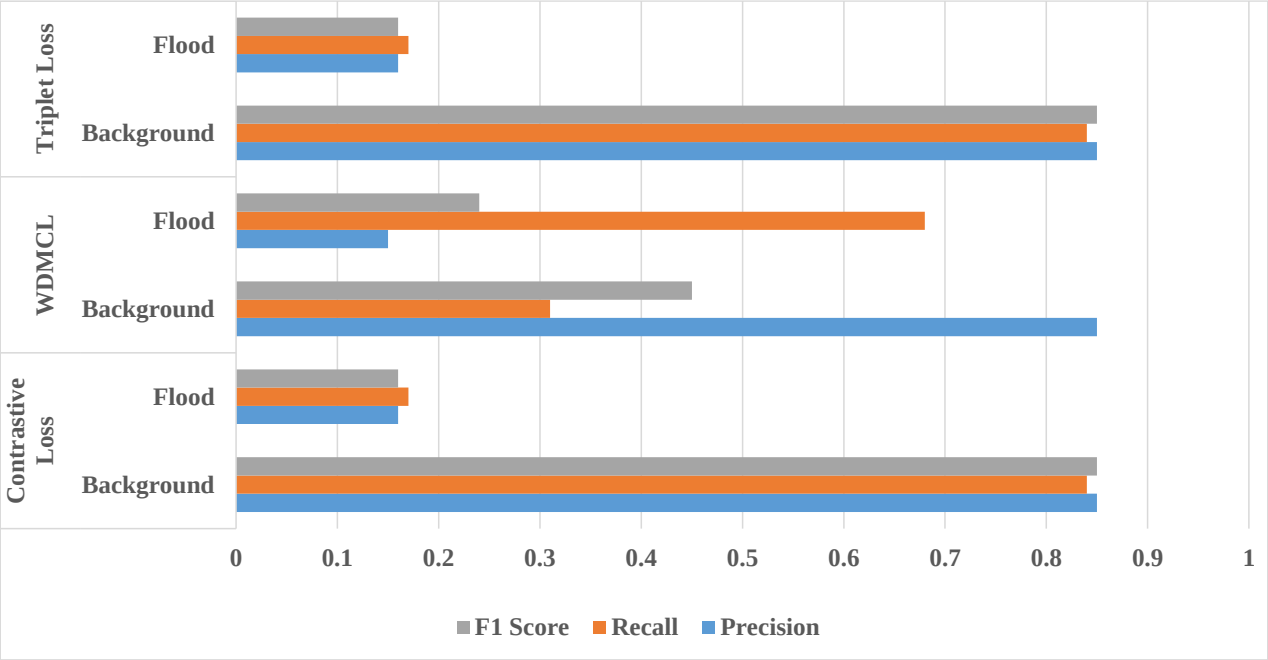


Figure 2-19: Background and Flood accuracy indices for Contrastive Loss, WDMCL, and Triplet loss functions in Abbotsford, BC

Figure 2-19 suggests that the F1 score achieved for the WDMCL is about 1%, and 6% higher than the corresponding values for the Contrastive and Triplet Loss functions, respectively. While Contrastive and Triplet Loss functions achieved an F1 score of about 23% and 16%, the WDMCL achieved a 24% because of its significantly higher recall value

of 68%. Another important point is that although adding more weight to the flood samples in the WDMCL improved the recall index, this strategy did not help increase the precision index for flooded areas in the SAR image. The flood recall rate improved by 0.51 compared to the Triplet and by 0.19 compared to the Contrastive Loss Functions. At the same time, after decreasing the emphasis on background class in WDMCL, the recall rate dropped by 0.21 and 0.32 compared to Contrastive and Triplet Loss functions, respectively.

Figure 2-20 shows the accuracy indices for the background and flood classes for the Leverkusen case study in terms of the Loss Function applied in the CSN. A similar trend to the Gatineau area and Abbotsford cases regarding flood recall value increase after changing the loss function from Triplet Loss to WDMCL can be seen. Precision values were not significantly affected by the loss function, and the index remained at the exact value of 0.93 and 0.15 for the background and flood classes, respectively. It can be inferred from the bar chart that although Contrastive Loss achieved the highest flood recall rate of 0.99 among the loss functions, it could not achieve a high flood precision, and its value remained as low as in other cases. On top of that, the Contrastive Loss function resulted in a background recall index of 0.52, which was lower than its counterparts for the WDMCL and Triplet Loss functions which were 0.68 and 0.88, respectively.

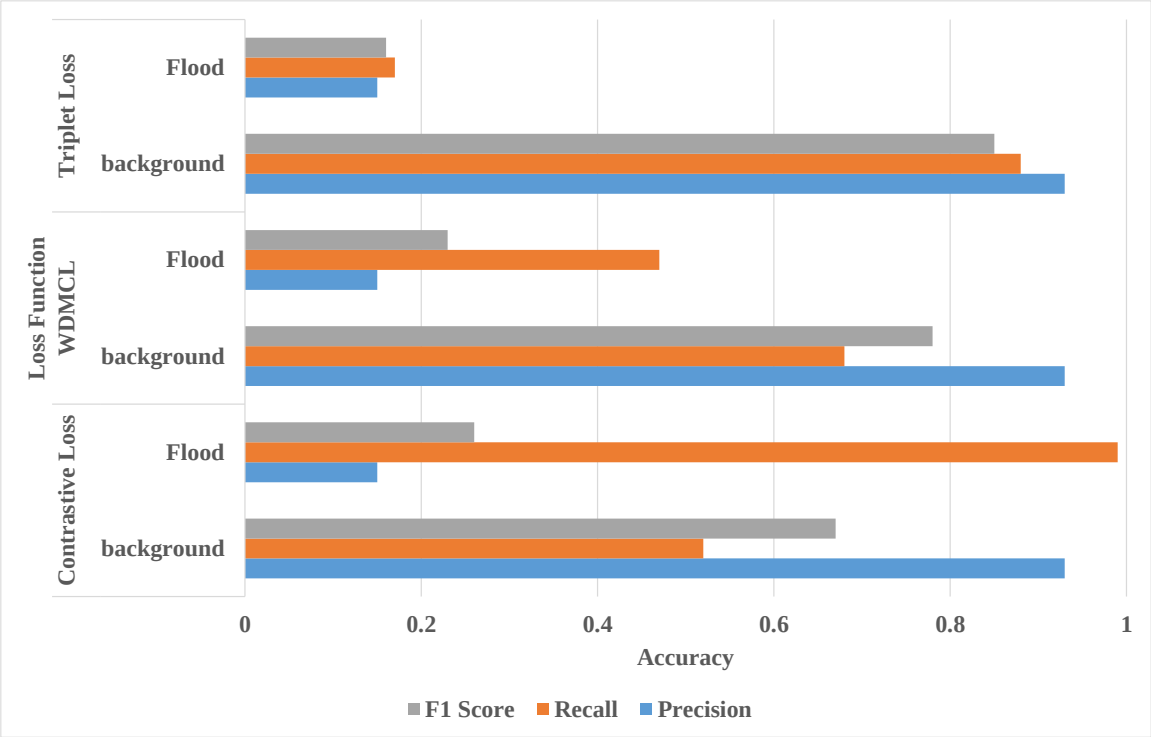


Figure 2-20: Background and flood accuracy indices for Contrastive Loss, WDMCL, and Triplet Loss functions in Leverkusen, Germany

Figure 2-21 shows the accuracy indices after applying different loss functions for SEN12-FLOOD and Sen1Floods11 datasets. For SEN12-FLOOD data, the highest flood accuracy indices were for the Contrastive Loss function (between 0.6-0.75). For background, the highest accuracies were for the Triplet Loss function. For the Sen1Floods11 dataset, the highest flood recall and F1 score indices were for the Contrastive Loss function (0.58 recall and 0.63 F1 score). In terms of flood precision for Sen1Floods11, WDMCL, considering higher weights during training for flood pixels (minority class), improved the precision index by 0.19 and 0.16 compared with the Triplet Loss and Contrastive Loss functions. On the other hand, using WDMCL did not improve flood precision for SEN12-FLOOD data. This contradiction stems from the different labeling formats between the two datasets. For the background class, Triplet Loss resulted in a higher F1 score and recall rate than WDMCL and Contrastive Loss functions (0.97 for F1 Score and 1 for recall).

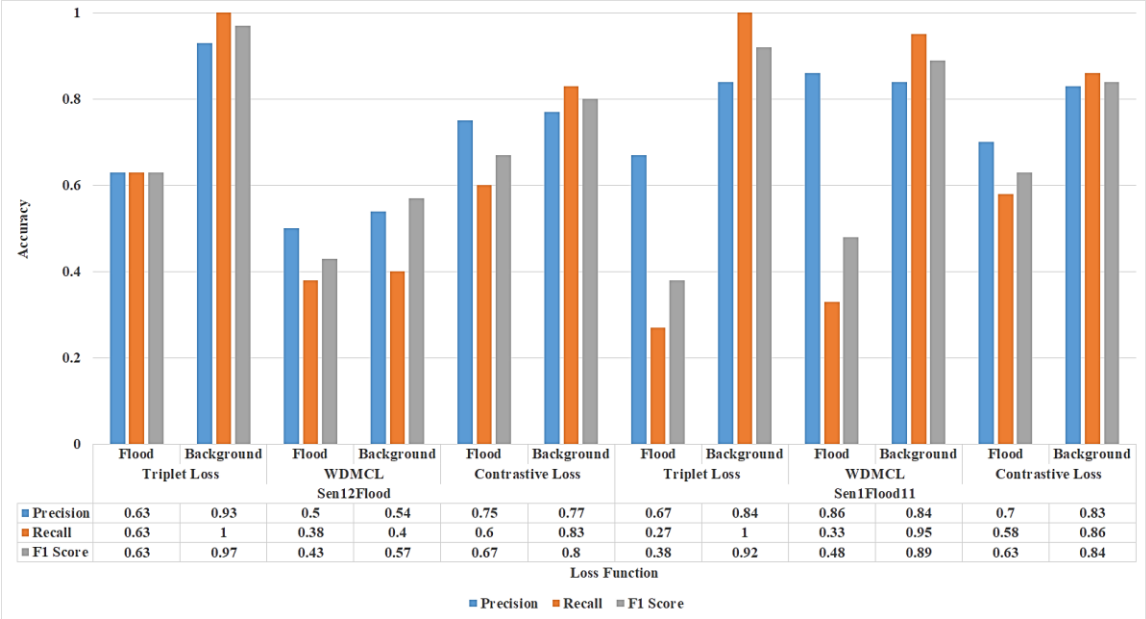


Figure 2-21: Background and flood accuracy indices for Contrastive Loss, WDMCL, and Triplet Loss functions for SEN12-FLOOD and Sen1Floods11 datasets

2.6.3 Comparison with Other Deep Learning Techniques

For all flood events in the comparison section, the backbone architecture was VGG16 and the Loss function was Contrastive Loss. The input data for all the methods in this section were Sentinel-1 intensity and coherency data and a 30m resolution SRTM DEM data to help the deep learning algorithms to differentiate between the low and high-lying lands.

Overall, after comparing the CSN with other DL algorithms, it was generally observed that, for the background class, higher recall and F1 score were achieved because of the higher number of training samples. The unbalanced training sample effects on recall and F1 score measures can be partly reduced by using a weighted loss function. It was also observed that the precision index for background class was higher than the flood in all cases whether the loss function be a weighted or non-weighted.

Figure 2-22 compares the proposed CSN with four other state-of-the-art deep learning algorithms, including Unet, Unet++, DeepLabV3+, and Siamese-Unet for the 2019 Ottawa River flood event. Based on the figure, Siamese-Unet and the proposed Siamese Network

performed better than other DL algorithms. Although Siamese-Unet showed higher flood recall (0.99), and F1 score (0.58), its low background recall rate, about 0.01, indicates flood overestimation. The proposed Siamese Network shows a higher background recall rate (about 0.42) than Siamese-Unet, and its accuracy indices are more balanced between flood and background classes.

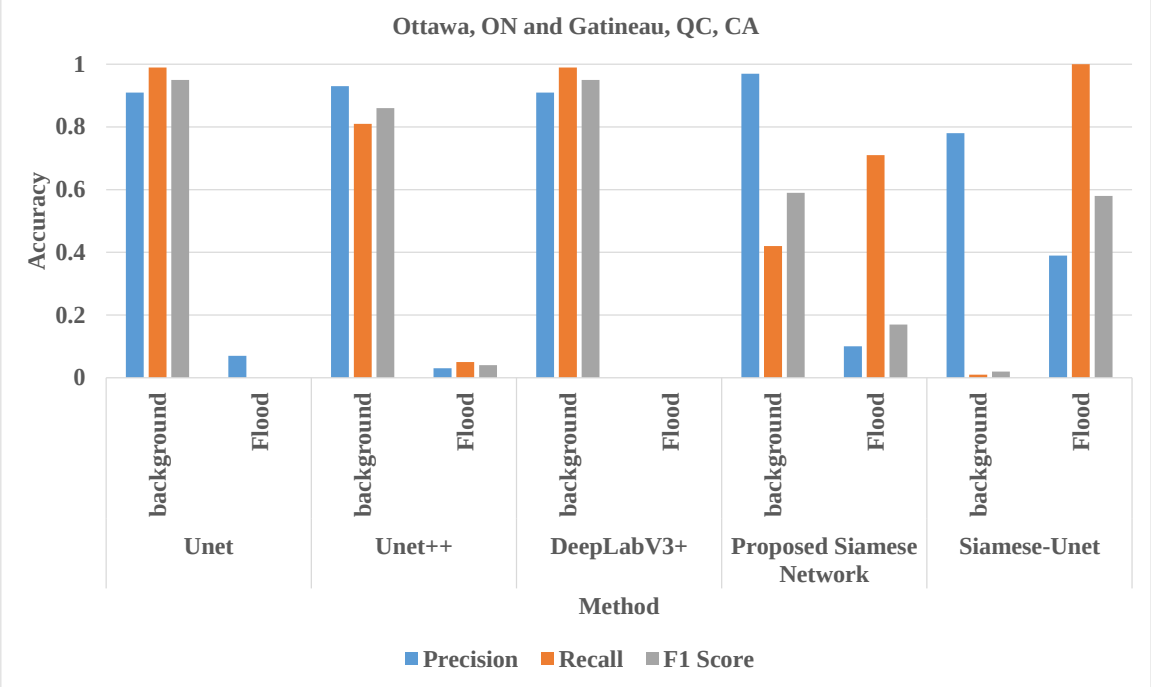


Figure 2-22: Comparison of the proposed CSN with Unet, Unet++, DeepLabV3+, and Siamese-Unet for Ottawa, ON and Gatineau, QC, CA

Figure 2-23 compares the proposed CSN with four other state-of-the-art deep learning algorithms, including Unet, Unet++, DeepLabV3+, and Siamese-Unet for the urban area in Abbotsford, BC. The input data for all the methods were dual-polarized HH and HV RCM intensity and a 25m resolution DEM data. Based on the bar plot, it is apparent that the accuracy indices achieved for the background areas were higher than flooded regions because of the higher number of train data available for the background class. Another reason for this might be the limitation of the dual-polarized RCM data used for Abbotsford city. The RCM images applied were in HH and HV polarization modes. Based on the

literature, the most suitable polarization in C-band SAR data for urban flood mapping is the VV mode (Pramanick et al., 2022). It is also notable that the within-class accuracy distribution in Unet, Siamese-Unet, and our proposed CSN was more balanced than the Unet++ and DeeplabV3+. In other words, the values achieved for the precision, recall, and F1 score in each class were closer in the Unet, Siamese-Unet, and the proposed method than Unet++ and DeepLabV3+. Although DeepLabV3+ achieved 0.85 precision for background class and 0.93 recall rate for flood regions, it is still unreliable because of its low recall value for background class and low precision value for flood class. Siamese-Unet had a comparable performance with the proposed CSN (because they both use change detection for flood map generation), but it achieved a lower recall rate on the background, about 5%, than the proposed CSN. Besides, it achieved a higher flood recall rate, about 4%, than the proposed CSN. The low accuracy indices achieved for the flood areas in all the applied deep learning algorithms confirm the SAR data limitation for urban flood mapping applications. This result is in agreement with previous studies that used SAR data for urban flood mapping (Li et al., 2019b; Lin et al., 2019; Hertel et al., 2023).

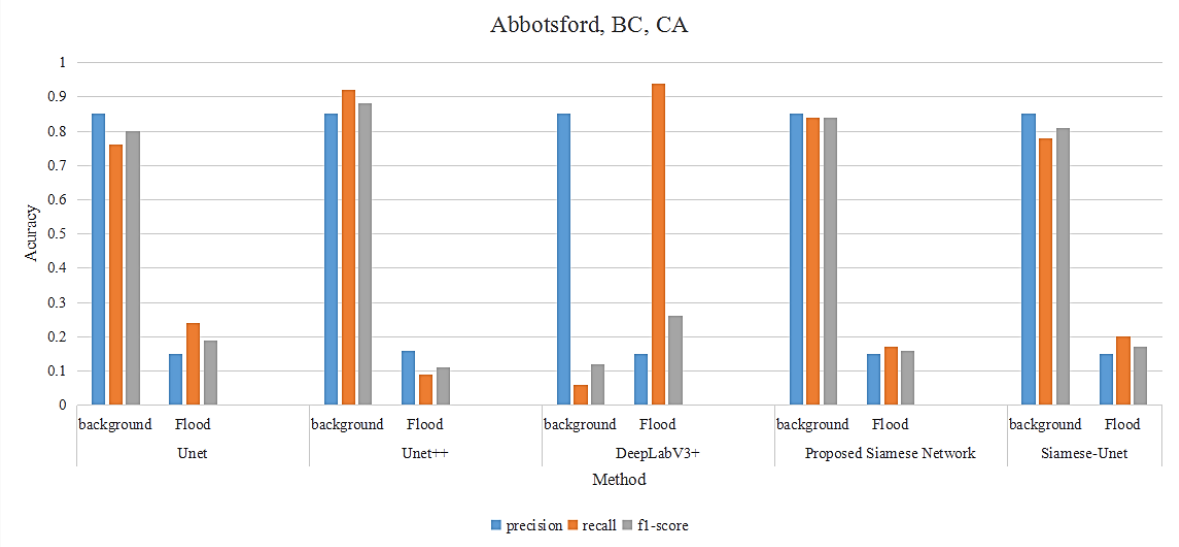


Figure 2-23: Comparison of the proposed CSN with Unet, Unet++, DeepLabV3+, and Siamese-Unet for Abbotsford, BC (Urban), CA

The method was also tested in an agricultural area near Abbotsford to further investigate the reliability of the proposed CSN. Figure 2-24 shows the flood mapping accuracy results for this area. While Unet++ and DeepLabV3+ resulted in low flood accuracies, the proposed CSN achieved precision and F1 score of 0.71 and 0.6, which were the highest among all the methods, and the method achieved more balanced accuracies between background and flood classes.

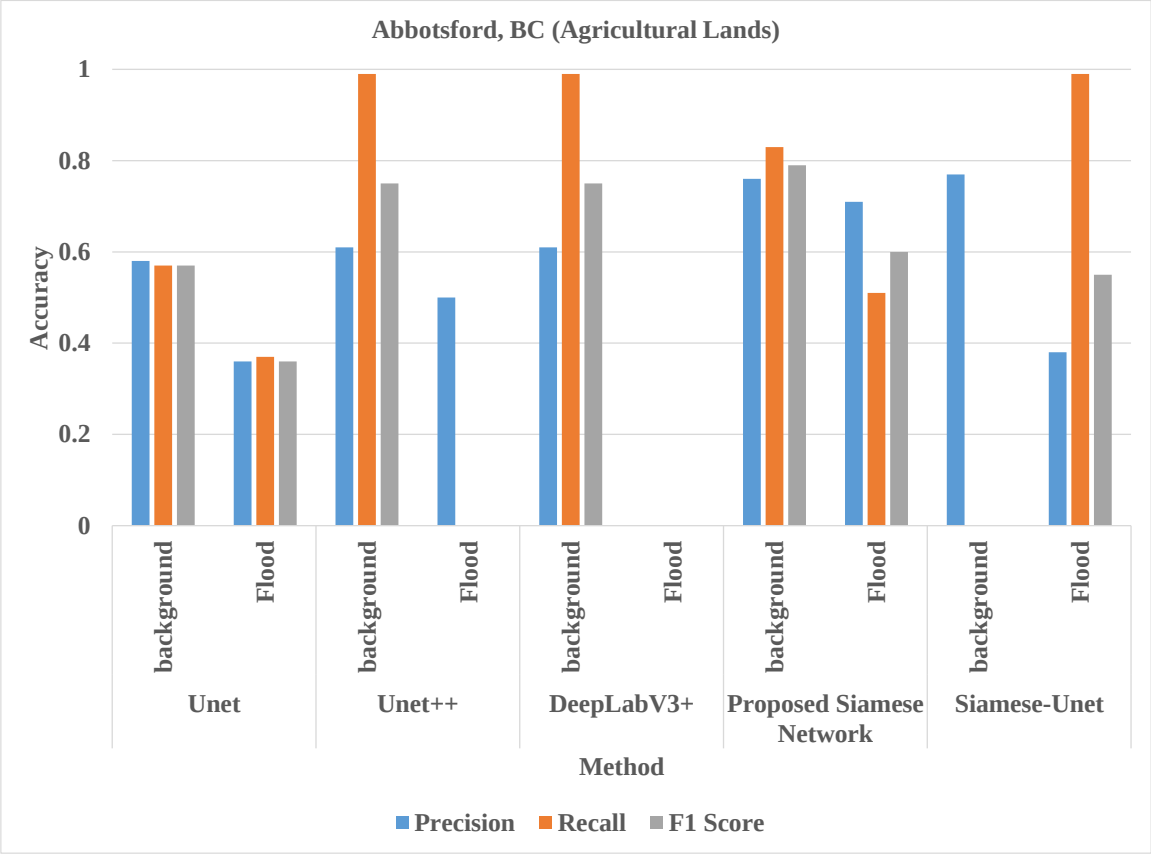


Figure 2-24: Comparison of the proposed CSN with Unet, Unet++, DeepLabV3+, and Siamese-Unet for an agricultural area near Abbotsford, BC, CA

Figure 2-25 shows the contingency map of the proposed CSN for this suburban area. The low recall rate value for the proposed CSN shown in the bar chart can be justified by so many missed flood areas (red regions) on the map, and the reported high precision index of 0.71 is because the number of false alarms is relatively low, and most predicted flood areas are consistent with the ground truth data.

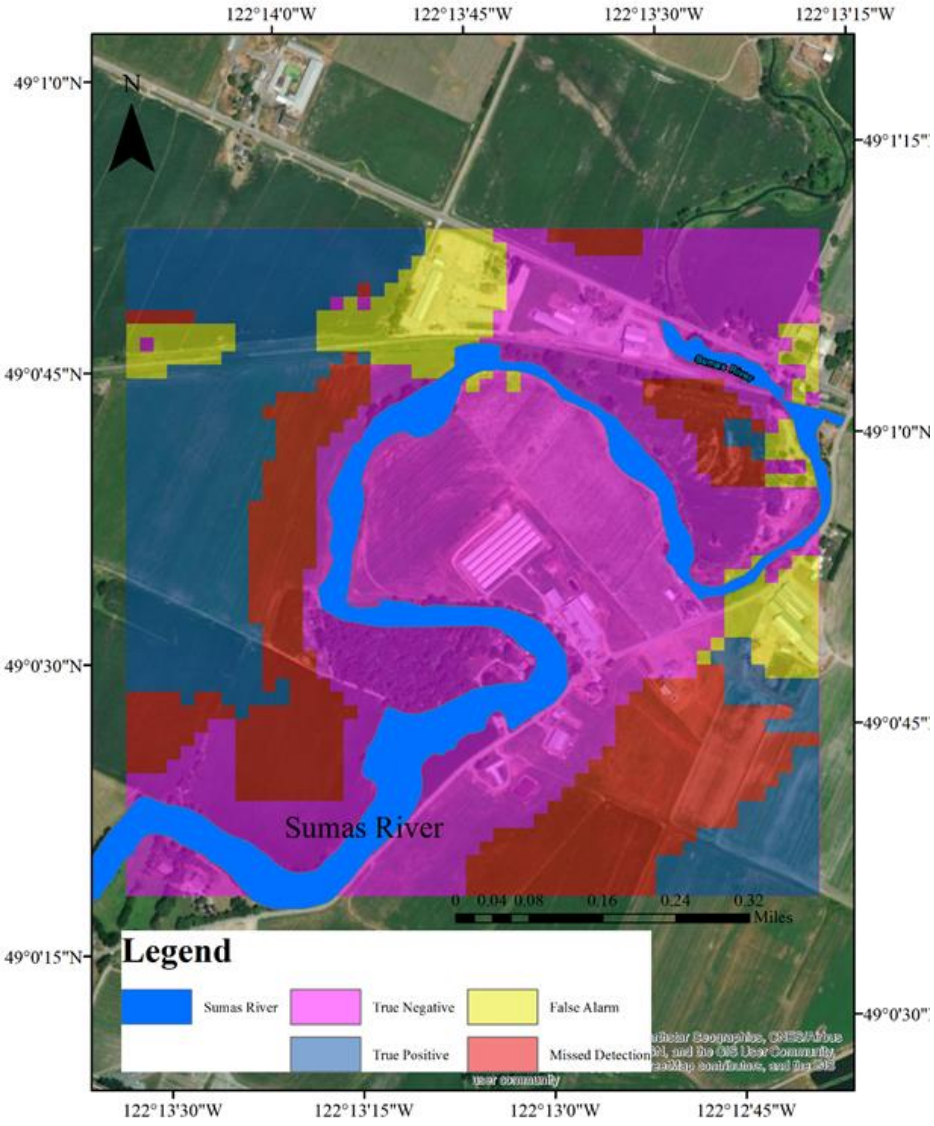


Figure 2-25: Contingency map for the agricultural area in Abbotsford, BC

Figure 2-26 compares the proposed CSN and the previously mentioned deep learning algorithms, including Unet, Unet++, DeepLabV3+, and Siamese-Unet for Leverkusen, Germany. It can be seen that Unet++ achieved acceptable precision, recall, and F1 score rates of 0.79, 0.95, and 0.86 for flood areas, but the method resulted in high false alarms and mixed many background pixels with the flood. Further, Unet acquired higher

background precision and recall indices than Unet++ and DeepLabV3+. DeepLabV3+ achieved a high recall rate of 1 for the background class, but it could not detect any flood pixel in the scene. The proposed CSN and Siamese-Unet achieved the highest background precision of 0.93 among all the deep learning methods and had comparable performance because they both use Siamese architecture for flood map generation. The proposed method could not achieve high precision, recall, and F1 score for the flood areas.

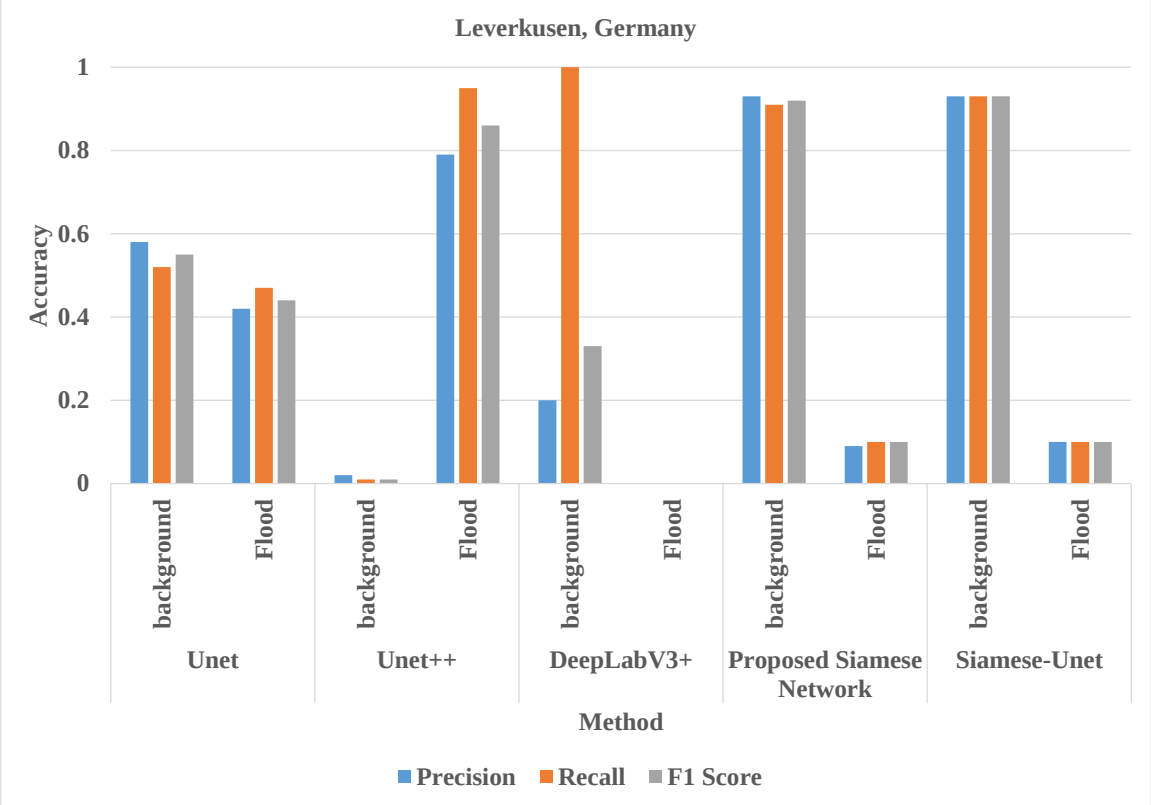
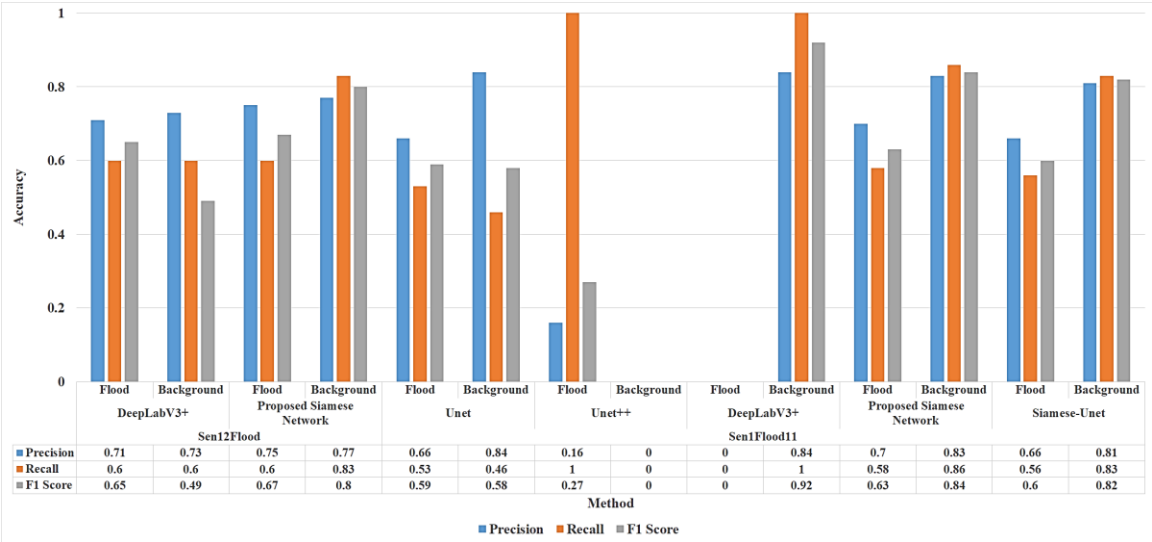


Figure 2-26: Comparison of the proposed CSN with Unet, Unet++, DeepLabV3+, and Siamese-Unet for Leverkusen, Germany

Figure 2-27 compares the proposed CSN with other DL methods in terms of the flood and background accuracy indices for Sen1Floods11 and SEN12-FLOOD datasets. Please note that Unet, Unet++, and Siamese-Unet are per-pixel predictors and can not be applied to the

SEN12-FLOOD dataset. Hence, only DeepLabV3+ and the proposed CSN were tested on this dataset. The proposed CSN is adaptable to both per-pixel and per-scene labeling formats by modifying the last layer (prediction layer). For the Sen1Floods11 dataset, the highest flood precision index, 0.7, was for the proposed CSN, about 0.04 higher than Unet and Siamse-Unet. Regarding the recall index, the proposed CSN also achieved a higher value of 0.58 compared with 0.53 and 0.56 for Unet and Siamese-Unet, respectively. The other DL methods, Unet++ and DeepLabV3+ showed significant flood overestimation and underestimation. For the SEN12-FLOOD dataset, the flood accuracy indices for the proposed CSN were higher than Sen1Floods11 dataset because of the difference between labeling formats. Besides, the proposed CSN resulted in higher flood precision, and F1 score than DeepLabV3+. While flood precision for DeepLabV3+ was 0.71, this index for the proposed CSN was 0.75.



	DeepLabV3+ Flood	DeepLabV3+ Background	Proposed Siamese Network Flood	Proposed Siamese Network Background	Unet Flood	Unet Background	Unet++ Flood	Unet++ Background	DeepLabV3+ Flood	DeepLabV3+ Background	Proposed Siamese Network Flood	Proposed Siamese Network Background	Siamese-Unet Flood	Siamese-Unet Background
Precision	0.71	0.73	0.75	0.77	0.66	0.84	0.16	0	0	0.84	0.7	0.83	0.66	0.81
Recall	0.6	0.6	0.6	0.83	0.53	0.46	1	0	0	1	0.58	0.86	0.56	0.83
F1 Score	0.65	0.49	0.67	0.8	0.59	0.58	0.27	0	0	0.92	0.63	0.84	0.6	0.82

Figure 2-27: Comparison of the proposed CSN with Unet, Unet++, DeepLabV3+, and Siamese-Unet for SEN12-FLOOD and Sen1Floods11 datasets

2.6.4 Effect of Adding DEM Data on Flood Mapping Accuracy

As mentioned in section 2.5, although the flood detection accuracy improved by 0.01 and 0.05 for the Gatineau case study, SEN12-FLOOD, and Sen1Floods11 after adding DEM to SAR images, it reduced by 0.03 for Abbotsford area, respectively. This result contrasts with Konapala et al. 2021 which combined SRTM DEM and Sentinel-1 and reported an improvement of about 0.11 for the F1 score. A reason for this contradiction can be related to sensor differences between the two studies. While Konapala et al. 2021 used Sentinel-1 for flood mapping, RCM data was used for the Abbotsford area. Sentinel-1 image applied in this work had VV and VH polarizations which were different from RCM images with HH and HV polarizations. Another critical point inferred from the results is that the background accuracy increase was more significant after adding DEM in Leverkusen than in Abbotsford and Gatineau because the study area in Leverkusen was larger, and the elevation variation was more significant than in the others. In other words, the elevation variation for the Leverkusen region, a catchment near the Rhine River, was more significant than the Abbotsford case study, which was a relatively small area with lower elevation variation than the Leverkusen region. The exact wording holds for the Gatineau area. The accuracies reported in section 2.5 for Gatineau were for a small area at the Gatineau Hull and Ottawa rivers with a smooth elevation variation.

2.7 Conclusions

In this study, urban flood mapping using SAR data through a CSN was explored and validated against ground truth data for three flood events in Ottawa, ON and Gatineau, QC, Abbotsford, BC, and Leverkusen, Germany. Also, CSN was compared with three state-of-the-art DL algorithms, including Unet, Unet++, and DeepLabV3+. For the sake of comparison with a network with similar architecture and taking advantage of Unet encoder-decoder architecture, Siamese-Unet was also compared with the proposed CSN. The results indicated that Sentinel-1 data can detect flooded areas with an average accuracy above 0.5. However, the complexities of urban infrastructures and the shadowing effect of high-rise buildings and the SAR sensor caused the flood mapping accuracies to be lower than in non-

urban regions. The reliability of the proposed CSN was also tested on two publicly available datasets SEN12-FLOOD and Sen1Floods11. A promising flood F1 score of 0.67 was achieved, which was higher than its counterpart for Sen1Floods11, with an F1 score of 0.63.

The variability of flood mapping accuracies among urban case studies can be related to 1- SAR system characterization, including band type, spatial resolution, polarization, and revisit time; 2- Flood extent and duration: longer lasting floods might be easier to capture using satellite images; 3- Buildings height and density: flood mapping in dense urban areas with high-rise buildings are more problematic; 4- Topography: metropolitan areas with varied topography, such as hills and valleys, can create additional challenges to SAR flood mapping.

The effect of loss function on the CSN performance was also examined. It was inferred from the results that CSN is sensitive to the loss function selection, and the use of WDMCL can improve flood precision and recall at the cost of deteriorating the background indices. Another important finding was that the loss function selection had less contribution to the precision index. Precision was more affected by the input data type and the normalization method applied to the input data.

The effect of adding DEM data on flood mapping accuracy was also explored. Although the highest F1 score improvement for the flood class was around 0.05, the results strongly confirmed the DEM data efficiency for improving the F1 score for the background class.

References

Aparna, A., & Sudha, N. (2022, May). SAR-FloodNet: A Patch-based Convolutional Neural Network for Flood Detection on SAR Images. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 195-200). IEEE.

Bonafilia, D., Tellman, B., Anderson, T., & Issenberg, E. (2020). Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 210-211).

Bouvet, A., Mermoz, S., Ballère, M., Koleck, T., & Le Toan, T. (2018). Use of the SAR shadowing effect for deforestation detection with Sentinel-1 time series. Remote Sensing, 10(8), 1250.

Chen, C., & Fan, L. (2021, August). Scene segmentation of remotely sensed images with data augmentation using U-net++. In *2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)* (pp. 201-205). IEEE.

Chen, J., Yuan, Z., Peng, J., Chen, L., Huang, H., Zhu, J., Liu, Y. and Li, H., 2020. DASNet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, pp.1194-1206.

Chen, T., Lu, Z., Yang, Y., Zhang, Y., Du, B., & Plaza, A. (2022). A Siamese Network Based U-Net for Change Detection in High Resolution Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 2357-2369.

Deng, Z., Dong, Z., Yang, F., & Xia, L. (2020, November). Data augmentation method of remote sensing image based on transfer learning and VGG model. In *AOPC 2020: Display Technology; Photonic MEMS, THz MEMS, and Metamaterials; and AI in Optics and Photonics* (Vol. 11565, pp. 172-179). SPIE.

European Union, 1995–2022. (2021). Copernicus Emergency Management Service - Mapping. Copernicus EMS - Mapping. https://emergency.copernicus.eu/mapping/

Ghorbanzadeh, O., Rostamzadeh, H., Blaschke, T., Gholaminia, K., & Aryal, J. (2018). A new GIS-based data mining technique using an adaptive neuro-fuzzy inference system (ANFIS) and k-fold cross-validation approach for land subsidence susceptibility mapping. *Natural Hazards*, *94*(2), 497-517.

Hänsch, R., Arndt, J., Lunga, D., Gibb, M., Pedelose, T., Boedihardjo, A., ... & Bacastow, T. M. (2022). Spacenet 8-the detection of flooded roads and buildings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1472-1480).

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hertel, V., Chow, C., Wani, O., Wieland, M., & Martinis, S. (2023). Probabilistic SAR-based water segmentation with adapted Bayesian convolutional neural network. *Remote Sensing of Environment*, *285*, 113388.

Jaisakthi, S. M., Dhanya, P. R., & Jitesh Kumar, S. (2021, March). Detection of Flooded Regions from Satellite Images Using Modified UNET. In *International Conference on Computational Intelligence in Data Science* (pp. 167-174). Springer, Cham.

Jiang, X., Li, G., Zhang, X.P. and He, Y., 2021. A Semisupervised Siamese Network for Efficient Change Detection in Heterogeneous Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing.

Konapala, G., Kumar, S. V., & Ahmad, S. K. (2021). Exploring Sentinel-1 and Sentinel-2 diversity for flood inundation mapping using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *180*, 163-173.

Lalitha, V., & Latha, B. (2022). A review on remote sensing imagery augmentation using deep learning. *Materials Today: Proceedings*, *62*, 4772-4778.

Liang, Z., Zhu, B., & Zhu, Y. (2022). High resolution representation-based Siamese network for remote sensing image change detection. *IET Image Processing*.

Li, Y., Martinis, S., & Wieland, M. (2019). Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. ISPRS Journal of Photogrammetry and Remote Sensing, 152, 178-191.

Li, Y., Martinis, S., Wieland, M., Schlaffer, S. and Natsuaki, R., 2019. Urban flood mapping using SAR intensity and interferometric coherence via Bayesian network fusion. *Remote Sensing*, *11*(19), p.2231.

Lin, Y.N., Yun, S.H., Bhardwaj, A. and Hill, E.M., 2019. Urban flood detection with Sentinel-1 multi-temporal synthetic aperture radar (SAR) observations in a Bayesian framework: a case study for Hurricane Matthew. *Remote Sensing*, *11*(15), p.1778.

Mateo-Garcia, G., Veitch-Michaelis, J., Smith, L., Oprea, S. V., Schumann, G., Gal, Y., ... & Backes, D. (2021). Towards global flood mapping onboard low cost satellites with machine learning. *Scientific reports*, *11*(1), 1-12.

Mayer, T., Poortinga, A., Bhandari, B., Nicolau, A. P., Markert, K., Thwal, N. S., ... & Saah, D. (2021). Deep learning approach for Sentinel-1 surface water mapping leveraging Google Earth Engine. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, *2*, 100005.

Muñoz, D. F., Muñoz, P., Moftakhari, H., & Moradkhani, H. (2021). From local to regional compound flood mapping with deep learning and data fusion techniques. *Science of The Total Environment*, *782*, 146927.

Nguyen, H. D., Fox, D., Dang, D. K., Pham, L. T., Viet Du, Q. V., Nguyen, T. H. T., ... & Petrisor, A. I. (2021). Predicting future urban flood risk using land change and hydraulic modeling in a river watershed in the central Province of Vietnam. Remote Sensing, 13(2), 262.

Olthof, I., & Svacina, N. (2020). Testing urban flood mapping approaches from satellite and in-situ data collected during 2017 and 2019 events in Eastern Canada. *Remote Sensing*, *12*(19), 3141.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.

Pierdicca, N., Pulvirenti, L., & Chini, M. (2018). Flood mapping in vegetated and urban areas and other challenges: models and methods. In Flood Monitoring through Remote Sensing (pp. 135-179). Springer, Cham.

Pramanick, N., Acharyya, R., Mukherjee, S., Mukherjee, S., Pal, I., Mitra, D., & Mukhopadhyay, A. (2022). SAR based flood risk analysis: A case study Kerala flood 2018. *Advances in Space Research*, *69*(4), 1915-1929.

Pulvirenti L, Chini M, Pierdicca N. InSAR Multitemporal Data over Persistent Scatterers to Detect Floodwater in Urban Areas: A Case Study in Beletweyne, Somalia. Remote Sensing. 2021 Jan;13(1):37.

Rambour, C., Audebert, N., Koeniguer, E., Le Saux, B., Crucianu, M., & Datcu, M. (2020). Flood detection in time series of optical and sar images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *43*, 1343-1346.

Shawky, O. A., Hagag, A., El-Dahshan, E. S. A., & Ismail, M. A. (2020). Remote sensing image scene classification using CNN-MLP with data augmentation. *Optik*, *221*, 165356.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

Tanim, A.H., McRae, C.B., Tavakol-Davani, H. and Goharian, E., 2022. Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning. *Water*, *14*(7), p.1140.

Wang, J., Wang, S., Wang, F., Zhou, Y., Wang, Z., Ji, J., ... & Zhao, Q. (2022). FWENet: a deep convolutional neural network for flood water body extraction based on SAR images. *International Journal of Digital Earth*, *15*(1), 345-361.

Wang, M., Tan, K., Jia, X., Wang, X. and Chen, Y., 2020. A deep siamese network with hybrid convolutional feature extraction module for change detection based on multi-sensor remote sensing images. *Remote Sensing*, *12*(2), p.205.

Wang, Z., Peng, C., Zhang, Y., Wang, N. and Luo, L., 2021. Fully convolutional siamese networks based change detection for optical aerial images with focal contrastive loss. Neurocomputing, 457, pp.155-167.

Yang, X., Hu, L., Zhang, Y., & Li, Y. (2021). MRA-SNet: Siamese Networks of Multiscale Residual and Attention for Change Detection in High-Resolution Remote Sensing Images. *Remote Sensing*, *13*(22), 4528.

Yang, L., Chen, Y., Song, S., Li, F. and Huang, G., 2021. Deep Siamese networks based change detection with remote sensing images. *Remote Sensing*, *13*(17), p.3394.

Yu, X., Wu, X., Luo, C., & Ren, P. (2017). Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. *GIScience & Remote Sensing*, *54*(5), 741-758.

Zhang, C., Feng, Y., Hu, L., Tapete, D., Pan, L., Liang, Z., ... & Yue, P. (2022). A domain adaptation neural network for change detection with heterogeneous optical and SAR remote sensing images. *International Journal of Applied Earth Observation and Geoinformation*, *109*, 102769.

Zhang, H., Qi, Z., Li, X., Chen, Y., Wang, X., & He, Y. (2021). An Urban Flooding Index for Unsupervised Inundated Urban Area Detection Using Sentinel-1 Polarimetric SAR Images. *Remote Sensing*, *13*(22), 4511.

Zhang, X., He, L., Qin, K., Dang, Q., Si, H., Tang, X., & Jiao, L. (2022). SMD-Net: Siamese Multi-Scale Difference-Enhancement Network for Change Detection in Remote Sensing. *Remote Sensing*, *14*(7), 1580.

Chapter 3

Automated First Floor Height Estimation for Flood Vulnerability Analysis Using Deep Learning and Google Street View

3.1 Introduction

Flooding can cause significant disruptions in cities and impact people, the economy, and the environment. These impacts may be exacerbated by climate and socio-economic changes. However, they can be reduced by creating a resilient community for which vulnerable areas are identified and reported to city planners and decision-makers to take the necessary measures (Hammond et al., 2013). Locating vulnerable areas at the building scale provides valuable information to conservation organizations and insurance industry. Previous studies have focused on flood vulnerability analysis at the building scale, for example, Milanesi et al. (2018) developed a physical model to examine flash flood vulnerability for masonry buildings in alpine areas. Percival et al. (2019) combined physical factors (such as population density, green areas, utilities, and dwellings) and socio-economic factors (such as age, household structure, illness, or disability) to model flood vulnerability and risk in Portsmouth, UK. Leal et al. (2021) combined building properties (such as materials and structure, condition status, and number of floors) with flow parameters such as velocity and depth to model flash flood vulnerability at the building scale in the Barcarena basin.

First Floor Height (FFH) or the height of the building's first lowest floor above the adjacent grade (FEMA), is one of the critical parameters for flood vulnerability analysis and damage/loss estimates in urban areas. FFH can be applied for locating vulnerable structures and helping decision-makers change flood management policies (Hampton Roads Planning District, 2020). Change in FFH by less than a foot can increase or decrease flood loss estimates by hundreds of structures and millions of dollars across the community. Paulik et al. (2022) used the Random Forest model and Spearman's Rank correlation test to find important variables contributing to residential buildings damage for five flood events in New Zealand between the years 2013-2017. The results showed that

the water depth above the first finished floor is strongly correlated with total building damage ratios. Some recent studies have applied FFH indirectly or directly along with the number of stories, basement existence, area, and unit area repair cost in USD to calculate the average annual loss in Louisiana, USA (Gnan et al., 2022; Al Assi et al., 2023a; Al Assi et al., 2023b). Another application of FFH is in flood insurance models and setting premium policies. For example, Flood Risk 2.0 is a flood insurance model introduced by FEMA. This model applies FFH as one of the building attributes contributing to flood insurance premiums (Rahim et al., 2023). Another similar parameter used in the literature for flood vulnerability modeling is First Floor Elevation (FFE). FFE is the height of the first finished floor relative to the vertical datum. Taghinezhad et al. (2020) addressed the missing FFE issue for Louisiana, USA using the imputation method. Wang and Sebastian, (2021) used the median of FFE, mean water depth, building age, and material, along with demographic and household income information to model flood vulnerability in North Carolina, USA.

One of the informative documents commonly prepared at the building scale by emergency organizations is an Elevation Certificate (EC). This document presents the level of compliance with floodplain regulations to determine the building vulnerability to flooding and the cost of flood insurance premiums. EC gives information about the Lowest Adjacent Grade (LAG) which is the lowest point of ground level immediately next to the building (FEMA, 1999) along with information about buildings' latitude and longitude foundation type, and flood zone. This document also includes information on the FFE and the Base Flood Elevation, which is the elevation of water surface resulting from a flood that has a 1% chance of equaling or exceeding that level in any given year (FEMA, 1999). The information in EC is commonly extracted through site inspection and ground survey, which is time-consuming and labor-intensive. Besides the ground survey method, other techniques, including statistical models and imagery-based analysis, were used in the literature for FFH estimation (Gordon and McFarlane, 2019; Du et al., 2020). Ning et al., (2021) developed a methodology based on Google Street View (GSV) images, Deep Learning (DL), and Tacheometric Surveying principals to estimate FFE for an area in the USA. This method was used in Gao et al., (2023) for FFE estimation in Galveston Islands, Texas, and the methodology expanded by estimating the cost of elevating FFE for buildings

at risk of flooding based on the FEMA formula. Statistical methods build a statistical relationship between the information in the EC, such as foundation type, structure age, building occupancy type, and difference in grade as explanatory variables, and the FFH parameter as the dependent variable. For example, Gordon and McFarlane (2019) developed a Random Forest model based on the foundation type, structure age, flood zone, and difference in grade to estimate FFH for Hampton and Chesapeake cities in Virginia State. One of the limitations of the RF algorithm was the FFH underestimation for high FFH values. Also, the developed RF model was based on the information derived from EC that is unavailable for most flood-prone areas worldwide, and its preparation is time-consuming and a tedious task. Diaz et al. (2022) used a positioning-enabled Unmanned Aerial System (UAS) and created a detailed 3D photogrammetric model to estimate the FFE of the buildings in Galveston Island, Texas. Although the reported mean absolute error was promising (0.16m), the method is costly and does not apply to large areas.

Computer vision techniques can be applied to reduce the amount of fieldwork for FFH estimation. These methods include image classification, clustering, object localization, and object detection. Image classification involves assigning a class label to each pixel in an image, whereas clustering considers image pixels as connected components and labels each connected component. Object localization involves drawing a bounding box around one or more objects in an image, whereas object detection is more challenging and combines classification and localization tasks (Brownlee, 2019). It creates a bounding box around each object of interest in the image and assigns it a class label. DL can be used for object detection, with its foundation rooted in neural networks. These networks are algorithms that mimic the workings of the human brain and are designed to recognize patterns. DL is a way of classifying, clustering, and predicting variables using a neural network trained on vast amounts of data. It creates many layers of neurons, attempting to learn a structured representation of big data layer by layer. Two types of DL algorithms can be applied to an object detection problem;:1- Region-Based Convolutional Neural Networks (R-CNNs) developed by Girshick et al. (2014); 2- You Only Look Once, or YOLO, a second family of DL algorithms used for object recognition and designed for speed and real-time use (Redmon et al., 2016). The errors in YOLO algorithm were improved by developing subsequent versions of the algorithm, including YOLOv2 (Redmon and Farhadi, 2017),

YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al. 2020), and YOLOv5 (Jocher, 2020). YOLOv5 has been used widely for object detection in natural and remote sensing images and has achieved promising results. Previous works used this algorithm for face masks, safety helmets, and fruit detection in natural images (Yang et al. 2020; Zhou et al. 2021; Yao et al. 2021). Wu et al. (2021) combined YOLOv5 with a Convolutional Neural Network (CNN) for detecting small targets in optical satellite images. Zhao et al. (2021) applied YOLOv5 for wheat spike detection using UAV images, and Mathew and Mahesh (2022) used the algorithm for leaf-based disease detection using the PlantVillage dataset.

DL algorithms have recently been used for FFH estimation using imagery-based analysis. For example, Du et al. (2020) proposed a CNN for FFH estimation. The CNN was designed for feature extraction from the images captured from the front view of the buildings, and a fully connected network was considered for FFH estimation. The extracted features were the distance between the ground and the lowest boundary of the Front Door (FD) and the FD size on the image. Ning et al. (2021) estimated First Floor Elevation (FFE) using a DL algorithm, YOLOv5 (Jocher, 2020), to detect FD. Based on the principles of the Tacheometric Surveying and the depth maps extracted from the Google Street View (GSV) panorama images, the FFE was estimated for Hampton, Virginia. The developed algorithm in this work did not estimate FFH, which is a critical parameter for flood vulnerability analysis. Most hydraulic software tools provide flood depth maps as the output, not flood elevation. In other words, flood vulnerability analysis can be achieved by comparing the flood depth values with FFH. In addition, most existing works for FFH or FFE estimation have been conducted for the state of Virginia in the USA. However, testing the methods on just one case study introduces limitations due to the specific architectural and structural characteristics prevalent in that area. Besides, models tested on just one case study lack generalizability. Notably, there exists a research gap for FFH estimation using the DL algorithm, particularly within the Canadian context. The objective of this study is to propose an automatic algorithm based on DL to estimate FFH using the YOLOv5s algorithm, and GSV images. The proposed method is demonstrated through a case study for Greater Toronto Area (GTA).

3.2 Case study and dataset for FFH and flood vulnerability prediction

The study region was the Greater Toronto Area (GTA), which includes the city of Toronto and four regional municipalities, including Durham, York, Peel, and Halton, and has a total land area of 7125 km^2. The city of Toronto has a population of 2.48 million, constituting about 45% of the whole GTA. Toronto Region Conservation Authority (TRCA) is responsible for improving the health and well-being of watershed areas by protecting and restoration of the natural environment. TRCA has divided the GTA area into ten watersheds, including Carruthers, Don, Duffins, Etobicoke, Frenchman's Bay, Highland, Humber, Mimico, Petticoat, and Rouge. The delineation of GTA watersheds and the location of selected buildings was shown in Figure 3-1.

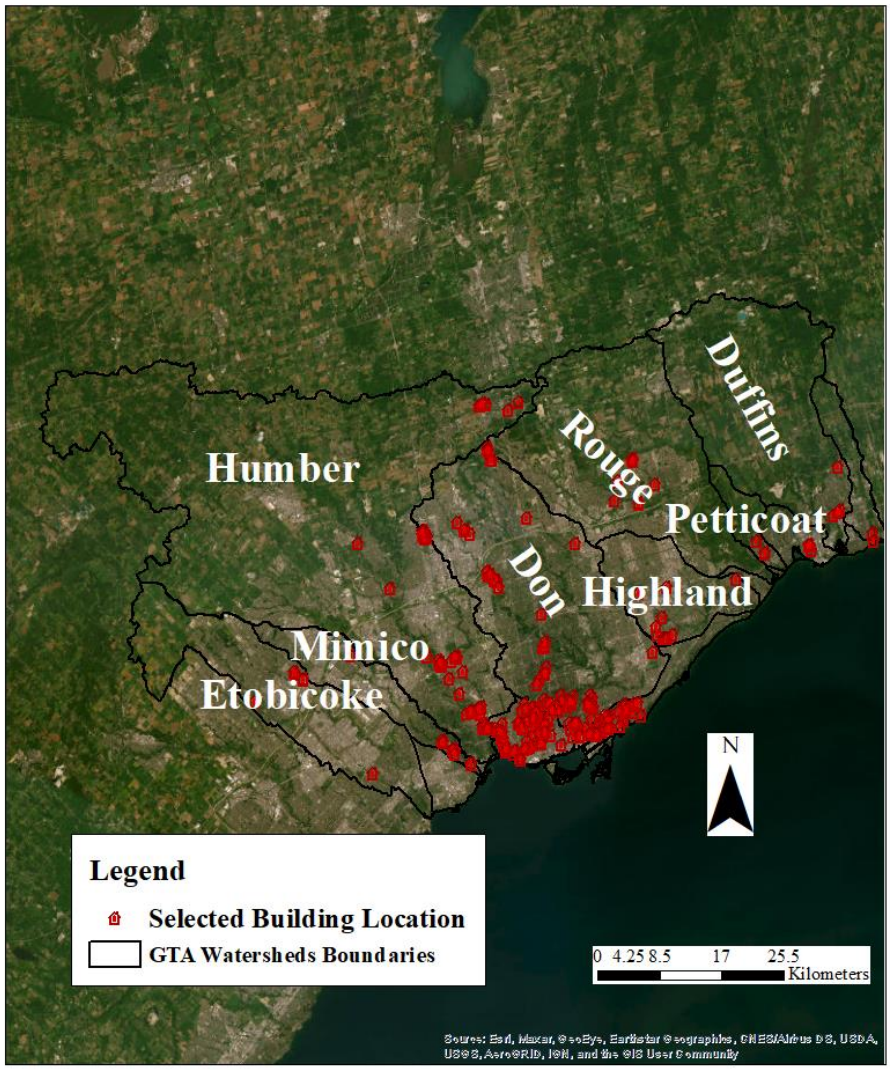


Figure 3-1: Delineation of GTA watersheds and selected buildings overlaid onto the Arc GIS base map

The dataset applied in this work included GSV images captured from the front view of buildings in the Greater Toronto area (GTA) and Virginia State, USA. The Virginia State case study was added as a comparison with the method in Ning et al. (2021) developed for the region. GSV images were downloaded automatically using Google Cloud Application

Programming Interface (API). GSV image parameters, including camera Field Of View (FOV) and heading angles, were set to 90 and 0 degrees for most locations. Locations where there was no GSV image available or the building FD was not fully visible in the image were discarded from further analysis. Finally, 760 images were separated for FFH estimation for the GTA and Virginia State areas. Ontario Classified Point Cloud (LiDAR-derived) (classified into Unclassified, Ground, Water, High, and Low Noise categories and consisted of non-overlapping 1km by 1km tiles) covering areas in southern Ontario and portions of northern Ontario and the 2013 USGS Lidar point cloud data for Norfolk, Virginia State were converted from LAZ to LAS using the LASzip tool and then imported into ArcMap. The last returns were extracted from the point cloud data, and Digital Elevation Models (DEM) were generated with a cell size equal to 25cm. The building footprint data for the GTA region was accessed via the Open Database of Buildings from the Statistics Canada website, and the FFH ground truth data for GTA was provided by TRCA. The building footprints and EC data, including the FFH ground truth values for the Virginia area, were downloaded from the Hampton Roads Planning District Commission (HRPDC) website. The flood depth map for the Lower Don area was accessed under the data sharing agreement between the first author and TRCA.

3.3 Methodology

The proposed FFH estimation method includes three steps: 1- Extracting bounding box coordinates for the FD and stairs; 2- GSV pixel size estimation in y-direction; and 3- FFH estimation. To accomplish this task, we considered two priorassumptions:

1- The camera's Line of Sight (LOS) is horizontal and does not have any rotation. This assumption makes it possible to measure the FFH using calculations in the y-direction. In other words, to be able to use the bounding box y (row) dimension to estimate the FFH value, the camera LOS must be horizontal.

2- The FD dimensions in American-style houses are usually 0.91m in width and 2.032m in height (Du et al., 2020). Ning et al. (2021) made the same fixed FD height assumption, and the RMSE for the estimated FFE was 15% of the assumed FD height. The achieved

RMSE was competitive among methods that do not use Ground Control Points (GCP). For this reason, this work also made the same assumption for the FD height.

Based on Figure 3-2a, the FFH in the image coordinate is the difference between the y (row) coordinates of the lower right corners of the stairs and FD bounding boxes. In Figure 3-2b, y and x are the row and column direction of GSV image. Based on the second assumption, the GSV pixel size in the y (row)-direction can be estimated using Equation (3-1). In this Equation, y_1 and y_2 refer to the upper left and lower right row numbers for the FD bounding box, respectively. After substituting the y_2 and y_1 parameters with the image coordinates of the lower right corners of the stairs and FD, the FFH can be estimated using Equation (3-2). In this equation, y_{Stairs}^{LR} is the row number for the lower right corner of the bounding box around the stairs, y_{FD}^{LR} is the FD bounding box lower right corner row number, and $pixel\ size_y$ is the GSV image pixel size in the y (row) direction, which is estimated based on the FD height (Figure 3-2b).

$$FD\ Height = (y_2 - y_1) \times pixels\ size_y \tag{3-1}$$

$$FFH = (\ y_{Stairs}^{LR} - y_{FD}^{LR}) \times pixel\ size_y \tag{3-2}$$

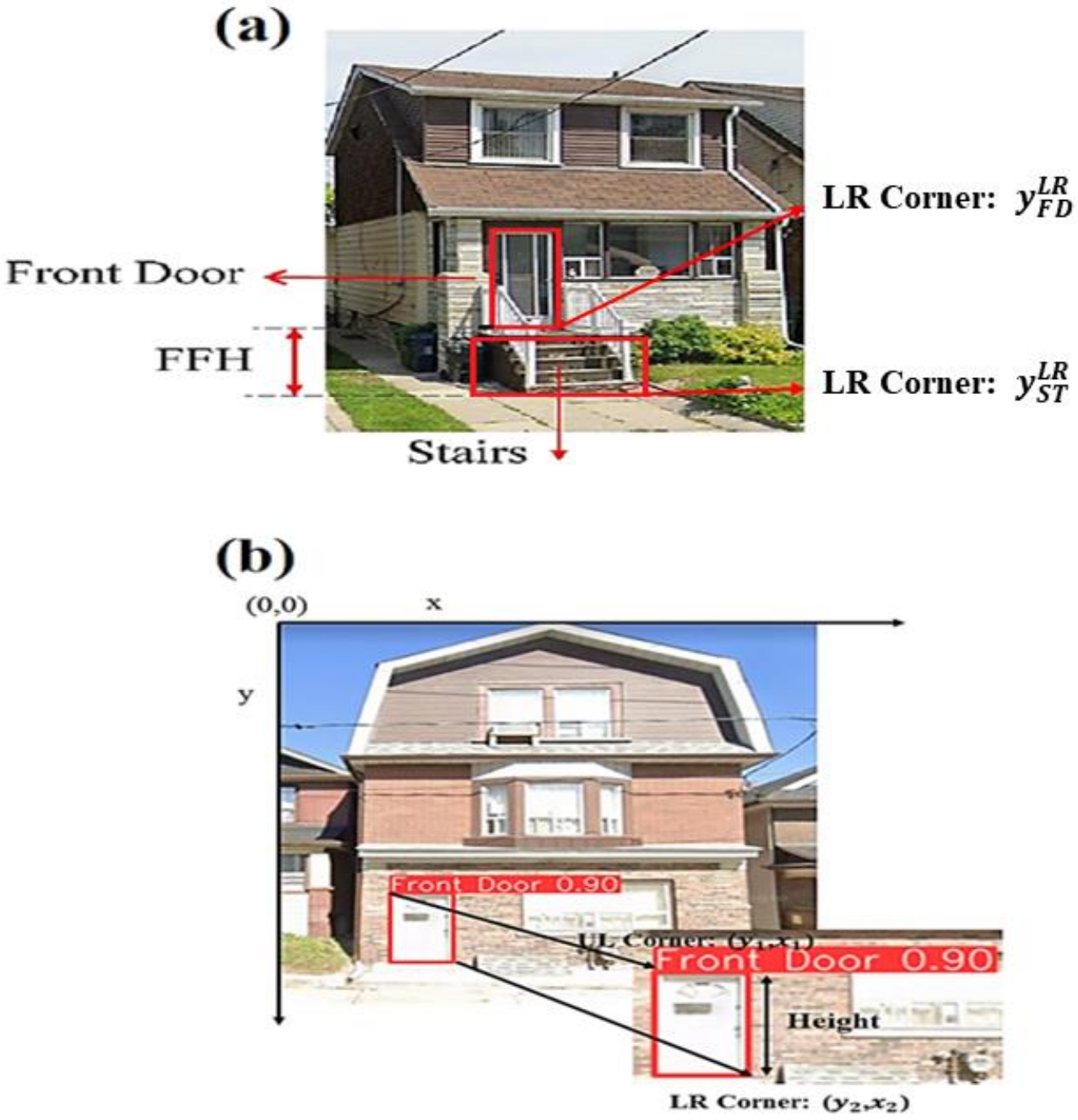


Figure 3-2: a) FFH definition b) GSV pixel size estimation in y-direction; 0.9 refers to the bounding box confidence value; UL and LR stands for the Upper Left and Lower Right corners

The next step was to predict flood vulnerability for selected buildings across GTA based on the estimated FFH using Equation (3-2) and the basement existence or lack of existence. If YOLOv5s detected a Basement Window (BW), a basement would be considered for the building. Figure 3-3 shows the steps for FFH estimation using the YOLOv5s DL algorithm and flood vulnerability prediction. The flood vulnerability for selected buildings was predicted using a water depth grid map, FFH, and basement information based on the YOLOv5s object detection algorithm.

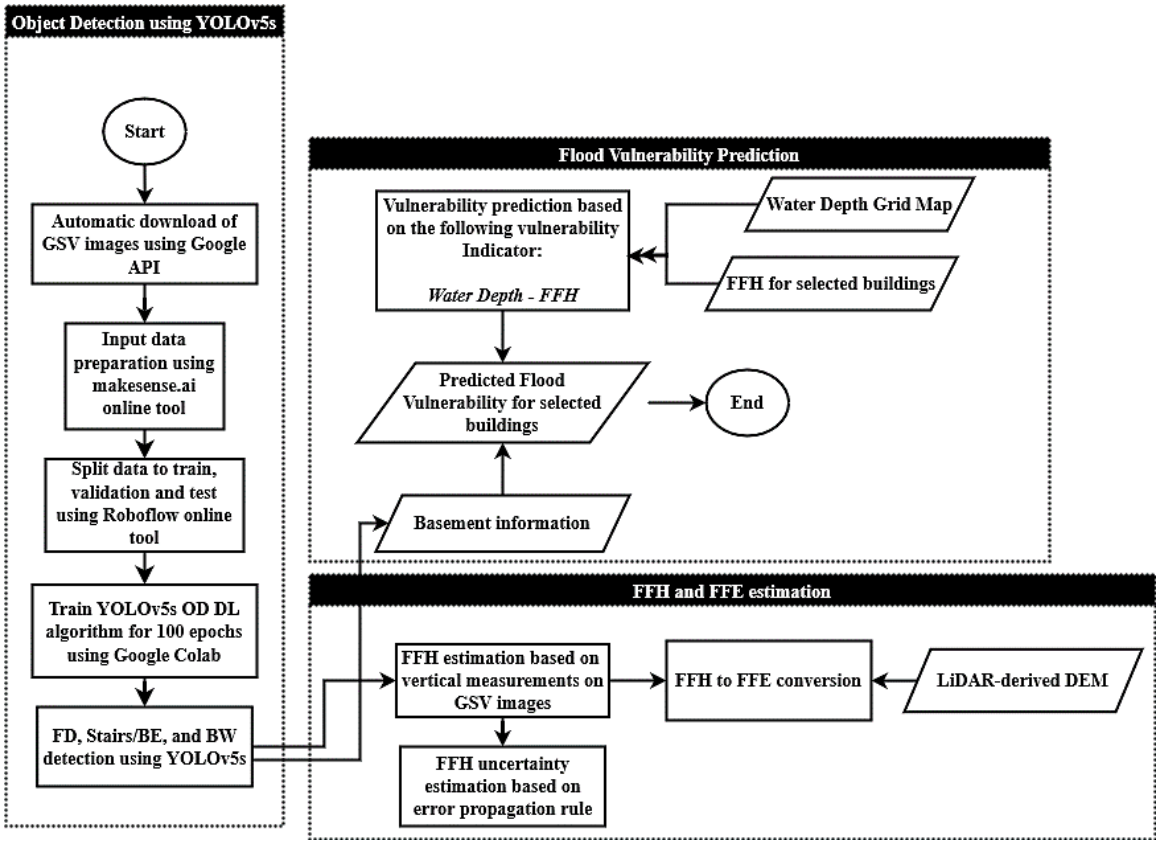


Figure 3-3: Flood vulnerability prediction based on FFH values estimated using GSV

3.3.1 Front Door and Stairs detection using YOLOv5s Deep Learning algorithm

3.3.1.1 YOLOv5s Deep Learning algorithm

YOLO, an acronym for '*You only look once*,' is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself. It is one of the most widely used object detection algorithms due to its speed and accuracy.

The GSV images were manually classified into High, Medium, and Low FFH based on the definition presented in Equation (3-3). The assumption was that all the stairs have a similar standard height of 15cm. Then, the Yolov5s model was trained separately on each set of images, and the 5-fold cross-validation method was used for training the models and accuracy estimation. Table 3-1 shows the number of GSV images used at each category.

$$\begin{cases} number\ of\ steps \leq 2 & Low\ FFH \\ 2 < number\ of\ steps \leq 5 & Medium\ FFH \\ number\ of\ steps > 5 & High\ FFH \end{cases} \quad (3\text{-}3)$$

Table 3-1: Number of train, validation, and test GSV images in each FFH category

Case Study	Category		Number of GSV images
GTA	Low FFH	Train	124
		Validation	10
		Test	65
	Medium FFH	Train	286
		Validation	30
		Test	85
	High FFH	Train	115
		Validation	15
		Test	30

The model weights were downloaded from GoogleColab, and the models were loaded sequentially on the local computer by importing the saved custom weights to the Pytorch Hub load function. After loading the models, the test data were imported, and FD and stairs in each image were detected. The outputs were some bounding boxes around each detected object with the bounding box coordinates of the upper left and lower right corners. In images with more than one FD and more than one stair set, the lower object (higher y) was selected, and objects with lower y coordinates were removed from FFH estimation because of the confusion arising when locating the corresponding FD and stairs. So, there were just two objects/bounding boxes, one for the FD and the other for the stairs.

After FFH estimation for each fold and image set using Equation (3-2), the addresses were geocoded by extracting the latitude and longitude values using the Python *requests* library and *positionstack* geocoding API. Four scenarios were considered for the Lowest Adjacent Grade (LAG) calculation, including *Point*, *Maximum*, *Minimum*, and *Mean*. In the *Point* case, the ground height at the address location was considered equivalent to the LAG parameter and extracted from the LiDAR-derived Digital Elevation Model (DEM), using *the Extract Values to Points* tool in ArcMap to convert FFH to FFE. In the *Maximum*, *Minimum*, and *Mean* cases, the *Maximum*, *Minimum*, and *Mean* DEM

height inside the buildings' footprints were calculated using the ArcMap *Zonal Statistics* tool. Finally, the FFE predictions for each fold and image set were combined and reported as the final result.

Pseudo Code for FFE estimation

Input: LiDAR-derived DEM, bounding boxes image coordinates for FD and Stairs were extracted using the YOLOV5s object detection algorithm

---

Part 1: GSV pixel size estimation in height (y) direction

Calculate FD height in image scale:

Extract the Upper Left (UL) and Lower Right (LR) coordinates of the FD bounding box and calculate FD height using formula *:

$$h_{FD}=\boldsymbol{y}_{\boldsymbol{LR}}^{\boldsymbol{FD}}-\boldsymbol{y}_{\boldsymbol{UL}}^{\boldsymbol{FD}} \quad *$$

Use the prior knowledge about standard FD height dimension (H_{FD}) and estimate GSV pixel size in y direction using Equation (3-1).

Part 2: FFH estimation

If no FD detected in the image:

Exclude the image from FFH estimation

If no Stairs detected in the image:

Consider the Building Extent bounding box LR corner instead of Stairs in Equation (3-2)

For all classes including, FD, Basement Window, and Stairs:

If more than one object of the same class was detected in the image:

Keep the object with higher y coordinate and remove other objects

Calculate FFH using Equation (3-2).

Part 3: Convert FFH to FFE

Geocoding the Addresses, finding their locations on the LiDAR-derived DEM, and extracting the height values at the corresponding locations (H_{DEM}), using one of the *Point*, *Maximum*, *Minimum*, and *Mean* methods. Then, calculating FFE using formula ** :

$FFE = H_{DEM} + FFH$ **

Output: FFE value and a flag if the image includes basement or not

3.3.2 Contribution of object detection uncertainty to FFH estimation

In this section, the error propagation equation for FFH is formulated based on the variance of estimated image coordinates for the FD, stairs, and Building Extent (BE). Based on the error propagation rule (Ogundare, 2018), the variance for FFH in Equation (3-2) can be estimated using Equation (3-4).

$$\sigma_{FFH}^2 = \alpha \times GSV_{pixel\ size_y}^2 + \left(\frac{FFH}{pixel\ size_y}\right)^2 \times \sigma_{GSV_{pixel\ size_y}}^2 \quad (3\text{-}4)$$

with, $\alpha = \left((\sigma_{y_{LR}}^2)^{FD} + (\sigma_{y_{LR}}^2)^{ST}\right)$, and $y_{LR}^{ST} - y_{LR}^{FD} = \frac{FFH}{pixel\ size_y}$

Based on Equation (3-1), the GSV pixel size in the y direction ($GSV_{pixel\ size_y}$) can be substituted with its value based on the real-world and image dimensions of the FD. Also, the variance for $GSV_{pixel\ size_y}$ can be estimated after taking the variance from both sides of Equation (3-1) using Equation (3-5):

$$\sigma^2_{GSV_{pixel\ size_y}} = (\frac{H_{FD}}{h^2_{FD}})^2 \sigma^2_{h_{FD}} = \beta \times \sigma \quad (3\text{-}5)$$

In Equation (3-5), $\beta = (\frac{H_{FD}}{(y^{FD}_{LL} - y^{FD}_{UL})^2})^2$, and $\sigma = \left((\sigma^2_{y_{LL}})^{FD} + (\sigma^2_{y_{UL}})^{FD}\right)$.The Equation above assumes that the real-world dimension of the FD (H_{FD}) is a fixed parameter. So, when calculating the $GSV_{pixel\ size_y}$ variance, the derivative with respect to this parameter is zero.

After substitution of Equation (3-5) in Equation (3-4), it can be reformulated as Equation (3-6):

$$\sigma^2_{FFH} = \alpha \times GSV^2_{pixel\ size_y} + \left(\frac{FFH}{pixel\ size_y}\right)^2 \times \beta \times \sigma \quad (3\text{-}6)$$

3.3.3 Flood vulnerability Prediction for selected buildings

A flood depth map covering the Lower Don region accessed from the TRCA was used to conduct a flood vulnerability analysis for selected buildings. To assess the flood vulnerability, firstly, the water depths at selected addresses were extracted using the ArcMap *Extract Values to Points* tool, and the parameter *water depth minus FFH*, along with information about the basement, were considered indicators of flood vulnerability. If the building had a basement, its flood vulnerability value would be classified as very high, no matter what the water depth value *minus FFH was.* The lower the water depth value *minus FFH*, the lower the flood vulnerability. A large negative value for this difference shows that water depth is far below the FFH, and the chances of the building

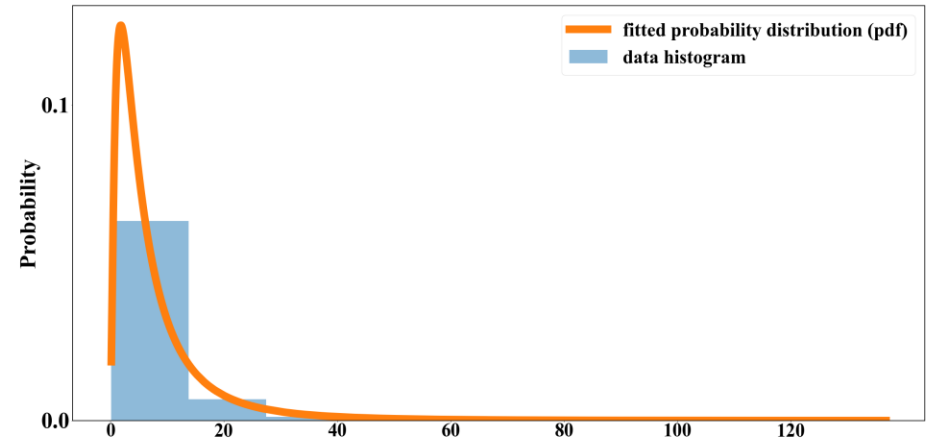
being flooded during a flood event are low and vice versa. After calculating the difference parameter at each location, the ArcMap *Natural Break* classification method was applied to the buildings where the DL algorithm did not detect any basement. Then, the vulnerability values were categorized into five flood categories, including very low, low, medium, high, and very high. The final output was flood vulnerability prediction showing the vulnerability values at selected buildings across GTA. Please note that any other classification method can be applied, and examining the flood vulnerability prediction sensitivity to the classification method was beyond the scope of this study.

3.4 Results

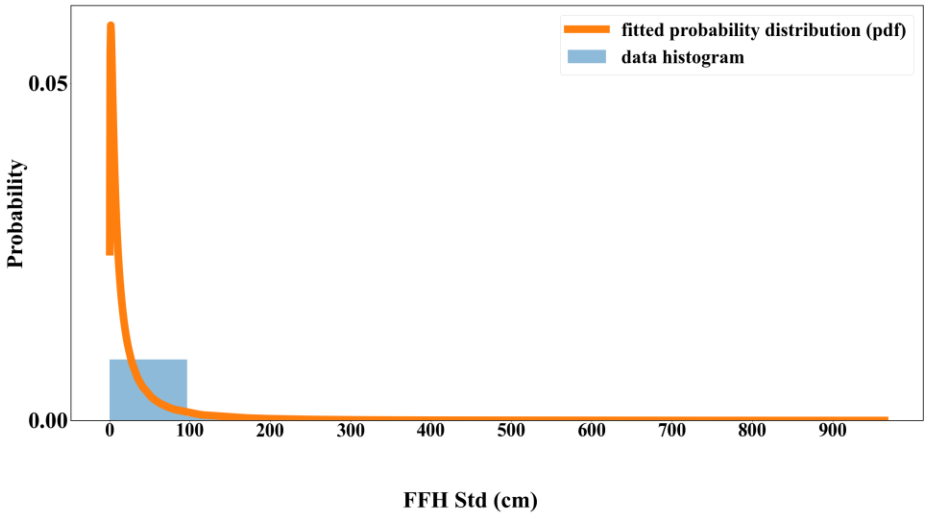
3.4.1 FFH uncertainty analysis

As mentioned in the methodology section, the uncertainty in object detection can contribute to the FFH estimation uncertainty based on Equation (3-7). This section will discuss the uncertainty analysis for the GTA and Virginia areas.

Based on Figure 3-4, the error distributions for both areas follow a lognormal distribution. It means the estimated FFH values are distributed around the mean, and the random error is around zero. The number of outliers for Virginia were higher than GTA because, for the GTA area, the downloaded GSV images were from a shorter distance to the building than in Virginia, and the building elements, including the FD and stairs/building extent, were more evident in the images compared to the Virginia case. Please note that a lower resolution was used for the data histogram than the fitted error distribution because of the limited number of samples.



(a)



(b)

Figure 3-4: Data probability distribution and fitted probability distribution for FFH standard deviation; a) GTA; b) Virginia

Scatterplots in Figures 3-5 and 3-6 show the FFH standard deviations for the GTA area in terms of standard deviations for the FD Upper Left (UL) and Lower Left (LL) corners

(Figure 3-5) and based on the FD and stairs/building extent LL corners (Figure 3-6). Standard deviation larger than 50 pixels in either FD or stairs/building extent corners resulted in FFH standard deviations larger than 1m. For example, based on Figure 3-5, in three cases, FD UL standard deviations of 58.52, 57.69, and 54.13 pixels were observed. For cases (a) and (c), where buildings have a medium FFH, the uncertainty can increase based on Equation (3-6). For case (b), the garbage basket in front of the building was erroneously detected as stairs, and FFH was overestimated. This overestimation can contribute to a significant FFH uncertainty. Figure 3-6 shows a case in which large stairs LL corner standard deviation (89.69 pixels) increased FFH uncertainty to about 1.37m. The immediate adjacent staircase was considered as ground truth, but the algorithm estimated FFH based on the lower stair set, and FFH was overestimated. So, in this case, large stairs LL standard deviation and consequently FFH overestimation both contributed to the large FFH uncertainty (higher than 1m). This case shows a challenging situation in FFH estimation using the proposed algorithm and confirms the importance of correctly locating the LAG when estimating the FFH.

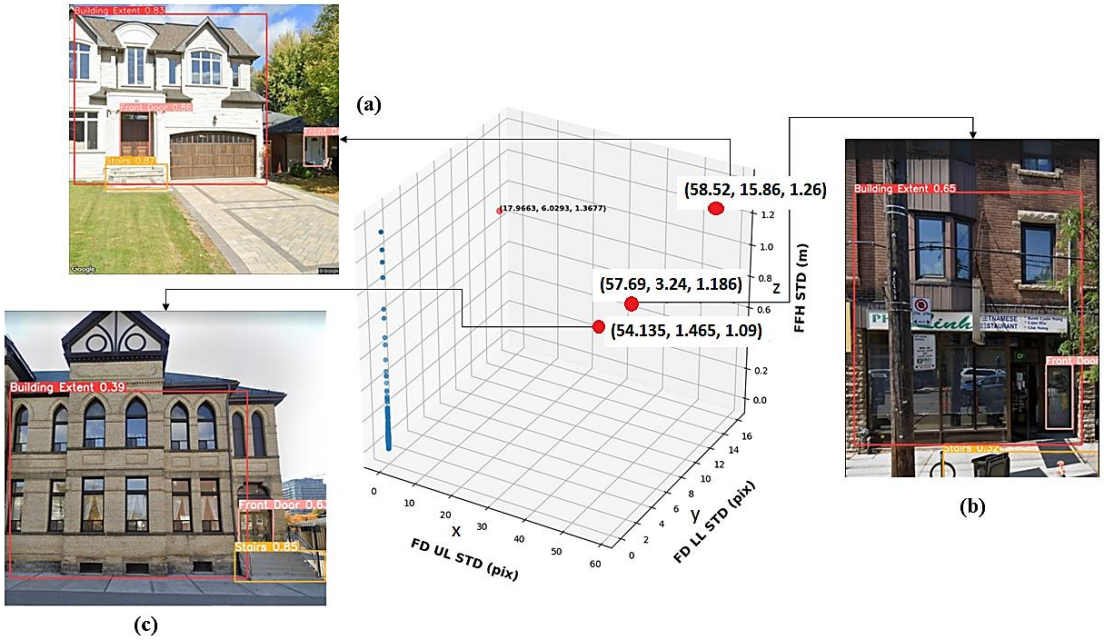


Figure 3-5: Examples of large FFH estimation uncertainty (outliers) based on the FD UL and LL corners (GTA) ; blue points show the records with FFH standard deviation less than 1m; red points are the records with FFH standard deviation larger than 1m; the red point not corresponding to any image also has FFH standard deviation larger than 1m and is replicating one of the shown scenarios on other GSV images

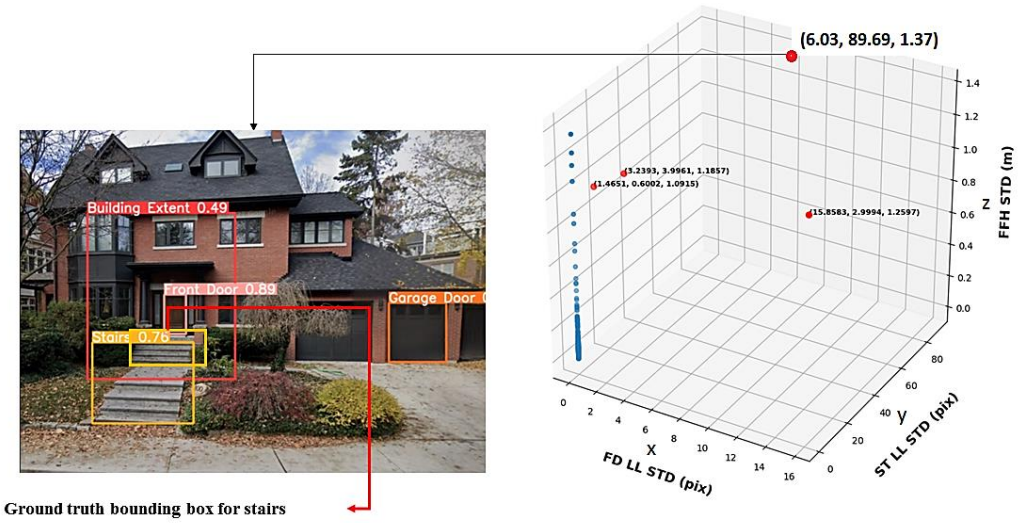
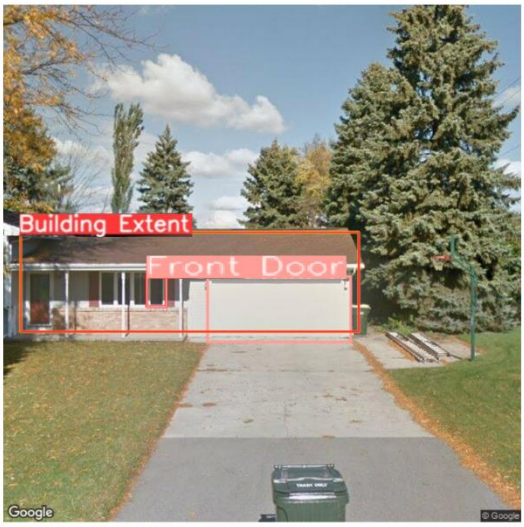


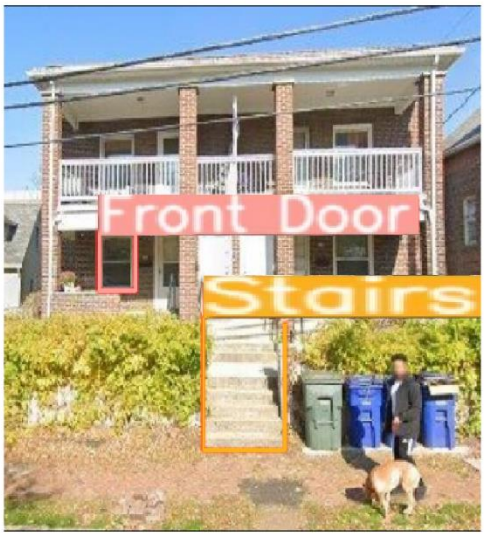
Figure 3-6: Examples of FFH predictions with large uncertainty based on FD and stairs/building extent LL corners (GTA) ; blue points show the records with FFH standard deviation less than 1m; red points are the records with FFH standard deviation larger than 1m; the red points not corresponding to any image also have FFH standard deviation larger than 1m and are replicating the scenario shown in the GSV image

One source of considerable FFH uncertainty can be detecting windows as the FD. This false detection can increase FFH uncertainty in two ways. 1- large standard deviations for the FD UL and LR corners; 2- windows are usually located at higher elevations than the FD, causing FFH overestimation, and based on Equation (3-6), FFH uncertainty also increases. Figures 3-7 shows examples of cases when the algorithm erroneously detected the window as the FD, resulting in high FFH uncertainty.



(a)

(b)



(c)

(d)

Figure 3-7: Example of YOLOv5s object detection results in which windows were erroneously detected as the FD ; a) 7 Black Oak Court (σ_{FFH} = 1.177m); b) 2207 Bayberry St (σ_{FFH} = 1.13m); c) 201 West Fourth Ave (σ_{FFH} = 1.705m); d) 198 Villa Drive (σ_{FFH} = 3.985m)

Figure 3-8 compares the FFH uncertainty values for three GSV images. Cases (a) and (b) have similar FD and stairs standard deviation values to case (c), but based on the scatter plot, cases (a) and (b) have considerably higher FFH uncertainty with values of 9.46m and 7.59m, respectively, compared with 1.19m for case (c). The reason is the difference between the estimated FFH values. For case (a), the building has a higher FFH value compared with case (c), so the FFH uncertainty based on Equation (3-6) is higher than (c). For case (b), the FFH has been overestimated because of erroneously considering the stairs of the adjacent building. These object detection errors usually happen when there is more than one building in the GSV image extent. The existence of more than one building object in the GSV increases the chance of errors in FFH estimation and its uncertainty.

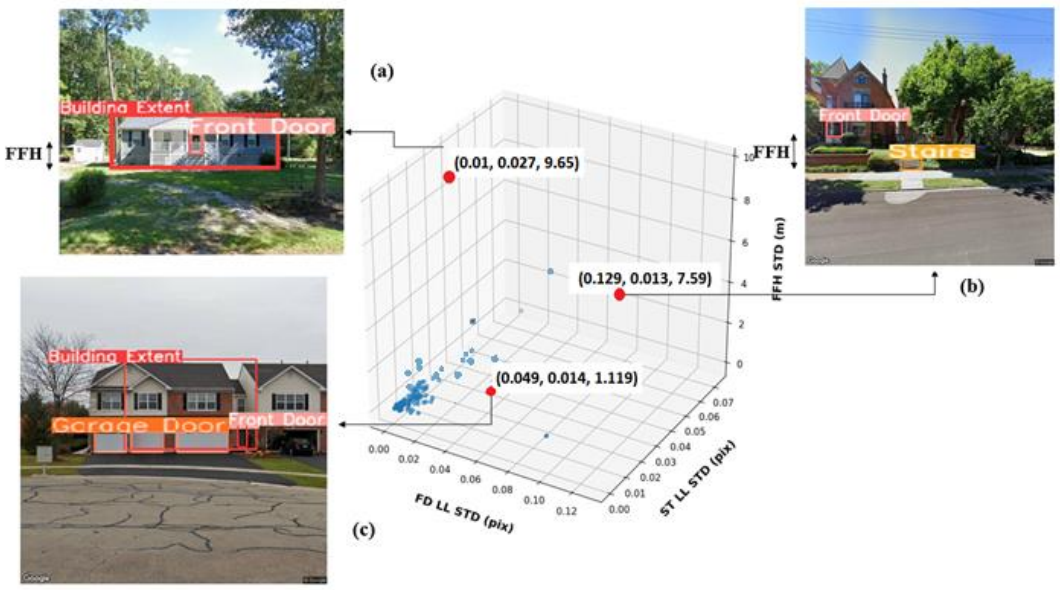


Figure 3-8: Cases a and b have higher FFH uncertainty than c because of larger estimated FFH; blue points show the records with FFH standard deviation less than 1m; red points are the records with FFH standard deviation larger than 1m

3.4.2 FFH prediction results

Figures 3-9 and 3-10 show the predicted FFH values vs. ground truth for the GTA and Virginia regions. For some buildings with FFH over 0.6m (GTA) and 1.5m (Virginia), which can be considered buildings with relatively high and very high FFH based on the definition presented in Equation (3-3) (four steps and above in front of the FD), the algorithm underestimated FFH. The negative Bias value for both case studies admits that the proposed algorithm overestimated the FFH by about 20-50cm in most cases.

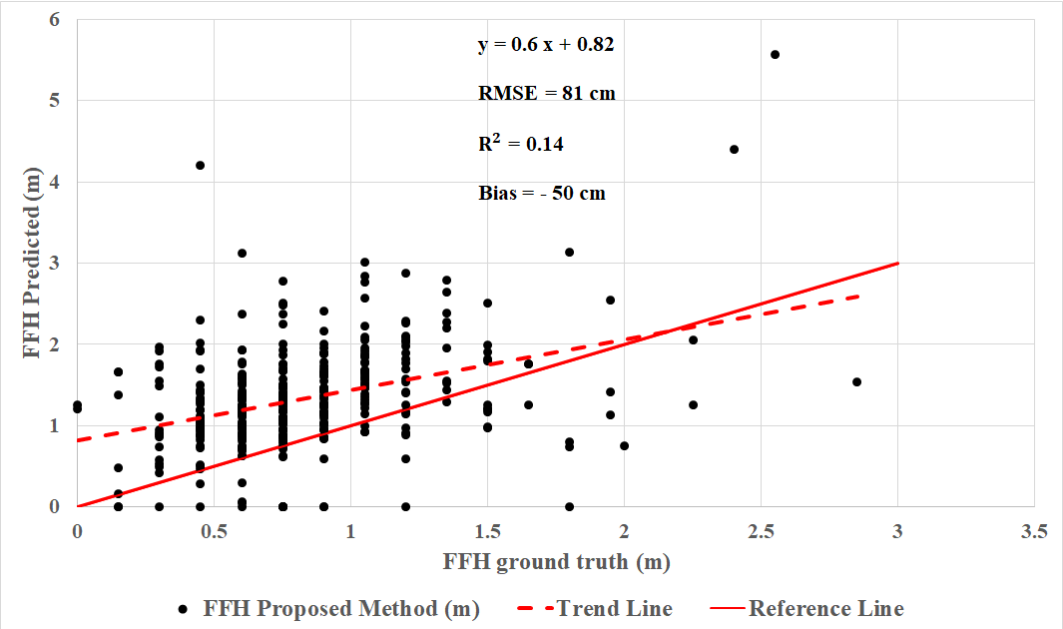


Figure 3-9: Scatter plot of FFH predicted values vs FFH ground truth for GTA; the solid line is y=x and the dotted line shows the trend line

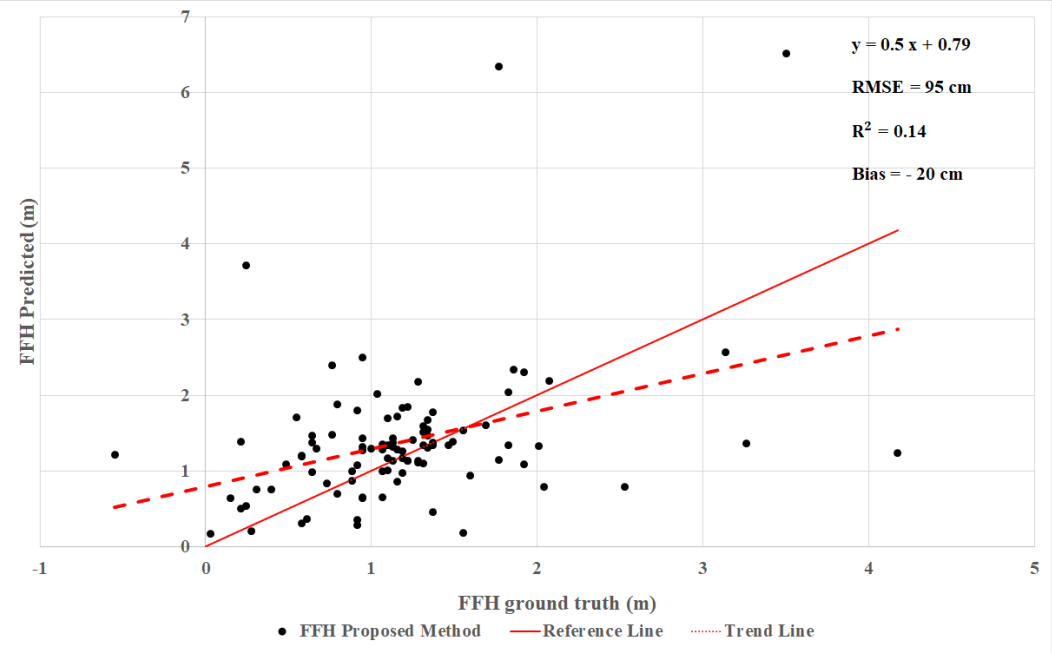


Figure 3-10: Scatter plot of FFH predicted values vs ground truth values for Virginia region; the solid line is y=x and the dotted line shows the trend line

3.4.3 Comparison with Tacheometric Surveying Method

Ning et al. 2021 used the vertical dimension of GSV images and depth maps extracted from the panorama images to calculate the FFE for the Hampton Roads area in Virginia, USA. Their methodology was based on the principles of Tacheometric Surveying and is referred to as Tacheometric Surveying hereafter.

3.4.3.1 FFE prediction for Virginia Region

Table 3-2 shows FFE RMSE, R^2, and Bias values for the Virginia region. Three cases were considered for the proposed method; in the first case, the latitude and longitude for addresses were used to extract the height from DEM (referred to as the *Point* method hereafter). In the second and third cases, the mean and minimum heights inside the building footprint were used to convert FFH to FFE (referred to as the *Mean* and *Minimum* methods

hereafter). The most accurate result was for the *Mean* method with an RMSE value of 1.04m, and the least biased estimator was the *Minimum* method, with a Bias value of 2cm. However, the reported indices were computed on a smaller test dataset than the *Point* method because some addresses had no building footprint data available. RMSE and R^2 values for the *Minimum* method were comparable to the *Mean*, with values of 1.08m and 0.83. The higher R^2 and lower RMSE values for the *Mean* and *Minimum* method compared with the *Point* method cannot be translated into a better FFE prediction because the results for these two methods were computed on fewer buildings because of the lack of access to building footprints for some locations. The negative Bias for the *methods* admits that the proposed algorithm overestimated the FFE.

Table 3-2: FFE RMSE, R^2, and Bias for the proposed and Tacheometric surveying methods for Virginia

	RMSE (m)	R^2	Bias (m)
Proposed Method (Point)	2.71	0.42	-0.68
Proposed Method (Mean)	1.04	0.86	-0.20
Proposed Method (Minimum)	1.08	0.83	-0.012
Tacheometric Surveying	2.97	0.44	-0.73

Figure 3-11 compares the Virginia FFE predictions vs. ground truth for the proposed and Tacheometric Surveying methods. The circle, square, and triangle points are for the *Point*, *Mean,* and *Minimum* methods, respectively. Based on the Figure, the Trend Line for the *Point* method is far above the Reference Line (the solid line), which shows overestimation. For the *Mean* method and the FFE values below 5m, the Trend Line was

above the Reference Line, but for buildings with very high FFE, the method slightly underestimated the FFE, and the Trend Line fell below the Reference Line. Because the concentration of test samples was for FFE values below 5m, the method overestimated FFE on average, and the overall Bias reported in Table 3-2 was a negative value below 0.7m. The Trend Line for the *Minimum* Method was far below the Reference Line as the method underestimated FFE values by about 1cm, as reported in Table 3-2. After using the *Mean* method, the Trend Line became closer to the Reference Line, and the R^2 value improved significantly by 0.44 and 0.42 compared to the *Point* and Tacheometric Surveying methods. The spike in the R^2 happened because no FFE estimation was available for the addresses for which the other two methods considerably overestimated. As mentioned before, there was no building footprint for these addresses. After using the proposed method, the RMSE, and Bias improved by 1.93m and 0.53m (considering the *Mean* method) compared to the Tacheometric Surveying method, which shows the reliability of the proposed algorithm.

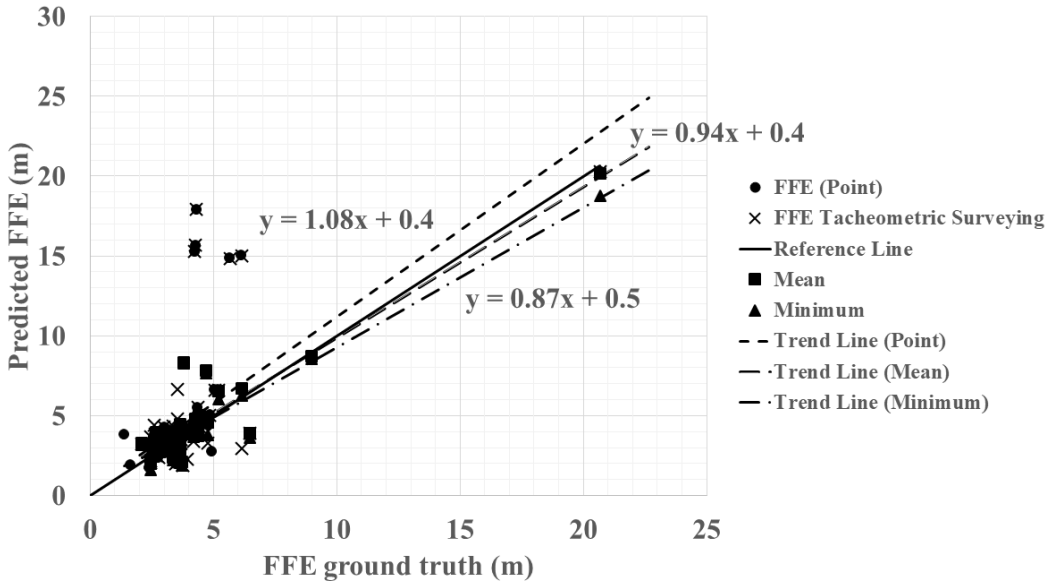


Figure 3-11: Scatter plot of predicted FFE values vs ground truth for the proposed and Tacheometric Surveying methods for Virginia ; the point, square and triangle marks show the predictions for the *Point, Mean*, and *Minimum* methods, respectively; the cross marks are for the Tacheometric Surveying method; the small

dashed line, and the large dashed lines are trend lines for the *Point* and *Mean* methods, respectively, and the dashed-dotted line is the trend line for *Minimum* method; the solid line is the reference line (y=x)

3.5 Discussion

3.5.1 Comparison of calculating FFE with different height extraction methods

Table 3-3 shows the difference between the statistical parameters of ground truth, *Point*, *Mean*, *Minimum*, and *Maximum* methods for the Virginia region. The statistical parameters include Min, Max, Median, interquartile distance (IQR), First Quartile, and Third Quartile. In terms of Min and IQR difference, the *Point* and *Mean* methods showed higher consistency to the ground truth than the *Minimum* and *Maximum* methods, with values of 12.57cm and 9.65cm, for Min and 16.2cm and 18cm for IQR, respectively. In terms of the other statistical indicators, including the Median, First Quartile, and Third Quartile, the *Minimum* method indicated the highest consistency with the difference of 14cm, 21cm, and 1.6cm, respectively. However, the difference in Min and Max for this method was about 48cm and 1.9m lower than the ground truth data. The high discrepancy in Min and Max values for this method shows a biased estimation and indicates that this method tends to underestimate FFE. Based on Tables 3-2 and 3-3, the most accurate result was for the *Point* and *Mean* methods. Figure 3-12, which shows the box plots for the ground truth, *Point*, *Mean*, *Minimum*, and *Maximum* methods, also confirms the superiority of the *Point* and *Mean* methods. In terms of comparison between *Point* and *Mean*, the analysis should be conducted on a more extensive test sample to make a reliable decision. However, the lack of access to building footprint data for all the addresses and the lack of good quality GSV images captured from the front view of the buildings at a suitable distance are some of the limitations to further examining the reliability of the proposed algorithm on more extensive test data.

Table 3-3: FFE difference statistics (St_{GT}-St_{method}; St and GT are acronyms for Statistics and Ground Truth) between *Point*, *Mean*, *Minimum*, and *Maximum* methods and ground truth data distribution (values are in meters)

	Point	Mean	Minimum	Maximum
Min	0.1257	0.0965	0.4852	-0.1962
Max	0.3699	0.5050	1.907	0.0217
Median	-0.2534	-0.2501	-0.1413	-0.3567
IQR	0.162	0.1799	0.2285	0.2299
First Quartile	-0.3991	-0.4162	-0.2116	-0.5493
Third Quartile	-0.2361	-0.2363	0.0169	-0.3193

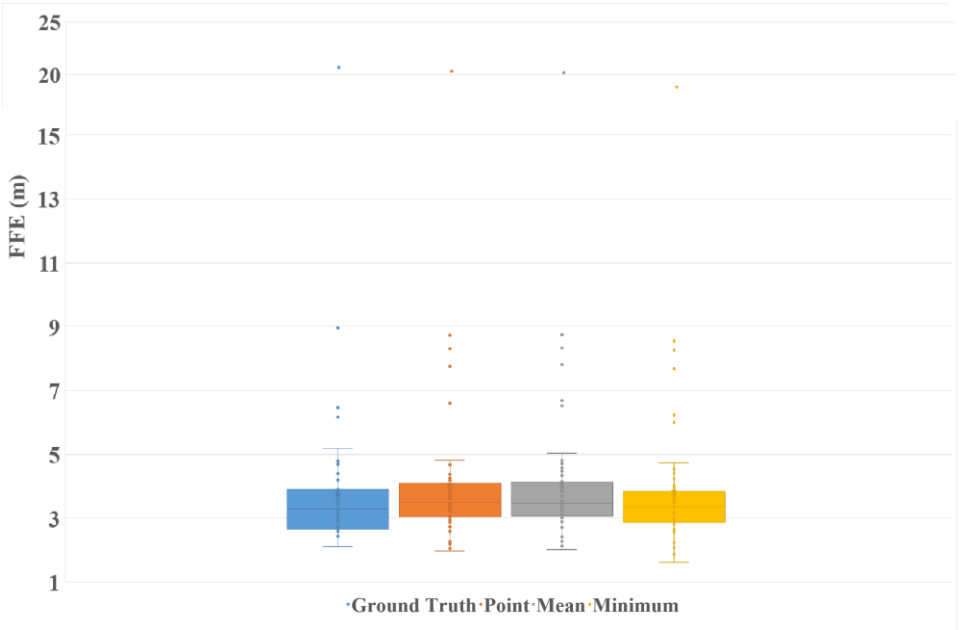
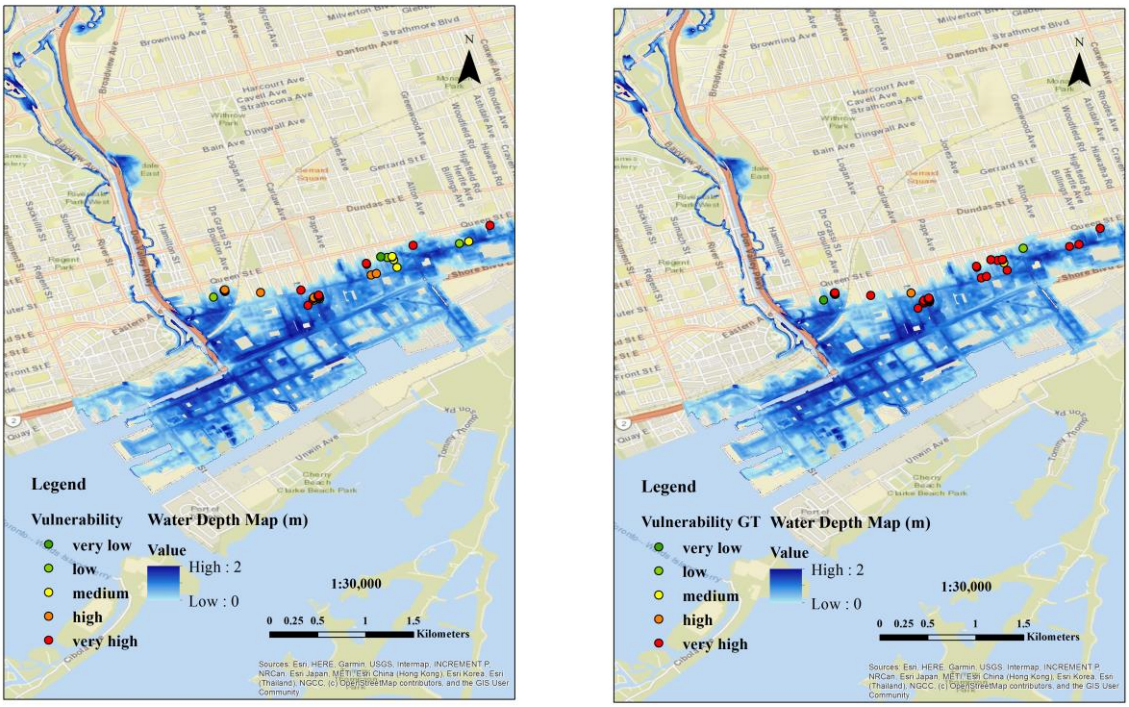


Figure 3-12: FFE box plots for ground truth, Point, Mean, Minimum and Maximum for Virginia

3.5.2 Flood vulnerability prediction

Figures 3-13a and 3-13b illustrate the predicted flood vulnerability and the reference, calculated based on TRCA-derived water depth grid and FFH values, for selected buildings across the Lower Don region. The water depth grid was simulated by the TRCA based on the flows from the Hurricane Hazel event in 1954. The software used for the northern part of the Lower Don region was a 1D Hydrological Engineering Center River Analysis System (HEC RAS) model. The modeling for this area was based on the GTA 2015 LiDAR collected DEM and was finalized in March 2020. For the other parts, a MIKE flood 2D model based on the GTA 2013 LiDAR collected DEM was used, and the modeling was finalized in February 2021 (Todd, personal communication, 2024). Many buildings on the reference map were classified as very highly flood-vulnerable because of the basement. However, they were categorized into the lower vulnerability classes because of the failure of the DL algorithm to detect the basement windows. The algorithm failed because there was no basement window in the GSV image (from the front view), and in some cases, the basement windows were blocked by the bushes in front of the buildings. The failure of the DL algorithm to detect the invisible basements in the front view GSV image is one of the limitations of the imagery-based analysis for flood vulnerability estimation.



(a) **(b)**

Figure 3-13: Flood vulnerability prediction for Lower Don region overlaid on the streets map from ArcGIS ; a) predicted flood vulnerability for selected buildings overlaid on the water depth map; b) reference flood vulnerability (calculated based on TRCA-derived flood depth grid and FFH values) for selected buildings overlaid on the water depth map.

3.6 Conclusion

This study employed imagery-based techniques and a DL-based object detection algorithm, YOLOV5s, to estimate FFH in both GTA and Virginia, US. Additionally, the basement windows were detected, and the buildings with a basement were marked. Subsequently, flood vulnerability was predicted for the Lower Don region, based on a water depth grid map accessed via the TRCA, building-level FFH values, and basement information derived from basement window existence. The study also conducted an

uncertainty analysis of FFH, showing that potential errors in FD and stairs/building extent bounding box coordinates contribute to FFH uncertainty. Although we assumed no camera rotation, this simplified assumption may lead to FFH overestimation or underestimation due to potential distortions in proportions of FD, stairs, and building extent. Therefore, accounting for image rotation should be considered in future work. Comparative analysis with a DL-based algorithm developed for Virginia demonstrated the superiority of the proposed method in terms of RMSE, R^2, and Bias values on the test data. The RMSE and Bias values reported for the proposed method were approximately 0.26-1.9m and 5-72cm smaller, respectively than the previous algorithm. The R^2 index was also 0.39-0.42 higher when using the *Mean* and *Min* methods for FFE estimation. Besides, comparing imagery-derived FFH with ground truth data, the proposed approach demonstrated promising results with an RMSE below 96cm and a Bias value below 55cm for both the GTA and Virginia. This makes the proposed approach suitable for applying to flood risk models and directing urban planning efforts to mitigate flood-related damages. However, it is worth noting that the basement detection using imagery-based techniques has limitations as basements are typically not visible from the front side of the building. Future research in flood vulnerability analysis using image-based techniques should explore coupling imagery-based analysis with ground survey data for more accurate basement detection.

3.7 Supplementary information

The code for the YOLOv5 model (available on GitHub) was used to train the model on the custom data using GoogleColab. Before training the model, labeled data are needed to train the object detection algorithm. The target objects in this work were the FD and stairs/building extent and these objects were critical elements in estimating the FFH parameter using the proposed method. Besides, the BW (if visible in the GSV image) were detected to identify the houses with a basement. The basement information is of interest for the flood vulnerability analysis. The assumption was that the existence of BW could be an indicator of a basement. After removing images in which stairs and FD were blocked by cars or trees, the remaining images were imported to the makesense.ai website, and the boundaries of the FD and stairs were delineated by drawing polygons. Then, the labeled

images were exported to YOLO format. This format contains one text file per image (with coordinates of the bounding boxes drawn on the image and a numeric/alphabetic label representation). The annotated data were split into train, validation, and test using the Roboflow framework, which is a computer vision framework for better data collection, preprocessing, and model training techniques and can easily read and write YOLO files.

References

Al Assi, A., Mostafiz, R. B., Friedland, C. J., Rahim, M. A., & Rohli, R. V. (2023). Flood risk assessment for residences at the neighborhood scale by owner/occupant type and first-floor height. *Frontiers in big Data*, *5*, 997447.

Al Assi, A., Mostafiz, R. B., Friedland, C. J., Rohli, R. V., & Rahim, M. A. (2023). Homeowner flood risk and risk reduction from home elevation between the limits of the 100-and 500-year floodplains. *Frontiers in Earth Science*, *11*, 1051546.

Brownlee, J. (2019). A Gentle Introduction to Object Recognition With Deep Learning. Machine Learning Mastery. https://machinelearningmastery.com/object-recognition-with-deep-learning/

Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Diaz, N. D., Highfield, W. E., Brody, S. D., & Fortenberry, B. R. (2022). Deriving First Floor Elevations within Residential Communities Located in Galveston Using UAS Based Data. *Drones*, *6*(4), 81.

Du W., Smith F. D., Brown B. T., 2020, US, SYSTEM, Computer Program Product and Method For Using a Convolutional Neural Network to Auto-Determine a Floor Height and Floor Height Elevation of a Building, 20200348132.

FEMA. (1999). *Appendix B - Glossary of Terms*. https://www.fema.gov/pdf/fima/pbuffd_appendix_b.pdf

Gao, G., Ye, X., Li, S., Huang, X., Ning, H., Retchless, D., & Li, Z. (2023). Exploring flood mitigation governance by estimating first-floor elevation via deep learning and google street view in coastal Texas. *Environment and Planning B: Urban Analytics and City Science*, 23998083231175681.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 580–587.

Gnan, E., Friedland, C. J., Rahim, M. A., Mostafiz, R. B., Rohli, R. V., Orooji, F., ... & McElwee, J. (2022). Improved building-specific flood risk assessment and implications of depth-damage function selection. *Frontiers in Water*, *4*, 919726.

Gordon, A., & McFarlane, B. (2019). Developing First Floor Elevation Data for Coastal Resilience Planning in Hampton Roads. Hampton Roads Planning District Commission.
https://d1wqtxts1xzle7.cloudfront.net/96351796/07A_Attachment_Developing_First_Flo

or_Elevation_Data_for_Coastal_Resilience_Planning_in_Hampton_Roads_HRPDC-libre.pdf

Hammond, M., Chen, A., Djordjević, S., Butler, D., & Mark, O. (2013). Urban flood impact assessment: A state-of-the-art review. *Urban Water Journal, 12*(1), 14–29.

Hampton Roads Planning District Commissions. (2020, February). *Applying First Floor Elevation Data to Flooding Vulnerability Assessments in Hampton Roads*.

https://www.hrpdcva.gov/uploads/docs/05A_Attachment_ApplyingFirstFloorElevationDatatoFloodingVu lnerabilityAssessmentsinHamptonRoads_CRC_26Jun2020.pdf

Jocher, G. (2020, May). YOLOv5. https://github.com/ultralytics/yolov5

Leal, M., Reis, E., Pereira, S., & Santos, P. P. (2021). Physical vulnerability assessment to flash floods using an indicator-based methodology based on building properties and flow parameters. *Journal of Flood Risk Management*, *14*(3), e12712.

Mathew, M.P. and Mahesh, T.Y., 2022. Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing*, *16*(3), pp.841-847.

Milanesi, L., Pilotti, M., Belleri, A., Marini, A., & Fuchs, S. (2018). Vulnerability to flash floods: a simplified structural model for masonry buildings. *Water Resources Research*, *54*(10), 7177-7197.

Ning, H., Li, Z., Ye, X., Wang, S., Wang, W., & Huang, X. (2021). Exploring the vertical dimension of street view image based on deep learning: a case study on lowest floor elevation estimation. *International Journal of Geographical Information Science*, 1-26.

Ogundare, J. O. (2018). *Understanding Least Squares Estimation and Geomatics Data Analysis*. John Wiley & Sons.

Paulik, R., Wild, A., Zorn, C., & Wotherspoon, L. (2022). Residential building flood damage: Insights on processes and implications for risk assessments. *Journal of Flood Risk Management*, *15*(4), e12832.

Percival, S., Gaterell, M., & Teeuw, R. (2019). Urban neighbourhood flood vulnerability and risk assessments at different diurnal levels. *Journal of Flood Risk Management*, *12*(3), e12466.

Rahim, M. A., Mostafiz, R. B., & Friedland, C. (2023). Disseminating Flood Risk Information in the USA through Risk Rating 2.0. *EGU23*, (EGU23-16893).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Taghinezhad, A., Friedland, C. J., Rohli, R. V., & Marx, B. D. (2020). An imputation of first-floor elevation data for the avoided loss analysis of flood-mitigated single-family homes in Louisiana, United States. *Frontiers in Built Environment*, *6*, 138.

Wang, Y., & Sebastian, A. (2021). Community flood vulnerability and risk assessment: An empirical predictive modeling approach. *Journal of Flood Risk Management*, *14*(3), e12739.

Wu, W., Liu, H., Li, L., Long, Y., Wang, X., Wang, Z., Li, J. and Chang, Y., 2021. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PloS One*, *16*(10), p.e0259283.

Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G. and Wang, W., 2020, December. Face mask recognition system with YOLOV5 based on image recognition. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)* (pp. 1398-1404). IEEE.

Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J. and Li, X., 2021. A real-time detection algorithm for Kiwifruit defects based on YOLOv5. *Electronics*, *10*(14), p.1711.

Zhao, J., Zhang, X., Yan, J., Qiu, X., Yao, X., Tian, Y., Zhu, Y. and Cao, W., 2021. A wheat spike detection method in UAV images based on improved YOLOv5. *Remote Sensing*, *13*(16), p.3095.

Zhou, F., Zhao, H. and Nie, Z., 2021, January. Safety helmet detection based on YOLOv5. In *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)* (pp. 6-11). IEEE.

Chapter 4

Building detection using a Dense Attention Network from LiDAR and image data

4.1 Introduction

Accurate building detection in urban areas is difficult due to nearby objects, such as trees, which can have similar height as buildings (Nahhas et al., 2018). LiDAR data have been widely used in previous studies for building detection because of their potential for extracting geometric features. The advantages of LiDAR data for building detection are: 1- collecting high-density point clouds over a relatively short time; 2- high vertical accuracy. Remote sensing and aerial images can provide spectral information that can help discriminate between trees and buildings. Therefore, considering both LiDAR point clouds and image data can be an essential step toward improving building detection quality.

Building extraction methods using remote sensing data can be divided into two categories: 1- stereoscopic methods, which use image matching techniques to develop 3D reconstruction of buildings; 2- machine learning algorithms, aka Artificial Intelligence techniques, which are based on automatic learning of building characteristics from various data sources like spectral and geometric features extracted from the spatial dataset. The first category is not automatic and can be labour intensive. Machine learning methods, however, are robust against artifacts inherent in images like shadows, occlusion, and against the surrounding environmental variations if they use sufficient training data and suitable parameters.

Among machine learning methods, Deep Learning (DL) algorithms such as Convolutional Neural Networks (CNN) allow for automatic extraction of the relevant features without requiring a separate stage for feature extraction and selection, contrary to conventional machine learning techniques such as Support Vector Machine (SVM) or neural networks. Maltezos et al. (2017) applied CNN and image matching techniques to detect building extents using normalized Digital Surface Model (nDSM) and orthoimages. The results

showed that the combination of image and height information can provide robust and efficient results. Maltezos et al. (2018) used CNN for building extent detection, based on the LiDAR data and LiDAR-derived geometrical features, including height variation, entropy, intensity, and normal vectors distribution. They concluded that higher accuracy can be achieved compared with using LiDAR and aerial images simultaneously. Pirasteh et al. (2019) used multi-temporal LiDAR data to extract building boundaries and conduct change detection analysis. Firefly and ant colony algorithms were applied for building extent extraction. The results were compared with the Mask RCNN DL method. They reported that comparable results could be achieved after using LiDAR data. Nahhas et al. (2018) applied autoencoders for feature fusion and reduction using LiDAR and orthoimages and then converted the low-level features to high-level using CNN. The maximum accuracy obtained was 86.19%, which was higher than the SVM. The authors stated that higher accuracy would be achieved if the hyperparameters were set in the DL model. Zhou and Gong (2018) imported a 2D height map derived from 3D LiDAR point cloud data to the VGG network. The problem with this study was that its accuracy was dependent on the edge-aware clustering technique proposed in the post-processing stage. Huang et al. (2019) used Gated Residual Refinement Network (GRNN) to extract building extents using high-resolution orthoimage with a resolution of 0.15-0.3m and LiDAR data. The advantage of using GRNN was considering features at different scales simultaneously for building boundary extraction. This method obtained comparable results to other versions of GRNN. Zhao et al. (2018) imported pan-sharped orthoimages to Mask RCNN network to extract building boundaries. The authors regularized building boundaries using the Douglas-Pauker algorithm and MDL optimization. In another work, 1m resolution images with Red (R), Green (G), Blue (B), and Infrared (IR) bands were used for building boundary detection. Two SegNets (a type of DL algorithm) were applied; one pre-trained with RGB bands, and another was trained after adding IR bands. After using the IR band and a sign-distance labeling technique, the authors concluded that better results could be achieved compared with state-of-the-art methods (Yang et al., 2018).

Accurate flood damage assessment in urban areas requires a detailed inventory of building footprints. DL algorithms can accurately detect building footprints because of their complicated architecture. However, these models face the vanishing gradient problem

because of their significant number of layers. One strategy to alleviate this issue is to apply skip connections in their architectures. It helps recover information from previous layers, improving the training process and keeping the information flow. Hence, a CNN based on dense attention blocks and skip connections was developed in this chapter to detect building footprints using MS and LiDAR data in Toronto and the widely known Massachusetts Building Dataset.

4.2 Case Studies and Datasets

4.2.1 Toronto Case Study

The first case study was in Toronto, Ontario, Canada, which has a mixture of high and low buildings. Two Tiles, one located near Riverdale Park East and the other near the East Chinatown area, were selected for analysis. The selected Tiles were near Don River and included different macro features, such as vegetation, trees, river, tall and short buildings, which make the building detection challenging in this area. Figure 4-1 shows the case study location on the map.

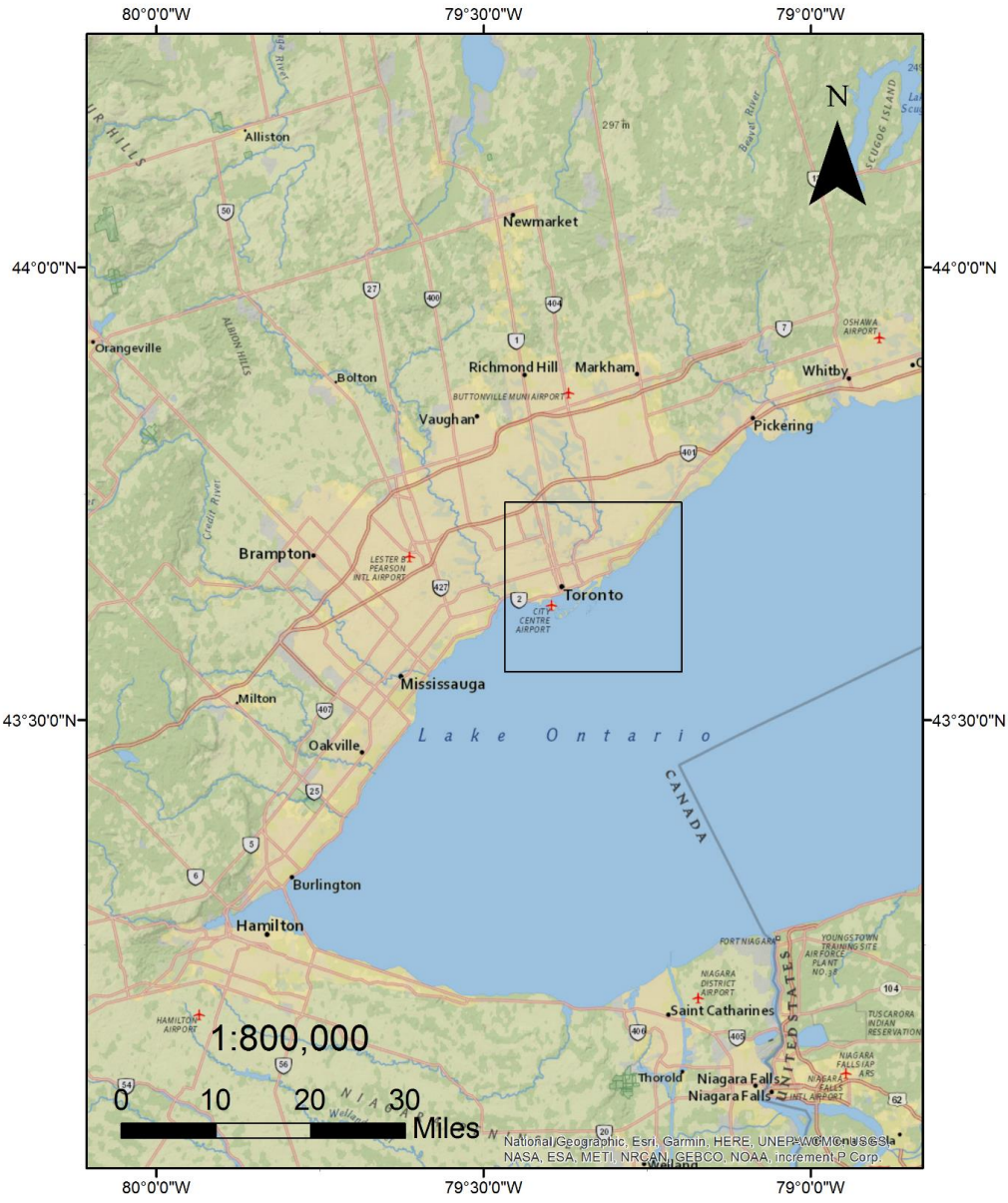


Figure 4-1: Toronto Case Study area (outlined in black).

The dataset used in this study included LiDAR and RapidEye analytic orthoimages. LiDAR data were acquired in 2015 covering an area of about 1km^2. LiDAR data point density for the last pulse was about 9.32 points/m^2 and 11.09 for the other LiDAR returns. Also, the

data had a sampling distance of about 0.3 m. The Rapid Eye images were acquired in the same year as LiDAR data and included four spectral bands, RGB, Red edge and NIR. The corresponding spatial resolution was about 5m. Tables 4-1 and 4-2 describe the dataset characteristics including data level, acquisition date, and spatial resolution. The DTM and DSM data are available for the case study through the Ontario Geo hub website.

Table 4-1: RapidEye technical characteristics

Name	Level	Acquisition Data	Resolution	Bands
Rapid Eye Analytic Ortho Tile	3A	23/09/2015	5 m	Blue (440-510
				Green (520-590
				Red (630-685
				Red edge (690-730
				NIR (760-850

Table 4-2: LiDAR data technical characteristics

Name	Year	Covered area (each tile)	Point density	spacing
1km176320483502015LGTA2015_CPC	2015	0.98 km^2	All returns 11.09 points/m^2, last only 9.32 points/m^2	All returns 0.3m, last only 0.33m
1km176320483602015LGTA2015_CPC				

4.2.2 Massachusetts Case Study

The Massachusetts building dataset includes 151 aerial images of the city of Boston, Massachusetts, USA. Each original image is 1500 × 1500 pixels for an area of 2.25 km^2, and the entire dataset covers approximately 340 km^2. The dataset consists of urban and suburban areas and buildings of all sizes, including individual houses and garages. Figure 4-2 illustrates three representative regions from the Massachusetts building dataset (Hinton & Mnih, 2013). The dataset used in this study was 512 × 512 pixels with a 1m spatial resolution obtained after cropping and resampling the original 1500 × 1500 pixel images. These data are the same dataset used for building detection in He et al. (2022) and contain 1065, 36, and 90 labelled images for training, validation, and testing, respectively.

Figure 4-2 (a-c): Three representative areas from the Massachusetts Building Dataset.

4.3 Methods

4.3.1 Train and Test Data Selection

Ground truth data were selected using Google Earth images. The data were sampled from building areas as class *building* and the river, vegetated, and ground areas as class *other*. Figure 4-3 shows the spatial distribution of ground truth data. The selected data spread all over the region. 70% of the data were selected randomly as train, and the remainder were selected as test for accuracy assessment. Also, 20% of the training data were considered for validation during training the network.

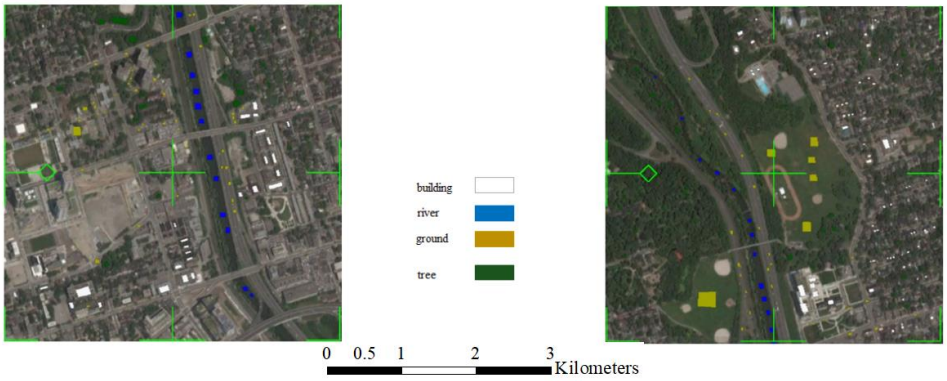


Figure 4-3: Spatial distribution of the ground truth data for Toronto case study

Homogenous polygonal areas were selected from buildings, vegetated areas, roads, and rivers using Google Earth Pro software. Homogenous here means that all the selected pixels inside a polygon come from the same class. This sampling strategy is different from some existing methods that sample the whole building extent and its surrounding. The number of train, test and validation data would be different based on the designated input shape in the CNN architecture; for example, if the input shape is (9×9), the square areas with the same shape were selected inside the original regions. Tables 4-3 and 4-4 show the number of the train, test, and validation data for Toronto and Massachusetts case studies, respectively.

Table 4-3: Number of train, test, and validation samples for Toronto case study

	Number of samples (in pixel)			
	Building		Background	
	First Tile	Second Tile	First Tile	Second Tile
Train data	4681	2554	7281	11744
Test data	2073	1122	3054	5007
Validation data	20% of the train data was set randomly as validation for adjusting the CNN network parameters during training.			

Table 4-4: Number of train, test, and validation samples for Massachusetts case study

	Number of samples (in pixel)
Train data	1065
Test data	90
Validation data	36

For Massachusetts Building Dataset, the data had already been divided into train, validation, and test data. So, it was not necessary to do further processing for data splitting.

4.3.2 Input Features

Input features to the network contained two categories; 1- LiDAR-derived features; 2- Spectral features. LiDAR-derived features included DSM, DTM, nDSM, hillshade, curvature, and slope. nDSM is the difference between DSM and DTM, which represents the ground features elevation. The spectral features included RapidEye analytical ortho tile spectral bands. LiDAR-derived DSM and DTM, as well as LiDAR data, were downloaded from the Ontario GeoHub website. Radiometric calibration and atmospheric correction of the RapidEye images were conducted using the ENVI software to convert digital numbers to ground surface reflectance. FLAASH atmospheric correction module in ENVI was used for atmospheric correction. Then, RapidEye bands were resampled to LiDAR-derived features' size for spatial resolution consistency.

4.3.3 Dense Attention Learning

Conventional CNN use the bottom-up approach to extract the image features. These networks focus on specific areas of the image and gradually convert low-level features to high-level features. CNN can extract multi-scale features when they become deeper, but one issue related to a deep CNN is that it becomes difficult for the succeeding layers to retrieve information from previous layers as the number of layers increases. This problem has been referred to as the vanishing gradient problem, and some recent studies have been conducted to alleviate this issue. The first strategy to tackle the vanishing gradient problem was residual networks, like ResNet50, and the second solution was introducing DANs (Zhang et al., 2020).

Suppose we have a convolutional feature map, *X*, extracted by convolutional layers, and *H*, a composite mapping function created by a dense attention block. The mapping function *H* can be a series of Batch Normalization (BN), convolutional, and pooling layers. The proposed mapping function in this study is formed by combining BN + Conv2D + drop out + average pooling + BN + Conv2D. Conv2D is the 2D convolution. The dense attention mapping can be formulated as Equation (4-1):

$$\{w_{ij}\} = H(X) \tag{4-1}$$

In Equation (1), w_{ij} is the weight value for the feature map at position (i,j). The weight values are computed during training, and based on these weight values, the mapping function *H* is computed and is convolved on the convolutional feature map as in Equation (4-2):

$$X'' = H(X) \otimes X \tag{4-2}$$

In Equation (4-2), $\otimes$ refers to the element-wise product, and X'' is the output feature map from the dense block.

As the extracted discriminative feature map via a dense attention block might interrupt the usual convolutional feature extraction process, a concatenation of all or some of the previous feature maps can retrieve the feature extraction process. Concatenation refers to the stacking of the outputs from several previous layers and passing them to the next layer. DANs fuse convolutional features, extracted by usual convolutional layers and discriminative features extracted by dense attention blocks to ease feature propagation through the layers. This fusion is achieved using concatenation layers embedded in the network and can be formulated as shown in Equation (4-3). In this equation, the subscripts refer to layer number; for example, X_l is the feature map in the *l*th layer, *[]* is the concatenation sign, and X''' is the output feature map from concatenation layer.

$$X''' = [X_a\ , H(\ X_a) \otimes X_a\ , X_b \otimes X_b, \dots\ , X_l\ , X_l \otimes H(X_l)] \tag{4-3}$$

4.3.4 Proposed Dense Attention Network Inputs and Parameters

The input data to the proposed network was LiDAR derived features and Rapid Eye spectral bands, and the training parameters included learning rate, batch size, the number of epochs, loss function, and optimizer. The assigned values for these parameters were reported in Table 4-5.

Table 4-5: Proposed Dense Attention Network (DAN) parameters

Parameter	Value
Learning rate	10^{-4}
Batch size	20
Number of epochs	400
Loss function	Binary cross entropy
Optimizer	Adam

These parameters were set by trial and error because it is more computationally efficient than using optimization algorithms like Genetic Algorithm applied in previous studies. The loss function was binary cross-entropy because the problem is a binary classification. The learning rate, and batch size were adjusted to 10^{-4}, and 20, respectively. The Adam optimizer function was considered during training, and the number of epochs was set to 400, but an early stopping condition was considered for faster training. This condition was set so that if the loss function on validation data had a tolerance not greater than 10^{-3} during 50 epochs, the training would be stopped.

4.3.5 Proposed Dense Attention Network Architecture

The proposed architecture was inspired by a DL method called DAN applied previously for building detection, but our method does not use an encoder-decoder structure to extract feature maps (Yang et al., 2018). It uses two dense attention blocks embedded in between

layers to convert low-level features to high-level features. Figure 4-4 shows the proposed architecture. The DAN block refers to a cascade of batch normalization, 2-D convolution, drop out, and average pooling layers in this figure. The last two dense layers assign either the label value of zero (*other*) or one (*building*) to the feature maps extracted by previous layers. The network patch (window) size was set to 9 because of the tradeoff between processing time and meaningful information content. Image patch size smaller than nine did not result in optimal classification because the small patch size might fail to include meaningful information for building detection.

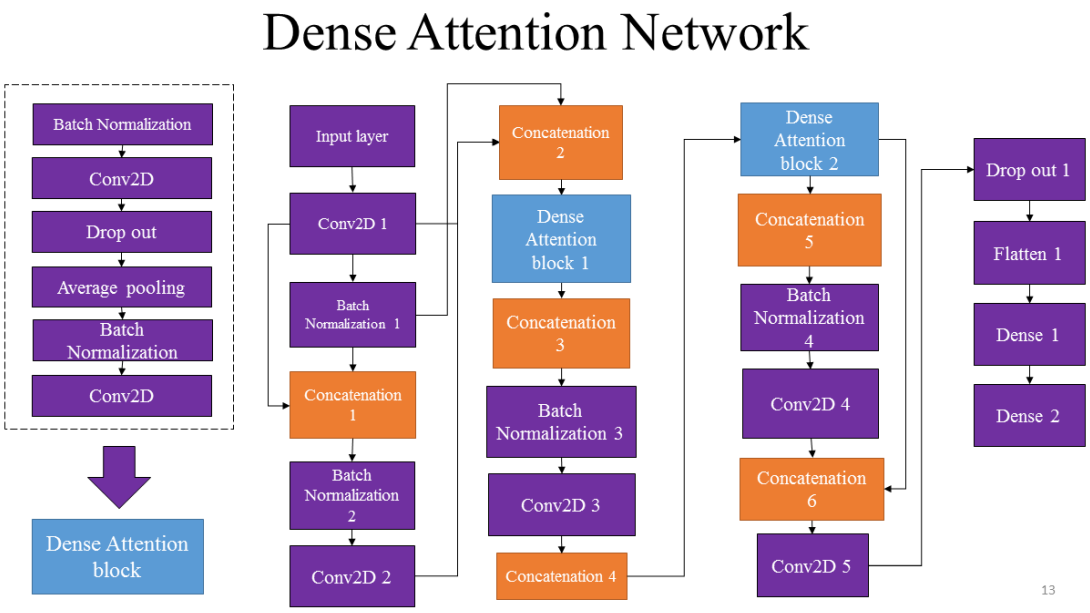


Figure 4-4: Proposed CNN architecture

Our method selects the samples at the polygon level, and the building detection stage labels the whole image scene at the pixel level by dividing the scene into overlapping areas with the same size as the input shape. After extracting the relevant features during the feature extraction stage, the FC layers, embedded on the top of the CNN network, assign a label to the central pixel inside the initially selected area. Figure 4-5 shows the flowchart of the study.

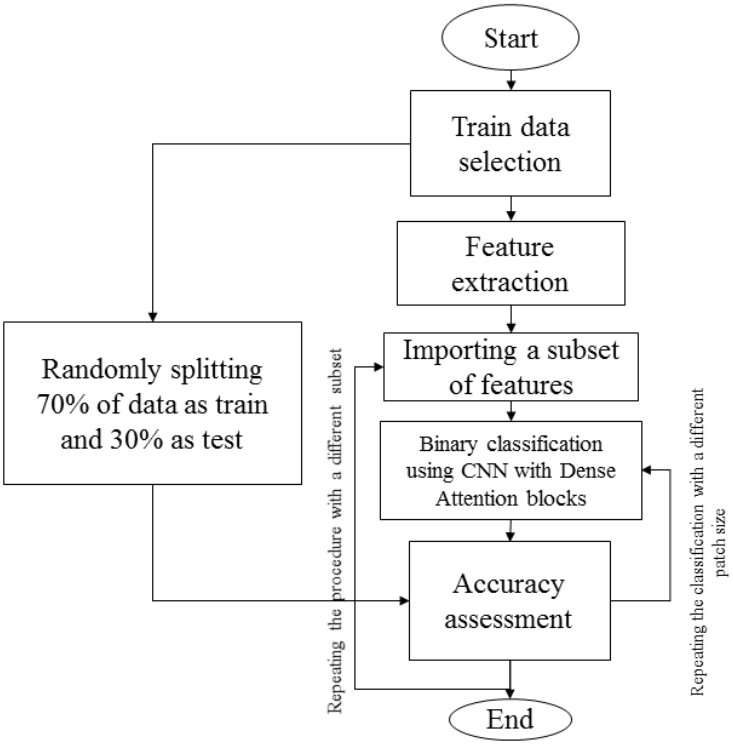


Figure 4-5: Flowchart of the study

4.3.6 Model Training

Figures 4-6 and 4-7 show the loss (cost) function and classification accuracy diagrams for training and validation data. Training stopped after about 140 epochs when the loss function and accuracy values for both training and validation data reached the same value. The train and validation accuracy reached to 100% at the last epochs.

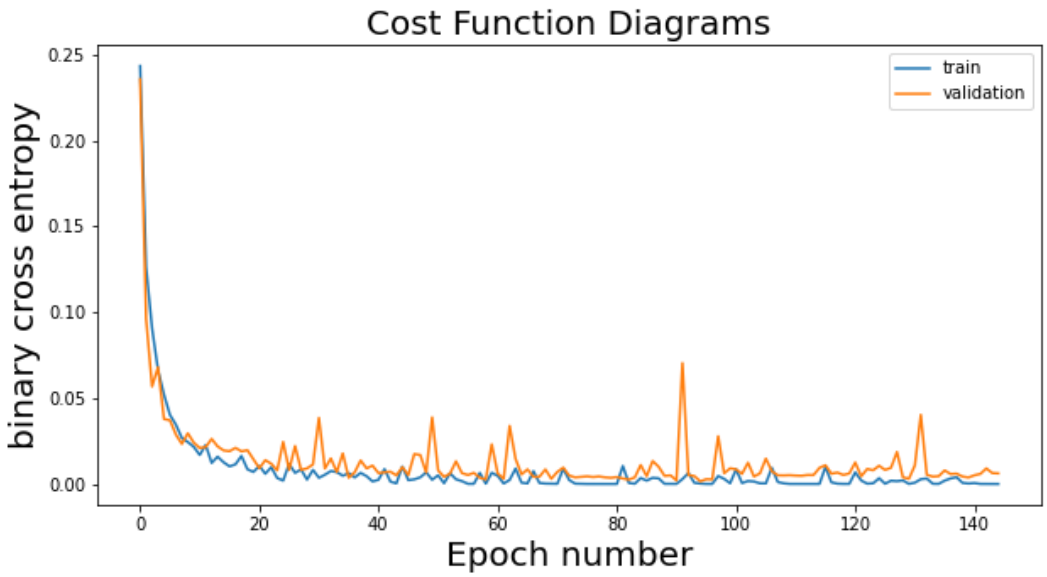


Figure 4-6: Loss function during training for train and validation data

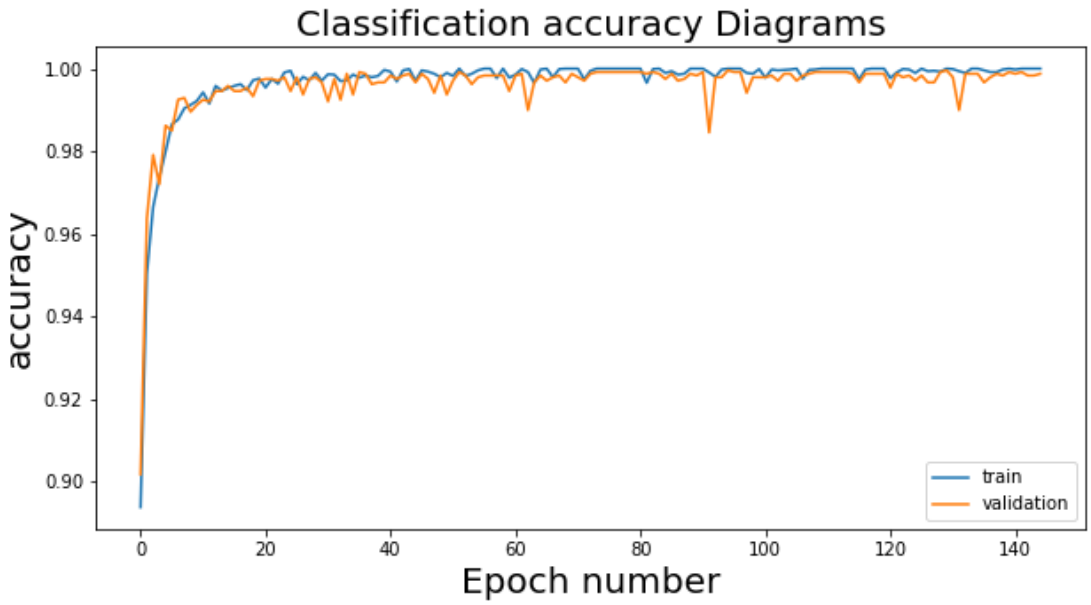


Figure 4-7: Classification accuracy during training for train and validation data

4.4 Results

This section analyzes the effects of changing the patch size and input features on building detection accuracy.

4.4.1 CNN results with different patch sizes

This section examines the proposed CNN results regarding classification accuracy indices and classification maps after changing the patch size parameter and keeping the same input features. Support, in Table 4-6, refers to the number of samples in each class and as a whole. The number of samples in each class seemed balanced because the class *other* constituted a larger part of the scene than the class *building*. All patch sizes acquired full accuracy on test data except for patch size 5. Accuracy indices for building pixels, including precision, recall, and F1 Score, dropped by 4%, 2%, and 3%, respectively, after changing the patch size parameter to 5 because larger patch sizes are more informative and contain more features than smaller patch sizes (Hamwood et al., 2018). The patch size 9 was selected for the network based on the outcome accuracy indices.

Table 4-6: CNN accuracy indices with different patch sizes (Tile 1)

Patch size	Class	Precision	Recall	F1 Score	Support
5	Other	0.99	0.98	0.98	5171
	building	0.96	0.98	0.97	3539
	accuracy			0.98	8710
7	other	1	1	1	3923
	building	1	1	1	2795
	accuracy	1	1	1	6718
9	other	1	1	1	3054
	building	1	1	1	2073
	accuracy			1	5127
11	other	1	1	1	2391
	building	1	1	1	1541
	accuracy	1	1	1	3932

Table 4-7 shows the same accuracy indices in Table 4-6 for the second Tile. The results for the second Tile are consistent with the first Tile. The accuracy indices were reduced by 5%, 6%, and 5% after changing the patch size to 5, and full accuracy parameters were achieved for the class *building* after changing the patch size to 7 and remained the same when testing larger patch sizes of 9 and 11. Table 4-8 shows IoU indices. The computed

values confirm the results obtained in Tables 4-6 and 4-7. It can be seen that the least IoU values were for patch size 5, with almost 0.94 and 0.89 for the first and second Tiles, respectively. Higher IoU values were achieved after changing the patch size to 7, 9, and 11. Looking at the support column in Tables 4-6 and 4-7, changing the patch size resulted in a change in the sample numbers in both classes, and the network building detection accuracy improved after decreasing the number of samples. In this case, a greater patch size means decreasing the number of samples because samples are selected from an area with a fixed size. This result implies that building detection accuracy is more dependent on the training data patch size than the number of samples because the sample patch size controls the information content transferring between CNN layers. Overall, it was expected that more training data would result in higher accuracy. Here, there was a reverse relationship between the number of samples and the patch size. It means that having a greater patch size would result in fewer training samples. A larger patch size caused more information content to be imported to the network and resulted in higher classification accuracy. It should be noted that while a small patch size can cause the loss of information content, a relatively large patch size can also result in inclusion of more than one object in patch size, however assigning a single classification label to more than one object is not correct.

Table 4-7: CNN accuracy indices with different patch sizes (Tile 2)

Patch size	class	Precision	Recall	F1 Score	Support
5	other	0.98	0.99	0.98	7140
	building	0.95	0.94	0.95	2097
	accuracy			0.98	9237
7	other	1	1	1	5875
	building	1	1	1	1540
	accuracy	1	1	1	7415
9	other	1	1	1	5007
	building	1	1	1	1122
	accuracy			1	6129
11	other	1	1	1	4377
	building	1	1	1	805
	accuracy	1	1	1	5182

Table 4-8: IoU values for different patch sizes

Patch size	IoU (Tile 1)	IoU (Tile 2)
5	0.95	0.90
7	1	1
9	1	1
11	1	1

Receiver Operative Characteristics (ROC) is applied to evaluate the diagnostic ability of one or multiple classifiers. As the Area Under the ROC Curve (AUC) increases, the classifier performance improves. Figure 4-8 shows the ROC curves for the Tiles. Again, full AUC values were obtained for patch sizes 7, 9, and 11, and this index was reduced to 0.97 and 0.96 after training the network with patch size 5. Here, patch size can be related to the size of the buildings in the scene. While small patch sizes contain no relevant information about building geometry and spectral characteristics, larger patch sizes are more likely to import meaningful information to the Dense Attention blocks.

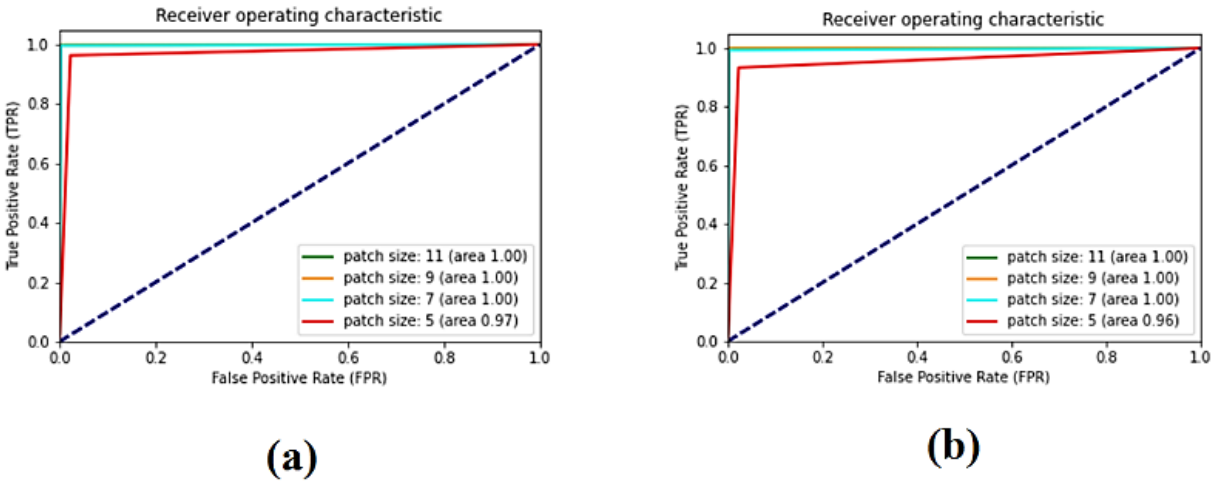


Figure 4-8: ROC curves with different patch sizes for first and second Tiles; dashed lines represent the ROC curves for random classifiers. a: first Tile; b: second Tile

4.4.2 CNN results with different input features

Tables 4-9, 4-10, and 4-11 show accuracy indices as well as IoU for the first and second Tiles, before adding hillshade, curvature, slope and spectral features, and after considering both LiDAR-derived and spectral features. Regardless of which kind of feature (LiDAR-derived or RapidEye spectral bands) was imported to the network, excellent result on the test data was achieved. This result verifies the effectiveness of the proposed Dense Attention Network (DAN) one more time.

Table 4-9: CNN accuracy indices with different features (Tile 1)

Features	class	Precision	Recall	F1 Score	Support
DTM, DSM, nDSM	other	1	1	1	3054
	building	1	1	1	2073
	accuracy			1	5127
DTM, DSM, nDSM, Hillshade	other	1	1	1	3054
	building	1	1	1	2073
	accuracy	1	1	1	5127
DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB	other	0.99	1	0.99	3054
	building	1	0.99	0.99	2073
	accuracy			0.99	5127
DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB, Red edge	other	1	1	1	3054
	building	1	1	1	2073
	accuracy	1	1	1	5127

Table 4-10: CNN accuracy indices with different features (Tile 2)

Features	class	Precision	Recall	F1 Score	Support
DTM, DSM, nDSM	other	1	1	1	5007
	building	1	1	1	1122
	accuracy			1	6129
DTM, DSM, nDSM, Hillshade	other	1	1	1	5007
	building	1	1	1	1122
	accuracy	1	1	1	6129
DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB	other	1	1	1	3054
	building	1	1	1	2073
	accuracy			1	5127
DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB, Red edge	other	1	1	1	3054
	building	1	1	1	2073
	accuracy	1	1	1	5127

Table 4-11: IoU values for different input features

Input Features	IoU (Tile 1)	IoU (Tile 2)	Input features	IoU (Tile 1)	IoU (Tile 2)
DTM, DSM, nDSM	1	1	DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB	0.98	1
DTM, DSM, nDSM, Hillshade	1	1	DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB, Red edge	1	1
DTM, DSM, nDSM, Hillshade, Curvature	1	1	DTM, DSM, nDSM, Hillshade, Curvature, Slope, RGB, Red edge, NIR	1	1
DTM, DSM, nDSM, Hillshade, Curvature, Slope	1	1			

Figure 4-9 illustrates the classification maps with different input features for the first Tile. There are some visual errors on the middle top and the bottom. Some high roads were labeled as building even after adding geometric LiDAR-derived features because although building and high roads edge pixels may have different slopes and curvature, these parameters are the same within the boundary area of both objects where the train data are

selected. Another reason for this commission error (erroneously classifying background pixels as buildings) would be integrating samples from rivers, roads, and vegetated regions as one class. This integration may cause difficulty in discriminating buildings from high roads due to high variance in class *other*, and the problem may be solved by considering a multi-class classification (Hamaguchi and Hikosaka, 2018). Multiclass classification was ignored because this study was focused on building detection. Test data are usually selected from pure pixels in the middle of an object, far from the edges. Therefore, these visual errors were not reflected in the accuracy indices.

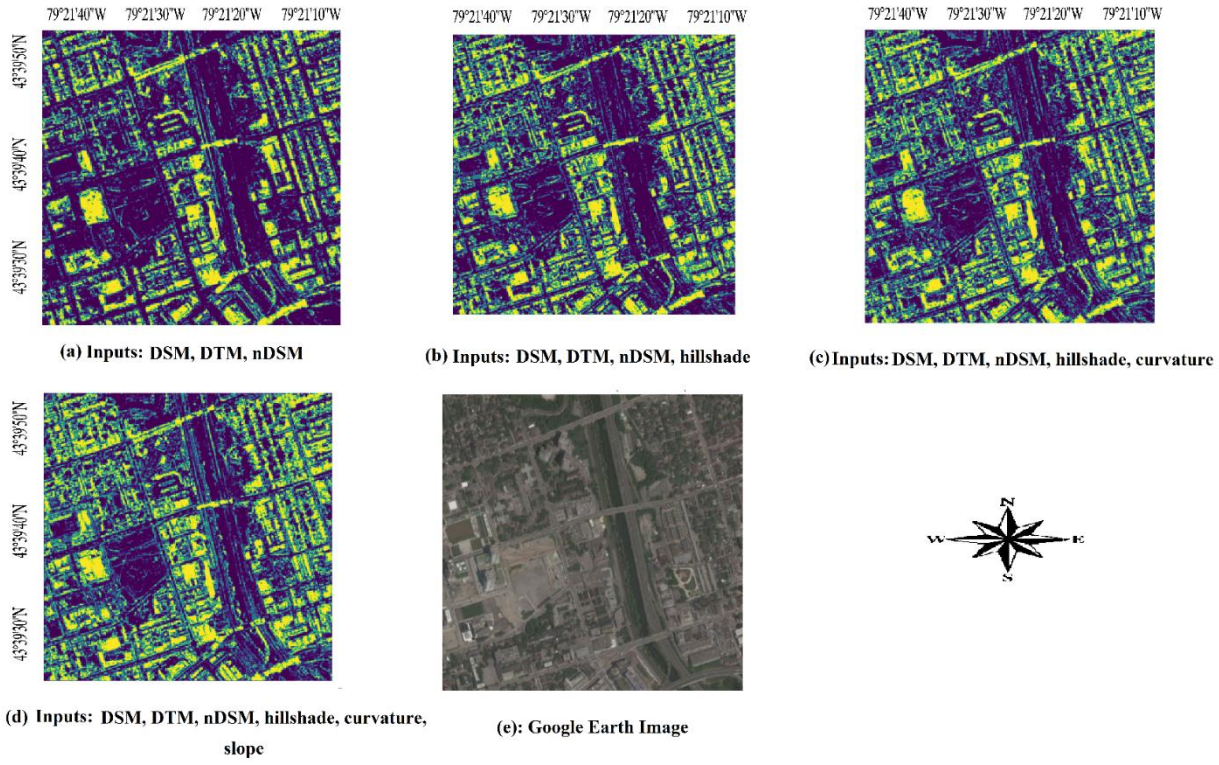


Figure 4-9: Building detection results with different input features (Tile 1); yellow areas show the detected building pixels. a: classification result with DSM, DTM, and nDSM ; b: DSM, DTM, nDSM and hillshade; c: DSM, DTM, nDSM, hillshade and curvature; d: DSM, DTM, nDSM, hillshade, curvature, and slope; e: Google Earth image

Figure 4-10 depicts the same result for the second Tile and also by adding RapidEye spectral features. Again, some visual errors exist on high roads, but another challenge here is the commission error of labeling the edge pixels between trees and roads as buildings. This kind of error may be solved by adding morphological operations as the post-processing step.

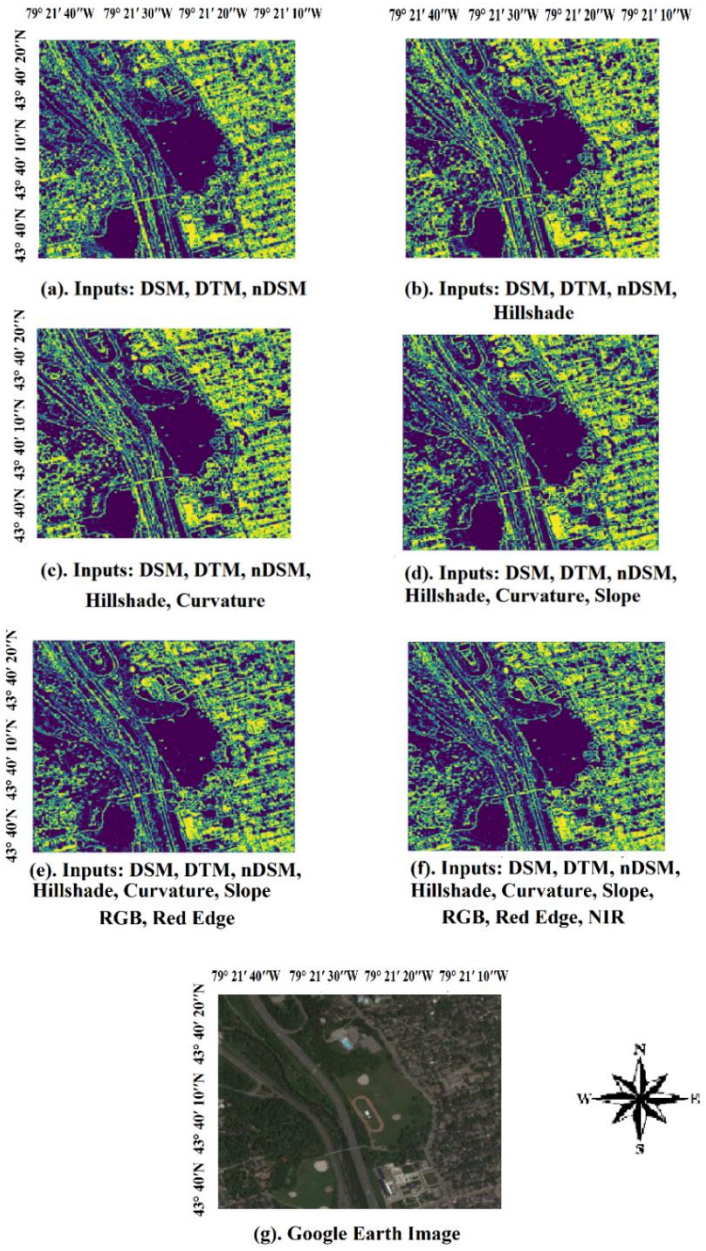


Figure 4-10: Building detection results with different input features (Tile 2); yellow areas show the detected building pixels. a: classification result with DSM, DTM, and nDSM ; b: DSM, DTM, nDSM and hillshade; c: DSM, DTM, nDSM, hillshade and curvature; d: DSM, DTM, nDSM, hillshade, curvature, slope, RGB, and Red edge; f: DSM, DTM, nDSM, hillshade, curvature, slope, RGB, Red Edge, and NIR; e: Google Earth image

4.5 Discussion

This section presents a comparison between some of the state-of-the-art DL techniques, including VGG16, ResNet50, Unet, ResUnet, and the proposed DAN network on two case studies, Toronto and Massachusetts Building Dataset. It also compares the building detection results of these DL methods with the building masks extracted from the Toronto Land Cover Map.

4.5.1 Comparison of building detection results with different patch sizes

Figure 4-11 shows the output building mask with different patch sizes, including 5, 7, 9, and 11 for the first Tile. As can be seen in the red square areas, some line objects in the road area are apparent in the building mask for smaller patch sizes, but these detailed features have been partially removed for larger patch sizes because the detailed objects such as edges are more likely to be captured by a smaller patch size than a larger one. As shown in Figure 4-12, the same situation can also be seen for the second Tile. Small vegetated areas in the vicinity of the river have been detected as buildings when masking the built-up areas with patch sizes 5 and 7, but these areas were gradually merged with the background by increasing the patch size.

Figure 4-11: Building masks for the first Tile with different patch sizes (the red square highlights a road edge that is more apparent in patch sizes 5 and 7 than 9 and 11) ; a: Patch Size 5; b: Patch Size 7; c: Patch Size 9; d: Patch Size 11; e: Google Earth image.

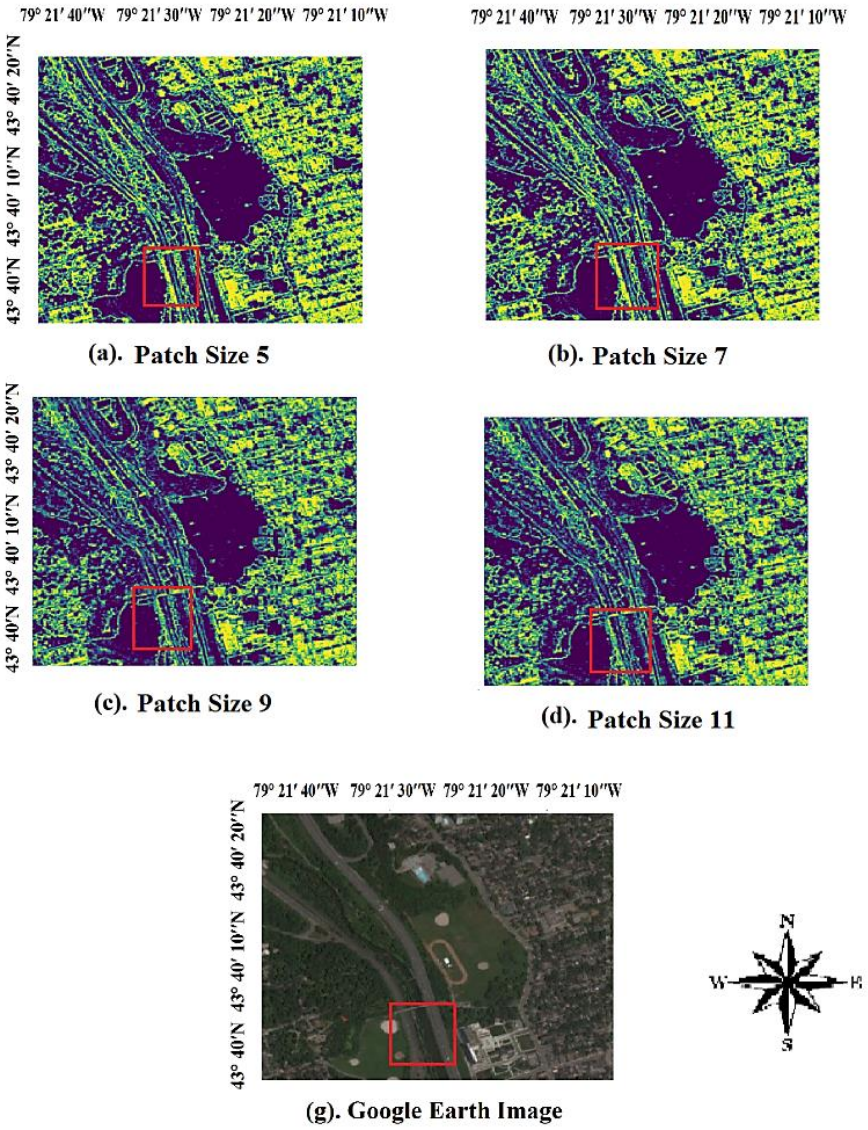


Figure 4-12: Building masks for the second Tile with different patch sizes (the red square shows a vegetated area in the vicinity of the river that has been detected as the building in patch sizes 5 and 7 but these areas are starting to merge with the background in patch sizes 9 and 11); a: Patch Size 5; b: Patch Size 5; c: Patch Size 9; d: Patch Size 11; e: Google Earth image

4.5.2 Comparison with other Deep Learning techniques

The proposed method was compared with two other well-known DL techniques, VGG16 and ResNet50 on the Toronto case study. VGG16 is a DL network developed by the Oxford Visual Geometry Group (VGG), which has sixteen weight layers and five convolutional blocks (Ünlü and Kiriş, 2021). VGG16 has been previously used for damaged building detection, classification, and pavement distress detection. ResNet50, with 50 layers, is based on the assumption that learning the residual function is more accessible than the actual mapping function (He et al., 2016). Although VGG16 and ResNet50 were previously trained, and their weights can be kept fixed and used for other classification problems, they have been trained on natural images having different semantic information to satellite images and LiDAR data (Jiang et al., 2021). Here, the preference was to retrain these networks on the train data to increase their accuracy. Two other recently developed state-of-the-art DL algorithms, including Unet and ResUnet (Ronneberger et al., 2015; Diakogiannis et al., 2020) were compared with our proposed DAN method on the Massachusetts Building Dataset to further explore the validity of the algorithm. Unet consists of an encoding-decoding architecture. The encoding part has the conventional architecture of a convolutional network. It consists of the repeated layers of two 3x3 convolutions, each followed by a Rectified Linear Unit (ReLU) function and a 2x2 max pooling operation with stride 2 for downsampling. Concatenation with the corresponding feature maps from the encoding part and two 3x3 convolutions, each followed by a ReLU, is conducted in each decoding step. Every step in the decoding part consists of an upsampling of the feature map followed by 2x2 Transpose Convolutional layers. ResUnet uses an encoder-decoder architecture. Instead of using standard convolution layers in the encoder-decoder part, it uses ResNet units containing multiple in-parallel atrous convolutions. Pyramid scene parsing pooling is embedded in the middle and end of the network. Unet and ResUnet were trained from scratch, and training parameters, including optimizer, initial learning rate and batch size, were set to Adam, 10^{-4}, and 20, respectively. The initial learning rates were decreased exponentially using the learning rate scheduler library in TensorFlow to prevent overfitting. The training parameters for these DL methods have been presented in Table 4-12.

Table 4-12: Training parameters for VGG16 and ResNet50

Method	Number of input features	Number of Epochs	Batch size	CNN patch size	Optimizer	Initial Learning rate	Loss	Early stopping condition
VGG16	3	400	20	33	SGD	10^{-1} (it was decreased exponentially)	Binary cross entropy	Same as DAN
ResNet50	11 (all of the LiDAR-derived and spectral features)	400	20	9	SGD	10^{-2} (it was decreased exponentially)	Binary cross entropy	Same as DAN
Unet	3	400	20	512	Adam	10^{-4} (it was decreased exponentially)	Dice loss	Same as DAN
ResUnet	3	400	20	512	Adam	10^{-4} (it was decreased exponentially)	Binary cross entropy	Same as DAN

4.5.3 Comparison with VGG16 and ResNet50 algorithms on Toronto case study

Figures 4-13 – 4-16 show the loss function and accuracy values on training and validation data for the first Tile. The loss and accuracy diagrams for the second Tile showed a similar trend as the first Tile and were not included for brevity purposes. In these figures, DAN refers to the Dense Attention Block after omitting the concatenation layers. The training diagram shows a descending trend for all the methods that proves the parameters, including

the learning rate, optimizer, batch size, number of epochs, and early stopping condition, have been set correctly. Our proposed technique shows lower loss values on both the train and validation datasets. On top of that, it has been converged to a minimum loss value more quickly than the VGG16 and ResNet50. VGG16 has a considerable number of parameters, and this characteristic increases its convergence time. To reduce model complexity in VGG16, it does not contain batch normalization layers which are beneficial for speeding up the learning process. So, the parameters, especially the learning rate, should be set carefully to converge the network. Also, the early stopping condition in VGG16 was set so that its training time decreases. VGG16 is quite sensitive to parameter setting, especially the learning rate value, and this can be a limitation for this method. Although newer models, like ResNet50, include batch normalization layers and converge more quickly, it can be seen that the proposed DAN has even less training time (better convergence) than ResNet50. The proposed DAN network has achieved smaller loss values than VGG16 and ResNet50 on the validation data, which shows its better performance in terms of generalization. Some evident fluctuations can be seen for VGG16 in figure 4-14. These fluctuations can be caused by the small number of validation data compared with the large number of parameters in VGG16.

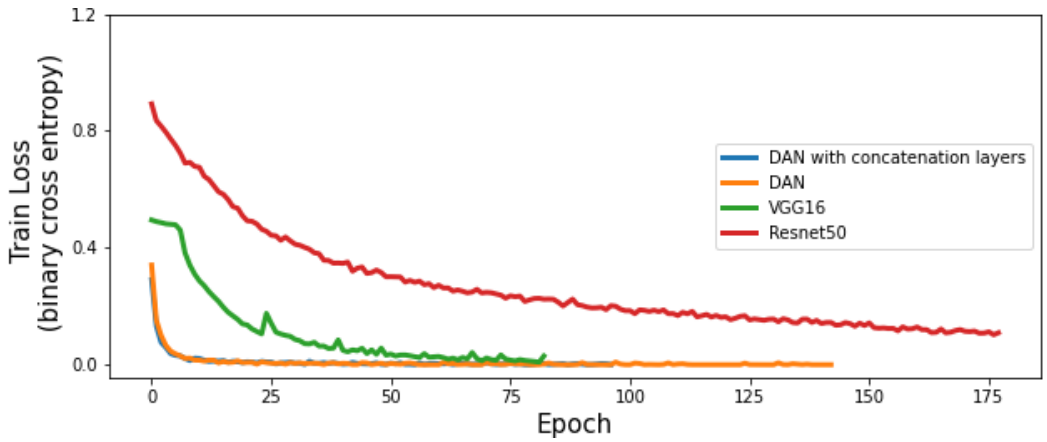


Figure 4-13: Train loss values during epochs

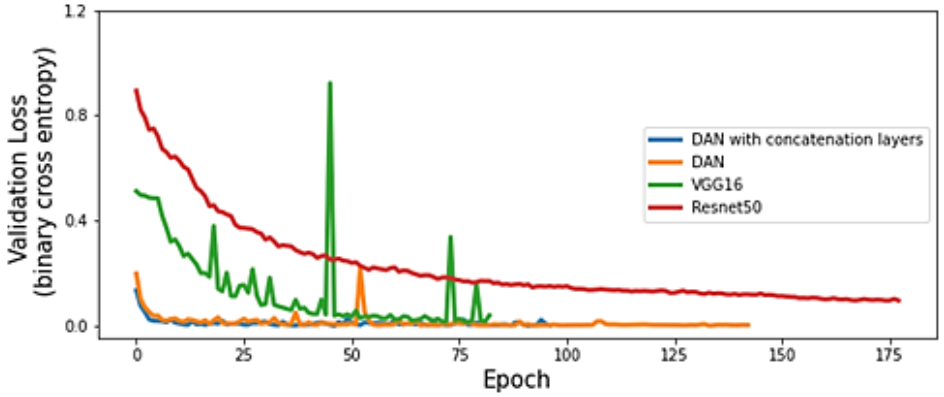


Figure 4-14: Validation loss values during epochs

According to figures 4-15 and 4-16, the proposed DAN network acquired higher accuracy values than VGG16 and ResNet50 on both the train and validation data, and the lowest accuracy was for ResNet50.

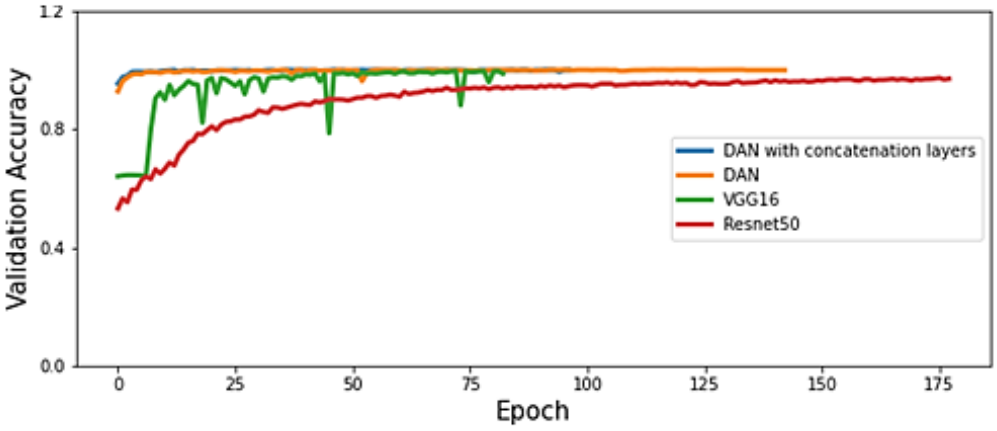


Figure 4-15: Training accuracy values during epochs

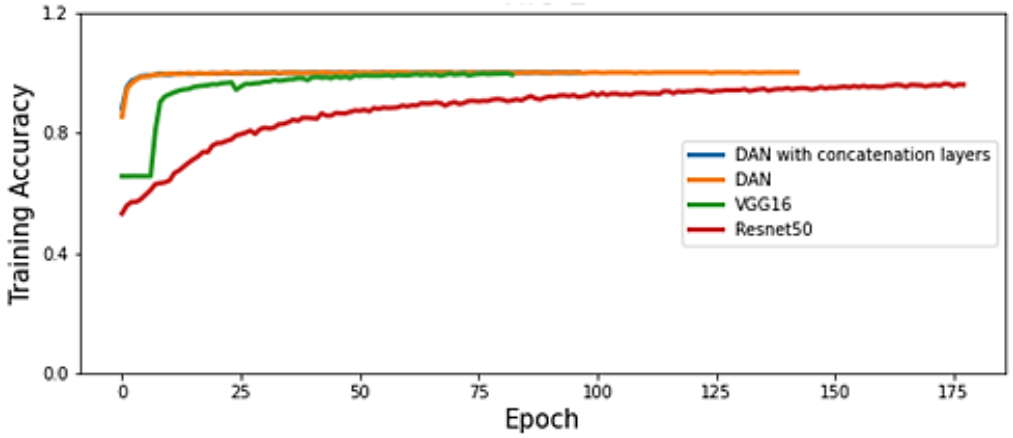


Figure 4-16: Validation accuracy values during epochs

Figures 4-17 and 4-18 show the bar plots for building detection accuracy indices for different techniques applied. It can be observed that the proposed method with or without concatenation layers can achieve high precision, recall, and F1 Score. However, after removing the concatenation layers in the first Tile, the accuracy indices dropped by 1%. ResNet50 showed lower recall and F1 Score values than VGG16 on test data because it has fewer parameters (23 million<138 million) and thus might fail when modelling complex mapping functions. Although VGG16 had comparable performance to the proposed DAN in terms of recall index, its precision was lower than both proposed DAN networks because of its higher false alarm rate. VGG16 has been designed to process RGB images; therefore, more than three bands cannot be imported to this network, reducing its efficiency and flexibility. In addition, it does not accept small image patches as input. This limitation can increase its processing time when the GPU is not available. Another issue regarding not accepting small patches is the mixed object problem, for example, building and vegetation in the same patch. So, the original 9×9 patches were resampled to 33×33 size using linear interpolation rather than dividing the image into 33×33 segments. Our proposed method solves this limitation, and it can accept any patch size.

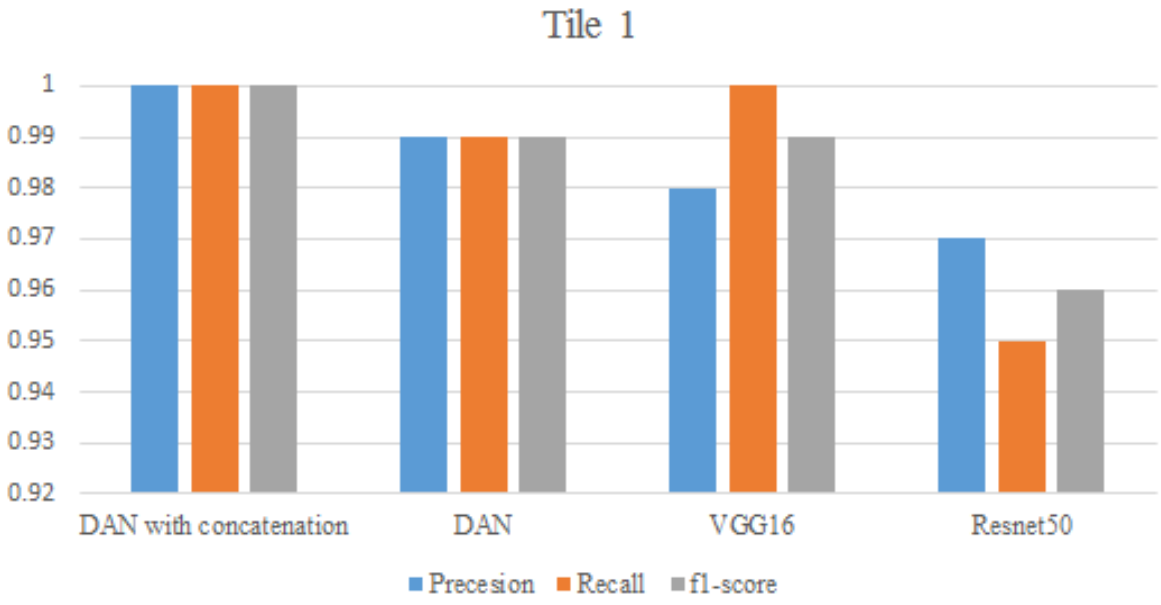


Figure 4-17: Building detection accuracy indices for the proposed method (both with and without concatenation), VGG16, and ResNet50 (first Tile)

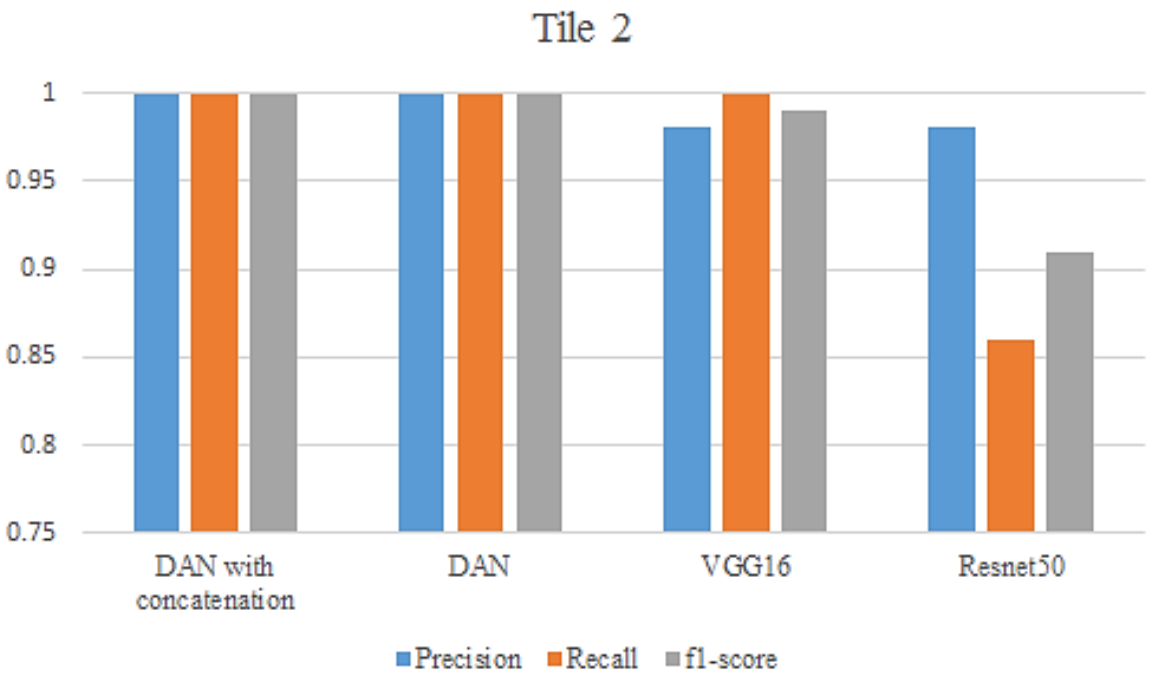


Figure 4-18: Building detection accuracy indices for the proposed method (both with and without concatenation), VGG16, and ResNet50 (second Tile)

Table 4-13 shows IoU values for different methods. It can be concluded that the highest agreement with the test data was achieved when using the proposed methods (both DAN and DAN with concatenation), and the IoU values were even improved by 1% after applying the concatenation technique. This Table also shows that although ResNet50 has more layers than VGG16 and DAN, it resulted in the lowest IoU values on both Tiles.

Table 4-13: IoU values for the proposed method (both with and without concatenation), VGG16, and ResNet50 for the first and second Tiles

Method	IoU (Tile 1)	IoU (Tile 2)
DAN with concatenation	1	1
DAN	0.99	0.99
VGG16	0.98	0.97
ResNet50	0.93	0.84

4.5.4 Comparison with Unet and ResUnet algorithms on Massachusetts case study

Based on the accuracy indices reported for the three methods in Figure 4-19, it is apparent that our proposed method acquired both higher precision and recall values than Unet and ResUnet. While the precision and recall values for the proposed technique were 0.81 and 0.71, respectively, the corresponding accuracy indices were 0.31 and 0.64 for Unet and 0.50 and 0.48 for ResUnet. Although the Unet method achieved a comparable accuracy value on the training dataset, its building detection ability was lower than DAN and ResUnet because of overfitting on the train data. Still, Unet acquired a 0.14 higher recall score than ResUnet, and there were more missed building blocks in ResUnet than Unet. On the other hand, our proposed technique resulted in 0.07 and 0.21 higher recall scores than other methods because of the use of dense attention blocks and importing semantic features from previous layers using concatenation layers. The F1 Score, the weighted average of the precision and recall values, was 0.76 for the proposed algorithm compared with the 0.49 and 0.42 values for ResUnet and Unet. Although Unet acquired a 0.14 higher recall value than ResUnet, its F1 Score is still 0.07 lower than ResUnet, which means

ResUnet shows higher overall building detection ability than the Unet algorithm on the Massachusetts Building Dataset.

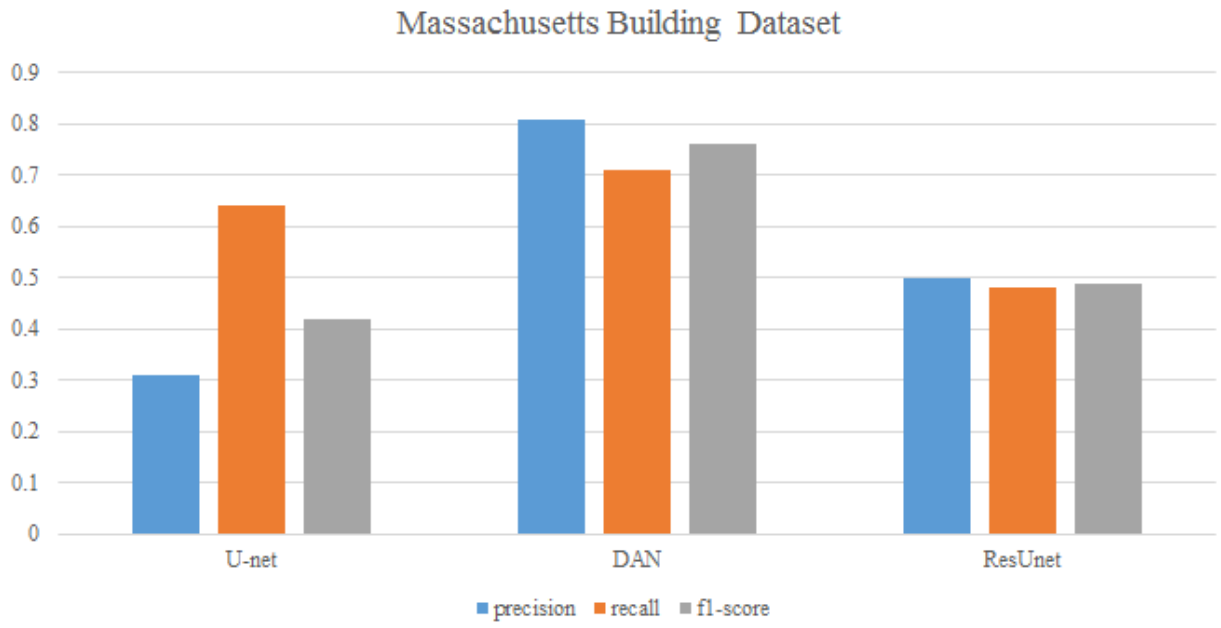


Figure 4-19: Accuracy indices for Unet, the proposed DAN, and ResUnet on the Massachusetts Building Dataset

Based on figures 4-20 and 4-21, it can be seen that our proposed DAN created building masks with a lower amount of background noise compared with ResUnet because this method is more sensitive to detailed background objects such as road edges because of its dependency to feature extraction based on residual function. Although ResUnet resulted in classification maps with higher precision than Unet, the method has merged some adjacent building blocks as one entity, as seen in areas highlighted by red squares on the building masks. It can be observed in figure 4-22 that this merging level was even more intense in Unet building masks than ResUnet because of the overfitting. In terms of the IoU index, the same trend as the F1 Score can be seen in Table 4-13; the proposed algorithm achieved

considerably higher IoU, about 0.61, than Unet and ResUnet, which acquired 0.26 and 0.33 F1 Score, respectively.

Figure 4-20: Building detection result on three representative areas using the proposed DAN on the Massachusetts Building Dataset. See text for explanation of red boxes.

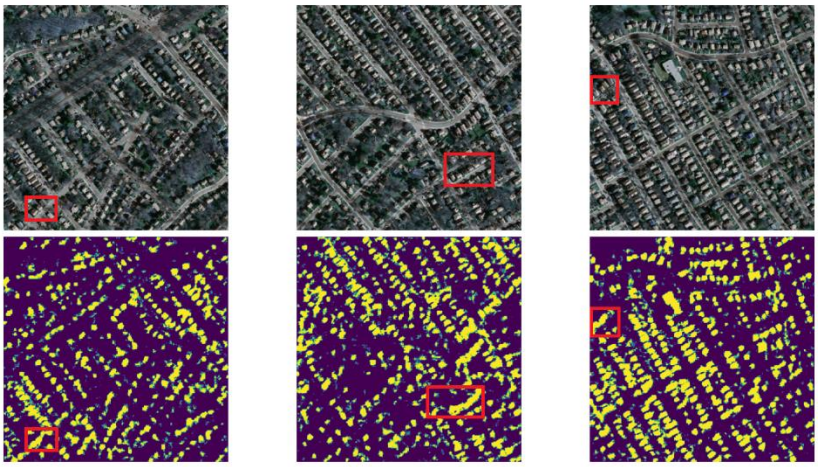


Figure 4-21: Building detection result on three representative areas using ResUnet on the Massachusetts Building Dataset

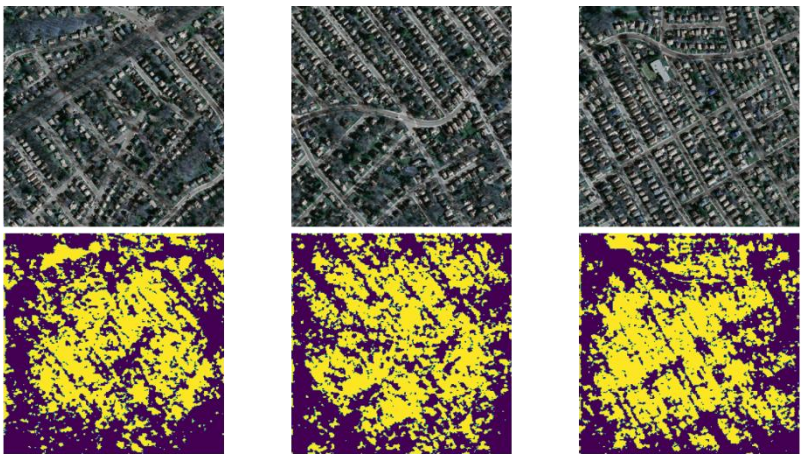


Figure 4-22: Building detection result on three representative areas using Unet on the Massachusetts Building Dataset

The Massachusetts Building dataset structure differed from the Toronto building data because the ground truth data were 2D label maps instead of pixel-based labels. So, a modification was carried out, and the FC layers in the initial architecture were replaced with the 2D Convolutional layers to produce label maps instead of pixel-based predictions. Regardless of the use of FC layers or 2D Convolutional layers in the prediction part, our proposed DAN achieved higher accuracy than the state-of-the-art methods in terms of precision, recall, and F1 Score. These results show the flexibility of the algorithm to both object-based and pixel-based predictions.

4.5.5 Comparison with building footprints from Toronto Land Cover Map

The most recent Toronto Land Cover Map was created in 2018, and it has been considered the most accurate land cover data of the area at the time of its creation. The dataset was created as part of the Toronto Tree Canopy Study, and it includes eight land cover classes:

(1) tree (2) grass (3) bare (4) water (5) building (6) road (7) other paved surfaces and (8) shrub. For this study, only building footprints were required, so other classes were merged as class *other*. Figure 4-23 shows building footprints after the combination of classes other than building.

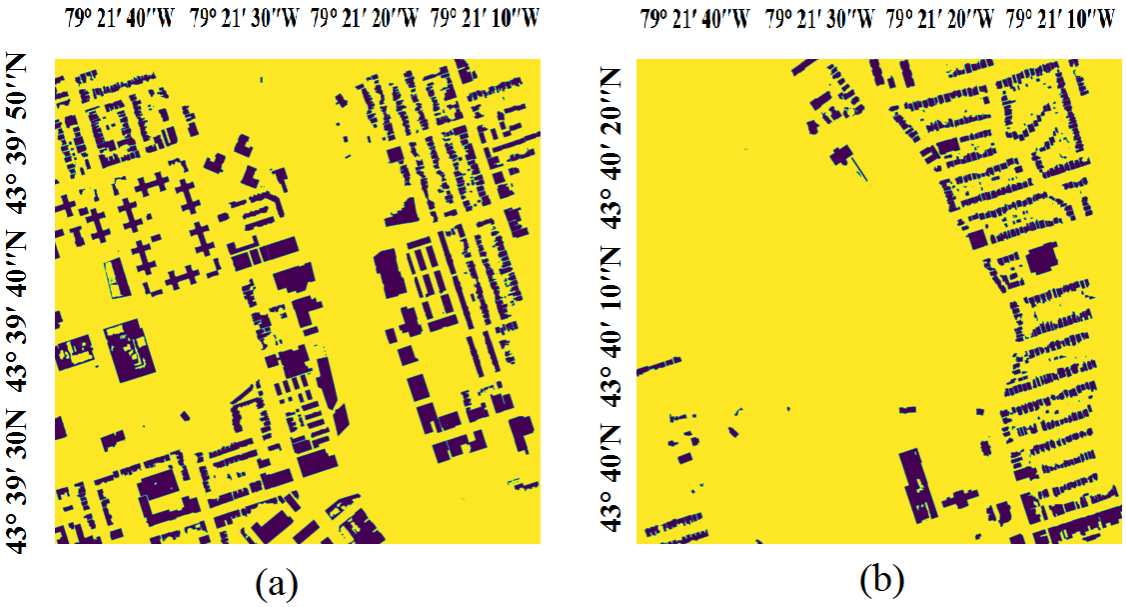


Figure 4-23: Building footprints extracted from 2018 Toronto Land Cover Map; a: building footprints for the first Tile; b: building footprints for the second Tile

The following average accuracy indices between building and background classes and tiles were computed after comparing the building footprints extracted from the DL methods with 2018 Toronto data. Figure 4-24 demonstrates the bar plots for average accuracy indices, including F1 Score, recall, and precision for proposed DAN, VGG16, ResNet50, Unet, and ResUnet. The vertical axis shows the method, and the horizontal axis refers to the average accuracy indices between classes and tiles. It can be observed that while all DL techniques have achieved agreements between about 50%-60%, the proposed DAN shows higher agreement with the reference data in terms of all the accuracy indices with a recall

value slightly above 60%. Higher agreements might be achieved using higher-resolution spectral data than Rapid Eye images.

Looking at the average F1 Score between building and background classes, VGG16 obtained a higher F1 Score by 4% than ResNet50 and ResUnet because its building detection results were more balanced between the first and second tiles. ResNet50 and ResUnet achieved lower building detection precision than VGG16 in the second tile, and hence their average F1 Score dropped to 41%. On the other hand, ResUnet achieved comparable performance to our proposed method in terms of recall score, with a recall value of 59%, because this method is a more recent DL algorithm than other techniques, and it takes advantage of both the residual learning strategy in ResNet networks and the Unet architecture.

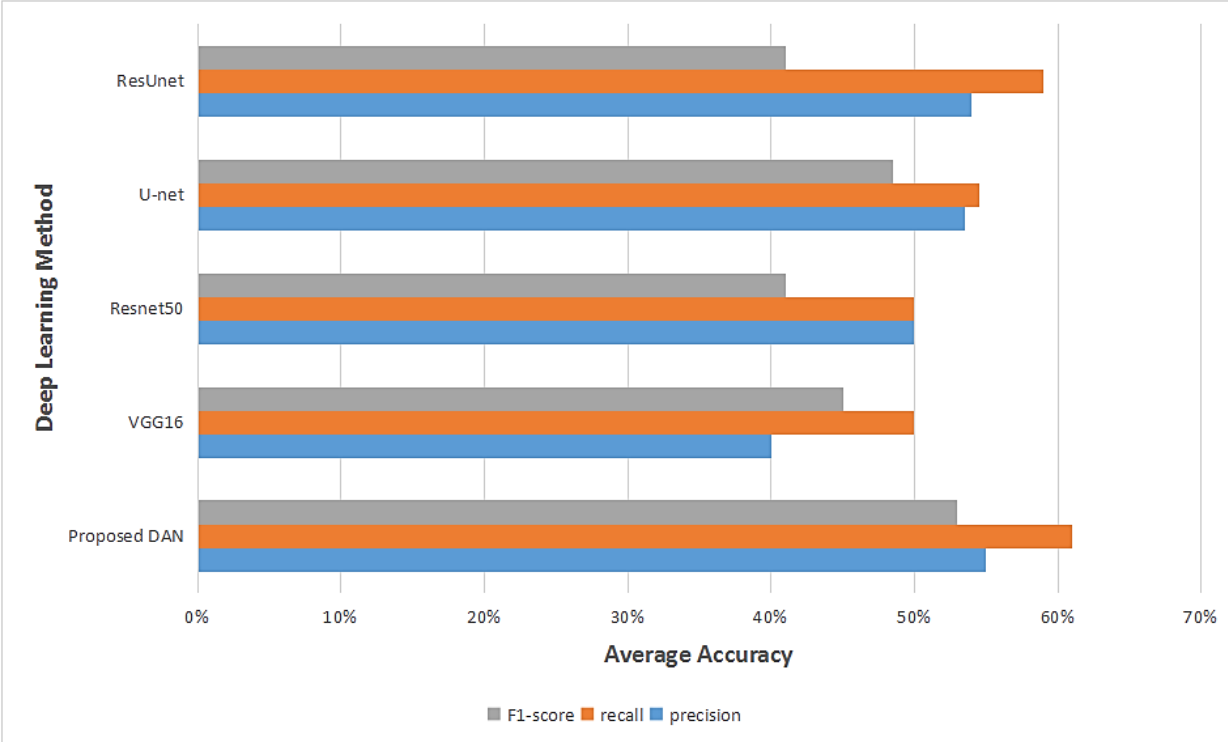


Figure 4-24: Average accuracy indices for ResUnet, Unet, ResNet50, VGG16, and Proposed DAN

4.6 Conclusion

This study proposed a CNN architecture based on the dense attention block concept for building detection in Toronto, Ontario, Canada. The inputs to the network were LiDAR-derived features, and the RapidEye spectral bands, resampled to the same resolution as LiDAR data. It was observed that excellent accuracy indices in terms of precision, recall, F1 Score, and IoU on test data were achieved using features extracted by LiDAR alone or after adding RapidEye spectral features. This result showed the capability of DAN networks to extract features at different scales and the efficiency of LiDAR-derived features in extracting building footprints with high accuracy. The results also showed that stacking high- and low-level features using concatenation layers slightly improved the accuracy indices. A comparison with four other state-of-the-art DL techniques, including

VGG16, ResNet50, Unet, and ResUnet, was also conducted to test the efficiency of the proposed DAN network. The results showed that the accuracy indices obtained were higher in most cases, which demonstrates the effectiveness of our proposed method. The building detection result of the proposed DAN was compared with the 2018 Toronto building footprint dataset. The results showed that the proposed DAN achieved a higher agreement than VGG16 and ResNet50 because VGG16 and ResNet50 were initially developed for RGB image processing, and their use for satellite image processing might not always be a good solution. Even efficient for satellite image processing, they need to be retrained for the new dataset, which takes a long time because of the high number of layers and parameters. Our proposed model, which has far fewer parameters (~200 000) than VGG16 and ResNet50 and fewer layers than ResNet50, can detect buildings with higher precision without requiring a post-processing step and is a suitable model for building detection in satellite images. In addition, the proposed method can predict at both the pixel and object levels. For the prediction at the pixel level, the flattening operation was embedded after the feature extraction stage, followed by the Fully Connected (FC) layers and a softmax function at the prediction layer. On the other hand, for prediction at the object level, the 2D convolutional layer was replaced with FC layers to create a 2D building mask at once instead of predicting the labels pixel by pixel. Therefore, it can be inferred that the proposed CNN is adjustable for prediction at both pixel and object levels.

The proposed training and test data preparation approach presented here is not similar to some state-of-the-art DL algorithms, such as Unet and ResUnet. While Unet and ResUnet accept input patches not smaller than 32×32, our proposed training data preparation procedure is limited to smaller patch sizes of up to 11×11 because of the memory issues arising when using larger patch sizes. In other words, our predictions are conducted on a smaller scale than those of Unet and ResUnet, which might limit the proposed training data preparation strategy. However, the proposed DAN network is still adjustable to object-level prediction using 2D convolution as the prediction layer, instead of using a dense layer.

References

Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*, 94-114.

Hamaguchi, R. and Hikosaka, S., 2018. Building detection from satellite imagery using ensemble of size-specific detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 187-191).

Hamwood, J., Alonso-Caneiro, D., Read, S.A., Vincent, S.J. and Collins, M.J., 2018. Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers. *Biomedical optics express*, *9*(7), pp.3049-3066.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

He, Y., Wang, J., Liao, C., Shan, B., & Zhou, X. (2022). ClassHyPer: ClassMix-Based Hybrid Perturbations for Deep Semi-Supervised Semantic Segmentation of Remote Sensing Imagery. *Remote Sensing*, *14*(4), 879.

Huang, J., Zhang, X., Xin, Q., Sun, Y. and Zhang, P., 2019. Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. *ISPRS Journal of Photogrammetry and Remote Rensing*, *151*, pp.91-105.

Jiang, X., Li, G., Zhang, X. P., & He, Y. (2021). A semisupervised Siamese network for efficient change detection in heterogeneous remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1-18.

Maltezos, E., Doulamis, A., Doulamis, N. and Ioannidis, C., 2018. Building extraction from LiDAR data applying deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, *16*(1), pp.155-159.

Hinton, G. E., & Mnih, V. (2013). Machine learning for aerial image labeling. https://tspace.library.utoronto.ca/bitstream/1807/35911/1/Mnih_Volodymyr_201306_PhD_thesis.pdf

Nahhas, F.H., Shafri, H.Z., Sameen, M.I., Pradhan, B. and Mansor, S., 2018. Deep learning approach for building detection using lidar–orthophoto fusion. *Journal of Sensors*, *2018*.

Pirasteh, S., Rashidi, P., Rastiveis, H., Huang, S., Zhu, Q., Liu, G., Li, Y., Li, J. and Seydipour, E., 2019. Developing an algorithm for buildings extraction and determining changes from airborne LiDAR, and comparing with R-CNN method from drone images. *Remote Sensing*, *11*(11), p.1272.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). Unet: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Springer, Cham.
Ünlü, R. and Kiriş, R., 2021. Detection of damaged buildings after an earthquake with convolutional neural networks in conjunction with image segmentation. *The Visual Computer*, pp.1-10.
Yang, H., Wu, P., Yao, X., Wu, Y., Wang, B. and Xu, Y., 2018. Building extraction in very high resolution imagery by dense-attention networks. *Remote Sensing*, *10*(11), p.1768.

Yang, H.L., Yuan, J., Lunga, D., Laverdiere, M., Rose, A. and Bhaduri, B., 2018. Building extraction at scale using convolutional neural network: Mapping of the united states. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(8), pp.2600-2614.
Zhang, H., Qin, K., Zhang, Y., Li, Z. and Xu, K., 2020, October. Dense Attention Convolutional Network for Image Classification. In *Journal of Physics: Conference Series* (Vol. 1651, No. 1, p. 012184). IOP Publishing.

Zhao, K., Kang, J., Jung, J. and Sohn, G., 2018. Building extraction from satellite images using mask R-CNN with building boundary regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 247-251).

Zhou, Z. and Gong, J., 2018. Automated residential building detection from airborne LiDAR data with deep neural networks. *Advanced Engineering Informatics*, *36*, pp.229-241.

Chapter 5

Fusion of Google Street View, LiDAR, and Orthophoto classifications based on a ranking method for building Land-Use type detection

5.1 Introduction

Building land-use type is valuable information for municipalities to assess flood damage and estimate the number of exposed people during or after the flood event, but this information is either unavailable or not in a standard format (Al-Habashna, 2022). The traditional method for collecting building land-use type information is ground surveying, which is labor-intensive. Some studies in the literature addressed this issue by automating this task using machine learning and EO data. Belgiu et al. 2014 used Light Detection and Ranging (LiDAR) data to categorize the buildings into residential/small, factory/industrial, and apartments. First, building footprints were extracted, and four kinds of features related to extent, shape, height, and slope were computed for each building footprint. Then, Random Forest (RF) classifier and some IF/THEN rules based on the building type ontology were applied to produce the final building type classification map. The results showed that although a high F1 score, about 98%, was achieved for residential/small buildings, the F1 scores for factory/commercial buildings and apartments were considerably lower, about 51% and 60%, respectively. Lu et al. 2014 used LiDAR data for building land-use type classification. They applied three feature types: 1- The basic statistical features such as minimum, maximum, and standard deviation computed on the first and last return pulses. 2- The Shape attributes such as length, width, and length-to-width ratio. 3- The spatial relationship between buildings together and between buildings and other landscape features. Wurm et al. 2016 applied 1D, 2D, and 3D building shape attributes and a Linear Discriminant Analysis method to classify buildings into different types, such as terraced houses/row houses, and detached/semi-detached. The results showed that shape features are unsuitable for discriminating similar building types, including perimeter block development, and block development. Although LiDAR can give a valuable source of information about building geometric features, which can help classify buildings into residential/commercial, it does not give any spectral information,

which can be helpful when discriminating other types of buildings. Orthophotos can provide spectral information, and the simultaneous use of LiDAR and orthophoto can improve identifying land-use type of buildings and reduce the chance of commission and omission errors. Meng et al. 2012 used LiDAR, aerial images, and a road network map to discriminate residential buildings from other types of buildings in Austin City, Texas using a C4.5 classification algorithm. The results were compared with the field survey data, and the accuracy of about 81% was achieved for residential buildings.

Machine Learning algorithms are a great choice when we have no prior assumption on the data distribution and can be used for various tasks, including multivariate non-linear non-parametric regression, supervised classification, and unsupervised classification (Lary et al., 2018). When doing supervised classification, we need a dataset, referred to as train data, large enough to span the parameter space as much as possible. In the remote sensing data context, supervised classification refers to when labeled data exists on the class membership of single pixels (Camps-Valls, 2009). These data are used to generalize the model to the whole image. Unsupervised classification is when the training step is skipped, and the image is partitioned into different parts based on the spatial or spectral characteristics of the input image. These tasks can be achieved using different algorithms, such as Neural Networks, Support Vector Machines, Decision Trees, and Random Forests.

While previous studies were extensively focused on using Machine Learning and Deep Learning (DL) algorithms for building footprint extraction (Yan et al., 2011; Abdollahi et al., 2020; Liu et al., 2022; Rastogi et al., 2022; Yu et al., 2023), recent efforts have been made for building land-use type classification using EO data. Xie and Zhou, 2017 extracted features from high-resolution satellite images at multiple resolutions using an Extended Multiresolution Segmentation (EMRS) algorithm and classified buildings into different functionality types with a Soft Classification using a Back Propagation (BP) network. The overall accuracy improved by 19.8% compared with a single-resolution segmentation space using a hard classification with the BP network. Huang et al. 2022 created a building roof type and functionality dataset from high-resolution satellite images for Beijing and Munich Cities. They examined DL based segmentation algorithms, including Mask R-CNN, Cascade Mask R-CNN, SOLOV2, and QueryInst, and achieved Average Precision

at Intersection Over Union = 0.5 ($AP_{0.5}$) of 23.5, 25.5 for Beijing and Munich Cities, respectively. Additionally, Google Street View (GSV) images have been used for building land-use type classification. Wang et al. 2017 used AlexNet to classify GSV images and achieved an overall test accuracy of about 90% on non-independent test data. Zhang et al. 2017 used LiDAR, high-resolution orthophotos, and GSV images for building land-use classification of an area in New York City, USA. It was concluded that although the overall accuracy did not improve significantly after using GSV images, the mixed commercial and residential class accuracy improved by 10%. Kang et al. 2018 used Convolutional Neural Networks (CNN) to classify buildings into apartments, churches, garages, houses, industrial, office buildings, retail, and roofs using façade information from GSV images. Four CNNs were tested, including AlexNet, Visual Geometry Group with 16 convolutional layers (VGG16), Residual Network with 18 (ResNet18) and 34 (ResNet34) deep layers. The best model was VGG16. Al-Habashna, 2022 used GSV images and CNNs for building land-use type classification. The models used in this study were VGG16, ResNet18, ResNet34, and Residual Network with 50 deep layers (ResNet50). The overall accuracies achieved were up to 78% when the train and test were from the same city, and up to 69% when the train and test data were from different cities. Laupheimer et al. 2018 used GSV images to classify buildings as commercial, hybrid, residential, special use, and under construction using four CNN models, VGG16, VGG19, ResNet50, and InceptionV3. The highest overall accuracy was achieved using InceptionV3, with a value of 64%. Wu et al. 2023 used GSV and OSM land use information to estimate mixed land use types throughout New York City. The image and text information were imported into two separate encoders, and the dot product of the resultant image and text embeddings was calculated as a similarity measure. The focus of this study was more on ground land-use type detection rather than building land-use. In another similar research, a CNN called Conv-Depth Block Res-Unet was proposed for land-use classification in three metropolitan areas in Korea. The proposed method combined convolution and depth-wise separable convolutions and achieved an overall accuracy of 83.7% on test samples. Although the proposed method achieved high overall accuracy, and the model outperformed existing CNNs, including DeepLab V3+, ResUnet, ResASPP-Unet, and Context-based ResUnet,

the classification maps were not detailed in terms of building land-use type (Yoo et al., 2022).

Some previous studies used the fusion of GSV, LiDAR, and aerial images for building land-use type classification. For example, Hoffmann et al. 2019 fused aerial and GSV images for classifying buildings into commercial, industrial, public, and residential. Three fusion methods based on DL were explored: 1- Use of a single DL network pre-trained on GSV dataset (*Places365*) and *ImageNet* data. 2- Using two streams of DL networks (VGG16), one for the GSV data and the other one for the aerial image. Then, the fine-tuning was used for training the DL networks. 3- Decision Level Fusion by combining the Softmax probabilities or the classification labels directly using model blending and stacking approaches. The highest overall F1 Score was for the Decision Level Fusion with a value of 75%. Cao et al. 2018 combined GSV and aerial images using the SegNet DL network. Two encoders were used for feature extraction, one for the GSV and the other for the aerial image. The extracted feature maps using the encoders were stacked and fed into the decoder part. The model was tested on the New York City aerial and GSV image dataset with an overall accuracy of up to 74% achieved after fusion. The highest per-class accuracy was for one and two-family buildings with a value of up to 84%.

This reviews shows that a comprehensive classification of buildings into building categories, like institutional, industrial, office buildings, and retail, remains deficient. Furthermore, most previous studies missed mixed building categories, such as mixed residential/commercial. Addressing these gaps, this Chapter introduces a fusion method using ranking classes based on the F1 Score index to produce a detailed building land-use type map for regions in Vancouver, BC and Fort Worth, Texas. The reason for selecting GSV is that it includes façade texture information. Also, LiDAR and orthophotos contain height and spectral information of buildings' façades and roofs. By combining CNNs trained on GSV, LiDAR, and Orthophoto using the proposed ranking method, this study aims to improve building land-use type detection accuracy.

5.2 Case Studies and Dataset

Three case studies were explored in this work. These areas were selected because of the ground truth data availability. The first case study included four cities in the Greater Toronto Area (GTA), Toronto, Vaughan, Richmond Hill, and Peel Region, with areas of 1829.05, 273.56, 100.79, and 1.25k square kilometers, and population densities of 3087.7, 1185, 2004.4, 1108.1 inhabitants per square km. The number of occupied private dwellings, reported in 2021, were 1160892, 107159, 69314, and 450746 for Toronto, Vaughan, Richmond Hill, and Peel Region, respectively (Government of Canada, Statistics Canada, 2023; Vaughan Economic Development, 2021; Government of Canada, Statistics Canada, 2017). Figure 5-1 shows the extent of the cities on the Open Street Map (OSM) map, along with the building samples used in this study. The zoomed areas show the building footprints.

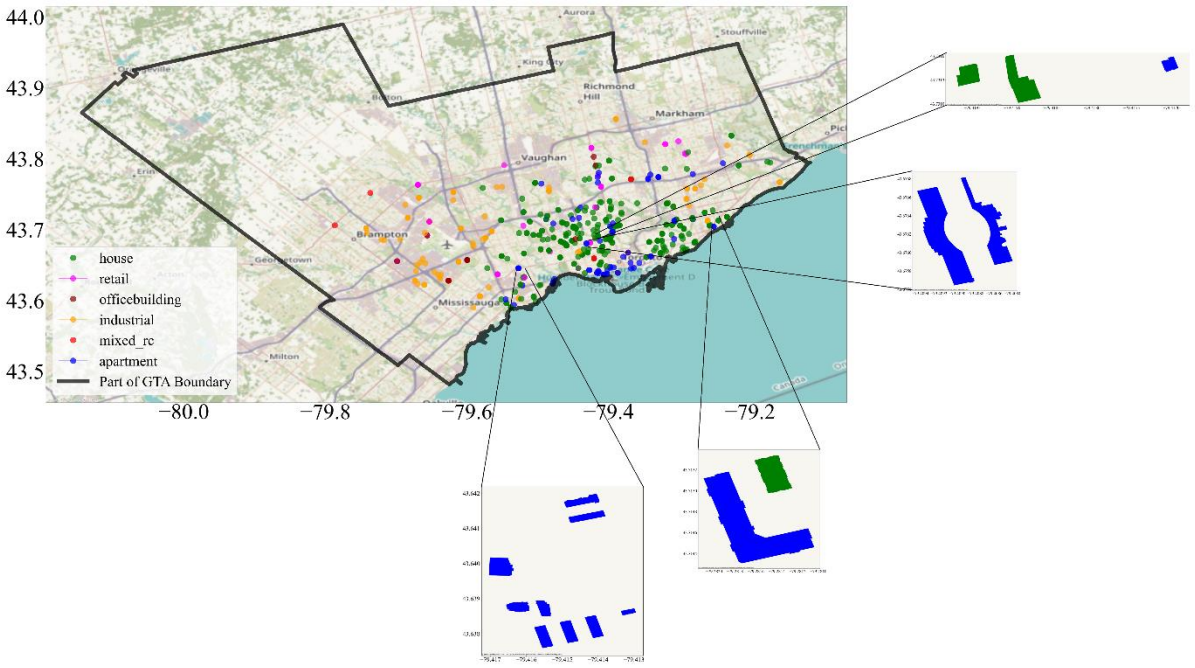


Figure 5-1: Greater Toronto Area (GTA) case study. The color dots indicate building samples within the boundary including the cities of Toronto, Markham, Vaughan, Richmond Hill, and the Peel region. The background map corresponds to OpenStreetMap (OSM).

Two independent test case studies, including the Vancouver and Fort Worth Cities, were explored to make the train and validation data as separate as possible from the test dataset. Vancouver City, with an area of 115 square kilometers, holds a population density of 5249 per square km. In 2021, of 1,043,320 occupied private dwellings in Vancouver about 28%, 24.5%, 19%, 16%, 10%, 2%, 0.4%, and 0.1% were single-detached houses, apartments with less than five stories, apartments with five stories and more, apartments or flats in a duplex, row houses, semi-detached houses, movable dwellings, and other single detached houses (Government of Canada, Statistics Canada, 2023a). Fort Worth City, located in Texas, US, with an area of 916.76 square kilometers, has a population density of 2677 per square mile, and 326647 households were living in the city between 2018-2022 (City of Fort Worth, 2019; United States Census Bureau, 2022). Figures 5-2 and 5-3 show the orthophoto images of the test case studies and their ground truth maps.

The GSV, LiDAR, and Orthophoto data for the GTA were used for training the DL models.

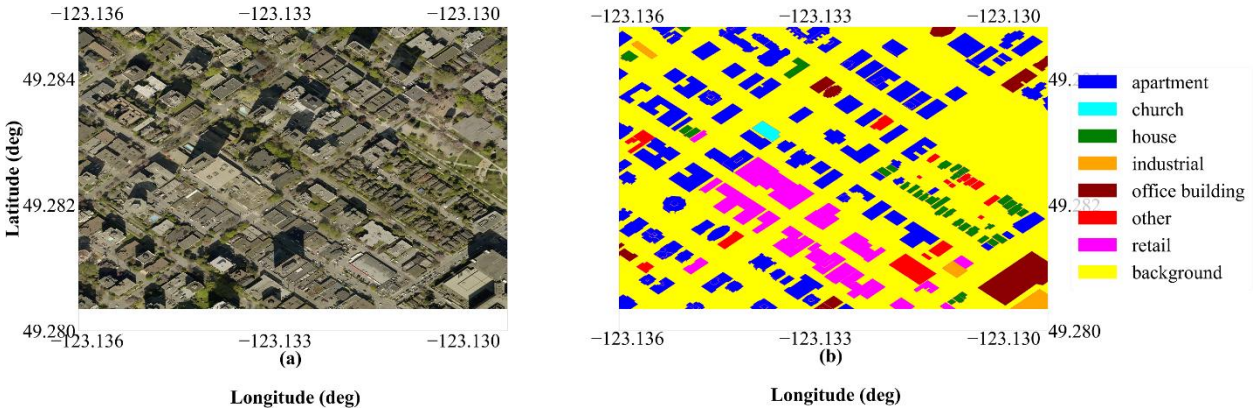


Figure 5-2: Vancouver test region; (a): orthophoto image; (b): ground truth map

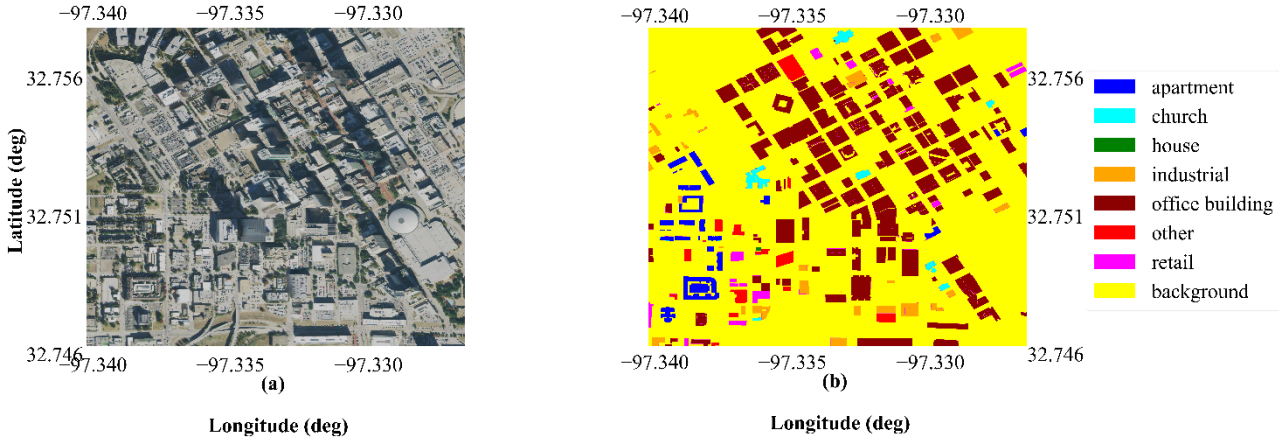


Figure 5-3: Fort Worth test region; (a): orthophoto image; (b): ground truth map

5.2.1 GSV Dataset

The building land-use type GSV image dataset created by Kang et al. 2018 was used to train the DL algorithms. The data included 17600 512×512 GSV images captured with a pitch angle of 10 degrees from cities across the U.S. and Canada, for example, Montreal, New York, and Denver. The ground truth labels for this data were extracted from OSM.

5.2.2 LiDAR Point Cloud Dataset

Ontario Classified Point Cloud data were accessed via the Scholars GeoPortal website. The data were in 1km×1km tiles in LAZ format, and the vertical accuracy was 20.76cm, with 6234 feet of flight height above the ground level. The horizontal spatial reference system of the data was Universal Transfer Mercator (UTM) Zone 17N, and the datum was the North American Datum 1983 Canadian Spatial Reference System. The vertical spatial reference system was the Canadian Geodetic Vertical Datum 2013.

For the Vancouver test area, the Classified LiDAR Point Cloud Data were downloaded from the Government of British Columbia (BC) website with 78cm vertical accuracy in non-vegetated areas. The LiDAR sensor flight height was 1850m, and the LiDAR point density was 8 points/m^2. The horizontal spatial reference system was UTM Zone 10N, with the North American Datum 1983. The vertical spatial reference system was the Canadian

Geodetic Vertical Datum of 2013. For the Fort Worth test area, the data were accessed via the Texas Natural Resources Information System (TNRIS) Hub, acquired during the United States Geological Survey (USGS) 2019 Pecos Dallas LiDAR Project. The data were in 21.2cm vertical accuracy in non-vegetated areas with the 6000 feet flight height, and the Average Nominal Pulse Density not less than 2 points/m^2. The horizontal spatial reference system was UTM Zone 14N, and the Datum was North American Datum 1983. The vertical spatial reference system was the North American Vertical Datum of 1988.

5.2.3 Orthophoto Dataset

Orthophoto images for the GTA were downloaded from the Scholars GeoPortal website. The data were acquired during the 2018 South Central Ontario Orthophotography Project (SCOOP 2018) and were provided in 1km×1km tiles. Besides, they included Red, Green, and Blue (RGB) and Near Infrared bands with 20cm spatial resolution. For the Vancouver test area, the data were downloaded from the Vancouver Open Data Portal. The images were captured in 2015 with 7.5cm spatial resolution and in RGB bands. For the Fort Worth test region, the data were accessed via the Texas Natural Resources Information System (TNRIS) Hub. The data were acquired by the National Agriculture Imagery Program (NAIP) between May 2018 and April 2019. Each image tile covered an area of 16 square miles, with 60cm spatial resolution. The images were in 4 bands (RGB and Infrared).

5.2.4. Building Footprint Data

The 2019 building footprint data for the GTA region was downloaded from the Statistics Canada website. The data included latitude, longitude, Building Identity Document (ID), building area, and length for buildings across GTA. The building footprint data for Vancouver were accessed via the city Open Data Portal and were generated from the city of Vancouver's 2015 orthophotos. The data included Building IDs for buildings across Vancouver. For Fort Worth city, the building footprint data were provided by the City of Fort Worth and included information such as ID, address, owner, building area, and length.

5.2.5 ImageNet Data

The *ImageNet* dataset contains 14197144 annotated RGB images. The annotation was conducted manually with two approaches: 1- image-level binary labeling and 2- object-based labeling with bounding boxes around the objects. The reported annotation precision was 99.7%. The images were from six subtrees, including mammals, vehicles, geo forms, furniture, birds, and musical instruments (Deng et al., 2009). These data have been used for training different types of CNNs. The CNN parameters trained on *ImageNet* data can be transferable to another classification problem.

5.3 Method

5.3.1 Preprocessing

Some modifications were applied to the GSV dataset to make it suitable for this study. The garage and roof classes were ignored, and the new mixed residential/commercial class was added. The final building land-use type classes included *apartment*, *church*, *house*, *industrial*, mixed residential/commercial (referred to as *mixed r/c* hereafter), *office building*, and *retail*. The addresses for the *mixed r/c* buildings were extracted using web scraping from real estate listing platforms, including CREXi, and LoopNet. The corresponding GSV images were downloaded using Google Application Programming Interface (API). Then, the GSV images were divided into five folds, and, 500, 100, and 200 images were considered as train, validation, and test in each fold. Because of data scarcity for the *mixed r/c* class, the original number of samples was lower, with 45, 10, and 10 images as train, validation, and test. The imbalance data problem was resolved using data augmentation (Chen and Fan, 2021). The samples in *mixed r/c* class were flipped horizontally to make the number of train, validation, and test images the same as other classes (C).

The LiDAR-derived statistics, including mean, minimum, maximum, standard deviation, and range were calculated in ArcGIS Pro based on products of Classified Point Cloud data,

including First Return (FR) pulse, Last Return (LR) pulse, Intensity, and Slope. Besides, normalized Digital Surface Model (nDSM) variance inside each building footprint was calculated because of the roof height variations among building land-use type classes. Figure 5-4 shows the screenshots of LiDAR-derived DSM showing building footprints.

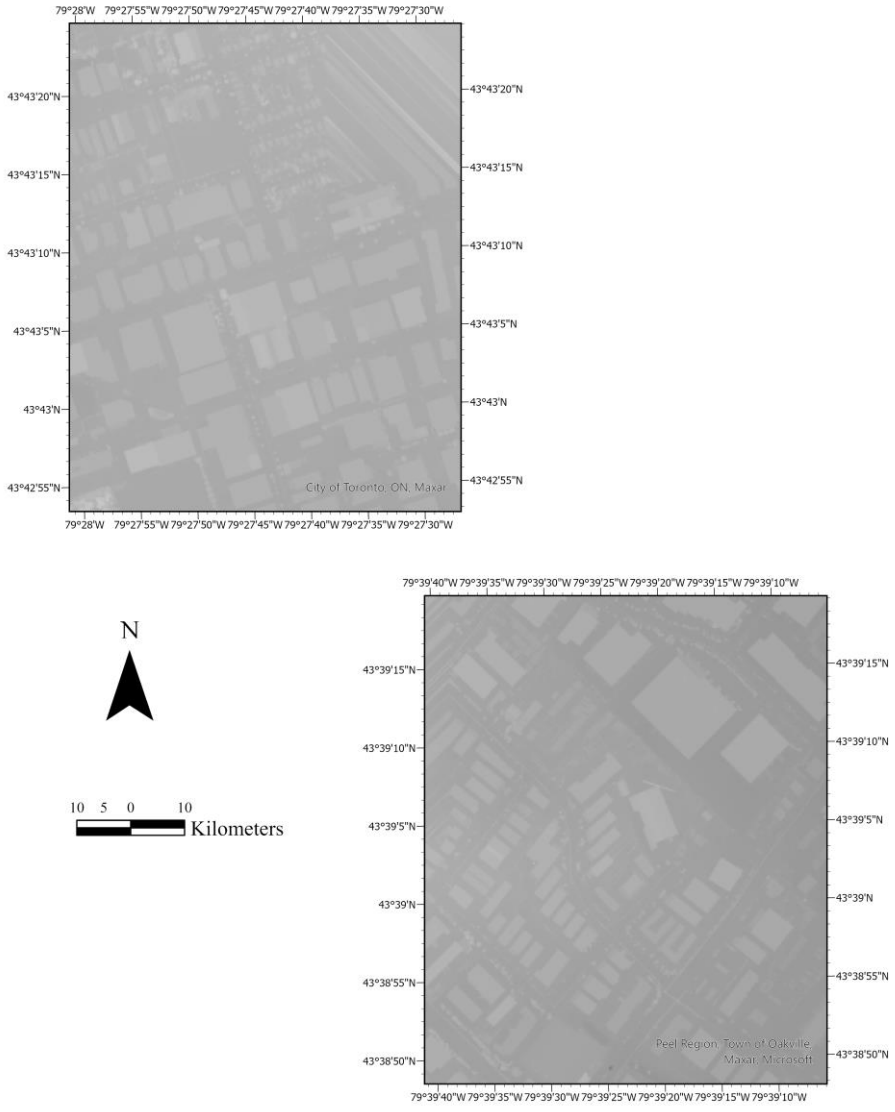


Figure 5-4: Screenshots of LiDAR-derived Digital Surface Model (DSM) from building footprints in two selected areas in Greater Toronto Area (GTA)

These statistics were calculated based on neighborhood analysis in a 3×3 window. The total number of LiDAR-derived features was 13, as listed in Table 5-1. Then, the features were clipped to the extent of each building footprint and reshaped into 512×512 for consistency with the GSV image size. The building land-use type labels were extracted using OSM. The number of samples in each class was made equal by over-sampling images in the minority classes and under-sampling images in the majority classes. After that, the created data were split into five folds. In each fold, 200, 40, and 60 feature bands, consisting of 13 LiDAR-derived features for each building footprint, were considered for train, validation, and testing. The OSM labels were relabeled for consistency with the GSV data. The commercial and government buildings were relabeled as *office buildings*. Also, the religious, educational, and military buildings were merged as *institutional*. Besides, some building footprints assigned irrelevant or general labels such as construction, brownfield, grass, recreation ground, and fairground were relabeled as *other*. Finally, six classes of *industrial*, *institutional*, *office building*, *other*, *residential*, and *retail* were considered for further analysis. The Principal Component Analysis (PCA) transform was applied to the input LiDAR-derived features to equal the number of feature bands to the *ImageNet* dataset. In other words, in order to achieve fine-tuning, the original 13 features were reduced to 3 using PCA transformation.

Table 5-1: Features and their corresponding statistics extracted from LiDAR Point Cloud data. For example, mean, maximum, and standard deviation were calculated in a 3 ×3 moving window in the First Return (FR) image.

Feature	Statistics
FR*	Mean
	Max
	Standard Deviation
LR**	Mean
	Max
	Standard Deviation
Intensity	Mean
	Standard Deviation
Slope	Min
	Mean
	Standard Deviation
	Range
nDSM	Variance

* First Return

**Last Return

5.3.2 Deep Learning models applied for building land-use type classification

CNNs with pre-trained parameters have been used in the literature for classification tasks (Karadal et al., 2021; Kumar et al., 2021; Liu et al., 2020). Some examples of these models are VGG16, MobileNetV2, and Residual Networks, such as ResNet18, ResNet34, and ResNet152. VGG16 was selected for the GSV dataset because Kang et al. 2018 applied this models, and the best results, compared with AlexNet, ResNet18, and ResNet34, were achieved. To explore the suitability of other CNNs, MobileNetV2, ResNet152, and InceptionV3 (Sandler et al., 2018; Simonyan and Zisserman, 2014; He et al., 2016; Szegedy et al., 2016) models were also applied to building land-use type classification. MobileNetV2 and VGG16 models were used for GSV images, and MobileNetv2, ResNet152, and InceptionV3 were tested for the Orthophoto and LiDAR datasets. The parameters, including the optimizer, and the initial learning rate were reported in Table 5-2.

Table 5-2: DL model parameters; optimizer, and initial learning rate; SGD, and Adam are acronyms for Stochastic Gradient Descent, and Adaptive Moment Estimation, respectively

Data	DL Model	Optimizers	Initial Learning Rate
GSV	MobilenetV2	SGD	10^{-1}
	VGG16	SGD	10^{-3}
Orthophoto	MobilenetV2	SGD	10^{-1}
	ResNet152	SGD	10^{-2}
	InceptionV3	Adam	10^{-3}

LiDAR	MobileNetV2	SGD	10^{-6}
	ResNet152	SGD	10^{-3}
	InceptionV3	Adam	10^{-3}

The loss function for all the DL models were set to the categorical cross-entropy. The initial learning rate in all the models was reduced exponentially with a decay rate and decay step of 0.9 and 500, respectively. Each DL model consisted of two parts: 1- Feature Extractor; and 2- Predictor. For all DL models, the Feature Extractor part was kept, and the predictor model was replaced with three layers, including, the Average Pooling, Drop out with 0.2 rate, and a Softmax function for class probability prediction. Finally, the arg max function was applied to the class probabilities, and the class with maximum probability was selected. For the MobileNetV2 model, just a Softmax function was added to the top of the Feature Extractor. Two scenarios were tested for training; 1- training the whole network, including the Feature Extractor and Predictor models, referred to as training from scratch; 2- initializing the weights in the primary layers of the Feature Extractor with pre-trained parameters based on the *ImageNet* dataset and training the other layers with train data, referred to as Transfer Learning.

5.3.2.1 MobileNetV2

MobileNetV2 uses 19 Residual Blocks in its architecture. Each Residual Block consists of three layers. The first layer is a 1×1 convolution layer with a Rectified Linear Unit6 (ReLU6) activation function. The second layer includes the depthwise convolution, and the last layer is a 1×1 convolution layer with a linear activation function. This model has 154 layers in the feature extractor. Figure 5-5 shows the network architecture.

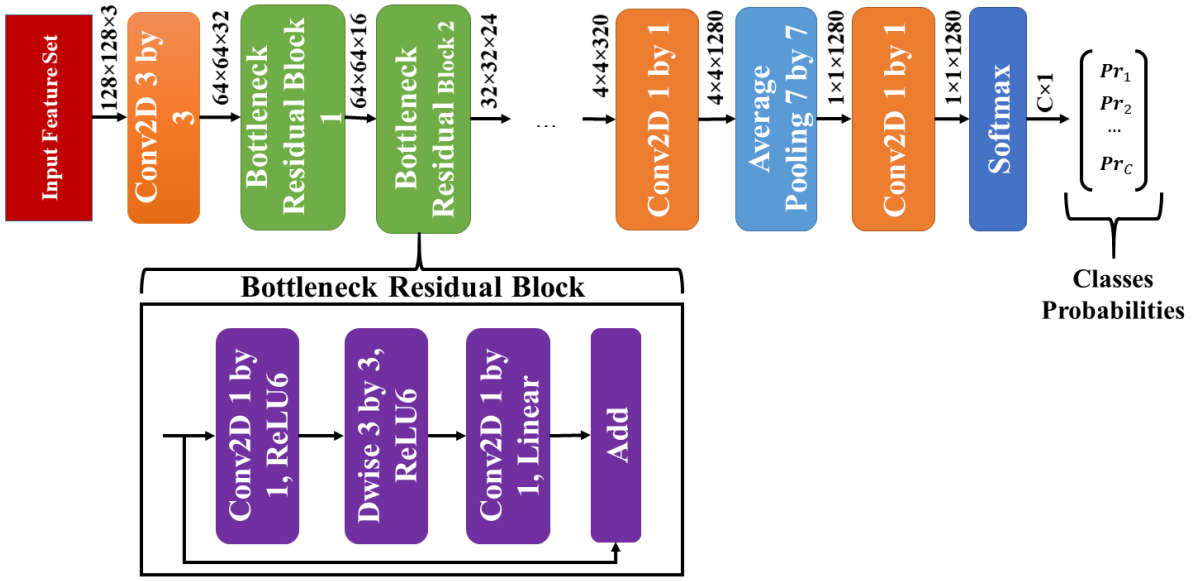


Figure 5-5: MobileNetV2 model architecture ; ReLU6, and Dwise refer to the Rectified Linear Unit function limited to the maximum value of 6, and Depth-wise separable convolutions, respectively; Pr_c represents the probability for class c

5.3.2.2 VGG16 Model

VGG16, VGG stands for Visual Geometry Group, and 16 refers to the number of convolutional layers. This model has a smaller receptive field than the earlier CNNs, AlexNet, and ZFNet (Krizhevsky et al., 2017; Zeiler and Fergus, 2014). In other words, instead of having a 7×7 or 11×11 convolution layer, VGG16 uses three layers of a 3×3 convolution layer but adds more depth to the network. The smaller receptive field has three advantages: 1- It makes the activation functions more discriminative and increases the network's ability to converge faster; 2- It reduces the number of the network parameters; 3- By replacing the 7×7 convolution layer with three layers of 3×3 convolution and adding ReLU non-linearity between layers, the chance of overfitting is reduced. This model has 13 layers in the feature extractor. Figure 5-6 shows the network architecture.

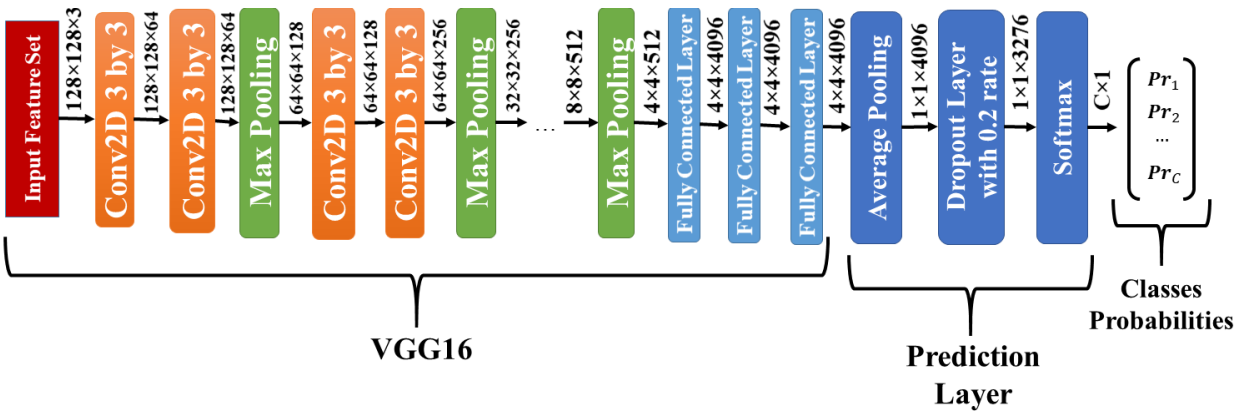


Figure 5-6: VGG16 model architecture; Prc represents the probability for class c

5.3.2.3 ResNet152

ResNet152 stands for Residual Network with 152 deep layers, and is deeper than its 34 or 101-layer counterparts. The main idea of Residual Networks is applying skip connections. It means the information flow can skip intermediate layers, and the feature maps can be connected to the following layers directly. There are two types of skip connections shown in Figure 5-7: 1- The solid line skip connection for the case when the feature map does not need upscaling; 2- The dotted-line skip connection: this connection is for when the feature map should be increased in size. This upscaling is accomplished by either the zero padding or a 1×1 convolution.

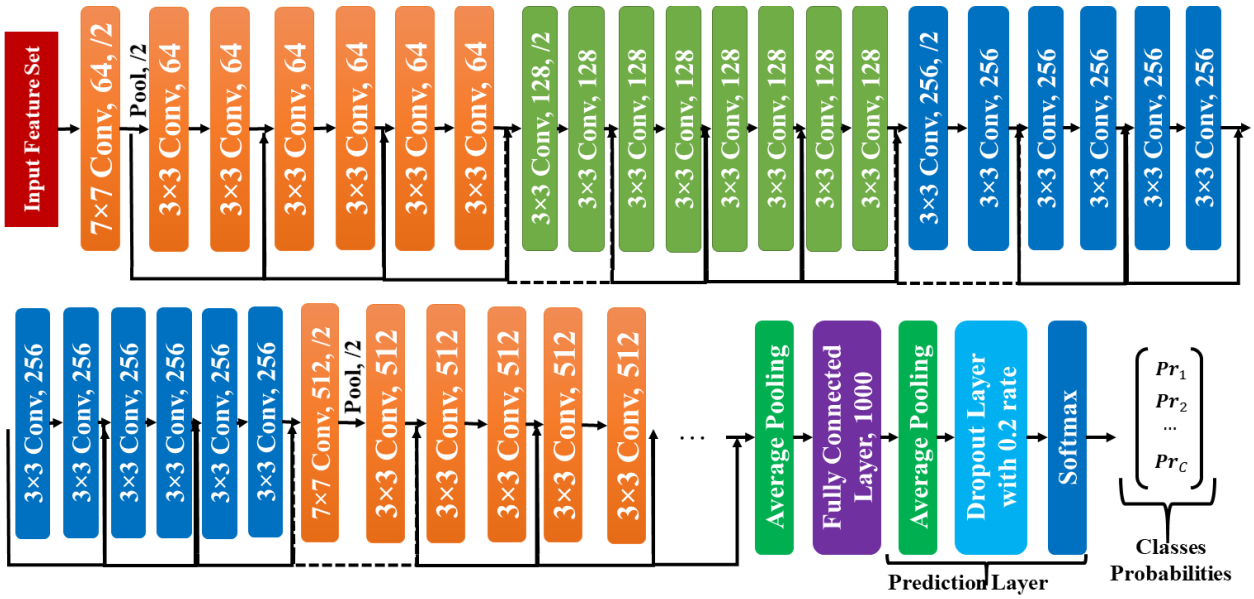


Figure 5-7: ResNet152 model architecture ; just 34 layers were shown for brevity purposes; the arrows between convolution blocks show the skip connections; the numbers after Conv, show the output feature map depth; for example, Conv, 64 means the layer outputs a feature map with 64 layers; Pol, /2 represents the pooling layer halving the output size; the prediction layer was added to the top of the feature extractor part; Pr_c represents the probability for class c

5.3.2.4 InceptionV3

InceptionV3 model is a variant of the Inception networks, which use the Inception modules in their architecture. Inception modules use convolution factorization in their structure. In other words, they replace n×n convolution layer with two layers of n×1 and 1×n convolutions. The convolution factorization can save computational power. Figure 5-8 shows the InceptionV3 model architecture. The network consists of the Inception, Reduction modules, and the Auxiliary Classifier Block, shown separately in Figure 5-9. The Reduction module is embedded into the network to avoid the representational bottleneck and reduce the computational burden. The Auxiliary Classifier Block improves the network convergence and pushes the useful gradients to the lower layers (Szegedy et al., 2016). The model has 313 layers in the feature extractor.

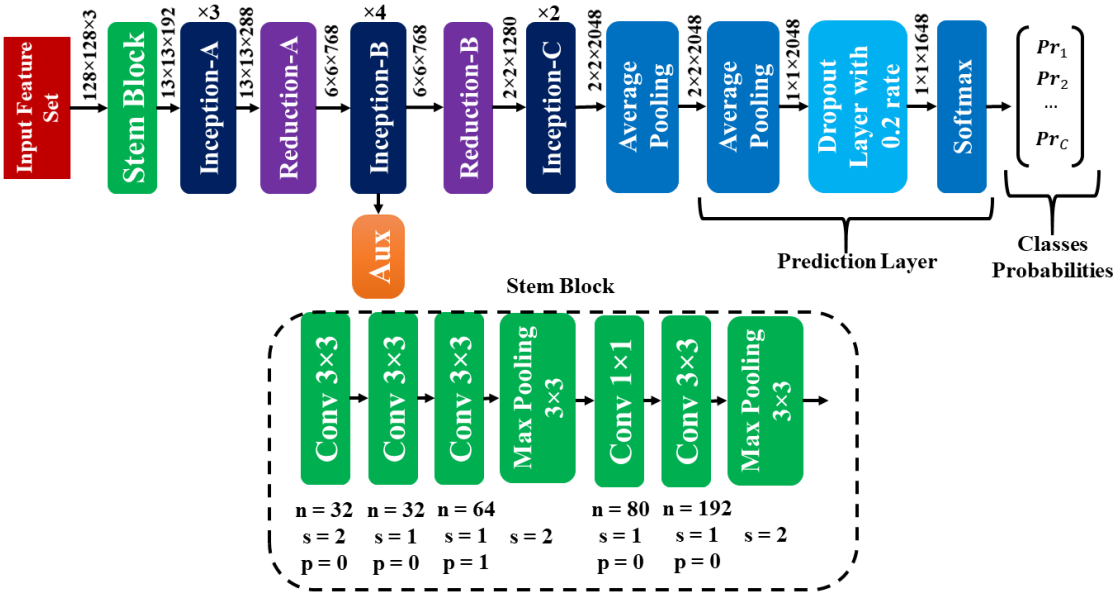


Figure 5-8: InceptionV3 model architecture with Stem block ; n, s, and p refer to the number of feature layers, stride, and padding parameters; Aux represents the Auxiliary Classifier Block; Pr_c represents the probability for class c

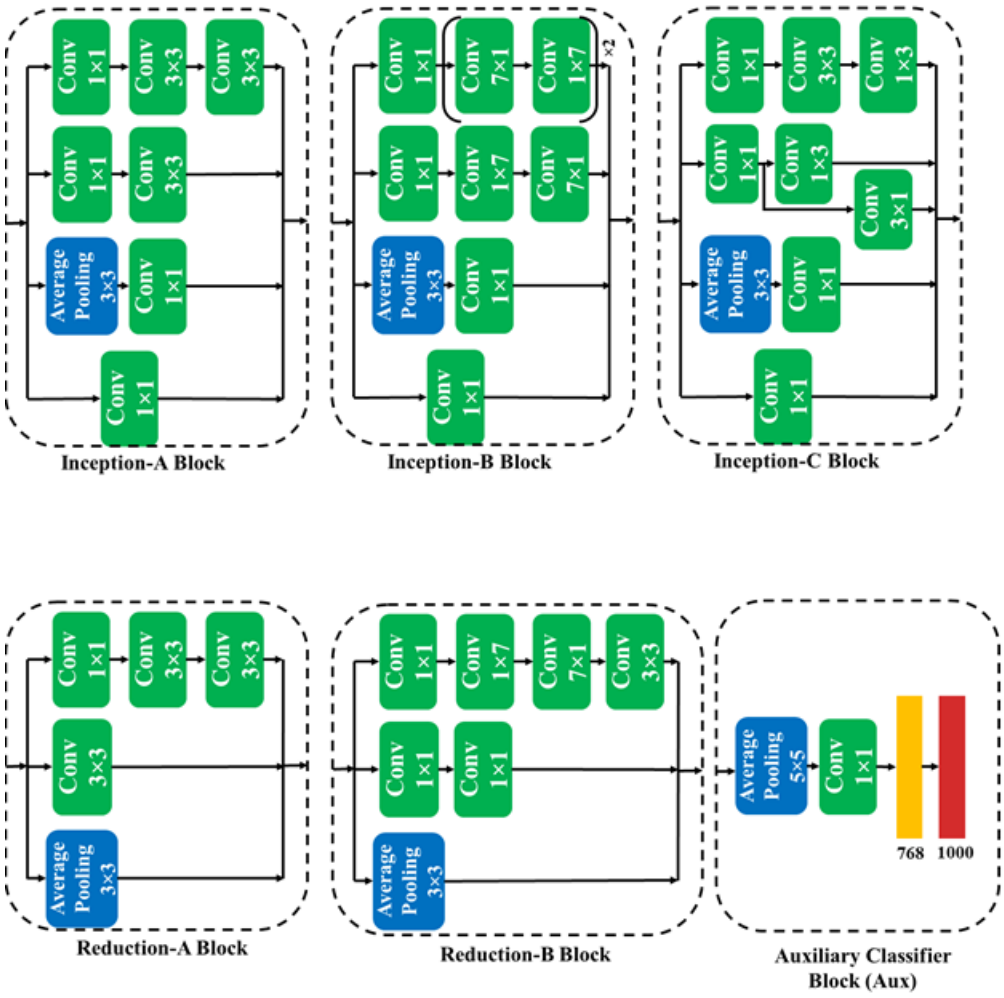


Figure 5-9: Inception, Reduction, and Auxiliary Classifier Blocks in InceptionV3 model

5.3.3 Fusion methods

The building land-use type classification maps from Orthophoto, LiDAR, and GSV images were combined using two fusion methods. We proposed a methodology based on ranking classes using the F1 Score. The second method applied the Fuzzy Fusion concept to combine classification maps. The methods were explained in this section.

5.3.3.1 Ranking classes based on F1 Score

DL models for classification problems are usually evaluated using Precision and Recall rates. Precision for class *c* in a multi-class classification problem is the ratio of correctly classified samples in that specific class divided by the total number of samples classified in class *c* by the DL model (Equation (5-1)). The recall rate for class *c* is the ratio of correctly classified samples in that specific class divided by the total number of samples in the ground truth data for that class (Equation (5-2)). In Equations (5-1) and (5-2), *True C* is the total number of samples correctly classified in class *c*, *False C* is the total number of samples erroneously predicted in class *c*, and *False NC* is the total number of samples belonging to class *c* but were incorrectly predicted in other classes.

$$Precision = \frac{True\ C}{True\ C + False\ C} \tag{5-1}$$

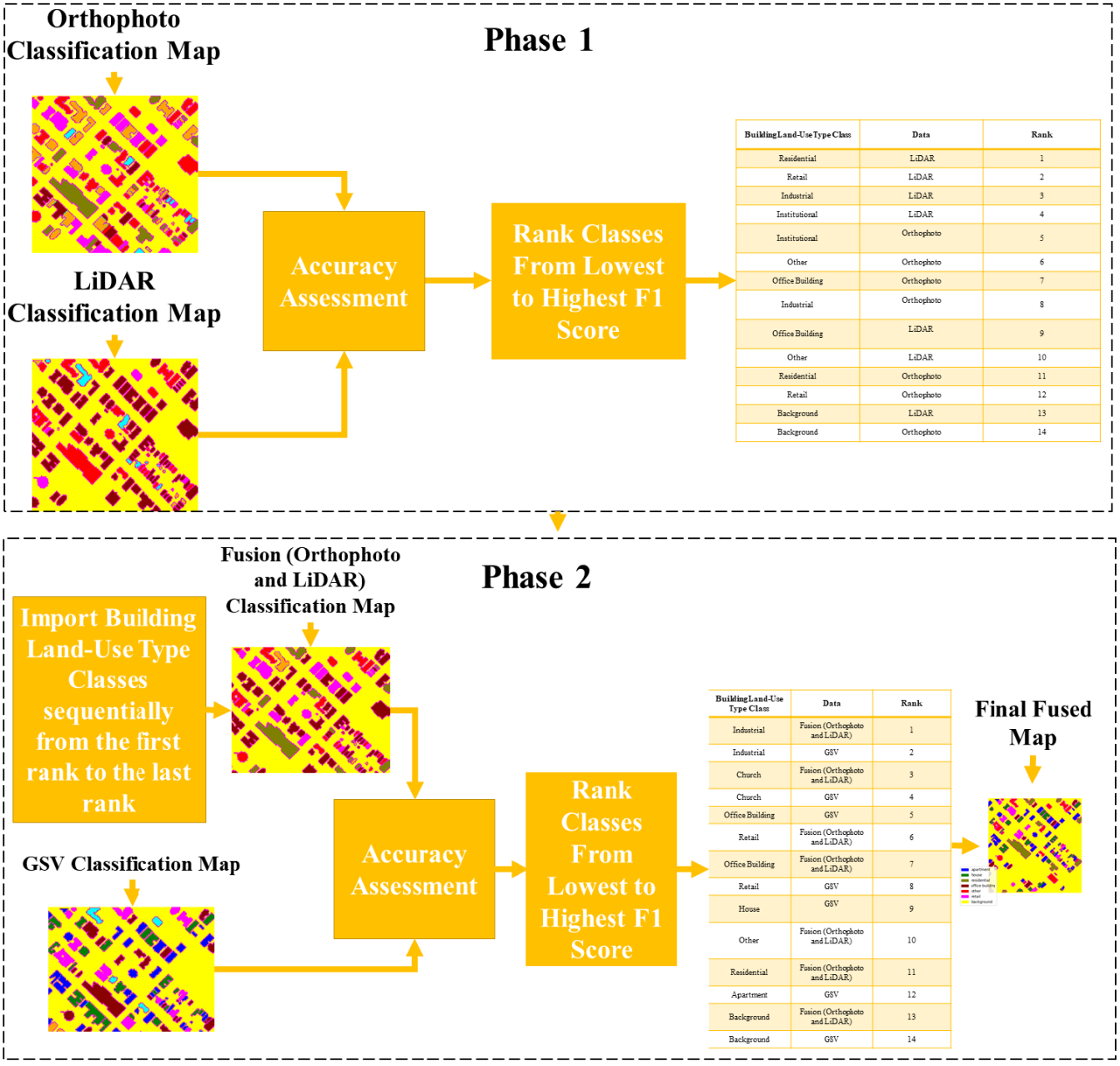
$$Recall = \frac{True\ C}{True\ C + Flse\ NC} \tag{5-2}$$

F1 Score combines Precision and Recall rates and takes the harmonic mean of these two indices as following:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5-3}$$

F1 Score is a suitable metric for evaluating a classifier performance because it considers both accuracy and predictive power of the DL model. Hence, it is a desired metric for per class performance evaluation. When there are multiple classifiers, ranking classes based

on F1 Score can be a useful fusion methodology to compare classes from different information sources. We presented a methodology based on this concept to combine building land-use type classification maps from three data sources, including, Orthophoto, LiDAR, and GSV. This method ranks classes according to their F1 Score values. The class with the lowest F1 Score was ranked 1, and the class with the highest F1 Score was ranked the last. The assigned rankings determine the order in which the pixels for each class is imported to the combined map. For example, if we consider the fused map as a two-dimensional empty array, first, the pixels for the class with the lowest F1 Score are imported to the array, and then the pixels for the class with the second lowest F1 Score are imported, and the last class imported is the class with the highest F1 Score. The sequential import of pixels for each building land-use type class allows classes with low scores to be corrected by classes with higher scores. The combination process was carried out in two phases due to two reasons. Firstly, the GSV ground truth labels were in agreement with the ground truth labels in Kang et al. 2018. However, the LiDAR and Orthophoto ground truth labels were different and were extracted from OSM. Secondly, the method was designed to be easily comprehensible to the end user. In the first phase, Orthophoto and LiDAR classification maps were combined based on the ranking methodology, and in the second phase, the output from Orthophoto and LiDAR maps combination was fused with the GSV classification map using the same F1 Score-based ranking methodology. Please note that GSV classification maps could have been used in the first phase (after matching the labels with either Orthophoto or LiDAR classification), and it should not affect the final classification results. Besides, the proposed fusion method is conducted at the pixel level. Figure 5-10 shows the proposed method, including phases 1 and 2.



Building Land-Use Type Class	Data	Rank
Residential	LiDAR	1
Retail	LiDAR	2
Industrial	LiDAR	3
Institutional	LiDAR	4
Institutional	Orthophoto	5
Other	Orthophoto	6
Office Building	Orthophoto	7
Industrial	Orthophoto	8
Office Building	LiDAR	9
Other	LiDAR	10
Residential	Orthophoto	11
Retail	Orthophoto	12
Background	LiDAR	13
Background	Orthophoto	14

Building Land-Use Type Class	Data	Rank
Industrial	Fusion (Orthophoto and LiDAR)	1
Industrial	GSV	2
Church	Fusion (Orthophoto and LiDAR)	3
Church	GSV	4
Office Building	GSV	5
Retail	Fusion (Orthophoto and LiDAR)	6
Office Building	Fusion (Orthophoto and LiDAR)	7
Retail	GSV	8
House	GSV	9
Other	Fusion (Orthophoto and LiDAR)	10
Residential	Fusion (Orthophoto and LiDAR)	11
Apartment	GSV	12
Background	Fusion (Orthophoto and LiDAR)	13
Background	GSV	14

Figure 5-10: A graphical depiction for F1 Score ranking fusion method

5.3.3.2 Fuzzy fusion based on Gompertz function

Assume we have *M* classifiers with Confidence Factors (*CFs*), $\{CF^1, CF^2, CF^3, \dots, CF^M\}$. These confidence factors resemble the degree of membership to a class, and their sum across different classes of a specific classifier should be equal to 1. In Equation (5-4), *C*, and *M* refer to the total number of classes, and the total number of classifiers, respectively. Also, CF_c^i represents the Confidence Factor for class *c* of *i*th classifier.

$$\sum_{c=1}^{C} CF_c^i = 1 \; \forall i = 1, \dots, M \tag{5-4}$$

The Fuzzy Rank value for class *c* of the *i*th classifier is calculated based on the Gompertz Function and by using the CF_c^i as in Equation (5-5):

$$R_c^i = \left(1 - \exp\left[-\exp\left[-2 \times CF_c^i\right]\right]\right), \forall i = 1, \dots, M; c = 1, \dots, C \tag{5-5}$$

The values of R_c^i lies in the range [0.127, 0.632]. The smallest value, 0.127, is analogous to the highest confidence value resulting in the lowest (best) rank.

The Final Decision Score (*FDS*) for class *c* is calculated based on the Fuzzy Rank Sum (*FRS*) for class *c* (Equation (5-6) and the Complement of Fuzzy Rank Sum (*CFRS*) of class *c* (Equation (5-7)) :

$$FRS_c = \begin{cases} R_c^i & if\ R_c^i \in K^i \\ P_c^i, & otherwise \end{cases} \quad (5\text{-}6)$$

$$CFRS_c = \frac{1}{M}\sum_{i=1}^{M} \begin{cases} CF_c^i & if\ R_c^i \in K^i \\ P_c^{CF}, & otherwse \end{cases} \quad (5\text{-}7)$$

In the Equations above, K^i refers to the top *k* ranks for class *c*, i.e. ranks from 1, …, *k*. P_c^i and P_c^{CF} are the penalty factors imposed on class *c* if it does not belong to the top *k* ranks. P_c^i is set to 0.632 and is achieved by putting $CF_c^i = 0$ in Equation (5-5). P_c^{CF} is set to 0.

The Final Decision Score (*FDS*) for class *c* is calculated using Equation (5-8) as follows:

$$FDS_c = FRS_c \times CFRS_c \quad (5\text{-}8)$$

In the Equation above, FRS_c refers to the Fuzzy Rank Sum for class *c* and $CFRS_c$ represents the Complement of Fuzzy Rank Sum for class *c*. The final predicted class of instance *I* of the dataset is calculated by taking the class with the minimum *FDS* score as in Equation (5-9):

$$class\ (I) = \underset{c = 1, 2, \ldots, C}{argmin}\ \{FDS_c\} \quad (5\text{-}9)$$

5.3.4 Accuracy assessment

We accomplished the accuracy assessment at two different scales; 1- Pixel-Based; and 2-Object-Based. In the Pixel-Based accuracy assessment, Precision, Recall, F1 Score, and Overall Accuracy were calculated by considering each pixel in the ground truth map as a test sample. For the Object-Based case, the accuracy metrics were achieved by assuming each building footprint in the ground truth map as a test sample. The Equations for the Precision, Recall, and F1 Score were mentioned in section 5.3.3.1, and Equation (5-10) presents the formula for the Overall Accuracy:

$$Overall\ Accuracy = \frac{\sum_{i=1}^{C} True\ i}{Total\ Number\ of\ Test\ Samples} \quad (5\text{-}10)$$

In the Equation above, *True i*, refers to the number of correctly classified samples in the *i*th class, and *C* is the total number of classes.

5.4 Experiments

This section includes the results and discussion on building land-use type classification using GSV, LiDAR, and Orthophoto separately and fusion of the classifiers. Per-class accuracies and learning curves for the best DL models were reported in the Appendices A-C.

5.4.1 Experiments on Google Street View Image

Two DL models were examined on GSV images, including MobileNetV2 and VGG16. MobileNetV2 has 154 layers in the feature extractor part, and five scenarios were examined based on the number of trained layers. In the first case, all the weights in all layers of the feature extractor network (154 layers) were trained. In the second, third, and fourth cases, 150, 100, and 50 layers out of 154 were trained, and the pre-trained weights based on

the *ImageNet* dataset were used for the rest of the layers. In the last scenario, all the weights in 154 layers were frozen, and none of the weights in the feature extractor were trained. In other words, just the weights in the prediction layers were trained, and the weights trained on *the ImageNet* dataset were used in the feature extractor. Table 5-3 shows the average training, validation, and test accuracies across five folds and the training time (in hours) for each above-mentioned transfer learning scenario. The number of trained layers in the Table refers to the number of trained layers in the feature extractor. The highest accuracies and the lowest training time have been bolded in the table. The highest overall training, validation, and test accuracies were for the MobileNetV2 with 150 trained layers, with accuracies of 89.63%, 72.17%, and 94.28%, respectively. These accuracies dropped to 50.48%, 58.87%, and 81.62% when none of the layers were trained. The training for the network with 150 trained layers took longer than other cases, about 15 hours longer than the fastest model. While it was expected that the fastest training time would be for the case when all the weights were frozen, the fastest model among all MobileNetV2 models was for the network with 100 trained layers. The reason is the early stopping condition defined when training the models. Based on the condition, the algorithm would be stopped if the validation accuracy did not change more than 1% over 100 epochs. In other words, in the model for which the validation accuracy converges more quickly, the training time would be the least. Although this model was the fastest, the training, validation, and test accuracies were about 0.5%, 1%, and 1.5% lower than the best MobileNetV2 model.

Table 5-3: Average accuracies across five folds and training times based on the number of trained layers. Bold values represent the highest accuracies and shortest training time in each column and method.

Model	Number of trained layers	Average training accuracy (%)	Average validation accuracy (%)	Average Test accuracy (%)	Training time (hours)
MobileNetV2	154 (from scratch)	89.26	72.78	93.87	22.07
	150	89.63	72.17	94.28	32.37
	100	88.94	71.08	92.76	17.61
	50	87.03	71.02	93.03	25.45
	0	50.48	58.87	81.62	19.33
VGG16	13 (from scratch)	72.66	71.27	92.15	23.96
	10	72.94	71.06	92.86	29.33
	5	72.17	71.38	92.19	22.09
	0	44	54.15	74.61	22.06

VGG16 model has 13 layers in the feature extractor part, and four transfer learning scenarios were explored. In the first scenario, all the layers in the feature extractor were trained, and in the second, and third scenarios, 10 and 5 layers out of 13 were trained,

respectively. The weights for the rest of the layers were kept frozen. In the last scenario, all the weights were kept frozen, and pre-trained weights based on *the ImageNet* dataset were used. The fastest VGG16 model, with a training time of 22.06 hours, was the model in which the whole parameters were kept frozen. This result can be justified based on the lower number of trainable parameters in this network compared with other VGG16 transfer learning cases. The highest train and test accuracies, with values of 72.94% and 92.86%, were for the model with 10 trained layers. Similar to MobileNetV2, the best model in terms of test accuracy, resulted in the highest training time, 29.33 hours, about 7 hours longer than the fastest VGG16 model. All the average accuracy indices dropped significantly, about 29%, 17%, and 18% for train, validation, and test accuracies after freezing the whole weights in the feature extractor and using the pre-trained weights based on the *ImageNet* dataset.

Based on Table 5-3, the train and validation accuracies were 4-30% lower than test accuracy because the internal cross validation was applied for accuracy assessment. It means the test data come from the same population as the train and validation data. To combat this issue, the experiments were repeated on independent test data from GTA.

5.4.1.1 Examining the generalization ability of DL models trained on GSV images for Greater Toronto Area

The generalization ability of the trained models was examined on an independent test dataset created for the GTA. The images were extracted from real-estate websites using the web scraping technique. These websites included LoopNet, Royallepage, and Remax. There was no image available for some addresses on the website. In these cases, the GSV images were used and downloaded using Google API. When downloading the GSV images, heading angles of 0, 90, 180, and 270 degrees and Field Of View (FOV) of 22.5, 45, and 90 degrees were traversed to expand the GTA database. The extra images created using this technique were labeled based on google maps and street view inspection. The labels for other images were extracted from the above-mentioned real-estate websites.

There was no building land-use type information available on the Remax website, so the labels were exploited from the OSM. Table 5-4 shows the number of images for each class.

Table 5-4: Number of images in each class for GTA dataset

Class	Number of images
apartment	149
house	465
industrial	95
mixed r/c	9
office building	28
retail	63

Based on the test accuracies in the previous section, the best MobileNetV2 and VGG16 models were selected for building land-use type prediction for GTA. Figure 5-11 shows the building land-use type classification results for GTA using MobileNetV2.

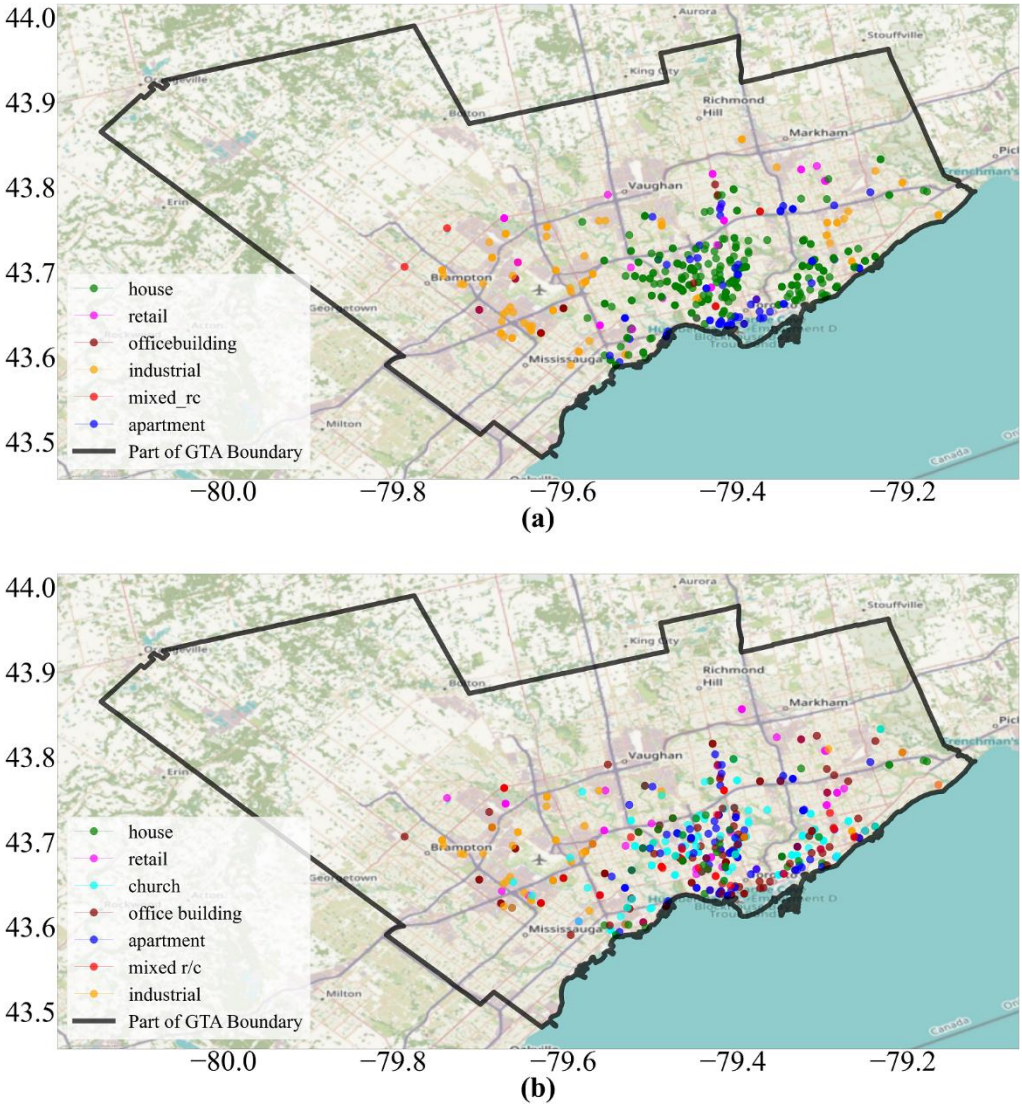


Figure 5-11: MobileNetV2 building land-use type classification result; many houses (green dots) were misclassified as churches (cyan dots); (a): Ground truth; (b) Predicted Map

Based on Figure 5-11, many houses were misclassified as churches. The reason for this confusion is the similarities between the two classes in GSV images. Another type of

building rarely discriminated from the house because of the similar appearance was *mixed r/c*. For example, Figure 5-12 shows some houses in GSV images misclassified as *mixed r/c*. On the first row, the images show the houses having a similar appearance (sloped roof between the first and second floor and some windows on the second floor) to the *mixed r/c* image in the second row.

Figure 5-12: Houses misclassified as *mixed r/c* because both building types have a sloped roof between the first and second floor and some windows on the second floor; the images on the first row are houses misclassified as *mixed r/c*; the image on the second row is a *mixed r/c* building

It is important to take into account that the accuracy values presented in section 5.4.1 (Table 5-3) may not accurately represent the confusions mentioned above. This is because the results were obtained using GSV images, which differ from the real estate website images used for evaluation. Although the images from real estate websites used for accuracy assessment in this section may appear similar to GSV images visually, they may have inconsistent image characteristics as they are from a different source.

5.4.2 Experiments on LiDAR-derived features

Because of the limitations of building land-use type detection using GSV images, for example similarities between apartments, and office buildings, and houses to churches in a GSV image, we also explored the building land-use type classification using LiDAR and Orthophoto images. The results for the analysis were reported in sections 5.4.2 and 5.4.3.

5.4.2.1 Influence of DL model and learning rate on building land-use type detection accuracies when training models from scratch

Three DL models were tested for building land-use type classification using LiDAR-derived features, including MobilenetV2, ResNet152, and InceptionV3. For each model, different learning rates, including, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1} were explored to find the suitable value for building-land use type classification. Two scenarios were tested for training, including, training from scratch and transfer learning.

Figure 5-13 shows the results for each DL model when training from scratch. InceptionV3 generally achieved superior performance compared with other DL methods. The classification accuracies for Residential buildings were higher than other building types, with a maximum accuracy of 64.33% when using InceptionV3 and setting the learning rate to 0.1. The building land-use type classification using LiDAR-derived features generally resulted in lower accuracies than GSV and Orthophoto.

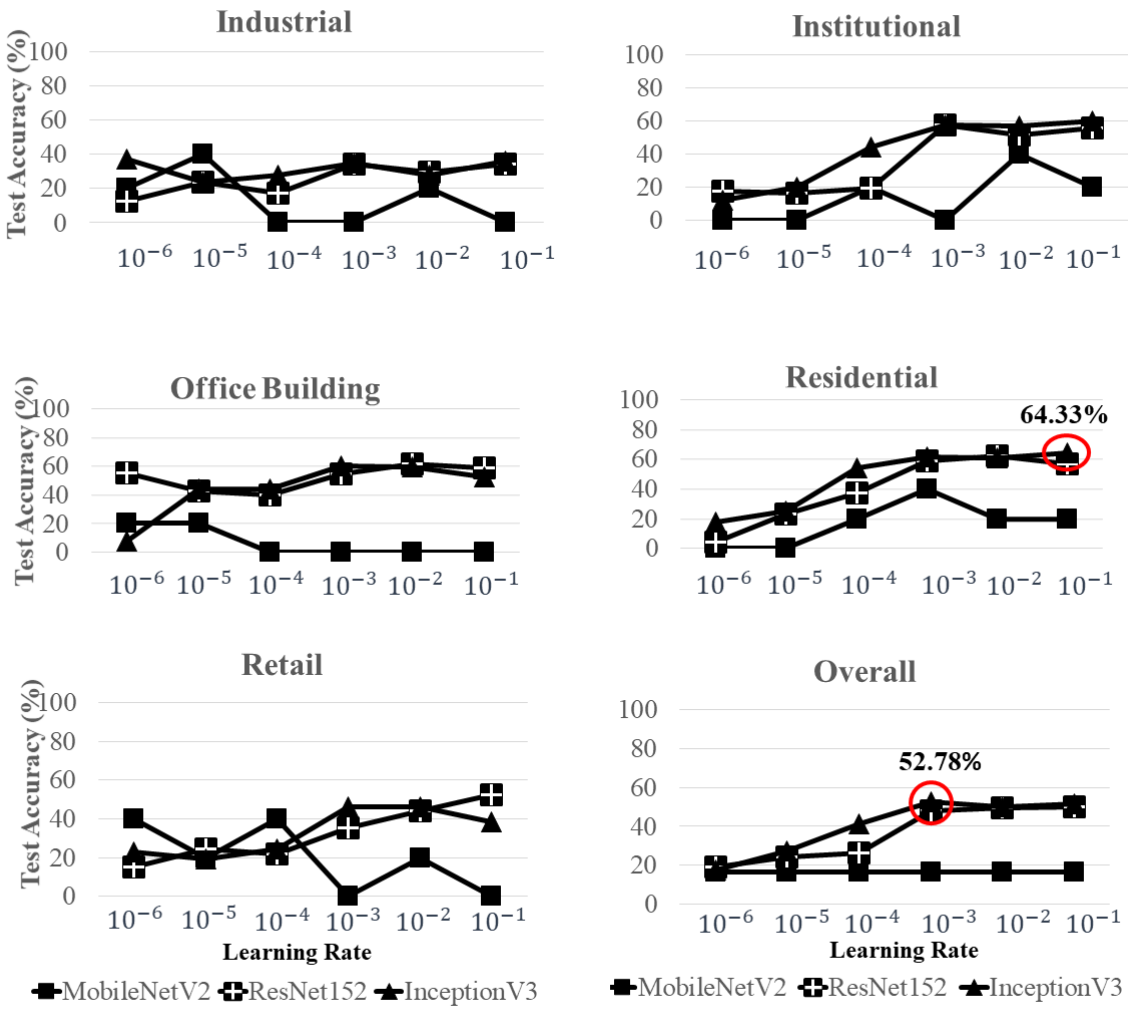


Figure 5-13: Building land-use type classification accuracies for LiDAR-derived features when training from scratch and using DL models, including, MobileNetV2, ResNet152, and InceptionV3 (the red circles show the highest accuracy for the residential class with 64. 33% test accuracy and the highest overall accuracy with a value of 52.78%)

Based on accuracies for training from scratch, the DL model with the highest overall accuracy, the InceptionV3, was selected for transfer learning. Different numbers of trained layers, including, 50, 100, 150, 200, 250, and 300 were examined for building land-use type classification, but transfer learning was not successful on LiDAR data, and all accuracies fell below 50%. Hence, DL models trained using the transfer learning strategy were not used in the fusion part.

5.4.3 Experiments on Orthophoto images

This section presents the building land-use type classification accuracies when using orthophoto images.

5.4.3.1 Influence of DL model and learning rate on building land-use type detection accuracies when training from scratch

Figure 5-14 shows the orthophoto results with learning rates, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}, when training all the parameters. The figure shows that the model performance depends on the learning rate parameter significantly. The highest accuracy was for the residential class, with 71% accuracy when using the ResNet152 model with a learning rate of 10^{-2}. This result was about 7% higher than LiDAR-derive features. The second highest accuracy was for the InceptionV3 model, with an accuracy of 70% for the residential class when setting the learning rate to 10^{-6}.

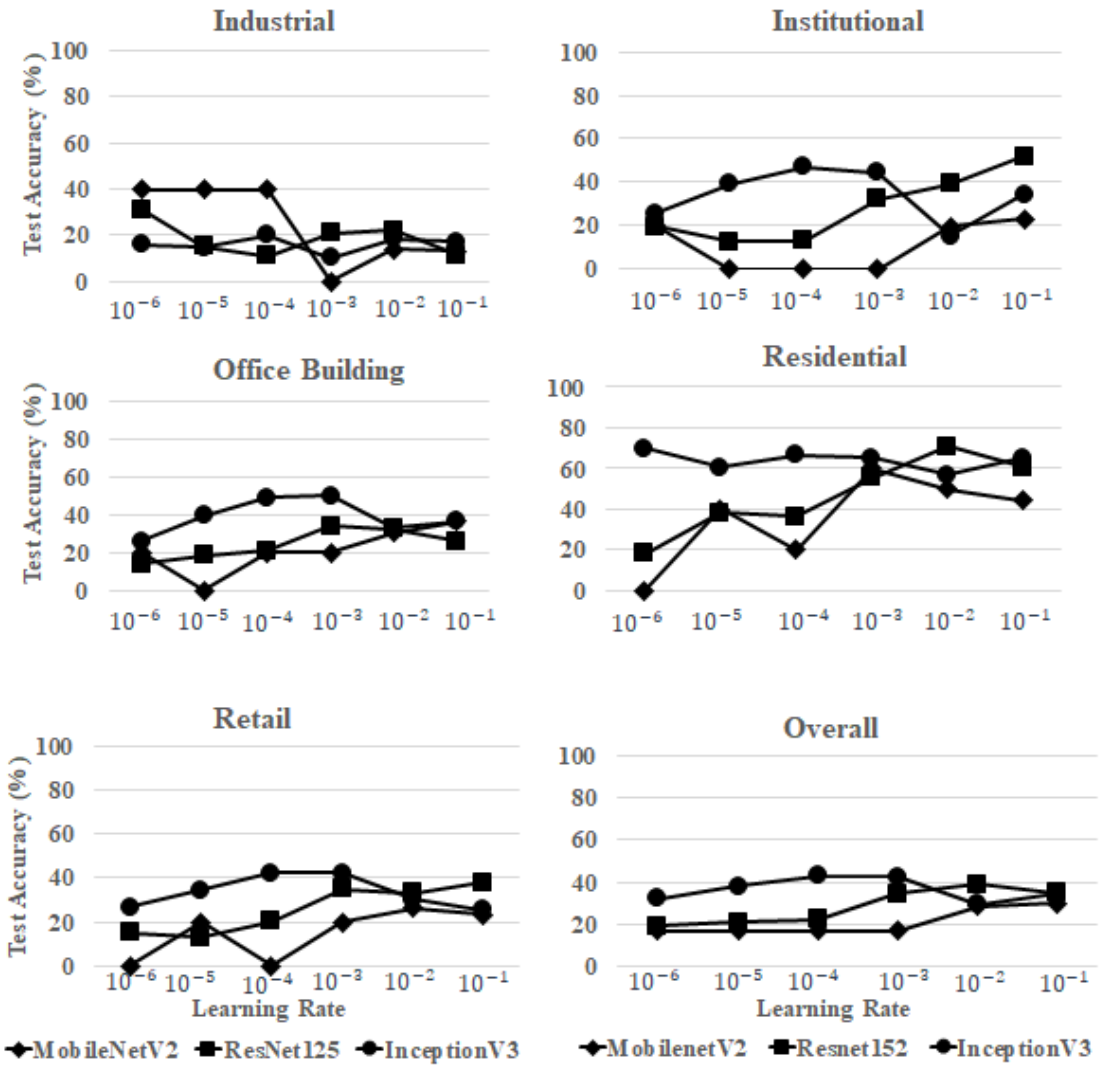


Figure 5-14: Building land-use type classification accuracies for orthophoto when training from scratch and using DL models, including, MobileNetV2, ResNet152, and InceptionV3 (the highest accuracy was for residential class with 71% test accuracy in the best case)

5.4.3.2 Influence of DL model and learning rate on building land-use type detection accuracies when using transfer learning

Transfer learning strategy was also examined on orthophoto dataset, and the results were shown in Figure 5-15. The transfer learning strategy was more efficient than LiDAR case because of the spectral similarity (RGB bands) between *ImageNet* data and orthophoto images. When training 150 layers, with learning rate 10^{-3}, the accuracy of 81.33% was achieved for residential class. The worst performance was for the case of using the default parameters (Szegedy et al., 2016).

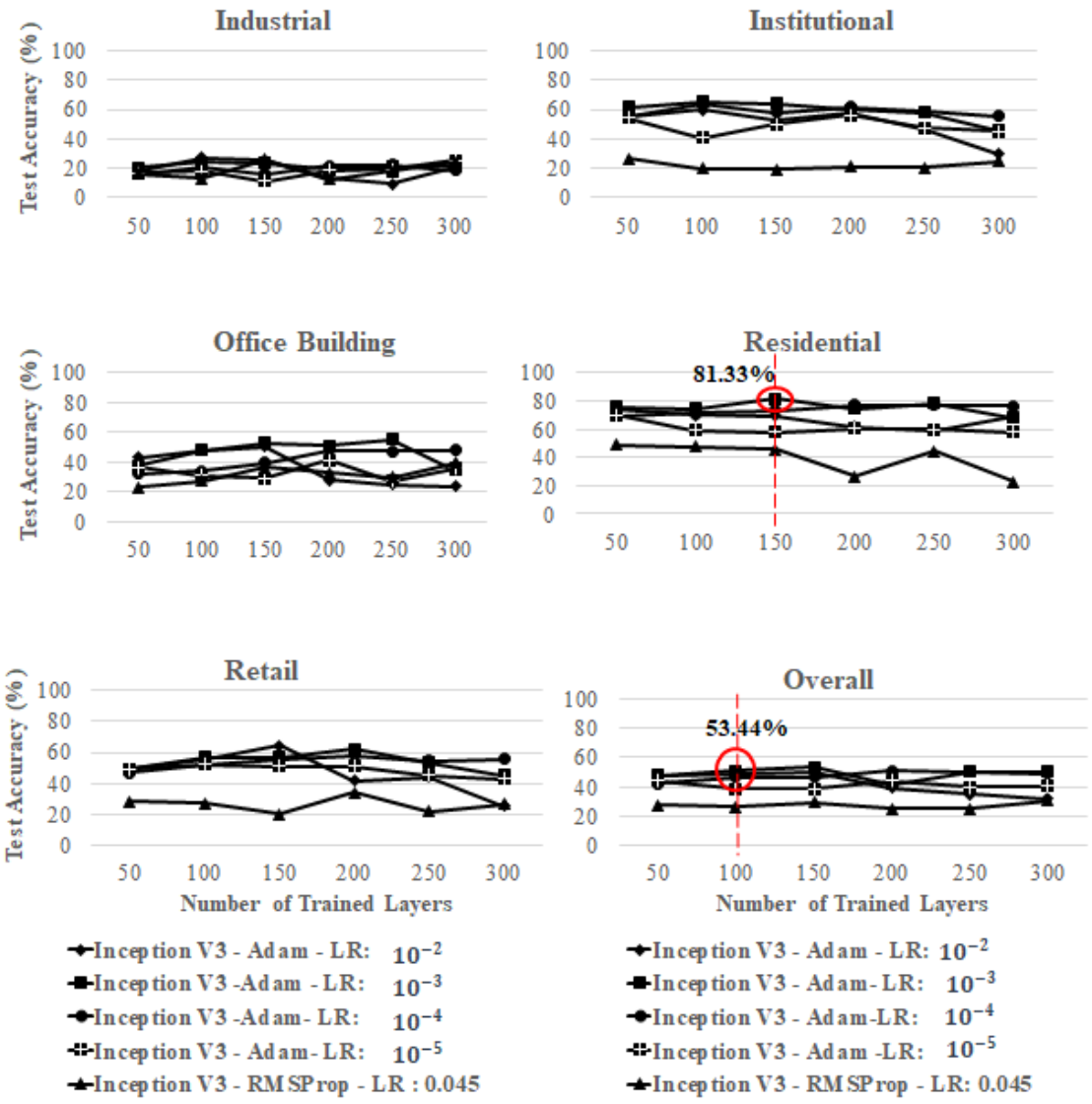


Figure 5-15: Building land-use type classification accuracies for orthophoto when using transfer learning and InceptionV3 (the red circles show the highest accuracies for the residential class with a value of about 81% and overall with a value of about 53%; the vertically dotted red line shows the number of trained layers that resulted in the highest accuracies); LR is an acronym for Learning Rate; Adam and RMSProp refer to the Adaptive Moment Estimation and Root Mean Square Propogation optimizers, respectively.

5.4.4 Deep Learning models training time

DL algorithms have millions of parameters, and training this huge amount of parameters is time-consuming. Finding the trade-off between training time and accuracy has always been one of the challenges in DL models. Figure 5-16 shows the training time when training all the parameters for LiDAR-derived features and Orthophoto images with learning rates, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}. Most models had training time of less than 1.5 hours except ResNet152 trained on orthophoto images, with training time fluctuating between 2-3 hours. DL model, GPU type, optimizer, and input dataset contribute to the training speed. The most time-consuming model was ResNet152 on orthophoto images because the model has 60.4 million parameters, the highest among the three models. The least training time was for MobileNetV2 on both LiDAR and orthophoto data because this model has the lowest number of parameters compared with InceptionV3 and ResNet152.

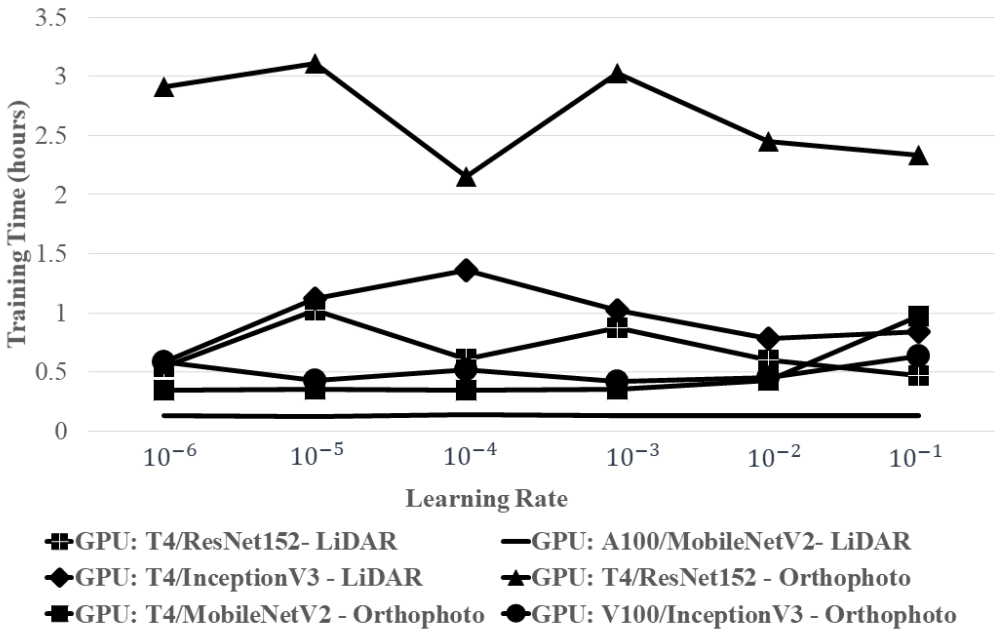


Figure 5-16: Training time for DL models; T4, A100, and V100 refer to the GPU types

5.4.5 Fusion of Orthophoto, LiDAR and GSV

The GSV, LiDAR, and orthophoto classification maps were fused to improve the generalization ability of GSV classifications. We examined two fusion methods. 1- Ranking Classes Based on the F1 Score, and 2- Fuzzy Fusion based on Gompertz Function (Kundu et al., 2021) mentioned in sections 5.3.3.1 and 5.3.3.2. Two ground truth data used by Kang et al. 2018 were tested in this section. The dataset included building land-use type labels for Vancouver and Fort Worth. The ground truth datasets are shown in Figure 5-17.

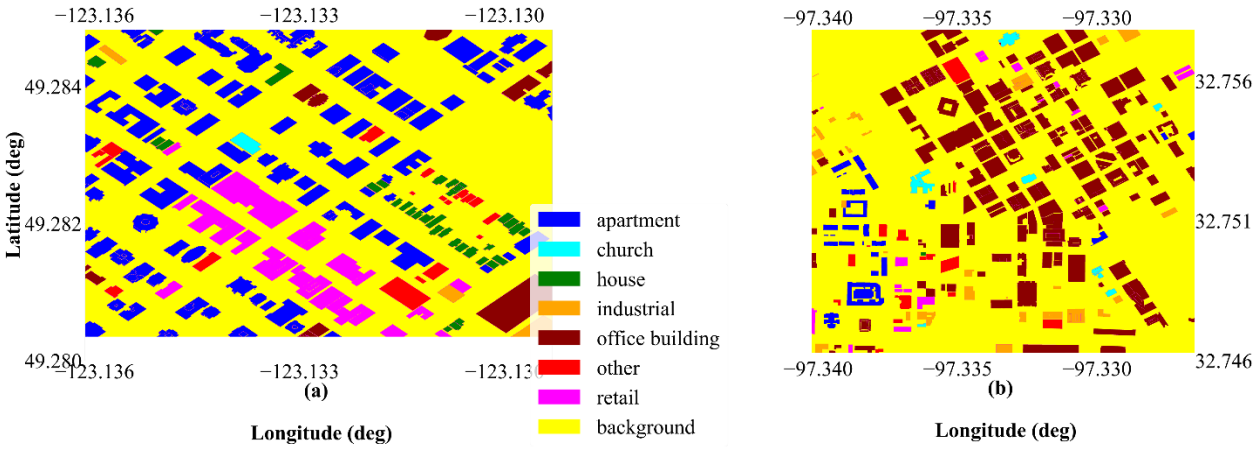


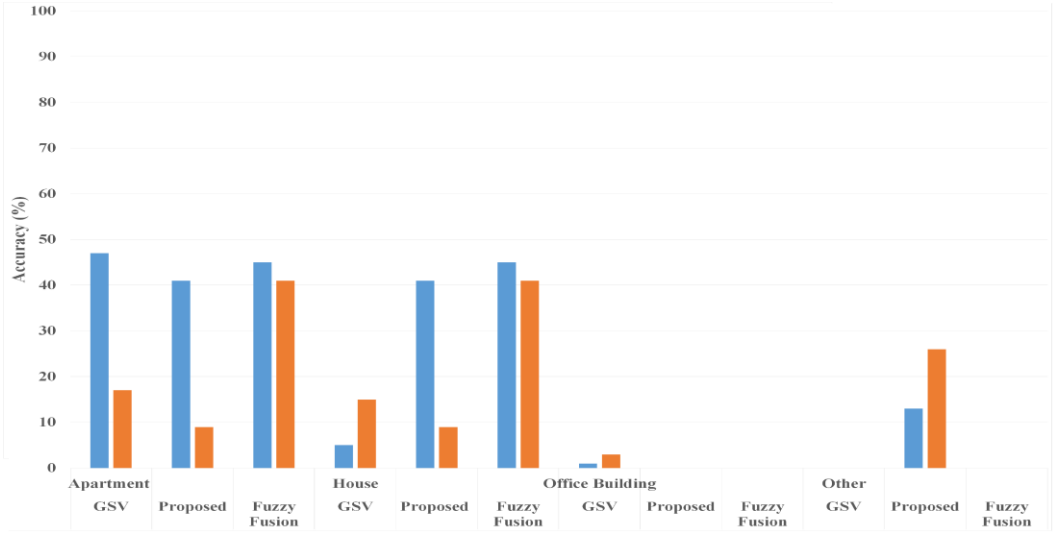
Figure 5-17: Ground truth labels for (a): Vancouver; (b): Fort Worth

The results in Figure 5-18 show the precision and recall rates for each building land-use class in the City of Vancouver, as well as the overall accuracy. In this experiment, the building types of church and industrial were not included because they had very few samples in the ground truth, and none of the classifiers were able to detect these building types before or after fusion.

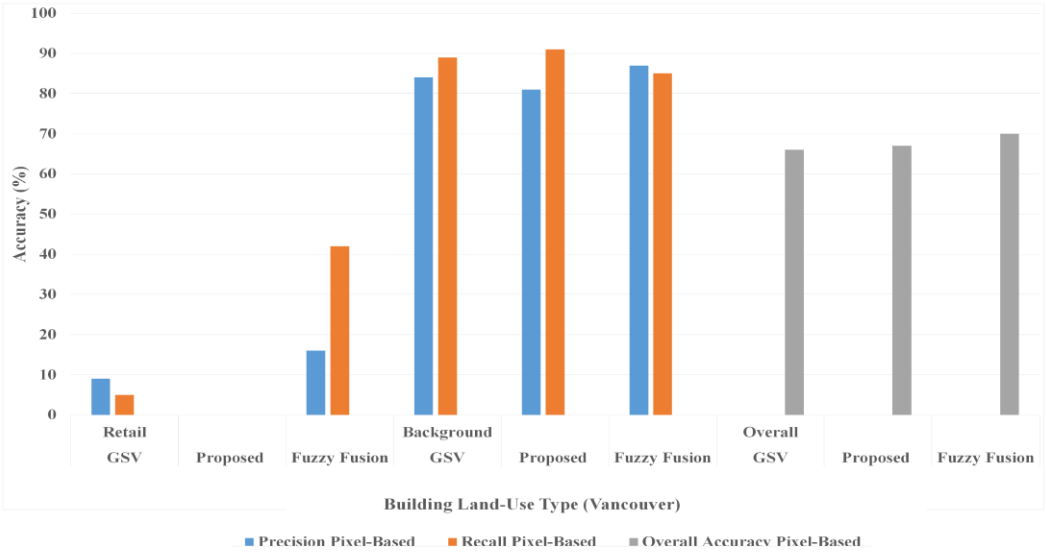
The proposed method was able to improve the precision of house detection by 36% compared to GSV, achieving a value of 41%. In addition, the proposed method was able

to detect buildings with building land-use class *other* with precision and recall rates of 13% and 26%, respectively, while GSV was not able to detect them at all. The *background* recall rate for the proposed method was 91%, which was 2% higher than the corresponding value for the GSV result. Overall accuracy also improved by 1% after using the proposed method, achieving a value of 67%.

Although the proposed method showed superior performance for *background* recall rate, with 6% higher rate than the fuzzy fusion method, and 13% and 26% precision and recall rates for class *other*, respectively, the fuzzy fusion method generally showed better performance for pixel-based accuracy indices in the City of Vancouver. The fuzzy method showed better performance for *apartment* and *house*, with 4% and 32% improvement in precision and recall rates, respectively. Additionally, precision and recall rates for class *retail* improved after using the fuzzy fusion method, with values of 16% and 42%, respectively. Another improvement when using the fuzzy fusion method was in terms of background precision, which improved from 81% to 87%. Furthermore, the overall accuracy also improved by 3%, achieving a value of 70% when using the fuzzy fusion method.



(a)



(b)

Figure 5-18: Pixel-based precision (blue bars), recall (orange bars), and overall accuracies (gray bars) for GSV, proposed method, and fuzzy fusion. (a): Precision and recall indices for building land-use type classes apartment, house, office building, and others. (b): Precision and recall indices for classes retail and background, and overall accuracies.

Object-based per-class precision, and recall, as well as overall accuracies in the City of Vancouver are presented in Figure 5-19. The *office building* class was excluded from the analysis due to the absence of building footprints in both pre and post-fusion stages. Similarly, the *background* class was not considered in the object-based analysis, as it is a pixel-based class and does not include any building footprint. After applying the proposed fusion method, there was a 21% and 11% increase in precision and recall rates for the *apartment* class, respectively, resulting in values of 72% and 39%. The proposed method also led to a 2% and 17% improvement in the precision and recall rates of the *house* class, respectively. However, the fusion method had an adverse effect on the *other* class, with no building footprint being classified under this category. The proposed method did not affect the performance of the *retail* class, and no buildings were detected in this group before or after fusion. Overall accuracy was enhanced by 20% after applying the proposed fusion method, with a more significant increase compared to the 1% improvement in the pixel-based overall accuracy.

Generally, the fuzzy fusion method resulted in a higher performance than the fuzzy method in terms of object-based accuracy indices. For example, the recall rate for *apartment* was 24% higher than the proposed method and achieved a value of 63%. Also, the precision and recall rates for *house* were 12% and 43% higher than the proposed method when using the fuzzy fusion algorithm, with values of 39% and 63%. However, the proposed method had a better performance than the fuzzy fusion method in terms of *apartment* precision. When using the proposed method, the index was higher by 33%. Although both fusion methods degraded the class *other* performance, the fuzzy fusion improved class *retail* precision and recall rates. Before fusion and when using the proposed method, no building footprint was detected in *retail*. However, when using the fuzzy fusion, the precision and recall rates of 35% were achieved. Because of the improvements in class performance when using the fuzzy fusion method, the overall accuracy when using this method was 2% higher than the proposed algorithm.

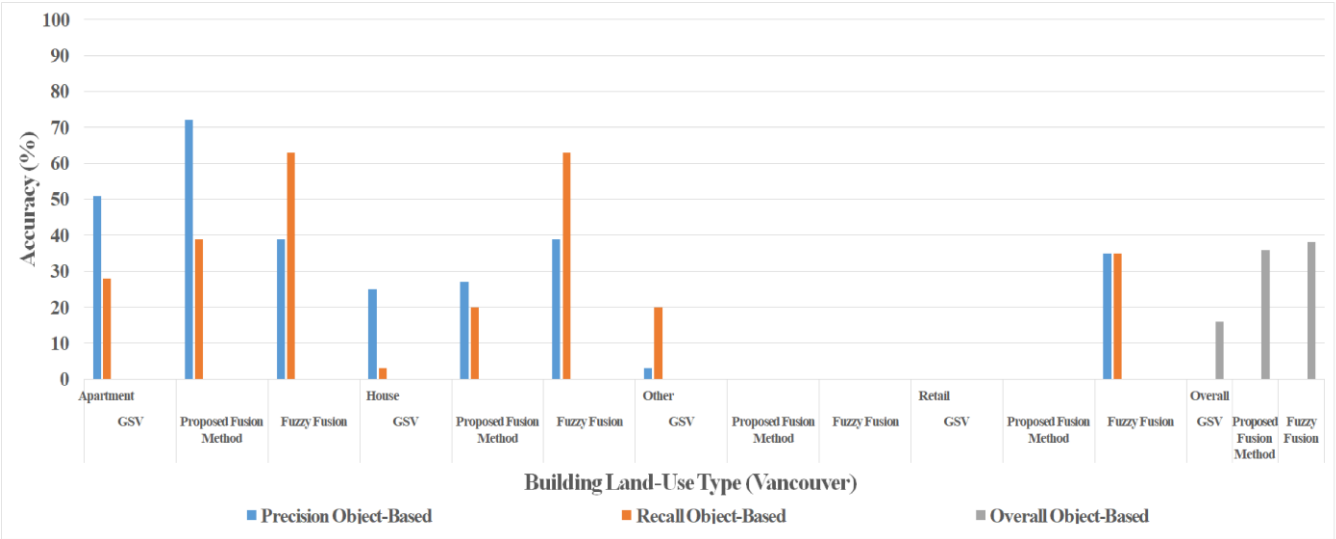


Figure 5-19: Object-based precision (blue bars), recall (orange bars) for each building land-use type and overall accuracy (gray bars) indices in the City of Vancouver. The results are for GSV, proposed fusion method and fuzzy fusion classifications.

Figure 5-20 shows the pixel-based per-class precision and recall indices along with overall accuracies in Fort Worth City. The proposed fusion method was more effective than GSV in improving several accuracy indices. Specifically, the proposed method achieved precision and recall rates of 9% for the *industrial* building class, which GSV could not detect. Furthermore, the precision and recall rates for *office building* improved by 2% and 8%, respectively, with values of 43% and 38% being achieved. Additionally, the *background* recall rate improved by 2% after using the proposed method, resulting in a value of 89%. Finally, the pixel-based overall accuracy increased by 3% after the proposed method was used, resulting in a value of 75%.

In terms of comparing the proposed fusion method with fuzzy fusion, the fuzzy fusion technique was unable to detect any pixel in the *industrial* class. However, the proposed fusion method achieved precision and recall rates of 9% and 10%, respectively. Moreover, the recall rates for *office building* and *background* were higher by 35% and 2% compared

to the fuzzy method. Overall, the proposed method improved the pixel-based accuracy by 7% with a value of 75% achieved, whereas the overall accuracy index of the fuzzy fusion was even lower than GSV classification. Although the proposed method generally showed better performance than the fuzzy fusion, there were instances where the fuzzy fusion demonstrated superior performance. For instance, the precision of the *office building* was 9% higher with a value of 52% achieved compared to the proposed method. Furthermore, while GSV and the proposed method could not detect any pixels in *other* and *residential* classes, fuzzy fusion achieved 80% precision and 14% recall rates for the *other* class and 7% precision and 31% recall rates for the *residential* class. In the *retail* class, the fuzzy fusion achieved higher precision and recall rates by 2% and 44%, respectively. Finally, the fuzzy fusion method achieved a higher precision of 89% for the *background* class, which was 3% higher than the proposed method.

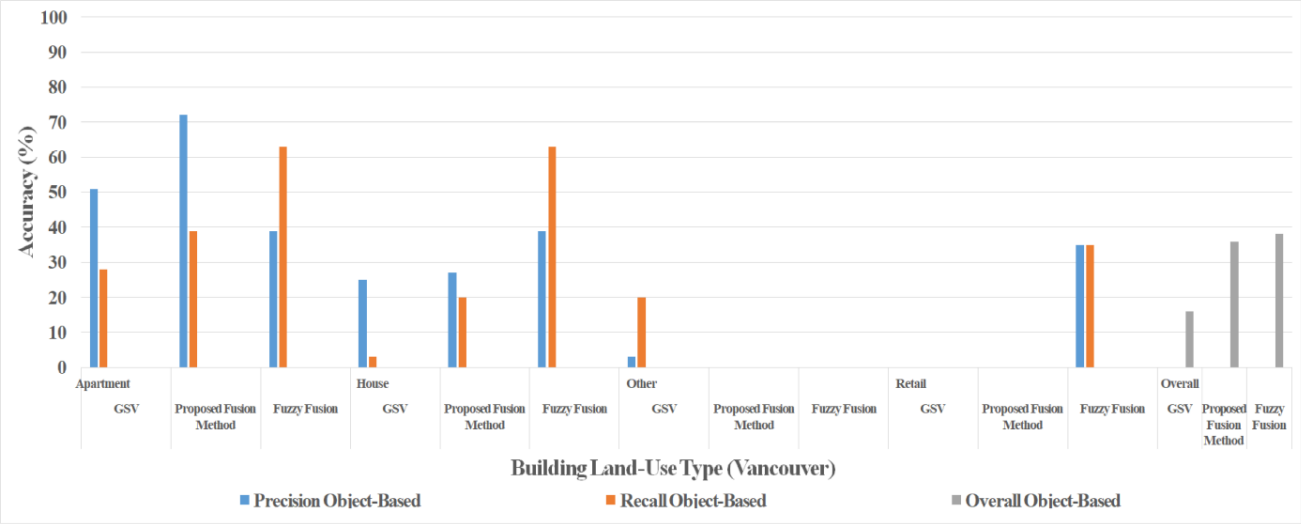


Figure 5-20: Per-class pixel-based precision (blue bars) and recall (orange bars) for GSV, proposed method, and fuzzy fusion classifications in Fort Worth City. The gray bars show overall accuracy indices for the classifications.

The object-based precision, recall rates, and overall accuracy for Fort Worth are depicted in Figure 5-21. After using the proposed method for fusion, there were notable

improvements in the precision and recall indices for *office buildings*, with a 18% and 46% increase resulting in values of 38% and 51% respectively. The class *retail* also experienced a 1% and 9% increase in precision and recall rates after the fusion process. Finally, the overall accuracy had a 10% improvement, resulting in a value of 25%. When compared with fuzzy fusion, the proposed method yielded even better results. For instance, *office building* precision and recall rates both experienced a 5% and 48% increase respectively. Additionally, the proposed method achieved a higher precision rate for the *retail* class by 4%. On the other hand, the overall accuracy for fuzzy fusion was lower, with a value of 7%, when compared to both GSV and the proposed method.

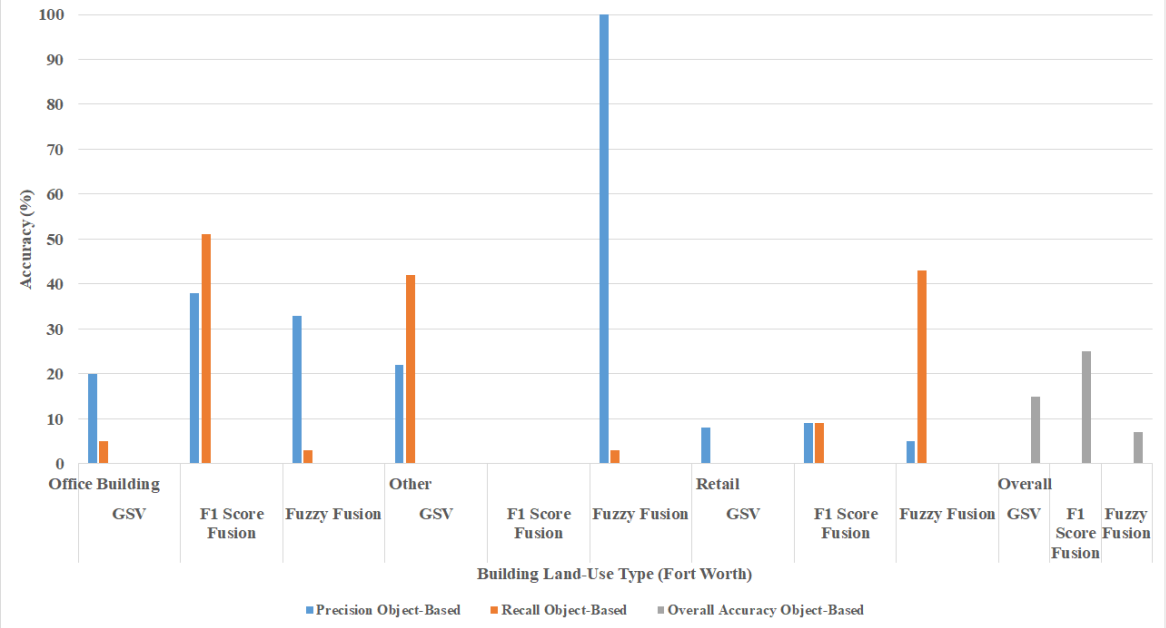


Figure 5-21: Per-class object-based precision (blue bars) and recall (orange bars) for GSV, proposed method, and fuzzy fusion classifications in Fort Worth City. Gray bars depict the overall accuracies for classifications.

The Fuzzy Fusion and F1 Score fusion achieved overall accuracies about 1% and 8% higher than the previous study using GSV and VGG16 model for the Fort Worth test region (Kang et al., 2018). The method applied in our work differs from the previous study in

terms of using an information fusion approach for building land-use type classification. While the previous work focused on GSV data, we combined three DL-based classifiers trained on GSV, LiDAR, and Orthophoto for classification.

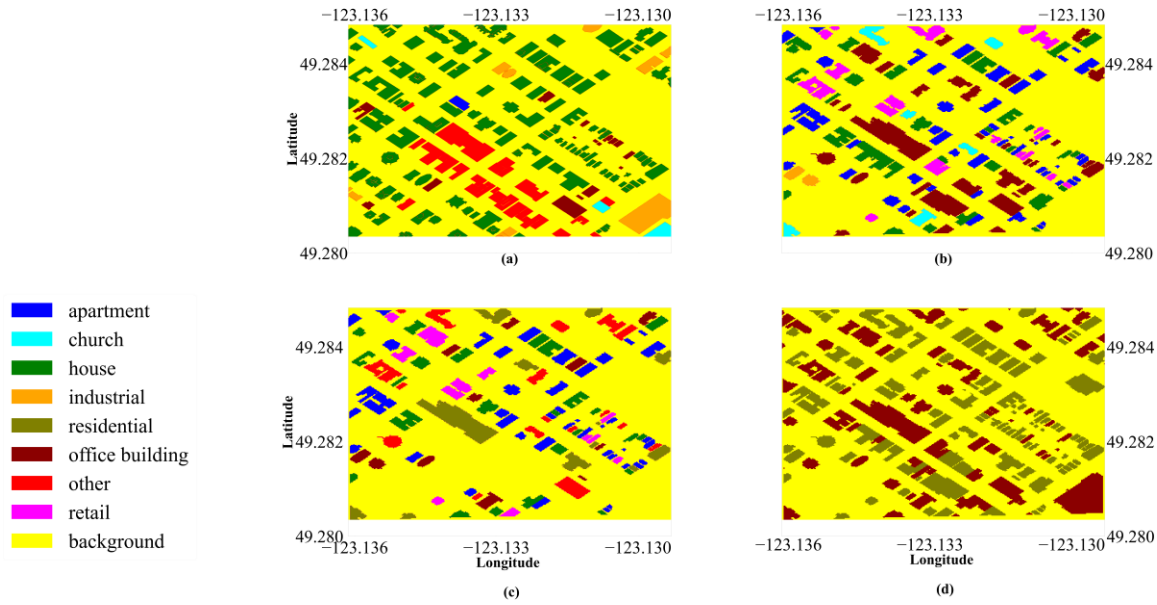


Figure 5-22: Building land-use type classification maps for Vancouver case study; (a): Ground truth; (b): GSV classification map; (c): Proposed Fusion Method; (d): Fuzzy Fusion.

Figure 5-22 shows the ground truth, GSV, and Fusion classification maps for Vancouver. Based on the F1 Score Fusion classification map (part (c)), many building footprints were merged into the *background* because this class was given the highest rank over other classes. While some buildings in Fusion maps were not in the ground truth map because of inconsistency between the 2015 Vancouver building footprint data and ground truth, most buildings in the Fuzzy Fusion map were labeled. The fuzzy fusion method faced difficulty in distinguishing between *houses* and *apartments* within residential buildings due to inconsistencies between the ground truth labels for Orthophoto-LiDAR and GSV images. The web scraping technique used for ground truth creation in GSV image classification could not be applied for LiDAR and Orthophoto classifications because it was not possible to find LiDAR and Orthophoto data for the same address as the addresses retrieved from web scraping. Although the proposed method did not show significant improvement

compared to GSV, it outperformed the fuzzy fusion method by accurately classifying buildings into houses and apartments. The classification maps in Figure 5-23 show building land-use type classification results in Fort Worth City. After comparing the building land-use type classification maps generated using GSV images and the proposed algorithm, some improvements can be observed after the fusion. For instance, in the GSV map, there was one building footprint on the top and three small building footprints on the right (rectangles) that were incorrectly detected as *other* or *industrial*, but the proposed method correctly classified them as *office buildings*. Additionally, while the fuzzy fusion mislabeled many *office buildings* as *retail*, the proposed method was able to correctly classify these buildings.

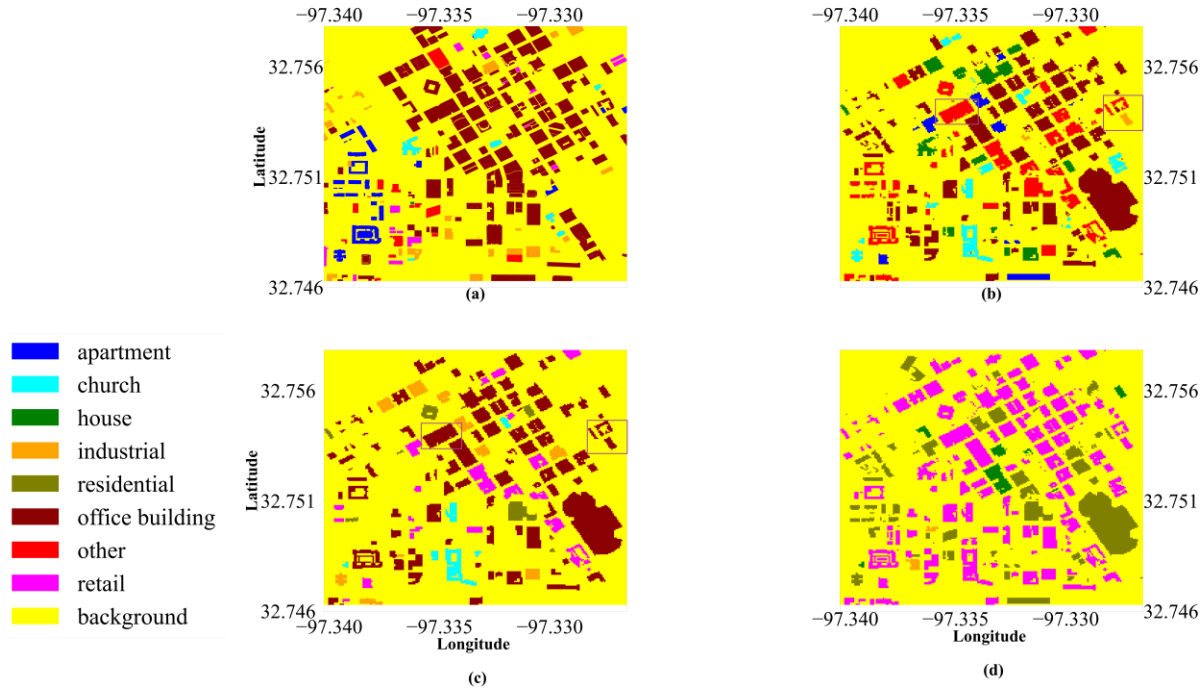


Figure 5-23: Building land-use type classification maps for Fort Worth case study; (a): Ground truth; (b): GSV classification map; (c): Proposed Fusion Method; (d): Fuzzy Fusion.

5.5 Conclusion

This study explored a detailed building land-use type classification. The building footprints were classified as *apartments*, *houses*, *churches*, *industrial*, *office buildings*, *other*, and *retail*. Also, *mixed r/c* buildings were detected using the GSV dataset. When no detailed ground truth information existed, apartments and houses were aggregated into residential. While reducing the number of classes might increase the overall accuracies, this study aimed to train models to classify buildings into more detailed land-use classes than previous studies. Class *other* included different land use/land cover types, including construction, brownfield, grass, recreation ground, and fairground, and had a significant intra-class class variability. While the intra-class variability would cause lower classification performance, the accuracies did not change significantly after excluding class *other*. The building land-use type classification was accomplished using the fusion of three DL-based classifiers trained on GSV, LiDAR, and Orthophoto data using the proposed F1 Score fusion and Fuzzy Fusion methods to improve the GSV classification result. A Fuzzy Fusion method based on the Gompertz function was tested for combining classifiers at the decision level. The results showed that although the fuzzy method improved recall rates and overall accuracies for Vancouver, almost all accuracy indices dropped for Fort Worth City. The proposed F1 Score method improved the Pixel-Based and the Object-Based Precision and Overall Accuracies for both independent test data. In comparison among GSV, LiDAR, and Orthophoto, the best test accuracy on non-independent test data was for the GSV data, but the accuracy degraded significantly after testing the DL models on the independent test data. The transfer learning method was efficient on GSV and Orthophoto datasets but was not successful on LiDAR data because of inconsistency between the *ImageNet* data and LiDAR-derived features. MobileNetV2 and InceptionV3 models achieved the highest test accuracies for GSV, LiDAR, and Orthophoto data, respectively. While this work introduced a data fusion method at the decision-level, using other types of fusion, like feature-level fusion, are worth exploring. Furthermore, preparing building-land use type data or developing the existing datasets for training DL models are crucial research directions to pursue.

References

Abdollahi, A., Pradhan, B., Gite, S., & Alamri, A. (2020). Building footprint extraction from high resolution aerial images using generative adversarial network (GAN) architecture. *IEEE Access*, *8*, 209517-209527.

Al-Habashna, A. (2022). Building Type Classification from Street-view Imagery using Convolutional Neural Networks. In *Statistics Canada*.

Bakhtiari S, Najafi MR, Goda K, Peerhossaini H. Integrated Bayesian Network and Strongest Path Method (BN-SPM) for effective multi-hazard risk assessment of interconnected infrastructure systems. Sustainable Cities and Society. 2024 May 1;104:105294.

Belgiu, M., Tomljenovic, I., Lampoltshammer, T.J., Blaschke, T. and Höfle, B., 2014. Ontology-based classification of building types detected from airborne laser scanning data. *Remote Sensing*, *6*(2), pp.1347-1366.

Camps-Valls, G., & Ieee. (2009, Sep 01-04). Machine Learning in Remote Sensing Data Processing. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP 2009), Grenoble, FRANCE.

Cannon AJ, Alford H, Shrestha RR, Kirchmeier-Young MC, Najafi MR. Canadian Large Ensembles Adjusted Dataset version 1 (CanLEADv1): Multivariate bias-corrected climate model outputs for terrestrial modelling and attribution studies in North America.

Cao, R., Zhu, J., Tu, W., Li, Q., Cao, J., Liu, B., ... & Qiu, G. (2018). Integrating aerial and street view images for urban land use classification. *Remote Sensing*, *10*(10), 1553.

Chen, C., & Fan, L. (2021, August). Scene segmentation of remotely sensed images with data augmentation using U-net++. In *2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)* (pp. 201-205). IEEE.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

Government of Canada, Statistics Canada. (2017, February 8). Focus on Geography Series, 2016 Census - Census division of Peel, RM (Ontario). https://www12.statcan.gc.ca/census-recensement/2016/as-sa/fogs-spg/Facts-cd-eng.cfm?LANG=Eng&GK=CD&GC=3521&TOPIC=1

Government of Canada, Statistics Canada. (2023, November 15). Profile table, Census Profile, 2021 Census of Population - Richmond Hill, Town (T) [Census subdivision], Ontario. https://www12.statcan.gc.ca/census-recensement/2021/dp-pd/prof/details/page.cfm?Lang=E&DGUIDlist=2021A00053519038&GENDERlist=1&STATISTIClist=1&HEADERlist=0

Government of Canada, Statistics Canada. (2023a, October 4). Focus on Geography Series, 2021 Census - Vancouver (Census metropolitan area). https://www12.statcan.gc.ca/census-recensement/2021/as-sa/fogs-spg/page.cfm?lang=E&topic=3&dguid=2021S0503933

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hoffmann, E. J., Wang, Y., Werner, M., Kang, J., & Zhu, X. X. (2019). Model fusion for building type classification from aerial and street view images. *Remote Sensing*, *11*(11), 1259.

Huang, X., Ren, L., Liu, C., Wang, Y., Yu, H., Schmitt, M., ... & Mayer, H. (2022). Urban Building Classification (UBC)-A Dataset for Individual Building Detection and Classification From Satellite Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1413-1421).

Islam MR, Fereshtehpour M, Najafi MR, Khaliq MN, Khan AA, Sushama L, Nguyen VT, Elshorbagy A, Roy R, Wilson A, Perdikaris J. Climate-resilience of dams and levees in Canada: a review. Discover Applied Sciences. 2024 Mar 28;6(4):174.

Jalili Pirani F, Najafi MR. Characterizing compound flooding potential and the corresponding driving mechanisms across coastal environments. Stochastic Environmental Research and Risk Assessment. 2023 May;37(5):1943-61.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., & Zhu, X. X. (2018). Building instance classification using street view images. *ISPRS journal of photogrammetry and remote sensing*, *145*, 44-59.

Karadal, C. H., Kaya, M. C., Tuncer, T., Dogan, S., & Acharya, U. R. (2021). Automated classification of remote sensing images using multileveled MobileNetV2 and DWT techniques. *Expert Systems with Applications*, *185*, 115659.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kumar, A., Abhishek, K., Kumar Singh, A., Nerurkar, P., Chandane, M., Bhirud, S., ... & Busnel, Y. (2021). Multilabel classification of remote sensed satellite imagery. *Transactions on emerging telecommunications technologies*, *32*(7), e3988.

Kundu, R., Basak, H., Singh, P. K., Ahmadian, A., Ferrara, M., & Sarkar, R. (2021). Fuzzy rank-based fusion of CNN models using Gompertz function for screening COVID-19 CT-scans. *Scientific reports*, *11*(1), 14133.

Lary, D. J., Zewdie, G. K., Liu, X., Wu, D., Levetin, E., Allee, R. J., Malakar, N. K., Walker, A. L., Mussa, H. Y., Mannino, A., & Aurin, D. (2018). Machine learning applications for earth observation. In Springer eBooks (pp. 165–218).

Laupheimer, D., Tutzauer, P., Haala, N., & Spicker, M. (2018). Neural networks for the classification of building use from street-view imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*, 177-184.

Liu, K., Yu, S., & Liu, S. (2020). An improved InceptionV3 network for obscured ship classification in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 4738-4747.

Liu, T., Yao, L., Qin, J., Lu, N., Jiang, H., Zhang, F., & Zhou, C. (2022). Multi-scale attention integrated hierarchical networks for high-resolution building footprint extraction. *International Journal of Applied Earth Observation and Geoinformation*, *109*, 102768.

Lu, Z., Im, J., Rhee, J. and Hodgson, M., 2014. Building type classification using spatial and landscape attributes derived from LiDAR remote sensing data. *Landscape and Urban Planning*, *130*, pp.134-148.

Mahmoudi MH, Najafi MR, Singh H, Schnorbus M. Spatial and temporal changes in climate extremes over northwestern North America: The influence of internal climate variability and external forcing. Climatic Change. 2021 Mar;165:1-9.

Meng, X., Currit, N., Wang, L. and Yang, X., 2012. Detect residential buildings from lidar and aerial photographs through object-oriented land-use classification. *Photogrammetric Engineering & Remote Sensing*, *78*(1), pp.35-44.

Moazami S, Na W, Najafi MR, de Souza C. Spatiotemporal bias adjustment of IMERG satellite precipitation data across Canada. Advances in Water Resources. 2022 Oct 1;168:104300.

Na W, Najafi MR. Rising risks of hydroclimatic swings: A large ensemble study of dry and wet spell transitions in North America. Global and Planetary Change. 2024 Jul 1;238:104476.

Ngui YD, Najafi MR, de Souza C, Sills DM. Probabilistic assessment of concurrent tornado and storm-related flash flood events. International Journal of Climatology. 2023 Jul;43(9):4231-47.

Pirani FJ, Najafi MR. Nonstationary frequency analysis of compound flooding in Canada's coastal zones. Coastal Engineering. 2023 Jun 1;182:104292.

RahimiMovaghar M, Fereshtehpour M, Najafi MR. Spatiotemporal pattern of successive hydro-hazards and the influence of low-frequency variability modes over Canada. Journal of Hydrology. 2024 May 1;634:131057.

Rastogi, K., Bodani, P., & Sharma, S. A. (2022). Automatic building footprint extraction from very high-resolution imagery using deep learning techniques. *Geocarto International*, *37*(5), 1501-1513.
Rezvani R, Na W, Najafi MR. Lagged compound dry and wet spells in Northwest North America under 1.5° C–4° C global warming levels. Atmospheric Research. 2023 Jul 15;290:106799.

Rezvani R, RahimiMovaghar M, Na W, Najafi MR. Accelerated lagged compound floods and droughts in northwest North America under 1.5° C− 4° C global warming levels. Journal of Hydrology. 2023 Sep 1;624:129906.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh H, Najafi MR, Cannon A. Evaluation and joint projection of temperature and precipitation extremes across Canada based on hierarchical Bayesian modelling and large ensembles of regional climate simulations. Weather and Climate Extremes. 2022 Jun 1;36:100443.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

United States Census Bureau. (2022). QuickFacts Fort Worth city, Texas; Texas. https://www.census.gov/quickfacts/fact/table/fortworthcitytexas,TX

Vaughan Economic Development. (2021). 2021 Census Insights and Findings – Population and Dwellings | Vaughan Economic Development. https://vaughanbusiness.ca/insights/2021-census-insights-and-findings-population-and-dwellings/

Wang, Q., Zhou, C., & Xu, N. (2017, March). Street view image classification based on convolutional neural network. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 1439-1443). IEEE.

Wu, M., Huang, Q., Gao, S., & Zhang, Z. (2023). Mixed land use measurement and mapping with street view images and spatial context-aware prompts via zero-shot multimodal learning. *International Journal of Applied Earth Observation and Geoinformation*, *125*, 103591.

Wurm, M., Schmitt, A., & Taubenböck, H. (2015). Building types' classification using shape-based features and linear discriminant functions. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(5), 1901-1912.

Xie, J., & Zhou, J. (2017). Classification of urban building type from high spatial resolution remote sensing imagery using extended MRS and soft BP network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(8), 3515-3528.

Yan, L., Zhang, J., Huang, G., & Zhao, Z. (2011, August). Building Footprints Extraction from PolSAR Image Using Multi-Features and Edge Information. In *2011 International Symposium on Image and Data Fusion* (pp. 1-5). IEEE.

Yoo, S., Lee, J., Farkoushi, M. G., Lee, E., & Sohn, H. G. (2022). Automatic generation of land use maps using aerial orthoimages and building floor data with a Conv-Depth Block (CDB) ResU-Net architecture. *International Journal of Applied Earth Observation and Geoinformation*, *107*, 102678.

Yu, H., Hu, H., Xu, B., Shang, Q., Wang, Z., & Zhu, Q. (2023). SuperpixelGraph: Semi-automatic generation of building footprint through semantic-sensitive superpixel and neural graph networks. *International Journal of Applied Earth Observation and Geoinformation*, *125*, 103556.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13* (pp. 818-833). Springer International Publishing.

Zhang, W., Li, W., Zhang, C., Hanink, D.M., Li, X. and Wang, W., 2017. Parcel-based urban land use classification in megacity using airborne LiDAR, high resolution orthoimagery, and Google Street View. *Computers, Environment and Urban Systems*, *64*, pp.215-228.

Chapter 6

6 Conclusions

6.1 Summary

This dissertation explored the development of DL-based algorithms for urban flood risk mapping. Two risk components, hazard and vulnerability, were estimated using DL-based algorithms. Four studies were conducted to achieve the goals stated in Chapter 1. Firstly, using the proposed CSN and SAR satellite images, flood extent maps were created. Secondly, a method was proposed that relied on the YOLOv5s object detection algorithm and vertical measurements on GSV images to estimate the FFH of buildings. Based on the FFH and water depth values, the buildings were then classified as highly vulnerable, moderately vulnerable, or low vulnerable. Thirdly, a Dense Attention Network was developed to detect building footprints using LiDAR and MS data. Finally, a fusion method called Ranking Classes Based on F1 Score was proposed to combine building land-use type classifications derived from LiDAR, Orthophoto, and GSV.

In Chapter 2, a DL-based change detection framework (CSN) was developed to accomplish flood extent mapping in urban areas using SAR satellite images. The SAR images applied were Sentinel-1 and dual-polarized RCM data. The applied data were captured in C-band, and their resolutions were 10m and 5m for Sentinel-1 and RCM, respectively. The intensity and coherence features were extracted from the pre-event and co-event SAR images and imported to the corresponding CNN in CSN. Three different loss functions were tested, including Contrastive Loss, WDMCL, and Triplet Loss functions. It was observed that after applying the WDMCL the flood precision and recall rates were improved. After adding DEM the flood F1 Score accuracy improved slightly, but the background F1 Score improvement was more significant. Comparison with other DL-based segmentation algorithms, including Unet, Unet++, DeepLabV3+, and Siamese-Unet, confirmed the reliability of the proposed CSN. Although a promising flood recall rate of about 0.7 was

achieved, it was inferred from the flood precision and F1 Score that medium resolution Sentinel-1 data might hinder its application for urban flood mapping. Further, RCM data were also tested in both urban and non-urban areas, and a precision of 0.79 was achieved for the non-urban case. Experiments on two existing datasets, SEN12-FLOOD and Sen1Floods11, showed that the proposed CSN achieved a higher precision index of 0.75 on SEN12-FLOOD compared to the Sen1Floods11 dataset, with a precision of 0.2, because Sen1Floods11 ground truth labels were per-pixel rather than per scene in SEN12-FLOOD dataset. Per-pixel classification is more complex than per-scene and requires satellite images with low radiometric distortions.

Chapter 3 presented a method to estimate the First Floor Height (FFH) using the YOLOv5s object detector and vertical measurements on the Google Street View (GSV) image. The standard size of the Front Door (FD) in American-style houses was used to convert the image measurements to real-world scale. The FFH was then converted to the First Floor Elevation (FFE) using the LiDAR-derived DEM. Four scenarios were tested to extract the Lowest Adjacent Grade (LAG) height from the DEM, namely *Point, Mean, Minimum,* and *Maximum*. The *Point* and *Mean* methods showed the best consistency in terms of the Interquartile (IQR) distance. Buildings were classified into Very High to Very Low vulnerability classes using the *Natural Break* classification method based on the difference between water depth and FFH. Finally, a vulnerability map was produced for selected buildings across the Lower Don region. The FFH variance equation was derived based on the error propagation rule, which showed the relationship between the FFH uncertainty and Lower Left (LL), Upper Left (UL), Lower Right (LR) bounding box coordinates for FD and LR bounding box coordinates for stairs/building extent, and GSV image pixel size. Additionally, the YOLOv5s algorithm was utilized to identify basement windows and assess basement existence. Experiments were conducted in both the GTA and the state of Virginia in the United States to validate the methodology. The results demonstrated an achievement of FFE RMSE and Bias values of 81 cm and -50 cm for GTA, and 95 cm and -20 cm for the Virginia region, respectively.

Chapter 4 presented a CNN for building footprint extraction from LiDAR and MS data. Two Dense Attention Blocks were embedded into the CNN architecture. Each Dense

Attention Block consisted of a cascade of BN, Conv2D, Dropout, and Average Pooling layers. Six concatenation layers were embedded into the CNN layers to prevent the vanishing gradient problem. The proposed method was tested on two case studies, including Toronto and Massachusetts Building Dataset. Compared with two widely used DL techniques, VGG16 and Resnet50, the proposed method had a simpler architecture and converged faster with higher accuracy. Also, a comparison with the two other state-of-the-art DL algorithms, including Unet and ResUnet, showed that the proposed technique could achieve a higher F1 Score (0.71), compared with those for Unet (0.42) and ResUnet (0.49).

In Chapter 5, buildings were classified into *apartment*, *house*, *industrial*, *mixed r/c*, *office building*, *retail* and *others* using GSV, LiDAR-derived features, and Orthophoto images. Features extracted from each dataset were imported to the corresponding DL network. The DL models were trained on building land-use type data for the GTA. The data was created using building land-use type labels from OSM and web scraping. Three DL-derived classification maps from GSV, LiDAR, and Orthophoto images were combined at the decision level using the proposed Ranking Classes Based on the F1 Score method. The method included two steps. In the first step, the classes in the LiDAR and Orthophoto classifications were ranked from the lowest to the highest F1 Score, and class labels were imported from the lowest to the highest ranks. Then, the F1 Score metric was calculated for the classification from the first step, and the classes for the combined and GSV classifications were ranked from the lowest to the highest F1 Score. Finally, the labels were imported the same as the previous step, and the final building land-use type classification map was produced. The proposed method was compared to the Fuzzy Fusion method based on the Gompertz Function, and improvements in terms of overall accuracy and precision were observed for residential and office building classes. The results of two independent case studies, Vancouver and Fort Worth, showed that the proposed fusion method could achieve an overall accuracy of 75%, up to 8% higher than Kang et al. 2018 using CNNs and the same ground truth data. Also, the results showed that while *mixed r/c* buildings were correctly detected using GSV images, the DL models confused many houses in the GTA with churches and *mixed r/c* because of their similar appearance in GSV images.

6.2 Conclusions and Contributions

This dissertation focused on DL algorithms to analyze urban flood risk mapping using EO data. The study estimated two risk components, hazard and vulnerability, and successfully achieved the main objective and three sub-objectives outlined in Chapter 1. The findings suggested that applying DL algorithms could help reduce the fieldwork required to collect urban flood risk-related parameters, such as FFH and building land-use type and also improve the flood extent mapping accuracy.

Based on the research questions and objectives outlined in Chapter 1, the following conclusions were obtained from the four studies:

1. Flood detection accuracy using CSN and SAR satellite images was lower in urban areas than in non-urban areas. Floods were overestimated due to the SAR shadowing effect, which was further exacerbated when using Sentinel-1 images. The selection of loss function had a significant impact on the CSN flood mapping accuracy, and using WDMCL improved flood precision and recall indices. The precision index was more affected by the input data type and normalization method than the loss function. The addition of DEM improved the SAR flood mapping F1 Score by 5%. Comparison with Unet, Unet++, DeepLabV3+, and Siamese-Unet showed comparable performance in terms of flood detection accuracy indices. Experiments on two publicly available datasets, Sen1Floods11 and SEN12-FLOOD, resulted in F1 Score values of 0.63 and 0.67, respectively.

2. While basement and FFH information were useful for flood vulnerability analysis at the building scale, vulnerability estimation solely based on image data resulted in flood vulnerability underestimation because some basements were not visible in GSV images. The FFH uncertainty analysis showed a strong relationship between the FD and stairs/building extent bounding boxes accuracies and FFH accuracies. FFE estimation using DL-based object detectors and vertical measurements on GSV images resulted in lower RMSE and Bias, and higher R^2 than the previous method using Tacheometric Surveying principles. Among four different scenarios examined for estimating FFE, and six statistical indices, including Min, Max, Median, IQR, First Quartile, and Third Quartile, compared with their ground truth counterparts, the *Mean* method resulted in the least

difference between FFE and ground truth distributions in terms of the Min index. The *Maximum* and *Minimum* methods showed the least discrepancies according to the Max and Median indices, respectively. The *Point* method achieved the best consistency based on IQR distance, and finally, the *Minimum* method achieved the least discrepancy in terms of the First and Third Quartile indices.

3. The use of Dense Attention Blocks in CNN architecture improved the building footprint detection accuracy using LiDAR and MS data. Additionally, the use of concatenation layers reduced the vanishing gradient problem and improved building footprint detection precision, recall, F1 Score, and IOU by 1%. Despite having a simpler architecture and fewer number of parameters, the proposed DAN achieved higher accuracies than the state-of-the-art DL models, including VGG16, Resnet50, Unet, and ResUnet.

4. The fusion of CNN-based building land-use type classifications extracted from GSV, LiDAR, and Orthophoto data achieved higher precision, recall, and overall accuracy than GSV alone after using Ranking Classes Based on the F1 Score method. The proposed method achieved higher precision and overall accuracy for Vancouver and Fort Worth case studies than the Fuzzy Fusion method based on the Gompertz Function. The *mixed r/c* class was successfully detected using GSV images. The building land-use type detection using GSV and Orthophoto images was improved by fine-tuning pre-trained DL models trained on *ImageNet*. However, the fine-tuned models using LiDAR data did not yield high building land-use type detection accuracies due to inconsistencies between *ImageNet* and LiDAR data. As a result, these models were discarded from the fusion step. The proposed fusion method, Ranking Classes Based on the F1 Score, was conducted in two steps to make it more meaningful to the end user. However, it is possible to achieve this method in one step by ranking the classes from the three classifications. In other words, the highest priority should be given to the class from the classification with the lowest F1 Score, the second priority to the class with the medium F1 Score, and the last priority should be given to the class with the highest F1 Score.

This dissertation has made multiple contributions.

1. This dissertation aimed to improve urban flood mapping using SAR satellite images through a change detection approach with the CSN DL algorithm. SAR data imposes challenges due to geometric distortions such as layover, shadowing effects, and speckle noise, compounded by the computational intensity of DL algorithms. The study addressed these complexities, making the first attempt to use SAR data and CSN for urban flood mapping.

2. A method based on the YOLOv5s object detection algorithm and GSV image measurement was proposed to estimate FFH and FFE in the GTA region. It was the first study estimating both FFH and FFE (using zonal statistics) and the first to use computer vision for basement detection and flood vulnerability prediction. Furthermore, it performed superior to Ning et al. 2021 in terms of RMSE, R^2, and Bias.

3. A new CNN based on multiple dense attention blocks and concatenation layers was proposed. The inspiration for the concatenation layers came from DAN (Yang et al., 2018). The issue with using dropout and pooling layers in the CNN is that these operations can cause the omission of useful features. To retrieve this information, the proposed CNN stacked information from previous layers and added it to the following layers through concatenation layers. These layers helped to improve the feature extraction ability of the simple CNN. While a simple CNN uses a cascade of feedforward convolutional and pooling layers for feature extraction, the proposed CNN was based on multiple dense attention blocks and concatenation layers to retrieve information from the previous layers and import them directly into the following layers. For the first time, DAN was used for building footprint extraction in Toronto. The proposed method could be adapted to both pixel-by-pixel and object-based predictions, and the latter could be achieved by replacing FC layers with 2D convolutional layers.

4. This dissertation introduced a Decision Level Fusion method, Ranking Classes Based on F1 Score, which combined three DL-based classifications trained on

GVS, LiDAR, and Orthophoto. To address the lack of building land-use type train data, the study utilized three techniques: 1- Web scraping to create labels for building types, 2- Testing different heading angles when downloading GSV images to increase sampling frequency, and 3- Applying transfer learning strategy to reduce the DL model dependency on train data. Also, it explored producing a comprehensive building land-use type classification map, and classified buildings into *apartments*, *houses*, *churches*, *industrial*, *office buildings*, *other*, and *retail.* Detection of *mixed r/c* buildings using GSV images was also another contribution of this study.

To enhance flood risk mapping, it is crucial to incorporate advancements in machine learning methodologies that utilize EO data. DL algorithms have been utilized for computer vision tasks since the mid-1960s. This thesis has implemented DL algorithms to assess their ability to map flood risk components in conjunction with EO data. In general, the four studies applied DL algorithms with EO data and improved the effectiveness of DL models for flood risk-related applications and building a flood-resilient community.

The proposed chapters can be integrated into a framework to estimate various flood risk components. For example, a flood extent map can be overlaid onto the building footprint map to count how many residential buildings were affected during the flood event or what percentage of highly vulnerable structures were inundated during the flood event. These initial analyses after the flood event can aid in prompt disaster response and equip policymakers with valuable insights to enhance preparedness for potential future events.

6.3 Limitations and Future Research

This dissertation has made significant progress in flood risk mapping and the estimation of flood risk-related parameters. However, there are still some areas of improvement and future research directions that are worth considering:

- The SAR satellite images used in this dissertation had spatial resolutions of 10 and 5m. To improve urban flood mapping, it is necessary to examine very high-resolution SAR satellite images with spatial resolutions of less than 5m (Popien et al., 2023; Baghermanesh et al., 2022). This study focused on flood extent mapping to estimate the hazard component. Other flood parameters, such as volume and depth, can be retrieved using RS data and are worth considering for future works (Santillan et al., 2016; Gao et al., 2018; Cohen et al., 2019; Popandopulo et al., 2023).
- SAR satellite images have limitations in accurate flood extent mapping due to speckle noise, and geometric distortions such as shadowing and layover effects. Urban flood mapping using SAR satellite images is challenging due to the water dynamic complexities in urban areas. One alternative to overcome these limitations is to fuse SAR and Optical satellite images (Yonghua et al., 2007; Irwin et al., 2017; Quang et al., 2019).
- We used pixel-based accuracy assessment for flood mapping. This method is more challenging than object-based accuracy assessment, especially for SAR flood mapping. Using an object-based approach might result in higher accuracy values.
- The DL parameters recommended in the chapters are case study specific, and practitioners should find the sub-optimal model parameters that work best for their case studies.
- One of the major limitations when using DL algorithms in a supervised manner is the training data. DL models because of having a huge number of layers and weights require a significant number of train data which might not be available. Besides, because of their black-box nature, their architecture cannot be physically interpreted. These limitations require future studies of semi-supervised learning and physically interpretable deep learning models.
- Using computer vision and image processing techniques for urban flood vulnerability mapping can lead to vulnerability underestimation because basements may not be fully visible in GSV images. The basements could be blocked by obstacles in front of the building or might not be visible in the building's front view.

An alternative is to use virtual reality and GSV image combinations for flood vulnerability analysis at the building scale.

- The rotation of GSV images can lead to geometric distortions, resulting in building elements appearing in exaggerated or understated relative sizes. In this study, the rotation of GSV images was not taken into account when estimating FFH, and this issue needs to be examined in future research. Another limitation of the proposed method for FFH estimation was the assumption of a fixed FD size for all buildings. While this may be reasonable for houses, apartments may have a different FD size, and the effect of the fixed FD size on FFH estimation should be investigated in future studies. Furthermore, the proposed method's accuracy was highly dependent on object detection accuracy. As mentioned in Chapter 3, α, β, and σ in Equation (3-6) were functions of the FD and stairs/building extent bounding box coordinates, and a significant FFH uncertainty of 1.37m was reported due to inaccurate object detection results. Besides, while this study focused on FFH uncertainty estimation, FFE also contributes to flood vulnerability analysis and its uncertainty needs to be explored (Bodoque et al., 2016).
- While acceptable object detection results were achieved using YOLOv5, more advanced versions of YOLO object detector families, such as YOLOv8, should be used in future works.
- One limitation of using GSV for FFH estimation is their unavailability for most developing countries, such as the Middle East, parts of Africa and Asia. Besides, they are not as accurate as drone images. Drone images provide a more precise geometry than GSV. Hence, their usage for FFH estimation is worth exploring.
- Creating a detailed building land-use type classification map using CNN requires high-quality and quantity training samples with accurate building land-use type labels. While the OSM provides labels for building land-use types and is accessible via GIS software, the quality of the labels is uncertain. In addition, web scraping using real estate websites cannot be fully substituted with the OSM as quite a few websites provide detailed building land-use type information. The lack of training data is even worse for mixed-building land-use types such as mixed residential/commercial and mixed retail/institutional. The use of semi-supervised

DL algorithms or developing building land-use type databases for supervised learning using GSV image inspection can be applied to resolve this issue (Xie et al., 2022; Bortoloti et al., 2022).

- The GSV, LiDAR, and orthophotos were not co-registered, resulting in spatial discrepancies. In other words, the pixels may not represent the same geographic locations across datasets, leading to errors and inaccuracies in the decision-level fusion process. Consequently, this misalignment can distort feature matching and data interpretation, affecting the reliability and accuracy of the fused output. Another limitation when combining information from different sources is the time discrepancy. While GSV images are usually the most recent images captured from the houses, the exact time of their acquisition is not evident. This ambiguity can be a considerable limitation, adding uncertainty to the fusion process.

- This dissertation aimed to investigate the estimation of hazard and vulnerability using RS data. However, it is important to note that this research did not cover the estimation of flood exposure, which is essential for a comprehensive flood risk analysis. Further research is required to explore this aspect (Ramesh et al., 2023).
- Urban flood mapping and capturing critical flooding time are essential for effective disaster management. The need for accurate and timely data to mitigate the impact of floods on urban areas cannot be overstated. However, the unavailability of high-resolution SAR data due to access restrictions, high cost, limited coverage, and technical complexities often creates a scarcity, hindering the process.
- If policymakers need to choose between ground-based measurements and high-resolution SAR images for disaster response, I recommend taking a hybrid approach. In other words, combining SAR data with ground-based observations to leverage the strengths of both methods is the optimal strategy. If budget constraints necessitate a choice, consider the following:
    - Immediate Need for Real-Time Data: Prioritize ground-based observing systems for their ability to provide real-time, localized data critical for emergency response.

- Need for Broad and Consistent Coverage: Prioritize SAR data if the goal is to monitor large urban areas and integrate data into long-term flood risk management and planning.

References

Baghermanesh, S.S., Jabari, S. and McGrath, H., 2022. Urban Flood Detection Using TerraSAR-X and SAR Simulated Reflectivity Maps. *Remote Sensing*, *14*(23), p.6154.

Bodoque, J.M., Guardiola-Albert, C., Aroca-Jiménez, E., Eguibar, M.Á. and Martínez-Chenoll, M.L., 2016. Flood damage analysis: First floor elevation uncertainty resulting from LiDAR-derived digital surface models. *Remote Sensing*, *8*(7), p.604.

Bortoloti, F.D., Tavares, J., Rauber, T.W., Ciarelli, P.M. and Botelho, R.C.G., 2022. An annotated image database of building facades categorized into land uses for object detection using deep learning: Case study for the city of Vila Velha-ES, Brazil. *Machine Vision and Applications*, *33*(5), p.80.

Cohen, S., Raney, A., Munasinghe, D., Loftis, J.D., Molthan, A., Bell, J., Rogers, L., Galantowicz, J., Brakenridge, G.R., Kettner, A.J. and Huang, Y.F., 2019. The Floodwater Depth Estimation Tool (FwDET v2. 0) for improved remote sensing analysis of coastal flooding. *Natural Hazards and Earth System Sciences*, *19*(9), pp.2053-2065.

Gao, W., Shen, Q., Zhou, Y. and Li, X., 2018. Analysis of flood inundation in ungauged basins based on multi-source remote sensing data. *Environmental monitoring and assessment*, *190*, pp.1-13.

Irwin, K., Beaulne, D., Braun, A. and Fotopoulos, G., 2017. Fusion of SAR, optical imagery and airborne LiDAR for surface water detection. *Remote Sensing*, *9*(9), p.890.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., & Zhu, X. X. (2018). Building instance classification using street view images. *ISPRS journal of photogrammetry and remote sensing*, *145*, 44-59.

Ning, H., Li, Z., Ye, X., Wang, S., Wang, W., & Huang, X. (2021). Exploring the vertical dimension of street view image based on deep learning: a case study on lowest floor elevation estimation. *International Journal of Geographical Information Science*, 1-26.

Popandopulo, G., Illarionova, S., Shadrin, D., Evteeva, K., Sotiriadi, N. and Burnaev, E., 2023. Flood extent and volume estimation using remote sensing data. *Remote Sensing*, *15*(18), p.4463.

Popien, P., D'Hondt, O., Sunkara, V. and Chakrabarti, S., 2023, July. Deep Learning Based Urban Flood Mapping From High Resolution Capella Space Sar Imagery. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1384-1387). IEEE.

Quang, N.H., Tuan, V.A., Hao, N.T.P., Hang, L.T.T., Hung, N.M., Anh, V.L., Phuong, L.T.M. and Carrie, R., 2019. Synthetic aperture radar and optical remote sensing image fusion for flood monitoring in the Vietnam lower Mekong basin: A prototype application for the Vietnam Open Data Cube. *European Journal of Remote Sensing*, *52*(1), pp.599-612.

Ramesh, B., Callender, R., Zaitchik, B.F., Jagger, M., Swarup, S. and Gohlke, J.M., 2023. Adverse Health Outcomes Following Hurricane Harvey: A Comparison of Remotely-Sensed and Self-Reported Flood Exposure Estimates. *GeoHealth*, *7*(4), p.e2022GH000710.

Santillan, J.R., Marqueso, J.T., Makinano-Santillan, M. and Serviano, J.L., 2016. Beyond flood hazard maps: Detailed flood characterization with remote sensing, gis and 2D modelling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, pp.315-323.

Xie, X., Liu, Y., Xu, Y., He, Z., Chen, X., Zheng, X. and Xie, Z., 2022. Building Function Recognition Using the Semi-Supervised Classification. *Applied Sciences*, *12*(19), p.9900.

Yang, H., Wu, P., Yao, X., Wu, Y., Wang, B. and Xu, Y., 2018. Building extraction in very high resolution imagery by dense-attention networks. *Remote Sensing*, *10*(11), p.1768.

Yonghua, S., Xiaojuan, L., Huili, G., Wenji, Z. and Zhaoning, G., 2007, July. A study on optical and SAR data fusion for extracting flooded area. In *2007 IEEE International Geoscience and Remote Sensing Symposium* (pp. 3086-3089). IEEE.

Appendices

Appendix A: Confusion matrices and learning curves for best building land-use type detection DL models trained on GSV

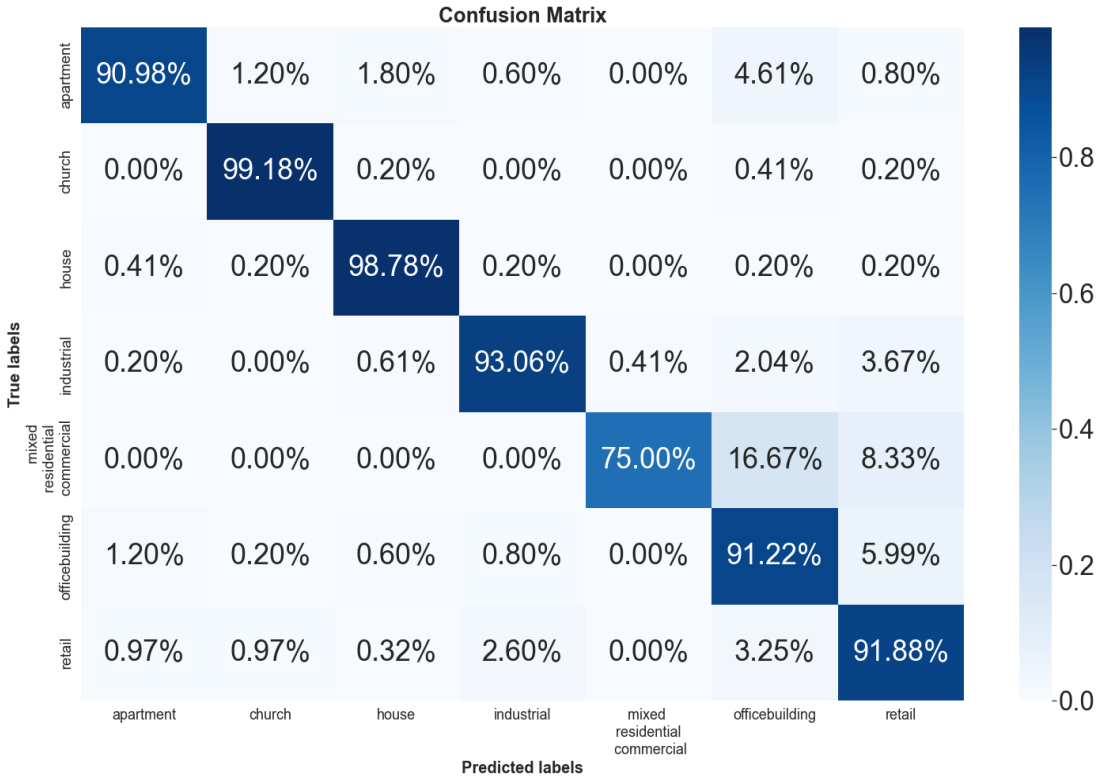


Figure A-1: MobileNetV2 confusion matrix for model with 150 trained layers

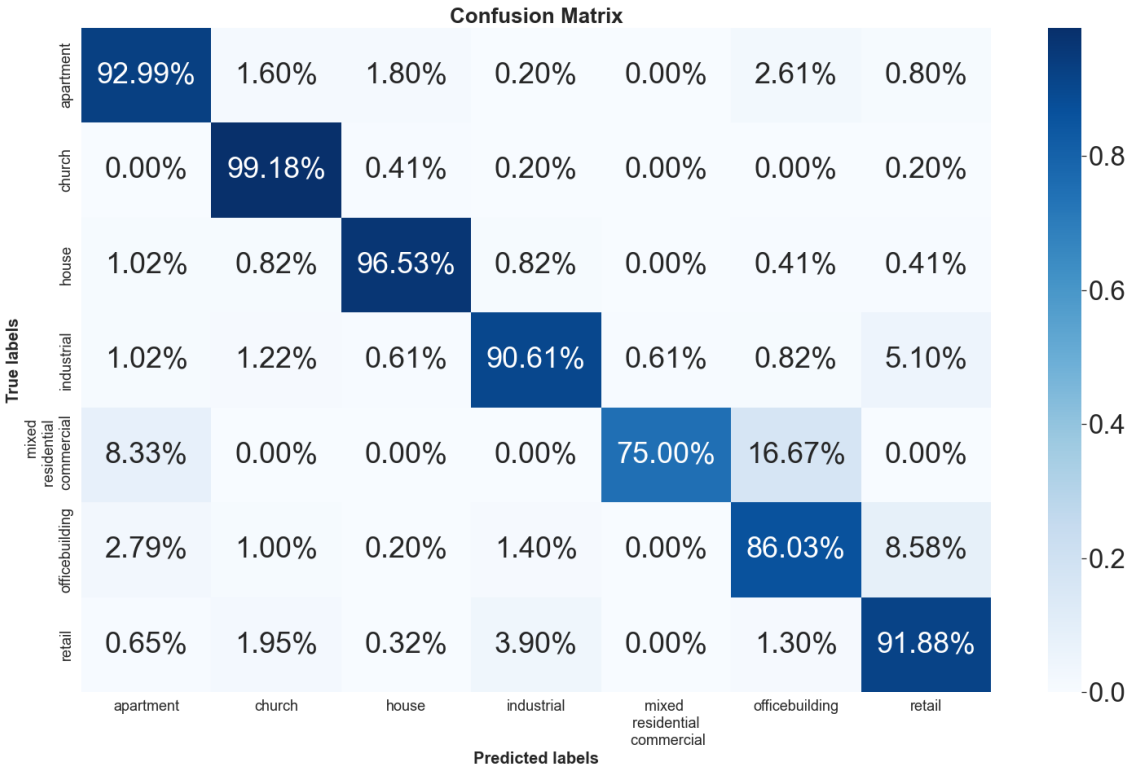


Figure A-2: MobileNetV2 confusion matrix for model with 100 trained layers

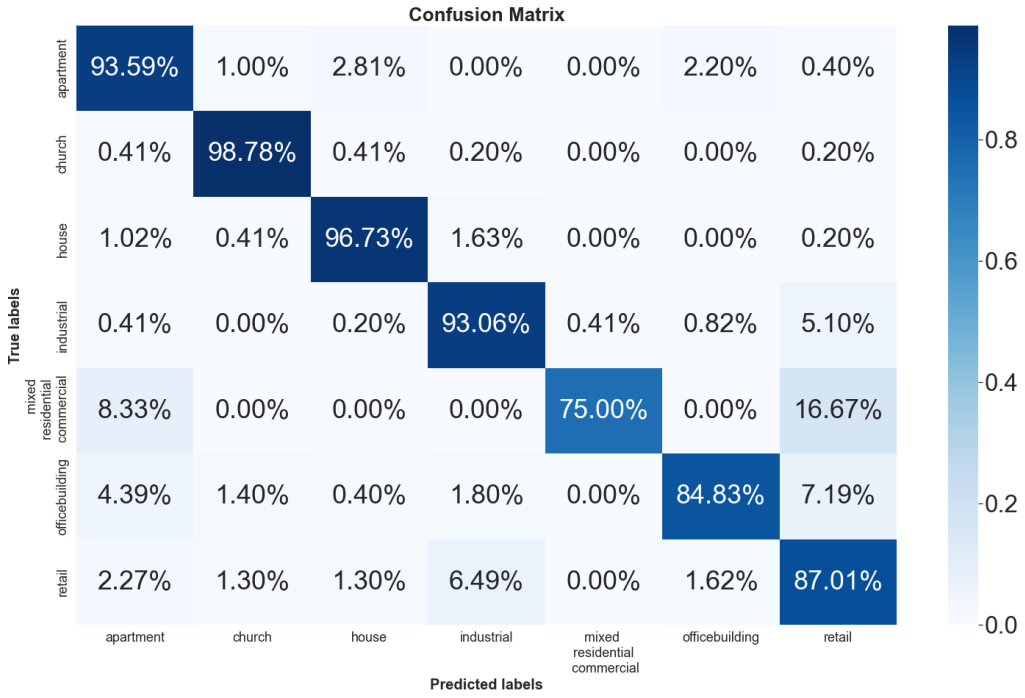


Figure A-3: MobileNetV2 confusion matrix with 50 trained layers

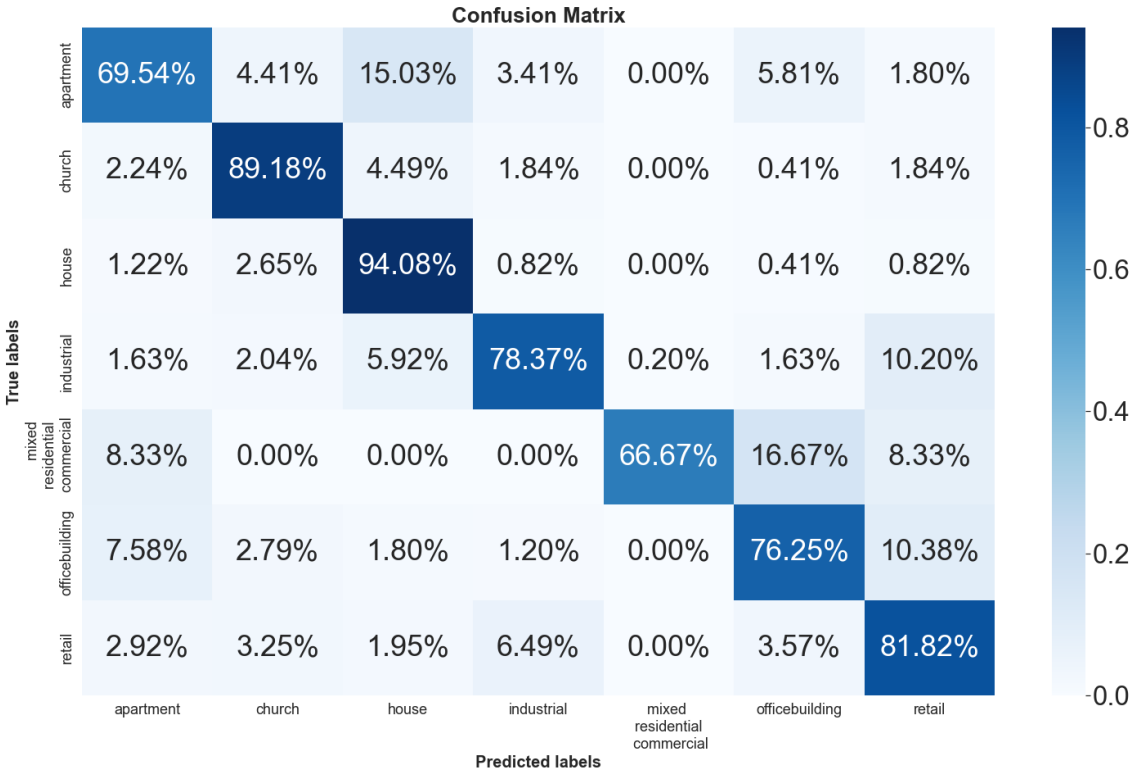


Figure A-4: MobileNetV2 confusion matrix for frozen model (0 trained layers)

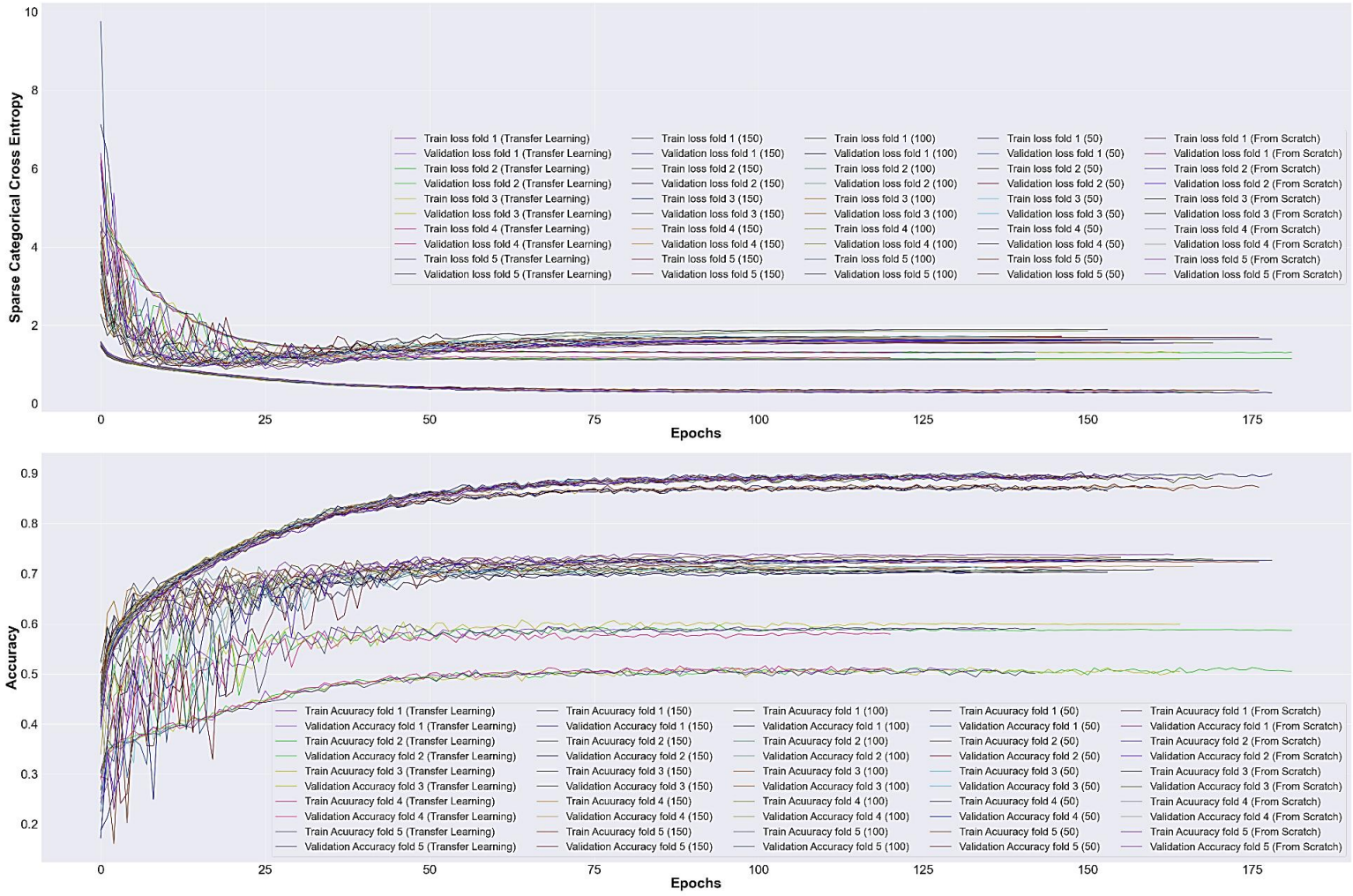


Figure A-5: Learning curves for MobileNetv2 (Codes used to create this figure are available at https://github.com/nafisegh/Building-Land-Use-Type).

Appendix B: Confusion matrices and learning curves for best building land-use type detection DL models trained on LiDAR data

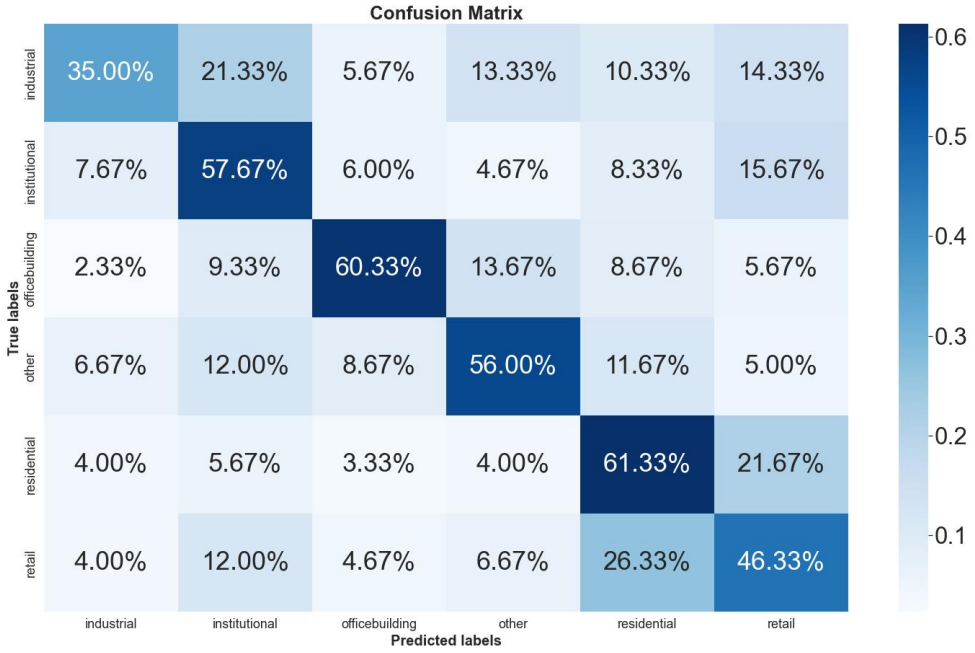


Figure B-1: InceptionV3 confusion matrix when training the model from scratch (LiDAR)

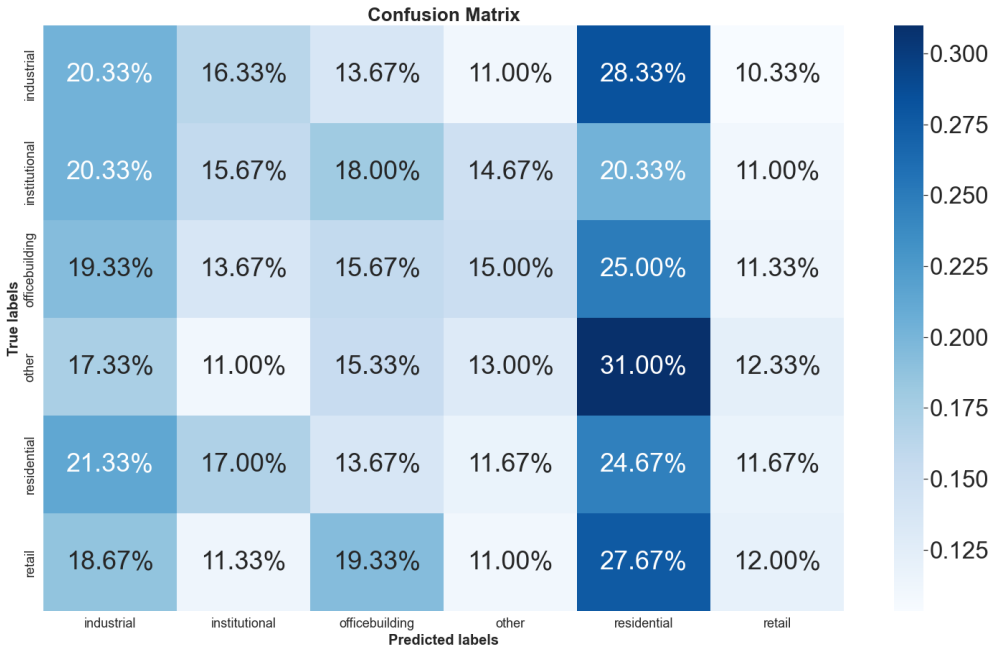


Figure B-2: InceptionV3 confusion matrix with 300 trained layers (LiDAR)

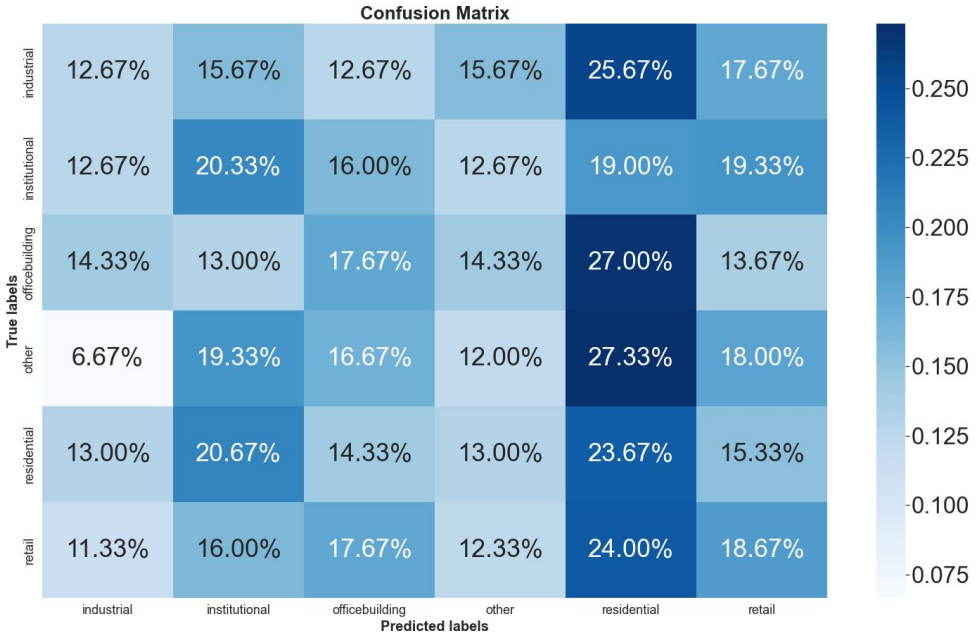


Figure B-3: InceptionV3 confusion matrix with 250 trained layers (LiDAR)

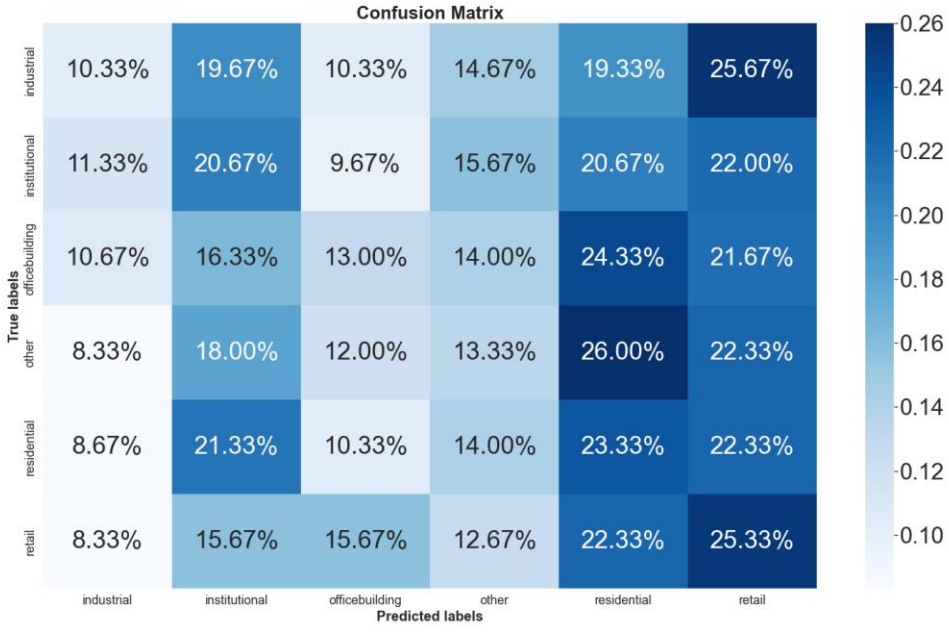


Figure B-4: InceptionV3 confusion matrix with 200 trained layers (LiDAR)

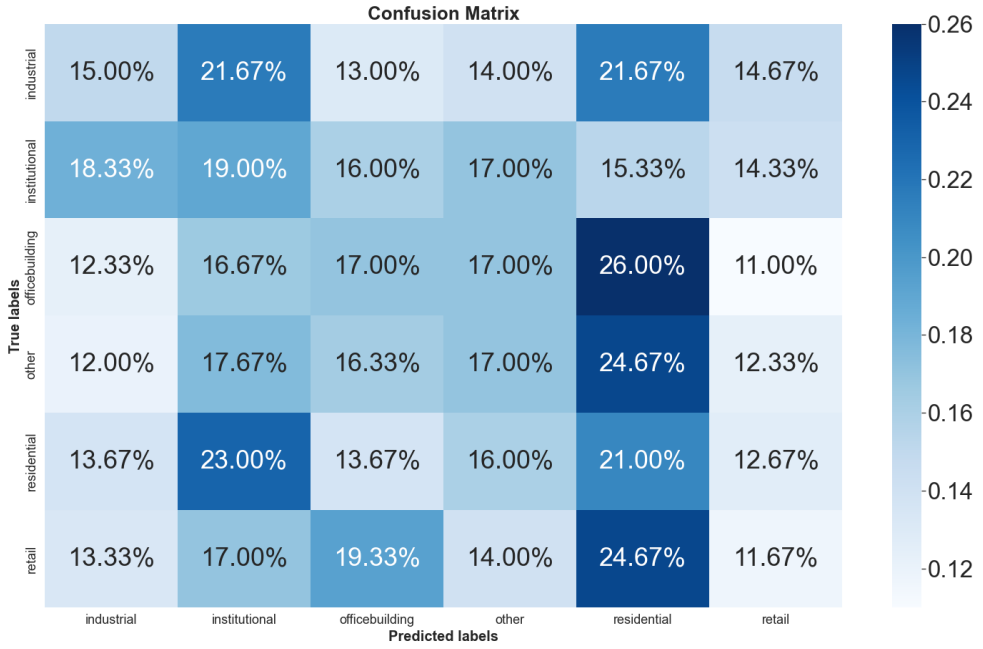


Figure B-5: InceptionV3 confusion matrix with 150 trained layers (LiDAR)

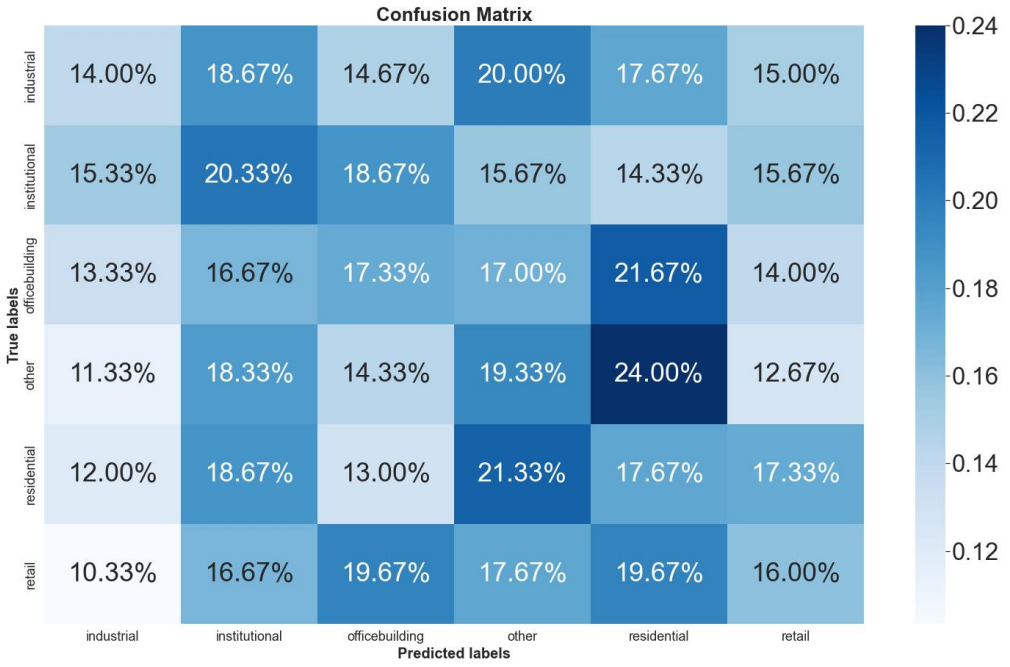


Figure B-6: InceptionV3 confusion matrix with 100 trained layers (LiDAR)

Figure B-7: InceptionV3 confusion matrix with 50 trained layers (LiDAR)

Figure B-8: Learning curves for InceptionV3 with the learning rate 10-3 (LiDAR)

Appendix C: Confusion matrices and learning curves for best building land-use type detection DL models trained on Orthophoto images

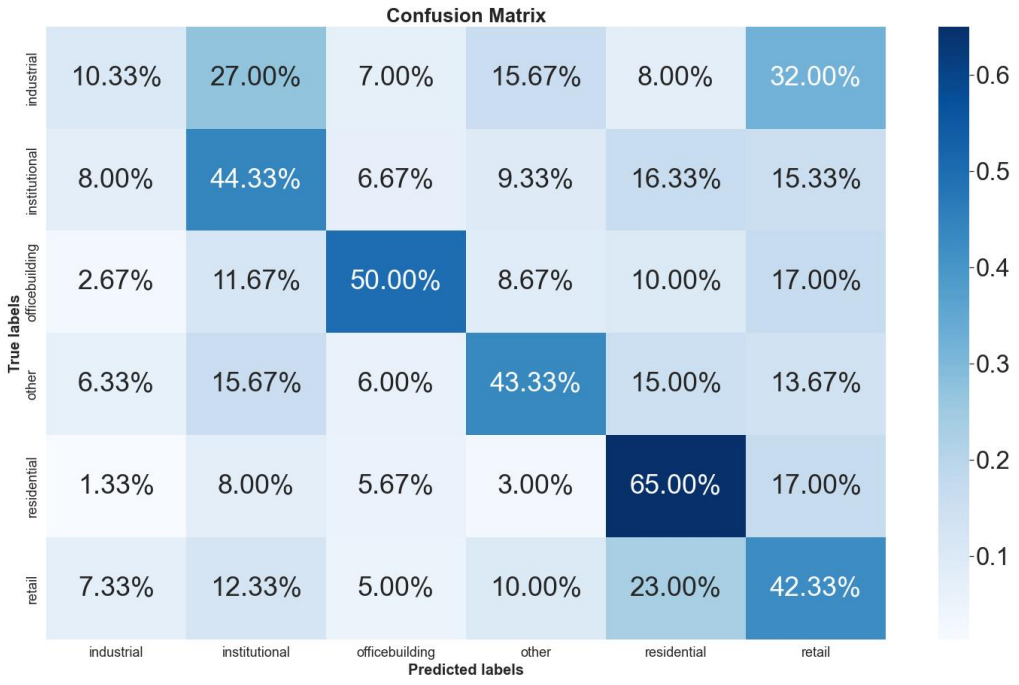


Figure C-1: InceptionV3 confusion matrix when training the model from scratch (Orthophoto)

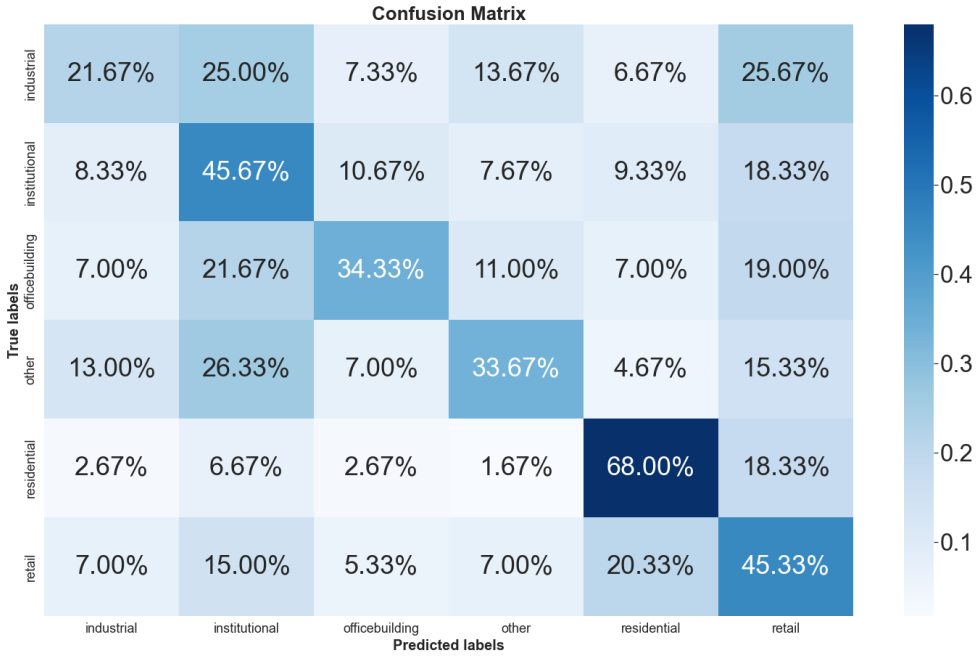


Figure C-2: InceptionV3 confusion matrix with 300 trained layers (Orthophoto)

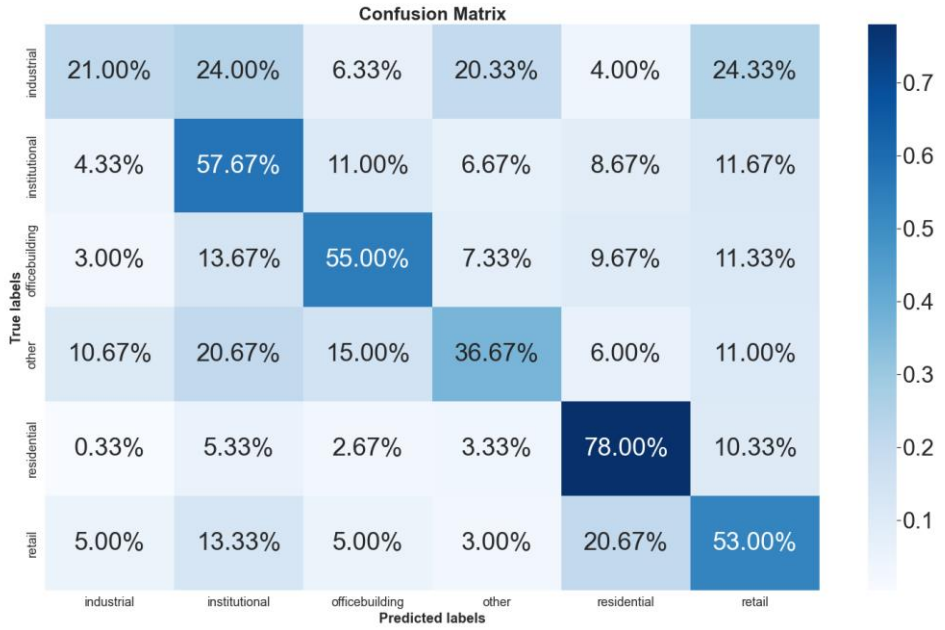


Figure C-3: InceptionV3 confusion matrix with 250 trained layers (Orthophoto)

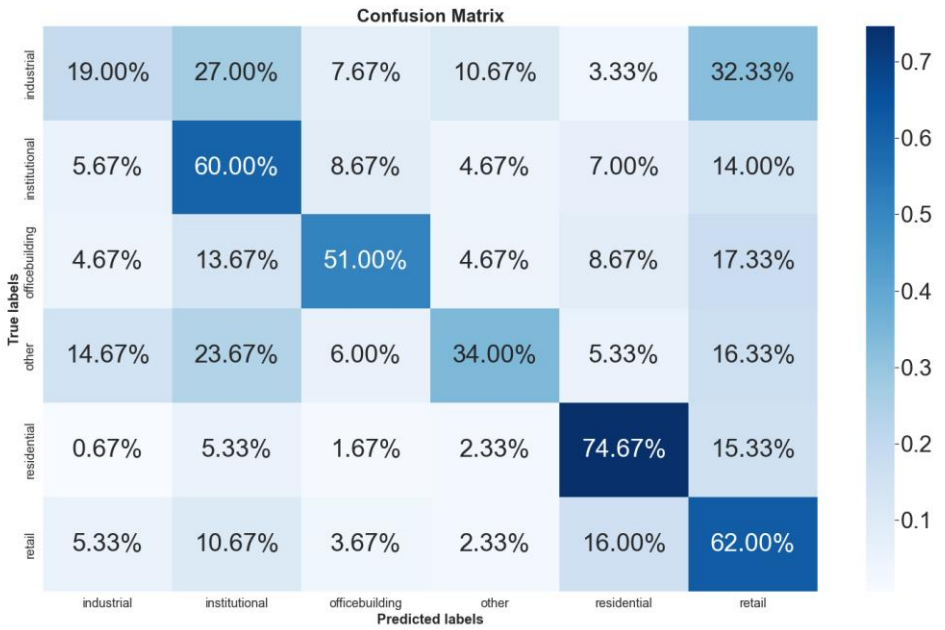


Figure C-4: InceptionV3 confusion matrix with 200 trained layers (Orthophoto)

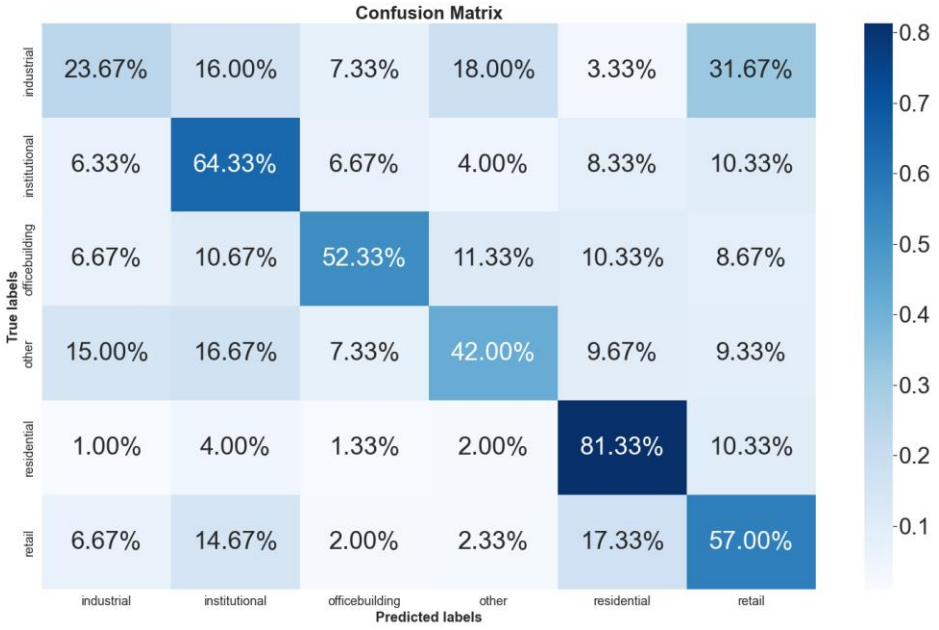


Figure C-5: InceptionV3 confusion matrix with 150 trained layers (Orthophoto)

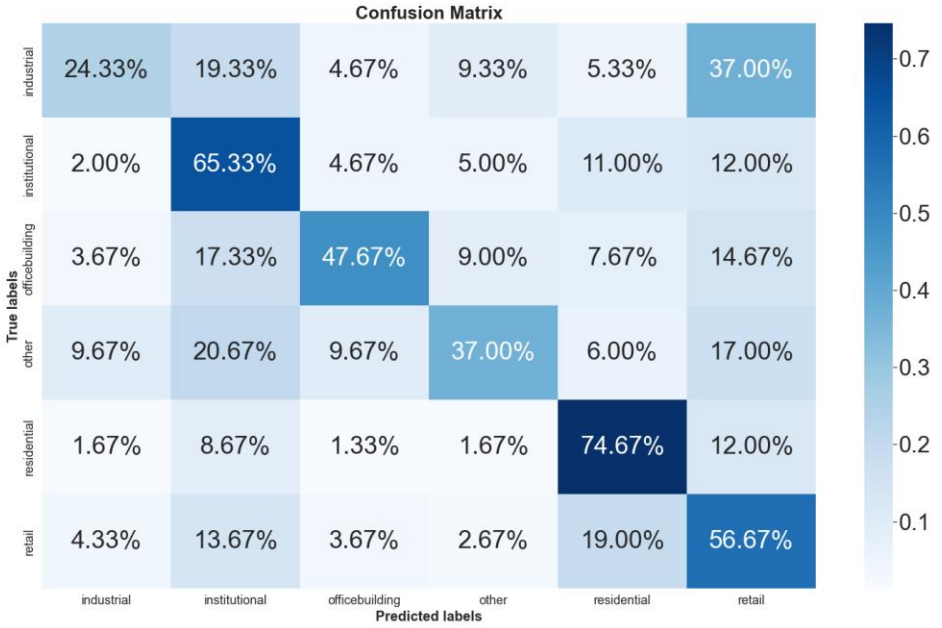


Figure C-6: InceptionV3 confusion matrix with 100 trained layers (Orthophoto)

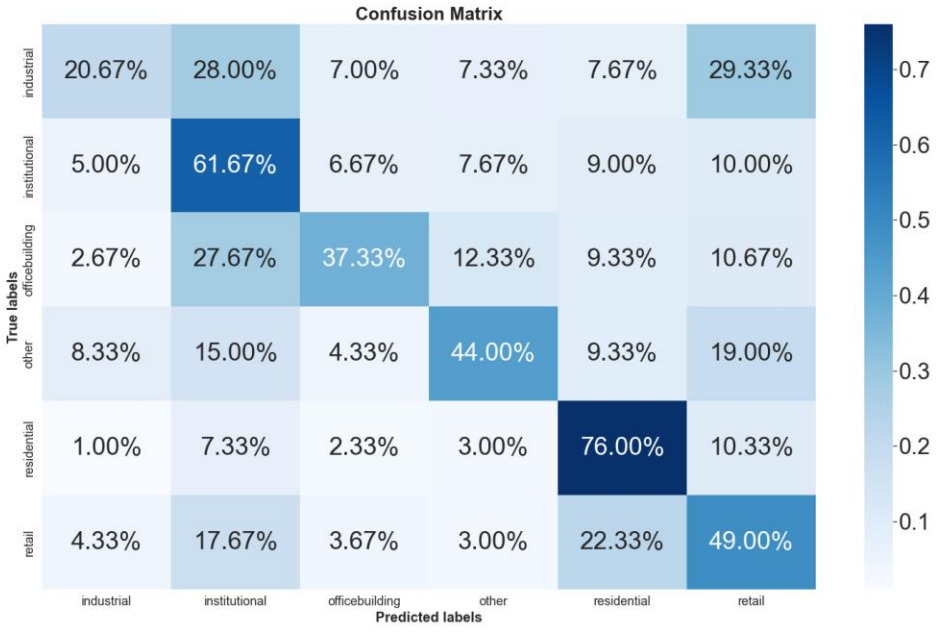


Figure C-7: InceptionV3 confusion matrix with 50 trained layers (Orthophoto)

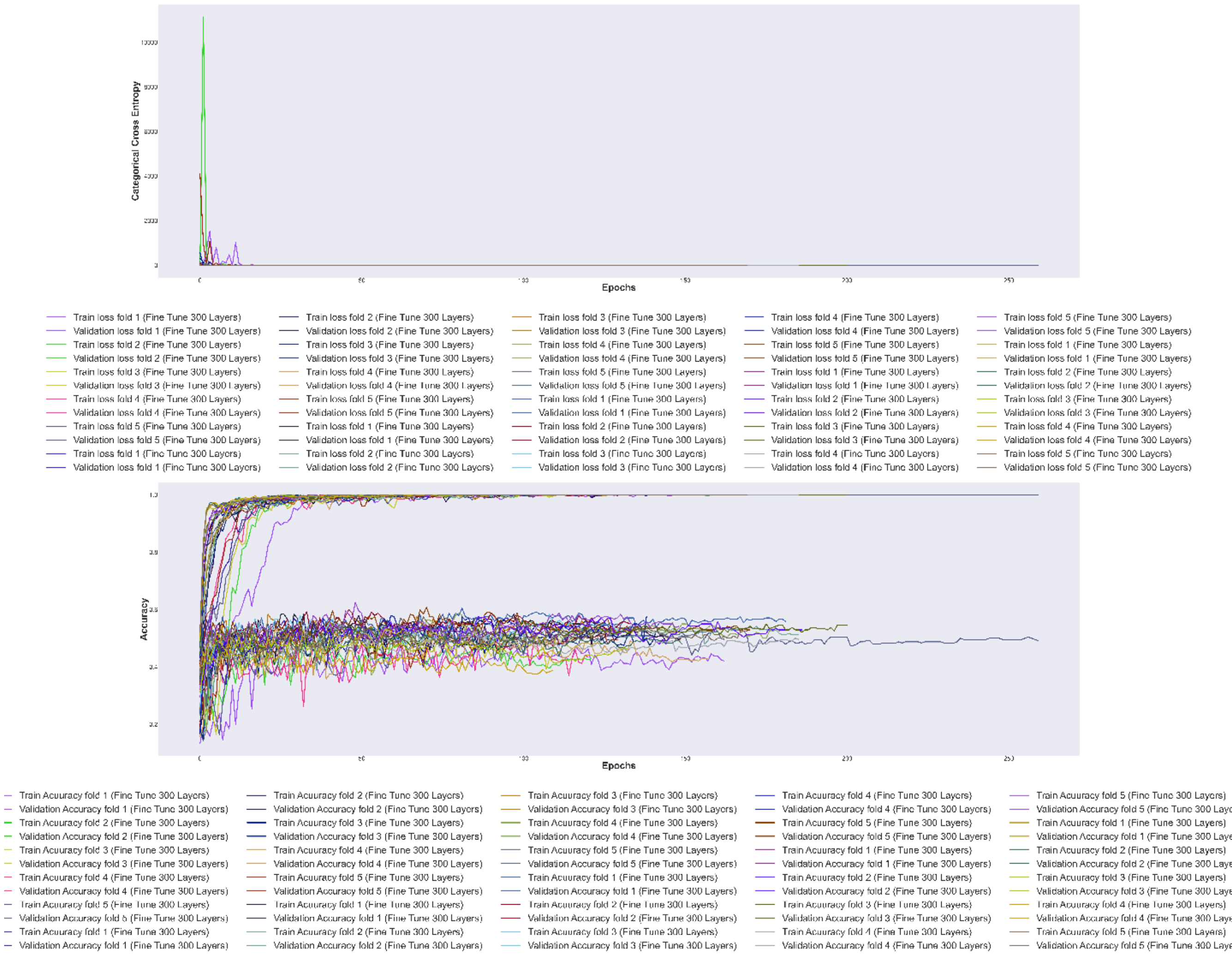


Figure C-8: Learning curves for InceptionV3 with learning rate 10-3 (Orthophoto)

Curriculum Vitae

Name: Nafiseh Ghasemian Sorboni

Postsecondary Education and Degrees:

University of Western Ontario
London, ON, Canada 2020-2024 Ph.D.

University of Tehran
Iran 2014-2016 M.Sc.

Babol Noshirvani University of Technology
Iran 2009-2014 B.E.

Related Work Experience:

Research Assistant

The University of Western Ontario, 2020-2024

Graduate Teaching Assistant

The University of Western Ontario, 2021-2024

Remote Sensing Specialist
Rayan Noor Tadbir Knowledge Foundation Co., Iran, 2020

University Instructor

Sariyan University, Iran, 2017-2018

Peer Reviewed Publications:

Ghasemian Sorboni, N., Wang, J., & Najafi, M. R. (2024). Fusion of Google Street View, LiDAR, and Orthophoto Classifications Using Ranking Classes Based on F1 Score for Building Land-Use Type Detection. *Remote Sensing, 16*(11), 2011.

Ghasemian, N., Wang, J., & Reza Najafi, M. (2024). Automated First Floor Height Estimation for Flood Vulnerability Analysis using Deep Learning and Google Street View. Journal of Flood Risk Management.

Ghasemian, N., Wang, J., & Reza Najafi, M. (2024). Urban flood mapping using Sentinel-1 and RADARSAT Constellation Mission image and Convolutional Siamese Network. Natural Hazard, 1-32.

Ghasemian, N., Wang, J., & Reza Najafi, M. (2022). Building detection using a dense attention network from LiDAR and image data. Geomatica, 1-28.

Selected Conference and Workshop Presentations:

American Geophysical Union (AGU) Fall 2022 meeting (12-16 Dec; attended as an online iposter presenter); Urban flood mapping using Sentinel-1 and RADARSAT Constellation Mission image based on a Convolutional Siamese Network

2022 Western – ICLR Multi-hazard Risk and Resilience Workshop (4-5 November; attended as a speaker); Building-scale flood risk analysis using a Deep Learning-based approach

10th International Conference on Argo-Geoinformatics and 43rd Canadian Symposium on Remote Sensing (ICAG-CSRS 2022) (July 2022, Quebec City, Canada; attended as an online presenter); Building detection using a dense attention network from LiDAR and image data

2021 Western – ICLR Multi-hazard Risk and Resilience Workshop (1-2 November; attended as a speaker); Flood Mapping Using Remote Sensing Data and a Siamese Network with a Weighted Loss Function