Cartography M.Sc.

Master thesis

Investigate Geographical Generalizability of GeoAI Methods for OpenStreetMap Missing Building Detection

Jiapan Wang

Technical University of Munich

TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology

TECHNISCHE UNIVERSITÄT DRESDEN

ITC UNIVERSITY OF TWENTE.

2023

Investigate Geographical Generalizability of GeoAI Methods for OpenStreetMap Missing Building Detection

submitted for the academic degree of Master of Science (M.Sc.)
conducted at the Department of Aerospace and Geodesy
Technical University of Munich

Author:	Jiapan Wang
Study course:	Cartography M.Sc.
Supervisor:	Dr. Hao Li (TUM)
Reviewer:	Assoc. Prof. Dr. Rolf de By (UT)

Chair of the Thesis
Assessment Board: Prof. Dr. Liqiu Meng

Date of submission: 07.09.2023

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

"Investigate Geographical Generalizability of GeoAI Methods for OpenStreetMap Missing Building Detection"

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Munich, 07.09.2023 Jiapan, Wang

Acknowledgments

I would like to express my gratitude to all those who have supported me throughout this challenging yet rewarding journey of completing my master's thesis.

First and foremost, I would like to express my sincere appreciation to Dr. Hao Li for his professional guidance, patience, and all-around support, which made me enjoy research, keep moving forward, and explore unlimited possibilities. Big thanks also to Prof. Martin Werner and Professorship for Big Geospatial Data Management for providing me with a workplace and the opportunity to participate in academic seminars. It was a very happy and informative journey to do my thesis within this group.

I would like to thank all my 11^{th} Cartography colleagues, who made this master program unique and wonderful. This will be an unforgettable and beautiful memory for me. Special thanks to Prof. Liqiu Meng, Juliane Cron, Barend Köbben, and all of the staff for keeping the program running smoothly and vibrantly.

Lastly, I would like to give my deep thanks and love to my family and friends who have always supported me and made me positive through this journey.

Thank you all for being a part of this significant milestone in my academic and life journey!

Abstract

OpenStreetMap (OSM), being the most successful Volunteered Geographic Information (VGI) project, holds a vital role in long-term open-sourced geospatial datasets, sustainable development, humanitarian activities as well as emergence response. Keeping OSM data up-to-date and complete in areas struggling with extreme poverty is an important mission for the mapping community. The development of Geographic Artificial Intelligence (GeoAI) and multimodal Earth Observation (EO) data in the last decade have shown opportunities for supporting automatic mapping processes. Moreover, it presents a promising prospect for assisting in humanitarian mapping, particularly in identifying areas lacking OSM buildings and estimating the quantity of missing buildings that need to be mapped.

Though conventional AI methods demand extensive training data, recent advancements in adapting pre-trained and task-agnostic AI models have yielded remarkable success in downstream tasks through fine-tuning, few-shot, or zero-shot learning. However, when adapting these pre-trained AI models to tackle geographic tasks, striking a balance between geographic generalizability and spatial heterogeneity of the model's performance remains a key challenge. For instance, detecting buildings across different regions of the world may require repeated training of GeoAI models, making it challenging to achieve model generalization across geographical space.

In this thesis, the Geographical Generalizability of GeoAI models was studied through the lenses of a case study of detecting OSM missing buildings across diverse regions in sub-Saharan Africa. Specifically, this study proposed a Geographical Weighted Model Ensemble (GWME) method, which began with training a Single-Shot Multibox Detetion (SSD) base model for OSM missing building detection in the source region (Kakola, Tanzania), and transferring this base model to a set of reference areas surrounding the target area (Babadjou, Cameroon) by multiple times Few-Shot Transfer Learning (FSTL), eventually ensemble multiple FSTL predictions according to unique weights, which represents the importance of reference areas to the target area. The determination of weights uses a pre-trained Vision Transformer model to simultaneously consider

both context and relative location information, called self-attention weights, compared with the other three weighting approaches (average, image similarity, and geographical distance). Intensive experiments showed the self-attention-based weighted model ensemble method achieved the highest performance with a 96.95% precision, 78.99% recall, and 0.8705 F1 score. The promising results shed inspiring light on improving the generalizability and replicability of GeoAI models across geographic space.

Furthermore, to explore GeoAI-enhanced web mapping applications, this thesis demonstrates a framework called GeoAI as a containerized microservice (GeoAIaaS), which utilizes microservice-based architecture with pre-defined mission recipes of handling geospatial data to lower the additional geography expertise barrier and improve the reusability of GeoAI solutions. This study conducts a building detection visualization application with standalone, distributed developable, and simply deployable microservices. The GeoAIaaS could inspire researchers with more possibilities in making web mapping applications across multiple platforms.

Keywords: GeoAI, OpenStreetMap, Model Ensemble, Object Detection, Microservice

Contents

List of Figures

List of Tables

Acronyms

API Application Programming Interface. 31, 32, 50

BBOX Bounding Box. 21, 35, 36, 38, 41, 45, 46, 48, 52

CNN Convolutional Neural Network. 13–15, 18

CV Computer Vision. 14, 18

DL Deep Learning. 11, 12, 26

EO Earth Observation. 8, 12, 21, 66

FCN Fully Connected Neural Network. 12

FSTL Few-Shot Transfer Learning. vii, 25, 33–35, 38–46, 48, 53, 55–57, 59

GAN Generative Adversarial Network. 19

GeoAI Geospatial Artificial Intelligence. 8, 9, 21, 23, 28, 32

GWME Geographical Weighted Model Ensemble. 33, 42, 44, 46–48, 53, 55, 59, 62

HOT Humanitarian OpenStreetMap Team. 7, 23, 54

HTTP Hypertext Transfer Protocol. 31

IOU Intersection Over Union. ix, 48, 49

LSTM Long- Short-Term Memory. 14–16

ML Machine Learning. 7, 27

MLaaS Machine Learning as a Service. 32

NLP Natural Language Processing. 14–16, 18

OGC Open Geospatial Consortium. vii, 30, 31, 51

OSM OpenStreetMap. 1, 2, 5, 12, 23, 33, 36, 50, 51, 55, 62

RNN Recurrent Neural Network. 15, 17

RS Remote Sensing. 11

SSD Single Short MultiBox Detector. 21

VGI Volunteered Geographic Information. 1, 5, 11, 23

VHR Very High Resolution. 21

ViT Vision Transformer. 18, 28, 35, 44, 45

WBF Weighted Boxes Fusion. 42, 45, 46, 59

WMS Web Map Service. 36, 51, 63

1. Introduction

1.1. Motivation and Problem Statement

Volunteered Geographic Information (VGI) is a valuable tool for a wide range of applications, including mapping, planning, and emergency response. OpenStreetMap (OSM), as one of the most successful VGI projects, plays an important role in supporting humanitarian mapping activities by providing open source and accurate geographical information (Herfort et al., 2021). While OSM data can be very valuable, it can also have some challenges. One of the main challenges with OSM data is that it's incomplete or out of date within some areas because it is collected and maintained by volunteers. As shown in Figure 1.1, the OSM building density varies across different regions in Africa [1].

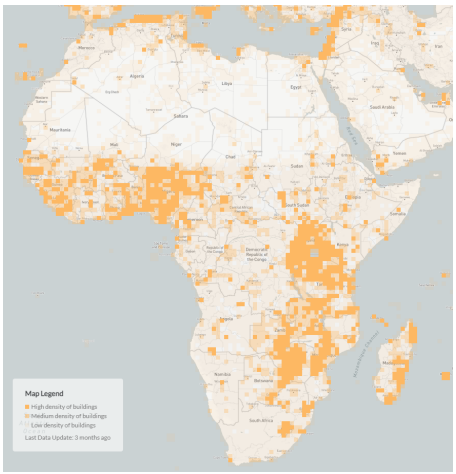


Figure 1.1.: The OSM building density map in Africa Region.

It is important to fill gaps in OSM data for a number of reasons. First and foremost, incomplete or out-of-date data can lead to misunderstandings or errors in humanitarian aid activities, which can have serious consequences. For example, if emergency responders are using outdated or inaccurate maps, they

[1] https://osm-analytics.org/#/

may have difficulty finding their way to the scene of an emergency, which could delay response times and put lives at risk. Similarly, if cartographers or planners are using incomplete or outdated data, they may make decisions based on inaccurate information, which could lead to inefficient use of resources or other problems.

It needs a huge amount of time to complete the OSM maps completely by volunteers. It's urgently needed to efficiently improve the existing mapping workflows. In the recent decade, the increasing availability of high-resolution satellite imagery allows for the enhancement and refinement of OSM data with machine learning (ML) techniques and offers a variety of promising solutions to this challenge currently encountered by humanitarian organizations (J. Chen & Zipf, 2017; Pisl et al., 2021; Vargas-Muñoz et al., 2019).

In contrast to traditional machine learning, which requires a large number of training samples, some few-shot learning methods have emerged in recent years to help accelerate the speed and accuracy of mapping in areas with a few map samples (B. Kang et al., 2019; Y. Wang et al., 2020). The FSTL method proposed by Li et al. has confirmed that in the sub-Saharan region, a pre-trained base model can perform well in a geographically remote area with few-shot or even one-shot training (H. Li, Herfort, et al., 2022). However, a remaining challenge for improving OSM missing buildings is the geographical generalizability of trained GeoAI models to detect spatially far areas. This research attempts to find a solution for improving the geographical generalizability of trained building detection models from the source region to the target region, which may have no training samples available. Furthermore, visualizing machine-generated geographical content and exploring how to integrate GeoAI solutions into a web mapping application is also an exciting challenge.

1.2. Research Objectives and Questions

The overall objective of this research is to investigate the geographical generalizability of GeoAI models for detecting OSM missing buildings and visualizing predicted geographical contents via a web client. In order to fulfill this research objective (RO), It was divided into the following sub-objectives:

- **RO1:** To implement Geospatial Artificial Intelligence (GeoAI) methods, which can be well-generalized for OpenStreetMap missing buildings detection across geographical space.
- **RO2:** Design a GeoAI application to efficiently manage, evaluate, and

visualize machine-generated geographic contents.

To meet the above research objectives, the following proposed research questions (RQ) and sub-questions need to be answered:

- **RQ1: How to improve OSM missing buildings by GeoAI methods across geographical space?**
  - RQ1.1: What is GeoAI?
  - RQ1.2: What GeoAI methods are commonly used for OSM missing building detection?
  - RQ1.3: How to enhance the geographical generalizability and evaluate the performance of the GeoAI solution to the geospatial object detection task?
- **RQ2: How to design and develop a GeoAI web application, especially for OSM building detection?**
  - RQ2.1: What is a web mapping application?
  - RQ2.2: How to efficiently visualize and manage machine-generated geographic content?
  - RQ2.3: How can GeoAI be integrated into web mapping applications?

1.3. Thesis Structure

The thesis structure is orchestrated as follows: Chapter 1 describes the background, motivation, and research objectives of the study. Chapter 2 reviews related work and introduces some basic theoretical background, including but not limited to VGI, GeoAI, and Web Mapping. Chapter 3 describes the detailed workflow of the proposed GWME methodology and depicts the GeoAIaaS-based infrastructure of the GeoAI-enhanced web mapping applications. Chapter 4 gives a brief summary of selected experiment areas, training datasets, and experimental configurations. Chapter 5 presents the results of the experiments and the user interface of the web mapping application. Chapter 6 discusses the solving ideas to research objectives across the overall thesis, as well as existing limitations and future perspectives of current work. Chapter 7 summarized the methodology and findings of the full thesis research.

2. Related Works and Theoretical Background

2.1. Volunteered Geographic Information

User-generated content and Web 2.0 (O'Reilly, 2005), make it possible for everyone to produce and share their own knowledge, and it is also the case for the generation and propagation of geographical data online (Gómez-Barrón et al., 2016). The utilization of Web 2.0 and crowdsourcing platforms for geographical data has led to the emergence and advancement of Volunteered Geographic Information (VGI), which was first termed by M. F. Goodchild (2007) in the article "Citizens as sensors". With the rapid development of communication technologies (low-cost GPS devices, mobile location based service (LBS) platforms, high-resolution satellite imagery) and geographic information integrated software, many well-known VGI projects appeared, including OpenStreetMap, Mapillary, Wikimapia, etc. Aiming at encouraging the public to create, assemble, and disseminate geographic information, VGI illustrates a great potential to be a significant digital source of geographic understanding of the Earth.

2.1.1. OpenStreetMap and Humanitarian Mapping

OpenStreetMap (OSM) has become the most successful crowdsourced volunteered geographic information project to date since it was founded in 2004 (Minghini & Frassinelli, 2019). It aims to create an open-source map dataset that is free to use, editable, and licensed under new copyright schemes (Haklay & Weber, 2008). In areas where access to geographic information is regarded as a national security problem, OSM may provide the cheapest source of geographic information, and sometimes the only source (M. F. Goodchild, 2007).

Due to its open-source nature, relatively complete data quality, and accurate geographic information, OSM is considered the most popular and widely used VGI platform, which can largely support humanitarian mapping and Sustainable Development Goals (SDG) as well as fill geographic data gaps for the entire world.

Herfort et al. (2021) have summarized the evolution of humanitarian mapping activities within OSM in Figure 2.1. The sketch illustrates that OSM highly contributed to humanitarian activities, like humanitarian mapping response in 2010 Haiti earthquake (Zook et al., 2010), West Africa Ebola outbreak in 2014 (Dittus et al., 2016), humanitarian mapping Cyclone Idai and Kenneth in Mozambique (H. Li et al., 2020), supporting global response to the COVID-19 pandemic (Minghini et al., 2020), and the 2023 Turkey Syria earthquake (OpenStreetMap Wiki, 2023). More and more humanitarian mapping activities have proven the huge value and potential of OSM for rapid emergency response when disaster comes. However, in order to provide the most accurate and fastest humanitarian aid response, it is important to boost the speed and accuracy of mapping up-to-date and complete, high-quality geographic information within OSM.

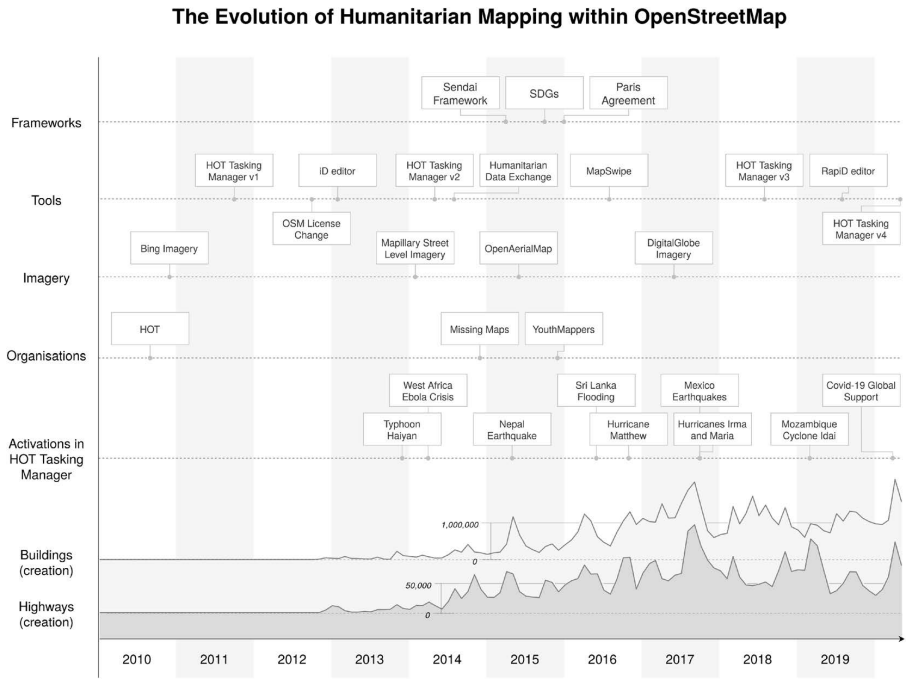


Figure 2.1.: The evolution of humanitarian mapping activities (until 2020) within OSM, the tools development and imagery sources. Plots at the bottom display the counts of buildings and highways added to OSM (Herfort et al., 2021).

2.1.2. Mapping Challenges for OSM

Although OSM data holds great value for emergency response, post-disaster mapping, and various humanitarian activities, accelerating the speed of humanitarian mappings with less volunteer effort and employing more automatic processes remains a significant challenge. This section explores the integration of very high resolution satellite imagery and artificial intelligence as auxiliary tools to assist volunteers in contributing data to OpenStreetMap.

Several platforms like Humanitarian OpenStreetMap Team (HOT), MapSwipe, Ushahidi, and RapiD have emerged to facilitate and streamline the mapping process. HOT (Soden & Palen, 2014) plays a vital role in coordinating volunteers to respond to mapping tasks efficiently. It streamlines the mapping workflow and ensures that the most critical areas are mapped promptly. MapSwipe (J. Chen & Zipf, 2017) is another tool that engages volunteers in the initial stage of mapping by asking them to identify areas that require mapping. This crowdsourced tool optimizes the allocation of mapping resources. Ushahidi (Ashley, 2009) focuses on collecting and managing information from various sources, including social media and SMS, to improve the accuracy and comprehensiveness of OSM data. RapiD (OpenStreetMap Wiki, 2019) leverages artificial intelligence to semi-automatically extract features from high-resolution satellite imagery, significantly reducing the manual effort required for mapping. Despite the exponential growth of OSM data contributed by these auxiliary tools and volunteer contributions, the issue of "geographical information is usually the least available where it is most needed" continues to present a significant challenge (D. Sui et al., 2013). The implications of this issue can lead to misunderstandings and errors in humanitarian aid activities, which can have serious consequences. Humanitarian organizations usually lack the precise location information of missing OSM data and the capacity of needed volunteer efforts. Particularly, vast rural regions in Sub-Saharan Africa remain unmapped (H. Li, Herfort, et al., 2022). Therefore, novel approaches are needed to optimize humanitarian mapping processes, aiming at reducing volunteer efforts and accelerating mapping speed.

Fortunately, the growing accessibility of high-resolution satellite imagery enables the enhancement and refinement of OSM data through the application of Machine Learning (ML) technologies, providing promising solutions to the existing challenge faced by OSM community (J. Chen & Zipf, 2017; Pisl et al., 2021; Vargas-Muñoz et al., 2019). Earlier research (J. Chen et al., 2019; Herfort et al., 2019; Huck et al., 2021) have found the speed and accuracy improvement of mapping OSM data leveraging ML technologies. The data quality of massive ML-based mapping results is still a concern from OSM

community ("Import/Guidelines OSM Wiki", 2023). To figure out this issue, machine-assisted mapping approaches are used instead to reduce the risk and error of ML mapping results (OpenStreetMap Wiki, 2019). Early works by Kaiser et al. (2017) and Mnih and Hinton (2012) explored the possibility of integrating OSM data as training samples used by training deep neural networks (DNNs) to solve segmentation tasks of streets and buildings. J. Chen et al. (2019) introduced a machine-assisted mapping method involving the concept of active learning from multiple crowds, which was subsequently validated in humanitarian activities conducted in Malawi. Nonetheless, ML-based methods demand large datasets for training. In regions with high OSM data density, the existing OSM data can be employed to generate training datasets. Conversely, in regions with limited OSM data density, like Sub-Saharan Africa, H. Li, Herfort, et al. (2022) introduced a method that involves transferring the pre-trained model from high-quality areas to distant geographic areas. This approach utilizes few-shot training samples from the target area to fine-tune the generalization of the pre-trained model. Their work showcased the transferability of AI models in tackling geographic problems with limited computational resources and efforts. However, a significant challenge persists in enhancing the generalizability of AI models and effectively replicating them in geographically remote areas with no OSM data. Research focused on adapting pre-trained AI models to low-resource remote regions (using few-shot and zero-shot techniques) is yet to be explored.

2.2. Geospatial Artificial Intelligence

Geospatial Artificial Intelligence (GeoAI) has emerged as a research hotspot and cutting-edge frontier for spatial analysis in the field of Geography. Significant advancements have been achieved in the innovative application and expansion of AI to solve geographic problems. Derived from computer science, AI research is dedicated to the advancement of computer systems that acquire machine intelligence, simulating human ways in perceiving, reasoning, and interacting with the world and with each other (Russell, 2010). There's no doubt that AI community significantly enhances research and applications in geography. Simultaneously, the geography community boasts a wealth of geographic data and expertise that facilitates data-driven research with AI technologies. For instance, multi-modal Earth Observation (EO), which provides a vast amount of remote sensing imagery data at high, spatial, temporal, and spectral resolution, making it a frequently used geospatial dataset in AI applications (J. Li et al., 2022).

The term "GeoAI" was first coined by Mao et al. (2018) and has since gained popularity among researchers working with data mining, machine learning, or high-performance computing in the field of (big) geospatial data analysis. W. Li (2020) illustrates a conceptual representation of GeoAI, depicting it as a convergence of AI, geospatial big data, and high-performance computing (HPC) in Figure 2.2. This presents GeoAI as a promising technological solution for data- or compute-driven geospatial challenges. Recent research demonstrates great potential for implementing GeoAI methods in the domain of Cartography and Mapping, especially deep learning for cartographic design and image colourization(X. Huang et al., 2019; Y. Kang et al., 2019; M. Wu et al., 2021), detection and extraction of map objects, symbols and texts (H. Li, Herfort, et al., 2022; Xie et al., 2020; Yan et al., 2021), and cartographic generalization (Feng et al., 2019; Touya et al., 2019). Multiple instances of AI and GI Science fusion have demonstrated the significance and forthcoming research potential in the GeoAI landscape.

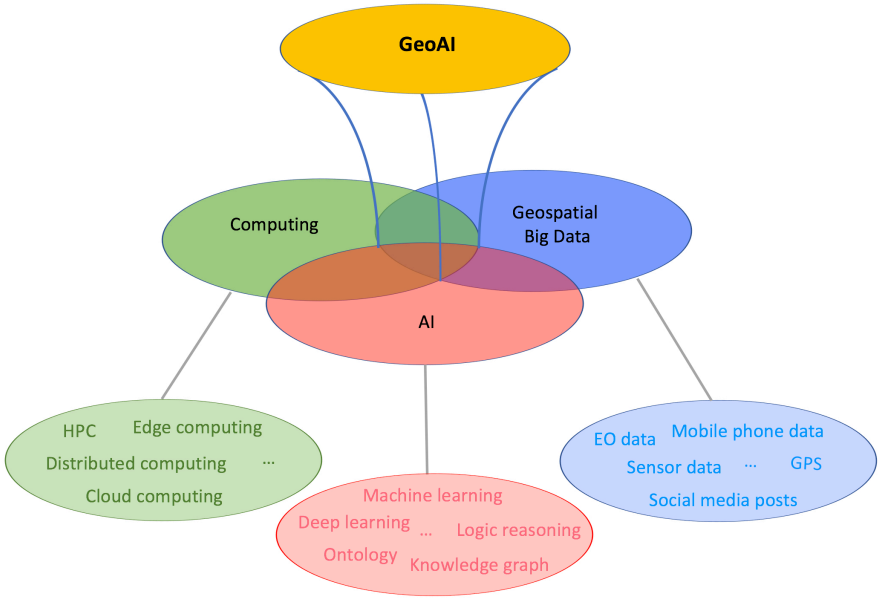


Figure 2.2.: A conceptual, three-pillar view of GeoAI (W. Li, 2020).

2.2.1. Big Geospatial Data and Machine Learning

GeoAI is an emerging interdisciplinary field focusing on the exploration and advancement of AI applications for geography-related analysis employing big geospatial data (W. Li & Hsu, 2022) as Figure 2.3 shown. To better investigate

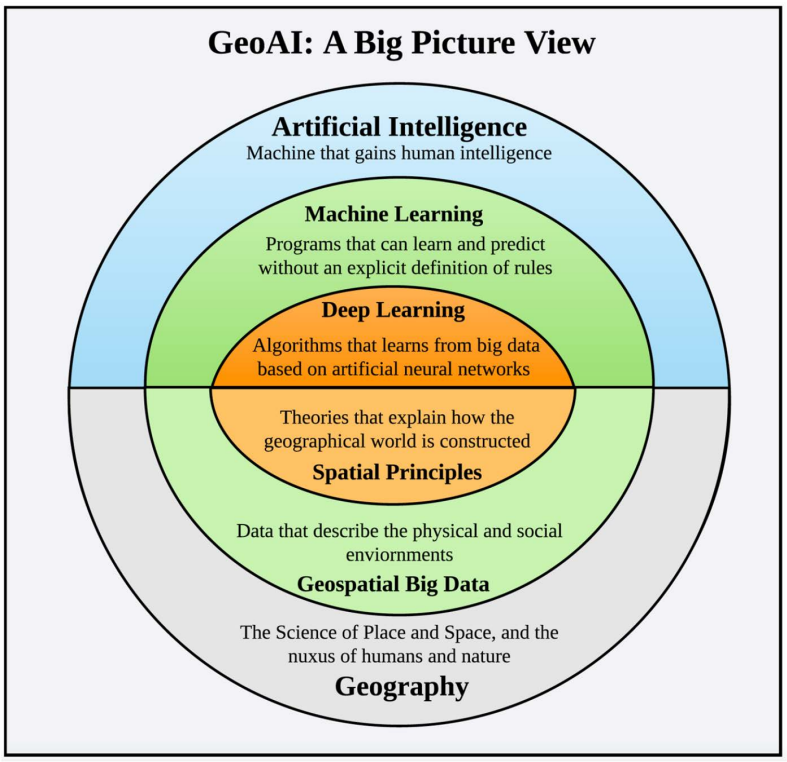


Figure 2.3.: A big picture view of GeoAI (W. Li & Hsu, 2022).

GeoAI, a robust convergence between AI and Geography is essential. Geography provides not only geospatial data sources but also a distinctive perspective for understanding and abstracting the world and society via established geographic principles like Tobler's first law of Geography (Tobler, 1970) and the second law of Geography (M. F. Goodchild, 2004). This knowledge from the Geography domain will extend current AI capabilities into spatially-explicit GeoAI techniques and solutions (Janowicz et al., 2020), enabling AI be more properly adapted to the geospatial domain. Recently, researchers in GeoAI have shown increased enthusiasm for utilizing deep learning methodologies, like convolutional neural networks (CNNs), to address geographical challenges through image analysis. The following sections briefly describe different types of big geospatial data for image analysis and mapping, popular deep learning methods, and GeoAI applications.

Big Geospatial Data for Image Analysis and Mapping

With the continuous advancement of hardware facilities and modern technology, diverse geographic-related data is swarming out. One of the key focuses of

GeoAI research is how to use geographic data in a rational and effective way. Some common geographic data categories are outlined next to enhance a better understanding of geographic data.

- **Remote Sensing (RS) Imagery** - Recognized as a highly utilized and significant geospatial data source, have a great potential for monitoring and management of diverse spatial data on large-scale areas. Multiple observation platforms like satellites, unmanned aviation vehicles (UAVs), aircraft, and diverse spatial, temporal, or spectral resolution sensors including multi- or hyper-spectral, LiDAR, synthetic aperture radar (SAR), enable RS data to be very valuable for Earth-related applications. With the rapidly growing accessibility of multi-modal RS data, researchers can easily extract information from Earth's surface and apply modern DL methodologies to solve geographic problems (J. Li et al., 2022).
- **Street View Images** - It has become a useful source to extract information from a human-centric perspective which contains not only geographic information but also the social environment (Y. Liu et al., 2015). Both big companies like Google, Tencent, and VGI platforms like Mapillary are interested in gathering and providing street view images to support GeoAI research and applications. For instance, H. Li, Yuan, et al. (2023) estimate building height leveraging street view images from a real-world perspective. As street view images become progressively richer, more and more GeoAI methods will derive meaningful information from such human-centric geographic data.
- **Geo-scientific Data** - Observations of physical phenomena on Earth's surface are of great significance for the study and advancement of human society. Such Geo-scientific Data are usually divided into two categories, sensor data, and simulated data. The former can be obtained from diverse environmental sensors, such as temperature, humidity, microwave, and infrared sensors, while the latter can be derived from Earth's environmental models, such as the water cycle, the atmospheric cycle, the carbon cycle (Carvalhais et al., 2014; Forkel et al., 2015). These two types of data can assist humans understand the principles of natural phenomena and predict future developments. For example, large-scale Geo-scientific data play an important role in weather prediction, flood observation, and fire detection. The rise of AI has provided a powerful aid to research in environment change in terms of handling multi-dimensional data, time series analysis, etc (Kadow et al., 2020).
- **Volunteered Geographic Information** - VGI is of immeasurable importance

for Geographic Information Science (GIScience) research across global regions. Especially in areas lacking official geographic data sources, open source, freely accessible, and relatively complete data like VGI stands as the only geographic data available to them. OSM, as the largest VGI platform, plays an important role in supporting humanitarian mapping, Sustainable Development Goals, and the daily use of maps (Herfort et al., 2021). The integration of OSM with Multi-modal EO data has emerged as a potent and robust geographical dataset for supporting GeoAI research (H. Li, Herfort, et al., 2022; H. Li, Yuan, et al., 2023; Minghini et al., 2020). In turn, GeoAI has fueled the growth and progression of VGI by minimizing human resources and augmenting computational efficiency (OpenStreetMap Wiki, 2019).

Deep Learning and Neural Networks

As another key focus of GeoAI research, a subset of machine learning, Deep Learning (DL) enables computational models with a multi-processing-layer structure to learn multi-level representations of data. DL performs well in discovering intricate patterns in big geospatial datasets by using the back-propagation algorithm to iteratively refine the internal parameters of the model, thereby guiding the evolution of representations from one layer to the subsequent layer (LeCun et al., 2015). Some state-of-the-art neural network architectures are briefly described as follows.

- **Fully Connected Neural Network (FCN)**

  Classical artificial neural network models are the basis for current popular complex network models. The Feedforward Neural Network (Figure 2.4) is one of the most basic models, which uses nodes and connections to simulate the signal propagation between human neurons (Shrestha & Mahmood, 2019). This model type is also called Fully Connected Neural Network (FCN), which contains an input layer, an output layer, and several hidden layers in between connecting them. Each node in each layer is connected to all nodes in the previous layer. During the training process, this network can learn the nonlinear relationships between input and output by adjusting the connections (weights) between nodes. Although FCNs were widely used in many classification algorithms leveraging their good capability for extracting features of input data, they always suffer from two major shortages: the network needs to manually design feature extractor to collect information from the input. Additionally, multiple neural layers have to be stacked When tackling intricate problems, and it is

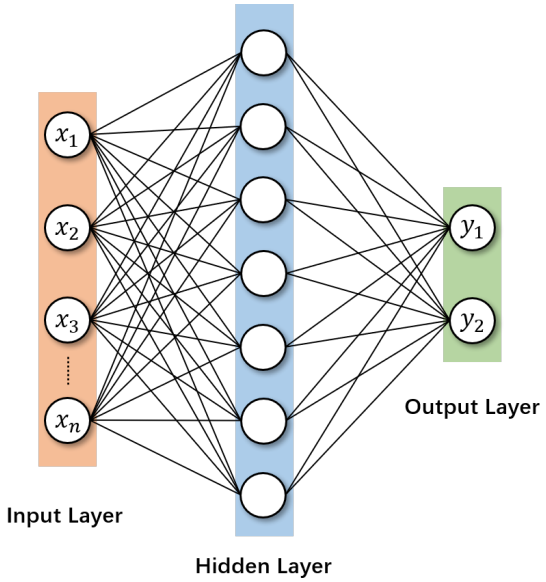


Figure 2.4.: A fully connected neural network with one hidden layer.

often incredibly computationally expensive and slow. To reduce learning parameters, newer neural networks have emerged consequently, one of which is Convolutional Neural Network.

- **Convolutional Neural Network (CNN)**

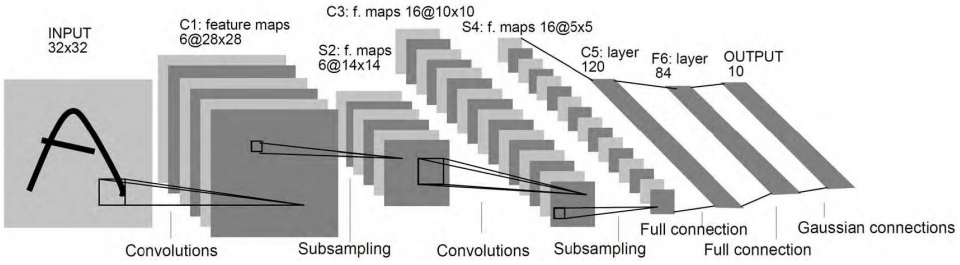


Figure 2.5.: LeNet-5: A 7-layer architecture of CNN for digit character recognition (LeCun et al., 1998).

The emergence of Convolutional Neural Network (CNN) has given a breakthrough in handling big data and computational speeds for AI. Instead of manually designed extractors, CNNs contain the automatic feature extractors to reduce tons of weights by leveraging a convolution operation that uses a sliding window to calculate the dot product of the input data, which could be 1D, 2D or 3D. Each convolution kernel (filter) can extract a specific feature from the original data, and generate a feature map,

which represents this feature's distribution across the whole area. The convolution kernels are learnable parameters, which are usually randomly initialized before training. During training, the kernels are updated aiming at minimizing the loss of the model, and are fixed after training. Normally a pooling layer (or called down-sampling) is connected after a convolution layer to ensure the prominent feature is preserved and also compress the feature map, thus reducing computation cost. Multiple CNN layers can be stacked to extract high-level features by semantically composing low-level features extracted at the beginning part. Following feature extraction, the network can be extended for diverse applications, thus, CNN can be regarded as a general automatic feature extractor.

Depending on the format of the data input to a CNN model, it can be categorized as 1D CNN, 2D CNN, or 3D CNN. The 1D CNN often applies on 1D vector space for processing sequential data, like language text, or audio segment. On the other hand, the 2D CNN has the ability to extract features from the planer spatial domain, like horizontal/vertical edges and corners, making it suitable for image processing. Furthermore, extending 2D CNN to 3D CNN can take care of temporal dimensional data, like video frames. As one of the best application scenarios with CNNs models, figure 2.5 illustrates the well-known 7-layer architecture of CNN model for digit recognition, called LeNet-5 (LeCun et al., 1998). The consequent convolution layers progressively refined feature extraction from input images to output layers, and fully connected layers perform digit classification according to the extracted high-level features. Subsampling layers are inserted between convolution layers to reduce parameters. This kind of network structure has since been widely used in a variety of image-based applications.

Because of the outstanding ability in considering the local receptive field instead of connecting all neurons and using shared weights for each neuron to look for the same pattern but in different parts of input images, CNN-based networks gain significantly improved performance in both accuracy and efficiency for both Natural Language Processing (NLP), and Computer Vision (CV) applications (Alzubaidi et al., 2021). Therefore, CNN becomes an essential building block for deep learning models.

- **Long- Short-Term Memory (LSTM)**

  While CNN has been widely used in various AI applications, especially in the CV field, such as image classification, object detection, and semantic segmentation, it still has limitations when dealing with sequential data.

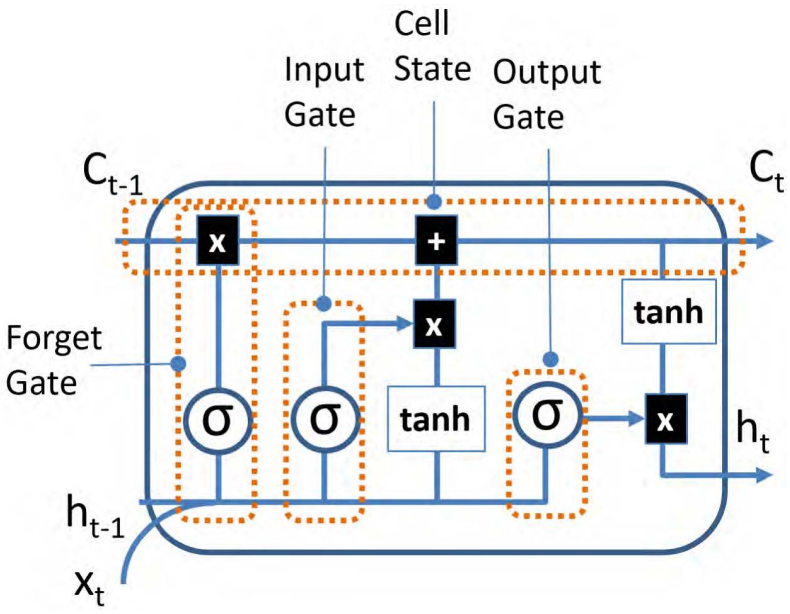


Figure 2.6.: A LSTM block with memory cell and gates (Shrestha & Mahmood, 2019).

Conventional CNN models typically require a fixed-size input and yield a fixed-size output, which is not a problem for image processing. However, when dealing with sequential data, CNN can not consider intrinsic interrelations and contextual influence among datasets, which is a major concern in addressing NLP challenges. Then Recurrent Neural Network (RNN) has been developed to use a hidden state as a form of memory that carries information from one step to the next. Unlike traditional feedforward neural networks, where data flows in one direction from input to output, RNNs have connections that loop back on themselves, enabling them to maintain the contextual memory (Cho et al., 2014). RNNs are particularly suitable for sequential tasks, such as time series data, NLP, and other tasks where the order of data points matters.

As an implementation of RNN, Long- Short-Term Memory (LSTM) was introduced by Hochreiter and Schmidhuber (1997) to address the vanishing gradients problem of RNNs. Figure 2.6 illustrates a LSTM block that consists of a memory cell state through which data flows while being controlled by input, forget, and output gates (Shrestha & Mahmood, 2019). In the iterative training process, the input gate determines the quantity of data from the prior state that should influence the current state. The forget gate evaluates the significance of different memory segments and discards

less important information. Meanwhile, the output gate orchestrates the combination of newly extracted memory with filtered memory to provide an accurate prediction for a further state (W. Li & Hsu, 2022). Due to its ability for time sequence predictions by dynamically capturing memories, it can serve for land use and land cover change by employing a time series of satellite images (Sherley et al., 2021). Another interesting research of LSTM in GeoAI is object detection (Hsu & Li, 2021). Despite a standalone image's absence of temporal information, the 2D image can be transformed into a 1D sequence via a scanning pattern. Then LSTM has the ability to capture the interrelations of the same object throughout the entire 1D sequence.

Various studies have shown that LSTM is well suited for the prediction of sequential data, such as speech recognition, text translation, and other NLP tasks, but it also shows some potential to solve GeoAI problems such as object detection, land use detection, etc. by utilizing satellite images.

- **Transformer**

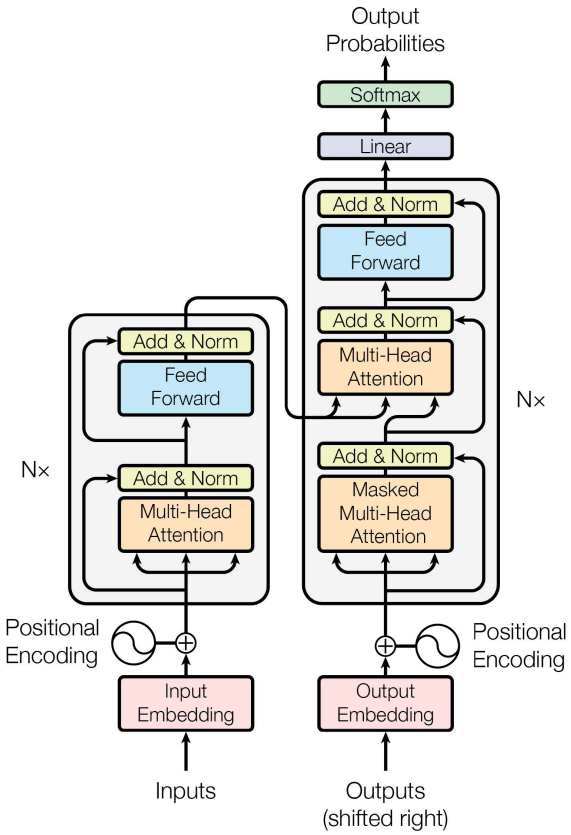


Figure 2.7.: The Transformer architecture (Vaswani et al., 2017).

The introduction of Transformer in the paper "Attention Is All You Need" has sparked a significant revolution in the AI field (Vaswani et al., 2017). The Transformer block is based on an encoder and decoder architecture that implements sequence-to-sequence learning, as shown in figure 2.7. In contrast to iteratively processing sequence segments from start to end of a sequence in RNN, Transformer possesses the ability in taking the entire sequence as input and capture the interrelationships between all word pairs in a sequence by assigning different weights to each word based on its relevance to other words in the sequence. This is called the attention mechanism, which enables the model to focus on the most relevant parts of the input sequence while performing various tasks. The Transformer model uses multiple attention heads in parallel, allowing it to capture different types of relationships and patterns within the data. The outputs from these attention heads are then concatenated and linearly transformed to obtain the final attention output for each word in the sequence. The attention module is repeatedly used in the encoder and decoder within the Transformer block.

Instead of directly processing the raw input, the encoder firstly converts the words into numeric representations with an input embedding and additionally uses a positional embedding to represent the position of each word within the input sequence. The embedded numeric representations will be given to the attention module, then connected with a feedforward layer for further steps. In summary, the encoder runs in parallel to derive attention scores that represent the semantics of each word in the input sequence.

The decoder part has a similar embedding and attention process as the encoder. It takes the output sequence with positional embedding and output embedding same as input embedding to get the numeric representations of a sequence of words. The embedded vectors will be delivered to an attention module, which is called masked attention. Differentiating from the attention module within the encoder, the masked attention module measures the attention score for each word only by considering the words before it rather than all words in a sequence, because the decoder is designed for predicting the forthcoming word in a sequence. After the masked attention module, the decoder also has another attention module, which will jointly take the embedded input sequence and embedded output sequence to calculate relations, or called to make predictions. The loss function will be used to minimize the difference between semantic representations of the input sequence and the predicted output sequence. Thus,

the Transformer could be widely used for various sequence-to-sequence tasks, such as machine translation, speech recognition, text generation, etc.

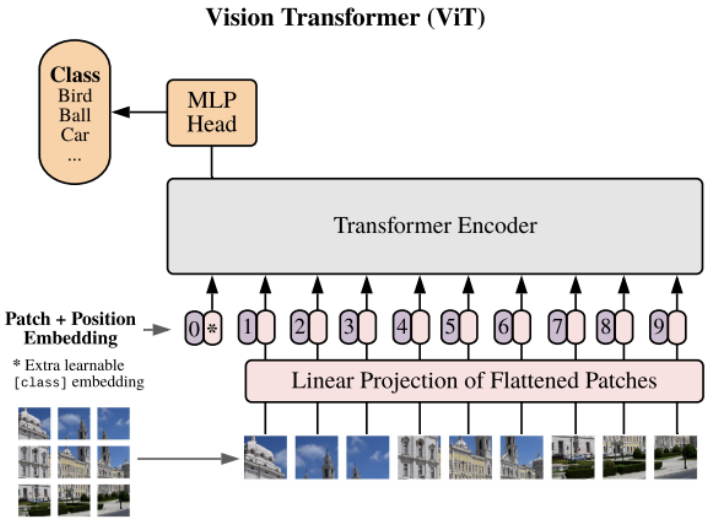


Figure 2.8.: The Vision Transformer architecture (Dosovitskiy et al., 2021).

Besides outstanding performance in NLP, Transformer models have been extended for processing images in CV domain, by splitting a full image into small patches and regarding an image patch inside an image as a word inside a sentence. In contrast to CNN focusing on the local receptive field of the image features, the Transformer can dynamically determine the size of the receptive field across the entire image. An exciting example is Vision Transformer (ViT) (Figure 2.8), enabling patch and positional embedding, which shows great potential on various computer vision tasks, including image classification, object detection, semantic segmentation, and more (Dosovitskiy et al., 2021). It can achieve similar or even better performance than CNN (Dosovitskiy et al., 2021). The success of ViT has sparked interest in exploring the potential of the Transformer architecture beyond natural language processing, highlighting its ability to capture long-range dependencies and spatial relationships in images effectively. Another trending research is combining language and images to build more complex models, such as text-to-image, image-to-text, semantic image retrieval, etc (Minderer et al., 2022; Radford et al., 2021). Undoubtedly, Transformer-based architecture gives researchers more chances to imagine and address challenging problems in the future.

In summary, these progressively emerging neural network models have revolutionized deep learning performance and applications. The foundational models mentioned above may be well suited for specific domains, such as sequence or/and image data, however, they can also be incorporated to build more complex models, such as Reinforcement Learning, Generative Adversarial Network (GAN), and Self-Supervised Learning. Their applications in the GeoAI domain will be reviewed in the following section.

GeoAI Applications

This section reviews and summarizes current applications of GeoAI, leveraging remote sensing imagery and ML/DL methods. Major GeoAI applications are categorized as object detection, classification, segmentation, height/depth estimation, image super-resolution, object tracking, change detection, forecasting, etc. Table 2.1 provides a brief summary in terms of task types, application scenarios, application methods, and references.

Table 2.1.: Review of GeoAI applications

Task	Application	ML/DL Approach	Reference
Object detection	Search and rescue operations	CNN	Bejiga et al. (2017)
	Aircraft detection	CNN	Zhou et al. (2021)
	Military object detection	CNN	Janakiramaiah et al. (2023)
	Terrain feature detection	CNN	W. Li and Hsu (2020)
	Building detection	CNN	H. Li, Herfort, et al. (2022)
Classification	Land use classification	CNN	Y. Yang and Newsam (2010)
	Multi-label scene	CNN	Kumar et al. (2021)
	Multi-label scene	GAN	Khan et al. (2019)
	Multi-label scene	CNN+GAN	Y. Li et al. (2020)
	Forest areas classification	Stacked auto-encoder	Haq et al. (2021)
Segmentation	Urban land cover	CNN	Kampffmeyer et al. (2016)
	Cloud segmentation	CNN	Mohajerani and Saeedi (2021)
	Road extraction	CNN	Grillo et al. (2021)
	Fire perimeter	3D U-Net	Doshi et al. (2019)
	Precision agriculture	CNN	Osco et al. (2021)
Height/Depth Estimation	Monocular depth estimation	Self-Supervised	Klingner et al. (2020)
	Height estimation	RCNN	Mou and Zhu (2018)
	Collapsed buildings	CNN	Amini Amirkolaee and Arefi (2019)
	Height estimation	CNN	Srivastava et al. (2017)
	DSM generation	Semi-global and block matching	W. Yang et al. (2020)
Image super resolution	Hyperspectral image	CNN	Fu et al. (2019)
	Hyperspectral image	CNN	X.-H. Han et al. (2018)
	Hyperspectral image	Spatial-spectral prior network	Jiang et al. (2020)
	DEM super-resolution	EfficientNetV2	Demiray et al. (2021)
	Lake area	Deep gradient network	Qin et al. (2020)
Object tracking	Satellite videos	Multiframe Optical Flow Tracker	Du et al. (2019)
	UAV aerial video	Saliency Enhanced MDnet	Bi et al. (2019)
	Satellite videos	Velocity correlation filter	Shao et al. (2019)
	Satellite videos	Rotation-adaptive correlation filter	Xuan et al. (2021)
	Multi-object tracking	Graph-Based Multitask Modeling	Q. He et al. (2022)
Change detection	Remote sensing	Fast RCNN	Q. Wang et al. (2018)
	Remote sensing	Attention Metric-Based	Shi et al. (2022)
	Land cover	LSTM+FCN	Sefrin et al. (2021)
	Remote sensing	Improved UNet++	Peng et al. (2019)
	Remote sensing	Transformer	H. Chen et al. (2022)
Forecasting	Drought prediction	LSTM	Poornima and Pushpalatha (2019)
	Wind Speed prediction	DBN	Wan et al. (2016)
	Drought prediction	EMD+DBN	Agana and Homaifar (2018)
	PM2.5 Prediction	FCN	Zamani Joharestani et al. (2019)
	LU/LC prediction	Transformer	Mohanrajan and Loganathan (2022)

2.2.2. Geospatial Object Detection

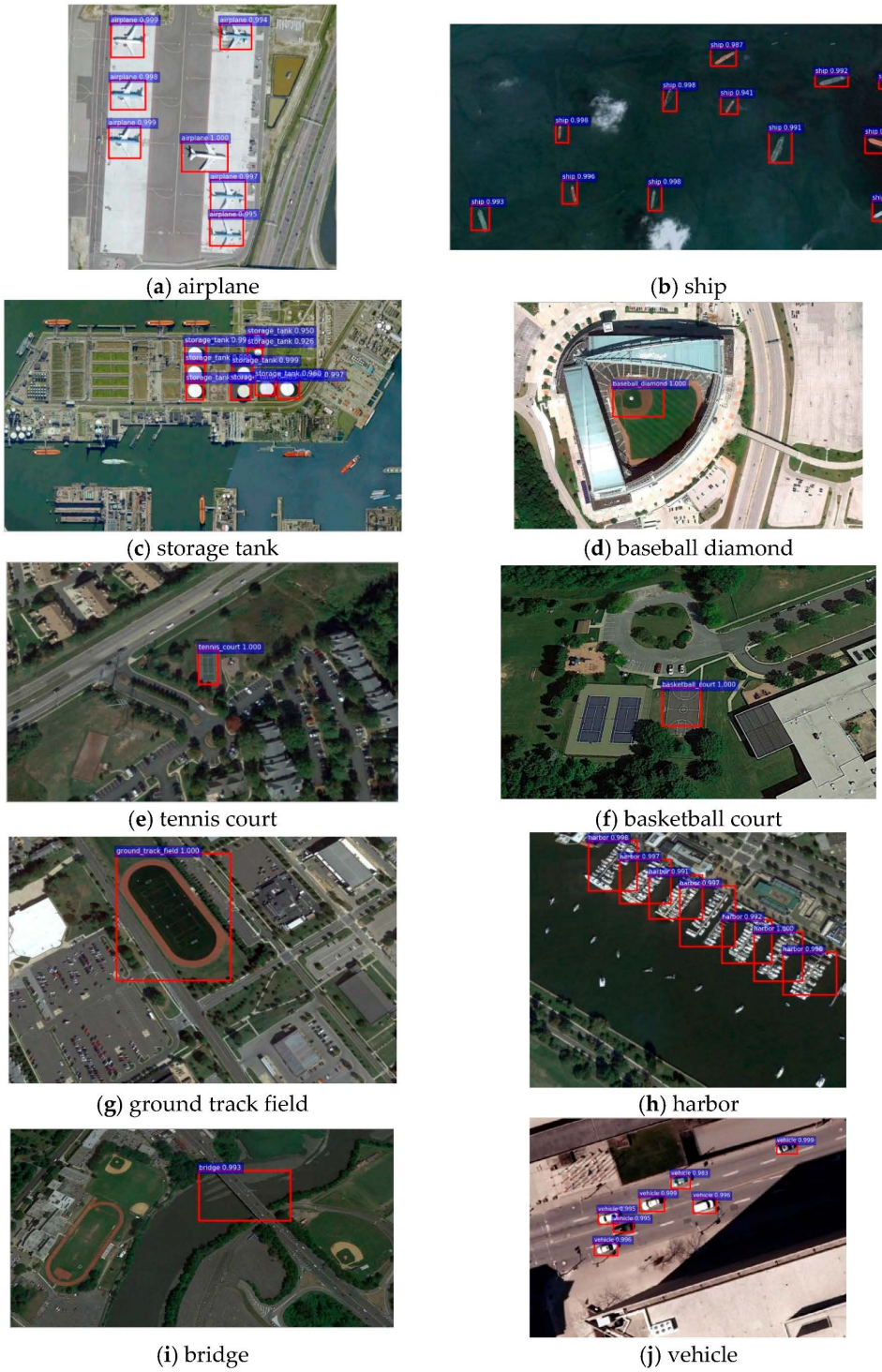
Object Detection

Object detection, as one of the most fundamental and challenging problems, aims to recognize objects within an image by determining their categories (classes) and locations via Bounding Box (BBOX). Generally, there are two primary categories of object detectors: region-based and regression-based (W. Li & Hsu, 2022). Within region-based methods, object detection is regarded as a classification problem, which is divided into three major phases: region proposal, feature extraction, and classification. Popular deep learning models within this group include Fast R-CNN (Ren et al., 2016), R-FCN (Dai et al., 2016), FPN (Lin et al., 2017), and RetinaNet (Lin et al., 2018). Regression-based models directly link image pixels with BBOX coordinates and class probabilities to save time in processing and transforming data between diverse layers. Due to its ability to respond quickly, many real-time applications prefer to use regression-based models, such as YOLO (Redmon et al., 2016), SSD (W. Liu et al., 2016), RefineDet (S. Zhang et al., 2018), M2Det (Zhao et al., 2019).

Geospatial Object Detection

Recently, the ever-growing availability of multi-modal Earth Observation (EO) data, including Very High Resolution (VHR) images, Multi-, and Hyper-spectral imagery, offers a promising data source for modern GeoAI methods to automatically detect and map geographic objects, ranging from artificial objects like buildings to natural objects such as trees, lakes. Figure 2.9 shows many examples of geospatial objects detection from high-resolution remote sensing images (X. Han et al., 2017). More successful examples of building detection include Global Urban Footprint (GUF) (Esch et al., 2013) from German Aerospace Center, High-Resolution Settlement Layer (HRSL) (Tiecke et al., 2017) from the Connectivity Lab at Meta, and the Google Open Building Layer (GOB) (Sirko et al., 2021). While these GeoAI methods provide an unparalleled capability to comprehensively monitor and map geospatial entities, the limitation of substantial training data at a large scale has emerged as a significant hindrance to the advancement of geospatial object detection (Ding et al., 2022; H. Li, Zech, et al., 2022). To address this issue, significant endeavors have been implemented towards creating benchmark datasets for multi-class geospatial object detection, such as NWPU VHR-10 (Cheng et al., 2016), DOTA (Ding et al., 2022), and FAIR1M (Sun et al., 2021).

To implement a geospatial object detection method, a large amount of training



(a) airplane (b) ship

(c) storage tank (d) baseball diamond

(e) tennis court (f) basketball court

(g) ground track field (h) harbor

(i) bridge (j) vehicle

Figure 2.9.: Geospatial detection examples for the NWPU VHR-10 dataset (X. Han et al., 2017).

Figure 2.10.: An example of building object detection.

data is typically needed, which should contain remote sensing images and corresponding location as well as meta information of the target objects. While remote sensing imagery at various temporal, spatial, and spectral resolutions is easy to obtain, manually annotating these images is super time-consuming. To speed up the data preparation process, researchers gave a lot of effort into integrating crowdsourced geospatial data from VGI platform, like OSM (J. Chen & Zipf, 2017; Herfort et al., 2019; Z. Wu et al., 2020). OSM provides a valuable geospatial data source with various object classes in millions of instances across the entire world. More importantly, with the efforts of volunteers and Humanitarian OpenStreetMap Team (HOT), OSM has provided many economically deprived regions with a wealth of geographic data that may be the only geographic data resource available to these regions. H. Li, Herfort, et al. (2022) successfully validated the method of using deep learning to accelerate building detection in Sub-Saharan Africa based on open-source remote sensing imagery and OSM data (Figure 2.10).

However, in social and environmental science, from variables of the existing process, we may observe spatial heterogeneity, which refers to the phenomenon that the expectation of a random variable varies across the Earth's surface (Anselin, 1989). Because of these phenomena, numerous GeoAI studies often face challenges when researchers attempt to replicate study findings in other regions, whether overlapping with the original area or not, without a notable performance drop (M. F. Goodchild & Li, 2021). Therefore, an increasing interest surrounds the potential of utilizing the expertise-embedded pre-trained GeoAI models across diverse geographic regions to achieve consistent performance in geospatial object detection without extensive additional training data.

2.2.3. Transfer Learning and Spatial Explicit AI

Transfer Learning

The objective of transfer learning is to enhance the performance of target learners of a specific domain by reusing the knowledge gained from disparate yet related domains (H. Li, Herfort, et al., 2022; Zhuang et al., 2021). There are several reasons why transfer learning is widely used in AI applications. The first is the lack of training data, for ideally traditional machine learning methods, the amount of labeled training data is often greater than or equal to the amount of test data, however, in many cases, collecting and annotating training data is very expensive, time-consuming, or unachievable. Second, the continuous development of foundation and pre-trained models in recent years has provided researchers with many base models that have already been trained on a large number of datasets, such as Microsoft COCO dataset (Lin et al., 2015), ImageNet dataset (Deng et al., 2009), and PASCAL VOC (Everingham et al., 2010). Meanwhile, the emergence of edge computing, and mobile AI technology is driving the demand for transfer learning, such low-computational platforms can not afford to train models with big data, but can realize the fine-tuning of the pre-trained models with a small number of data. In short, transfer learning partly addresses challenges of lack of training data, knowledge transfer of pre-trained models, and reduction of training computation.

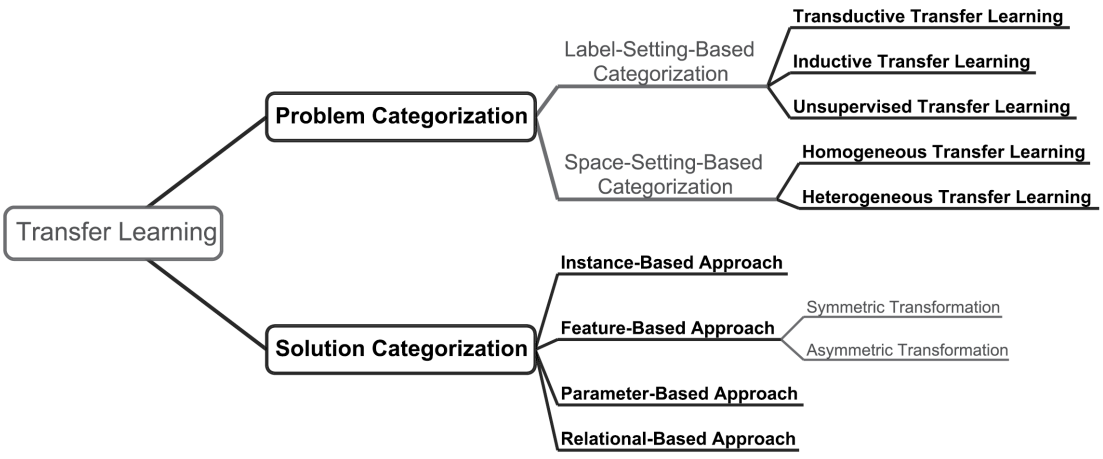


Figure 2.11.: Categories of Transfer Learning (Zhuang et al., 2021).

The categories of transfer learning are shown in Figure 2.11 (Zhuang et al., 2021). According to the categorization criteria from Pan and Yang (2010), transfer learning problems are classified into three classes: transductive, inductive, and unsupervised transfer learning. These classes are summarized as label-setting-

based aspects by Zhuang et al. (2021). In brief, transductive transfer learning refers to cases when the label information only comes from the source domain. If label information for target-domain instances is accessible, the scenario is classified as inductive transfer learning. In cases when the label information remains unknown for both the source and target domains, the scenario is recognized as unsupervised transfer learning. Another categorization classified transfer learning as homogeneous and heterogeneous transfer learning based on if feature spaces and label spaces are the same or not between the source and the target domains. If categorized in terms of solutions, transfer learning approaches can be categorized into four classes: instance-based, feature-based, parameter-based, and relational-based approaches (Pan & Yang, 2010; Zhuang et al., 2021). Instance-based approaches focus on the instance weighting strategy. Feature-based approaches transfer the original features into new feature representations, and they can be further classified into symmetric transformation and asymmetric transformation. The former approaches want to find a common potential feature space from the source and the target domain, and then transfer both into the new feature representations. The latter approach tries to transfer the source features to match the target features. Parameter-based approaches transfer the knowledge at the model/parameter level. Relational-based approaches transfer the logical relationships or rules acquired from the source domain to the target domain.

Transfer learning is widely used in image-related tasks, especially when there are few training samples in the target domain, researchers attempt to use a few-shot learning approach to transfer pre-trained deep learning models to the target domain (Y. Wang et al., 2020). In the geospatial object detection domain, early attempts successfully integrated meta-learning methods into few-shot object detection through re-weighting features within an object detection model (B. Kang et al., 2019; X. Li et al., 2022). X. Wang et al. (2020) introduced an efficient few-shot learning approach in object detection and confirmed its remarkable performance across the state-of-the-art benchmarks. Inspired by this, (H. Li, Herfort, et al., 2022) proposed a model-agnostic Few-Shot Transfer Learning (FSTL) method to improve the performance of the building detection model across different regions in Sub-Saharan Africa. However, the generalization of deep learning models to diverse regions across the world remains a challenge to be addressed.

Geographic Generalization of GeoAI Models

A frequently observed instance in GeoAI research is the poor performance of a deep learning model pre-trained in a specific region when applied to different geographic regions. This is also a reflection of the spatial autocorrelation for DL models, as stated in Tobler's first law of geography (D. Z. Sui, 2004). Figure 2.12 demonstrates Tobler's first law of geography that as spatial distance increases, the relationship between things decreases. This may limit GeoAI models to be reused with good performance only within a certain range of regions from the area being trained. Herein, the generalization capability of a GeoAI model to be reused or replicated across spatial space is called geographic generalizability (Mai, Huang, et al., 2023), or replicability across space (M. F. Goodchild & Li, 2021).

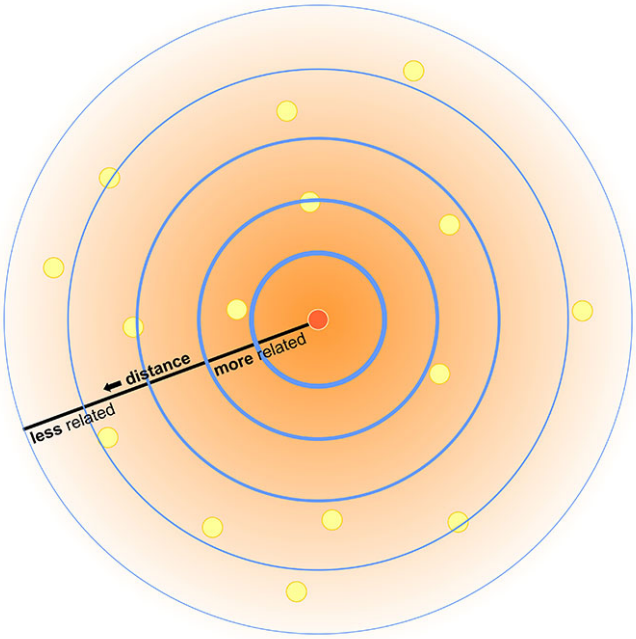


Figure 2.12.: A graphic illustration of Tobler's first law of geography (Anthony C. Robinson, n.d.).

In the AI domain, the reproducibility and replicability of DL models have always been a worthy concern for scientists (M. F. Goodchild & Li, 2021; Janowicz et al., 2020). Reproducibility refers to the ability of other researchers to get the same findings using the same data and methodologies, while replicability refers to the ability to duplicate the research findings proven by prior studies using the same methodologies but new data. As the data gradually becomes an integral

part (or called knowledge) of the model, it is hard to derive the same model without access to the original dataset used by the model's author. The availability of many public, domain-advanced large datasets, such as Microsoft COCO dataset (Lin et al., 2015), ImageNet dataset (Deng et al., 2009), and PASCAL VOC (Everingham et al., 2010), can partially help researchers achieve model reproducibility and replicability, however, in the actual training process, different data structures, training methods, or hyper-parameters can lead to trained models do not exactly match the expectation. Therefore, large companies and organizations have introduced many pre-trained DL models, and the parameters of these models are usually tuned by the large public databases mentioned above, which makes these models perform stably. These pre-trained task-agnostic models lead to great success in downstream tasks via fine-tuning, few-shot, or zero-shot learning.

While adapting pre-trained AI models to solve downstream problems, it is still challenging to balance the model's generalization and specialization. From the perspective of ML, the discussion about the generalization and specialization capability of AI models has a long history (Bousquet & Elisseeff, 2002; C. Zhang et al., 2021). The generalization means machine learning models generalize the knowledge learned from the training dataset to new unseen data. Conversely, specialization means models can effectively learn and solve a specific task, it probably is over-fitting or cannot generalize. Typically, specialization should be avoided, and ML models should be towards better generalization, which also means increasing the generalization capability or called generalizability. Many efforts have been made to achieve better generalized models, such as improving the gradient descent optimization procedure (Hardt et al., 2016), using large-scale datasets, developing more powerful model architectures like the Transformer family (Dosovitskiy et al., 2021; Vaswani et al., 2017), or meta-learning techniques (Rußwurm et al., 2020).

Although most AI applications tend to favor higher generalizability models, spatial phenomena (e.g., autocorrelation and spatial heterogeneity) present additional scenarios for AI, which can also be called spatially explicit AI (M. Goodchild, 2001; Janowicz et al., 2020). To deal with such scenarios, one needs to balance the geographic generalizability (or called replicability across space) and geographic specialization of GeoAI models, especially when targeting large-scale applications across geographic space (Mai, Huang, et al., 2023). For instance, many current GeoAI models have specialization in specific training regions, and these models perform poorly when applied to regions beyond the range of geographic autocorrelation. However, improving the performance of these GeoAI models across diverse regions is still a very challenging task. Such a

problem in the GeoAI domain can be defined as finding the "sweet spot" between geographic generalizability and spatial heterogeneity (Mai, Huang, et al., 2023).

Many existing studies have attempted to tackle such geographic generalizability challenges, which can be categorized into three main branches. The first common practice is to divide the interesting space into diverse regions based on the underlying data process and train individual models for each region respectively (Xie et al., 2021; Y. Zhang et al., 2017). The obvious drawback of these approaches is the large amount of parameters and training datasets needed for different regions. The second common solution is to apply transfer learning across space, such as urban-to-rural transfer (Elmustafa et al., 2022), city-to-city transfer (L. Wang et al., 2018), and country-to-country transfer (H. Li, Herfort, et al., 2022). In the third category, early attempts have sought to incorporate representation learning methods into a range of GeoAI applications, such as place recognition (Yin et al., 2019), trajectory prediction (Xu et al., 2018), point cloud segmentation (Qi et al., 2017), and geo-aware image classification (Mac Aodha et al., 2019; Mai et al., 2020; Mai, Lao, et al., 2023), in which spatial locations are encoded into a high-dimensional embedding space to capture the spatial heterogeneity features across space to facilitate downstream tasks. For a review of location encoding in GeoAI, see (Mai, Janowicz, et al., 2022). Recently, the concept of position/location embedding has achieved excellent performance in general AI tasks with the popularity of the Transformer and Vision Transformer models (Dosovitskiy et al., 2021; Vaswani et al., 2017), where a self-attention mechanism is employed to capture the relationships between different elements (like words, image patches, or audio segments) within the same sequence (like sentence, image, or speech). Inspired by these exciting studies, this thesis tends to explore the possibility of combining transfer learning and representation learning to improve the cross-spatial generalizability of GeoAI models.

2.3. Web Mapping

To interactively visualize AI-generated geospatial data, a browser is absolutely one of the most common and useful platforms. Before designing a user-centered GeoAI application, some knowledge of web mapping is needed. This chapter introduces the evolution of web mapping, the standards of web mapping services, and the GeoAI based mapping services.

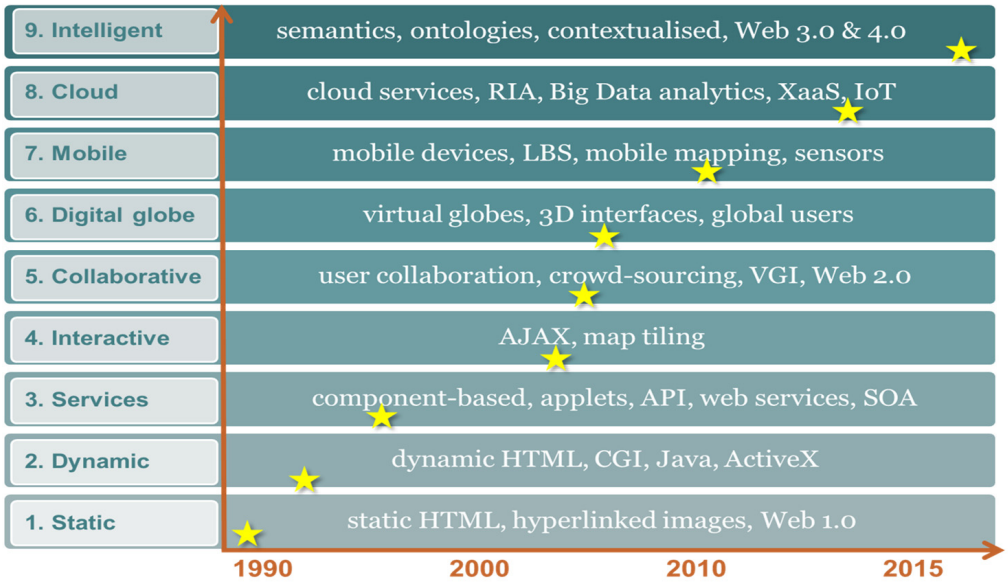


Figure 2.13.: Framework of web mapping eras (Veenendaal et al., 2017).

2.3.1. Eras of Web Mapping

With the progressive evolution of big geospatial data and smart mobile devices, web mapping has led to revolutionary advancements in recent decades. Whether humans realize it or not, our daily lives have become inseparable from web mapping and location-based applications. Defined by Neumann (2008), Web mapping is the process of designing, implementing, generating and delivering maps on the World Wide Web. Three foundational elements of web mapping are geospatial data and their map-based visualization, geospatial software, and the World Wide Web (Veenendaal et al., 2017). Over the past decades, the constant development of forms and technologies of these elements has resulted in continuously updating eras of web mapping.

Nine web mapping eras were identified and arranged through a timeline to mark the milestones of web mapping developments, as shown in figure 2.13. From static web mapping to current intelligent web mapping, there are several important concepts or technologies that appeared in each era, where a star mark indicating the approximate emergence of the developments. No definitive "end" has yet been defined, as many of these advancements have either continued, been embedded, or expanded in subsequent eras. The key developments within each era can be found in the paper of Veenendaal et al. (2017). Notably, the computer revolution of the 21st century has driven the advancements of various web technologies and concepts, as well as the rapid

development of web mapping at the same time. In particular, the growth of various cloud services and AI technologies in recent years has inspired more imagination and application scenarios for increasingly available big geospatial data. However, for the development of complicated web mapping applications and the processing and management of big geospatial data, a set of standards is extremely necessary.

2.3.2. OGC Standards

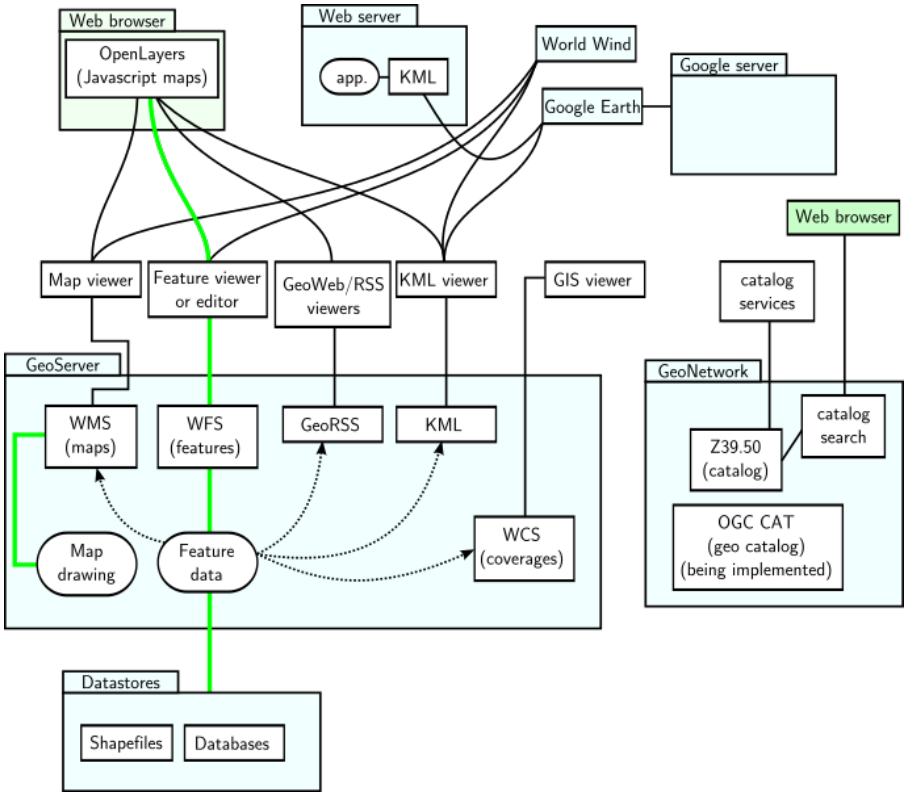


Figure 2.14.: Relationship between clients/servers and OGC protocols ("Open Geospatial Consortium", 2023).

Open Geospatial Consortium (OGC) is an international voluntary consensus standards organization founded in 1994, working on developing standards for geospatial content, location-based services, sensor web, Internet of Things, GIS data processing and data sharing ("Open Geospatial Consortium", 2023). The OGC standard provides specification standards and application paradigms for web mapping development, greatly improving the robustness, ease of maintenance, and reusability of web map applications. Figure 2.14 demonstrates a

common OGC-based web mapping development framework, integrating web browser, web server, GeoServer, and other map services providers. This protocol uses some popular geospatial standards from OGC, like Web Map Service (WMS), Web Feature Service (WFC), Web Coverage Service (WCS), Keyhole Markup Language (KML), etc.

The WMS standard allows clients to obtain map images, as well as some map-related metadata, via the Hypertext Transfer Protocol (HTTP) protocol. The advantages of WMS services are their simplicity and flexibility, and the ability to provide customized map styles, annotations, and so on. The WFS standard allows clients to access detailed information about geographic data, including elemental attributes and geometry, via the HTTP protocol. The advantage of the WFS service is that it provides more detailed geographic data, allowing clients to perform more complex data analysis and processing. The WCS standard allows clients to obtain detailed information about the coverage, including spatial data on topography, vegetation, and other geographic information, via the HTTP protocol. The advantage of the WCS service is that it provides high-resolution geographic data, which allows the client to analyze and process the data in a more precise manner.

In addition to the above three commonly used standards, OGC has also formulated many other standardized interfaces for web mapping development, which enables better data sharing and interaction between different GIS systems. In the actual web mapping development, open-source GIS software (e.g., GeoServer, MapServer, etc.) can be employed to realize the support of OGC standards, so as to provide services such as WMS, WFS, WCS, etc. Meanwhile, users can also develop their own OGC-based interfaces to meet specific business needs.

With the growing trend of separating front-end development and back-end development, developers preferred to use JSON/GeoJSON ("GeoJSON", 2023; "JSON", 2023) for data exchange in web mapping development, thus, the OGC organization put a lot of effort into developing a new set of Application Programming Interface (API) standard, called OGC API. The most notable features of OGC API can be summarized as follows: the interface style is REST ("Representational State Transfer", 2023), and the data exchange is in JSON format by default.

Therefore, based on the OGC standards and modern web mapping technology, researchers can build a variety of web mapping applications according to specific needs. The continuous advancement of GeoAI also promotes the thinking of how to use web technology and AI algorithms to improve GeoAI research and address geographic challenges.

2.3.3. GeoAI and Machine Learning as a Service

With the increasing availability of satellite imagery and advancement of deep learning techniques, a series of GeoAI applications were discovered, such as image-level classification, object detection, semantic segmentation, etc (W. Li & Hsu, 2022). Demand for GeoAI applications is growing not only in the research domain but also in practical production.

A major problem with applying GeoAI today is the reusability of methods for geospatial data preparation and processing. For instance, while applying deep learning methodologies to remote sensing data, satellite imagery often contains additional spectral bands beyond RGB and potentially integrates with other geospatial data which may have diverse coordinate systems. The complexity of appropriately handling various geospatial data should not be underestimated. Large companies have realized this general problem in the AI field, and offer easy-to-use APIs (Ribeiro et al., 2015), such as Hugging Face, JINA AI, etc. Recent research in formulating the application of GeoAI into high-level programming libraries has risen to lower the barrier of geography-related knowledge and programming efforts across various deep learning frameworks. TorchGeo is an excellent instance of integrating geospatial data into the PyTorch ecosystem (Stewart et al., 2022). However, these APIs have four significant limitations: (a) Tailoring them for specific applications isn't straightforward. (b) They are usually cloud-based. (c) They are complex to understand and use due to the lack of tutorial workflows or incomplete documentation. (d) They lack support for reusability and configuration sharing across different applications (Pahl & Loipfinger, 2018).

One method for overcoming these challenges is a service-oriented approach that enables individual models to operate and interact with others using web services (J. Wang, n.d.). Inspired by Machine Learning as a Service (MLaaS) and the reproducible capability of Docker (Boettiger, 2015), GeoAI methods can also be developed as an individually developable, locally runnable, reusable, and simply deployable service. Once a complete GeoAI service was built targeting a specific task, it can be effortlessly deployed on local workstations, cloud-based infrastructures, or edge computing platforms.

3. Methodology

This chapter presents the whole workflow and detailed steps of the proposed GWME method and related algorithms and formulas used. Section 3.1 describes the definition of the problem and the concepts used in the solution. Section 3.2 introduces the preparation of the training datasets. Section 3.3 shows how to implement Few-Shot Transfer Learning (FSTL) for a geospatial object detection model. Section 3.4 brings the core steps of GWME methods, proposed by this thesis. Section 3.5 provides the metrics and the algorithms for evaluating the performance of geospatial object detection models. The last section 2.3 demonstrates the infrastructure of a web mapping application for the visualization of results.

3.1. Definitions and Preliminaries

This thesis rethinks the geographic generalizability problem and proposes to solve it by an unsupervised self-attention model ensemble method, namely the Geographical Weighted Model Ensemble (GWME). An interesting case study of detecting OSM missing buildings across different counties in sub-Saharan Africa is conducted to demonstrate the effectiveness of GWME. The overall method is illustrated in Figure 3.1. This main target is to transfer a building detection model trained at the source region (A_{bm}) to the target region (A_{target}) and to improve the model's performance at the target region with the geographic information from nearby regions of the target region.

The general assumption is that although there's completely no OSM building data at the target region, it is still possible to find some nearby regions with a small number of OSM data that can be used as training samples, which is useful to help extend the generalization of the base model ($\mathbb{M}_{bm}$). Firstly, a base model $\mathbb{M}_{bm}$ was trained by the OSM-labeled training dataset $\mathbb{S}_{bm}$ at an OSM data-rich area A_{bm} (in Tanzania), which is called the source region. Then a few reference areas ($A_j, j = 1, \ldots, T$) are selected in the proximity of the target region A_{target} (in Cameroon), which is geographically far away from area A_{bm} and consists of

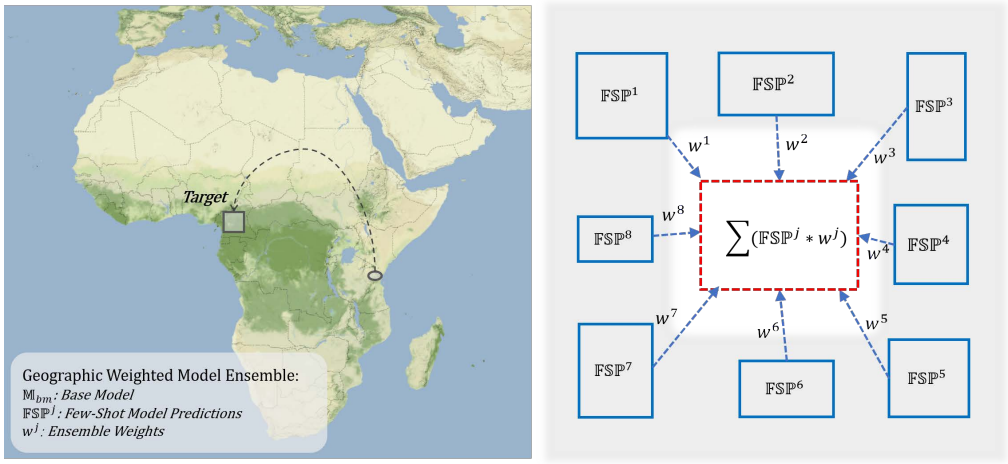


Figure 3.1.: Overview of Geographic Weighted Model Ensemble (GWME).

very diverse landscapes and building structures.

Next, a Few-Shot Transfer Learning (FSTL) technique (H. Li, Herfort, et al., 2022) is applied to extrapolate $\mathbb{M}_{bm}$ to those reference areas neighboring the target area. After the FSTL, a few less accurate models ($\mathbb{M}_{fs}$) are generated close to the target area. These $\mathbb{M}_{fs}$ were used to detect buildings at the target region respectively and each $\mathbb{M}_{fs}$ can detect a lot of building candidates (called $\mathbb{FSP}^j$, where $j = 1, 2 \ldots, T$). Then the weights between the target region and reference regions are calculated to determine how important these reference regions are to the target region. In the end, all of the building candidates $\mathbb{FSP}$ are ensembled according to different weights to generate a more accurate and comprehensive prediction $\mathbb{P}$ at the target region A_{target}. Therefore, the determination of weights is crucial for the model ensemble, which represents not only the importance of the reference regions $A_j, j = 1, \ldots, T$ for the target region A_{target} but also the generalization capability of $\mathbb{M}_{fs}$ for the detection $\mathbb{P}$ at the target region.

Therefore, the objective of measuring the model's geographic generalizability is achieved by minimizing the discrepancy between the GWME predictions ($\mathbb{P}$) and the ground truth label ($\mathbb{Y}_{target}$) of the test dataset ($\mathbb{S}_{target}$). Different weighting strategies are conducted: 1) average weighting (*average*), 2) image similarity weighting (*similarity*), 3) geographic distance weighting (*distance*), and 4) self-attention weighting (*attention*). A more detailed description of the methodology and how it is implemented is given in the following sections and also presented in (H. Li, Wang, et al., 2023).

Additionally, some important symbols and definitions are listed in Table 3.1.

Table 3.1.: Glossary of methodology.

Symbol	Definition
A_{bm}	The source area.
$\mathbb{S}_{bm}$	The training samples from the source area.
$\mathbb{M}_{bm}$	The base model for the GeoAI task.
T	The number of reference areas.
A^j	The reference area j.
$\mathbb{S}^j_{fs}$	The training samples from the reference area j.
$\mathbb{M}^j_{fs}$	The FSTL model trained at reference area j.
A_{target}	The target area
$\mathbb{S}_{target}$	The test dataset from the target area.
$\mathbb{Y}_{target}$	The ground truth label of the target area.
$\mathbf{M}_{ViT}$	The pre-trained ViT model.
$\mathbf{w}^j$	Corresponding weight for $\mathbb{M}^j_{fs}$.
$\mathbf{x}_i$	Image tile i in the satellite image set.
$\mathbf{y}_i$	List of object BBOX within the image tile i.
$\mathbf{g}_i$	Geographic information of the image tile i.
$\mathbb{FSP}$	List of BBOX and scores predicted from $\mathbb{M}_{fs}$.
$\mathbb{P}$	List of ensembled BBOX and scores.
$\mathbf{TH}$	Hyperparameters (the threshold of confidential scores).
$\mathcal{Q}$	Weighted boxes fusion function.
HIS	The function that calculates the histogram of an image.
COS	The function that calculates cosine similarity between two histograms.
CEN	The function that calculates the geometric center of a region.
DIS	The function that calculates the great circle distance.

3.2. Data Preparation

This thesis assumes an OSM-labeled training dataset for the base object detection model (BM) to be a triplet $\mathbb{S}_{bm} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{g}_i)\}$ with $i = 1, \ldots, N$ in an OSM data-rich area A_{bm}. Here, $\mathbf{x}_i$ is a satellite image, $\mathbf{y}_i$ is a set of object bounding boxes (BBOX) within this image, and $\mathbf{g}_i$ refers to the location information (e.g., longitude and latitude) and optionally geographic distances to the hold-out test dataset $\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}$ with $i = 1, \ldots, M$.

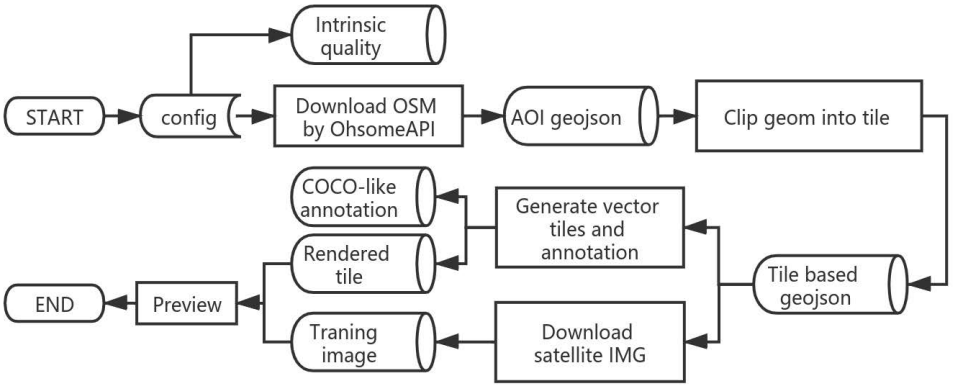
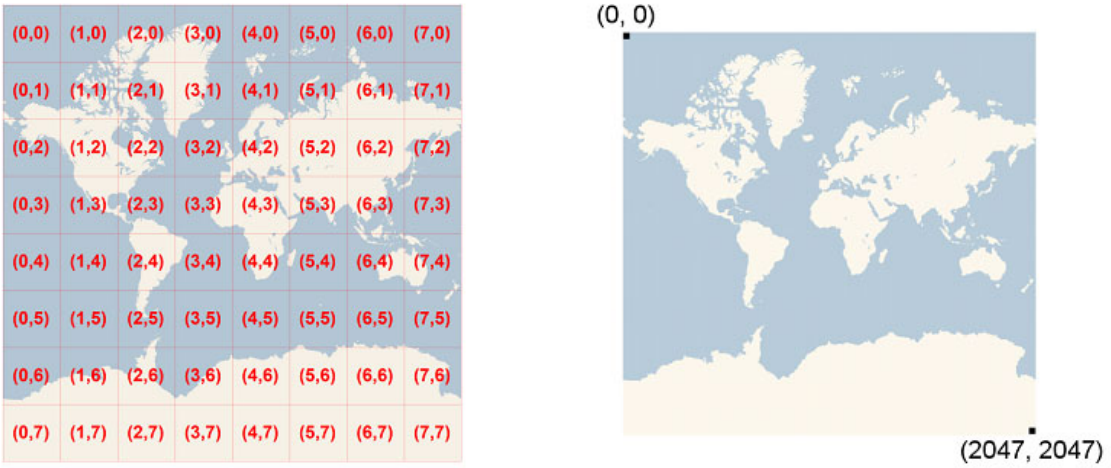


Figure 3.2.: The workflow of ohsome2label (H. Li & Zipf, 2022).

The data preparation steps in this thesis were automatically implemented by *ohsome2label* (Z. Wu et al., 2020), an open-source package for wrapping queried satellite imagery from open WMS and queried OSM features for the corresponding area into training samples by a tile-based manner. The detailed workflow of *ohsome2label* is shown in Figure 3.2. It starts with a configuration file, where the query metadata are stored, and downloads OSM features by *OhsomeAPI* according to the metadata. After geometric data within the area of interest (AOI) are downloaded in GeoJSON format, which is an open standard format designed for representing simple geographical features, the whole GeoJSON feature collection will be clipped into a list of tiles. Then the tile-based GeoJSON will be forwarded to generate vector tiles, which can be used for making COCO-like annotations (Lin et al., 2015), and also to download satellite images. Eventually, tile-based satellite images with the size of 256*256 pixels and the corresponding rendered label tiles can be compiled for the preview of training samples.

This thesis used Bing Maps Aerial Imagery as the satellite image source, which provides open-source and high-resolution satellite image WMS via a REST API (Bing™ Maps Imagery API). One of the most important steps in



(a) Tile coordinates. (b) Pixel coordinates.

Figure 3.3.: Bing maps tile system at level 3 (rbrundritt, 2022).

data preparation is the coordinates transformation. Whether converting the geographic coordinates of a satellite image to image coordinates when creating training samples or converting the image coordinates of predictions obtained by the trained model to geographic coordinates, the conversion between the geographic coordinate system and the image coordinate system is a very essential step. Figure 3.3 illustrates a level 3 example of the Bing maps tile system, which includes tile coordinates and pixel coordinates in a Web Mercator projection ("Mercator Projection", 2023). Given latitude and longitude in degrees (on the WGS 84 datum), and the level of detail, the pixel XY coordinates can be calculated as follows:

$$
\begin{aligned}
&sinLatitude = \sin(latitude * \pi / 180) \\
&pixelX = ((longitude + 180)/360) * 256 * level \\
&pixelY = (0.5 - \log((1 + sinLatitude)/(1 - sinLatitude))/(4 * \pi)) * 256 * level
\end{aligned}
\tag{3.1}
$$

To optimize the index of the map, Bing maps cut the map into tiles of 256*256 pixels each. The number of tiles is determined by the level of detail:

$$
mapWidth = mapHeight = 2^{level}\ tiles \tag{3.2}
$$

Given a pair of pixel XY coordinates, the tile XY coordinates of the tile containing that pixel can be easily determined by:

$$\begin{aligned} tileX &= floor(pixelX/256) \\ tileY &= floor(pixelY/256) \end{aligned} \tag{3.3}$$

With the Bing maps tile system, the desired satellite images can be stored as plenty of image tiles with the size of 256*256 pixels, attached with the tile coordinates. Furthermore, the latitude and longitude coordinates of the object BBOX can be converted to pixel coordinates within the range between (0, 0) to (256, 256). Meanwhile, the pixel coordinates of predicted BBOX can be converted back to geographic coordinates.

Figure 3.4.: Example Bing aerial image tiles (top), the corresponding OSM labels (middle), and the preview of training samples (bottom).

Figure 3.4 demonstrates three *ohsome2label* generated training samples composed of Bing aerial images and OSM building features. As long as in areas where OSM data is relatively complete, *ohsome2label* can generate well-prepared COCO-like training samples, which feed for the training process of object detection models. Besides the base training dataset $\mathbb{S}_{bm} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{g}_i)\}$ $(i = 1, \ldots, N)$ at the source area A_{bm}, a list of reference training datasets $\mathbb{S}^j_{fs} = \{(\mathbf{x}^j_i, \mathbf{y}^j_i, \mathbf{g}^j_i)\}$ $(i = 1, \ldots, n, j = 1, \ldots, T)$ were prepared from T reference areas A_j $(j = 1, \ldots, T)$ in the proximity of the target area A_{target}. These training datasets will be fed into the further FSTL process.

3.3. Few-Shot Transfer Learning

Base Model Training

The base deep learning model was trained using Tensorflow Object Detection API (J. Huang et al., 2017) which is an open-source framework built on top of TensorFlow (Abadi et al., 2016) that makes it easy to configure, train and deploy object detection models. Especially a one-stage SSD-based (W. Liu et al., 2016) object detection model which was pre-trained by Microsoft COCO datasets (Lin et al., 2015) was picked due to its relatively high speed and COCO mAP. More pre-trained models can be found from TensorFlow 2 Detection Model Zoo (J. Huang et al., n.d.).

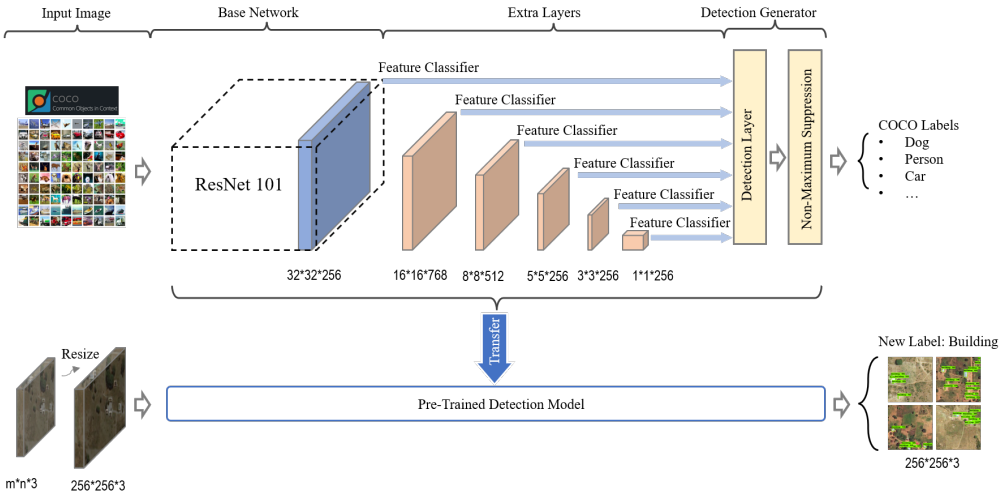


Figure 3.5.: The architecture of SSD-ResNet101 for building detection.

This thesis considered a pre-trained SSD-based model as an object detection base model (BM) (W. Liu et al., 2016). The feature extractor VGG used in the origin SSD model was replaced with a ResNet layer, which effectively solved the problem of precision reduction of deep network (K. He et al., 2015). Then the base model was further trained by optimizing the loss function $\mathcal{L}$ via gradient descent in a supervised manner with the training dataset $\mathbb{S}_{bm} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{g}_i)\}$ $(i = 1, \ldots, N)$. The overall objective loss function $\mathcal{L}$ is a weighted sum of the localization loss (loc) and the confidence loss (conf) (W. Liu et al., 2016):

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{conf} \tag{3.4}$$

where $\mathcal{L}_{loc}$ uses smooth L1 loss, and $\mathcal{L}_{conf}$ uses sigmoid focal loss.

Figure 3.5 illustrates the base training process, which can also be described as:

$$\mathcal{F}(\mathbb{S}_{bm}) \rightarrow \mathbb{M}_{bm} \tag{3.5}$$

where $\mathbb{M}_{bm}$ represents the BM for a specific GeoAI task (e.g. an OSM building detection model). In a word, an adequate geographic training dataset from the source region A_{bm} can extend the capabilities of a pre-trained model from detecting objects within the COCO label dataset to detecting building objects on high-resolution satellite images.

Multiple Few-Shot Transfer Learning

To improve the geographic generalizability of the OSM building detection base model $\mathbb{M}_{bm}$ from source region A_{bm} to a geographically far away target region A_{target}, the FSTL method is an efficient way in case there are some training samples available at the target region (H. Li, Herfort, et al., 2022). However, if the target region is completely missing training samples, this thesis assumes that it is still possible to find some neighboring areas with a small amount of "training shots". These neighboring areas, or called reference areas, have a higher spatial correlation with the target region depending on the distance. To ensure that the full surrounding spatial relevance of the target region can be taken into account, this thesis took the direction of searching for eligible reference regions towards the 8-neighborhood space of the target region, as the Figure 3.1 shows. The training samples from each reference area are expected to partly help extend the geographic generalizability of the base model.

Leveraging this idea, a given OSM building detection base model $\mathbb{M}_{bm}$, fed with several reference few-shot training samples $\{\mathbb{S}^j_{fs}\}^T_{j=1}$ generated from T reference areas $\{A_j\}^T_{j=1}$, can be replicated as multiple FSTL models $\{\mathbb{M}^j_{fs}\}^T_{j=1}$. This fine-tuning process can be formulated as:

$$\mathcal{F}(\{\mathbb{S}^j_{fs}\}^T_{j=1}) \rightarrow \{\mathbb{M}^j_{fs}\}^T_{j=1} \tag{3.6}$$

One step further, conducting these FSTL models forward inference, represented as $\mathcal{P}$, with the test dataset $\mathbb{S}_{target}$ at the target region A_{target} can generate plenty of object predictions derived from different FSTL models as follows:

$$\mathbb{FSP}^j = \bigcup_{(\mathbf{x}_i, \mathbf{g}_i) \in \mathbb{S}_{target}} \mathcal{P}(\mathbb{M}^j_{fs}, (\mathbf{x}_i, \mathbf{g}_i)). \tag{3.7}$$

Where $\{\mathbb{FSP}^j\}_{j=1}^T$ refers to the corresponding set of predictions, which includes predicted BBOX and corresponding confidence scores, obtained from T models in the target area $\mathbb{S}_{target}$, and $\{\mathbb{M}_{fs}^j\}_{j=1}^T$ are a set of FSTL models.

In the case of OSM building detection, after the SSD-based base model being trained with the base training dataset $\mathbb{S}_{bm}$ at the source region A_{bm}, multiple Few-Shot Transfer Learning were implemented to extrapolate $\mathbb{M}_{bm}$ to many less accurate FSTL models $\{\mathbb{M}_{fs}^j\}_{j=1}^T$, which were then employed to predict the missing OSM buildings with the test dataset $\mathbb{S}_{target}$ at the target region A_{target}. The pseudo-code of multiple FSTL and predictions is shown in Algorithm 1.

Algorithm 1 Multiple Few-Shot Transfer Learning and Predictions

Input:
$\mathbb{M}_{bm}$: the base model;
T: number of reference areas neighbouring the test area;
$\mathbb{S}_{fs}^j = \{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j)\}$ $(i = 1, \ldots, n, j = 1, \ldots, T)$: FSTL samples from reference areas;
$\mathbb{M}_{fs}^j \leftarrow \{\}$: few-shot models fine-tuned on reference areas;
$\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}$ $(i = 1, \ldots, M)$: dataset from the test area;
$\mathbb{FSP}^j, j = 1, \ldots, T \leftarrow []$: predictions from single FSTL models;
for dataset $\mathbb{S}_{fs}^j$ of each reference area A_j in $\{\mathbb{S}_{fs}^j\}_{j=1}^T$ **do**
    few-shot model $\mathbb{M}_{fs}^j \leftarrow \mathcal{F}(\mathbb{S}_{fs}^j, \theta)$;
    for each $(\mathbf{x}_i, \mathbf{g}_i)$ in $\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}_{i=1}^M$ **do**
        update $\mathbb{FSP}_i^j \leftarrow \mathcal{P}(\mathbb{M}_{fs}^j, (\mathbf{x}_i, \mathbf{g}_i))$;
    end for
end for
Output:
$\{\mathbb{FSP}^j\}_{j=1}^T$: list of objects and scores predicted from reference few-shot models;

Obviously, the performance of FSTL models is still limited by factors, such as the amount and quality of "training shots", the spatial correlation between regions, and the distance to the target region, which are all essential to the geographic generalizability of models. In the next section, a further step will be taken towards combining all of these FSTL predictions by establishing an effective weighting strategy to ensemble diverse FSTL models.

3.4. Geographical Weighted Model Ensemble

Since each FSTL model was obtained by fine-tuning the base model $\mathbb{M}_{bm}$ by the reference datasets $\{\mathbb{S}_{fs}^{j}\}_{j=1}^{T}$ from T reference regions surrounding the target region, the prediction ability of each FSTL model at the target region will be more or less influenced by these training samples. For instance, this thesis assumes that if the correlation between these reference training samples and the test data in the target region is higher, then the prediction ability or the geographic generalizability of the FSTL model in the target region will also be stronger (M. F. Goodchild & Li, 2021; H. Li, Herfort, et al., 2022; Mai, Huang, et al., 2023). This correlation could be the correlation of the data itself, such as the similarity of the satellite images, or it could also be the geospatial autocorrelation, such as geographic distance. Therefore, the Geographical Weighted Model Ensemble (GWME) method was proposed to ensemble the geographic generalizability of all FSTL models by jointly considering the vision representation and geospatial correlation between reference regions and the target region (H. Li, Wang, et al., 2023).

The specific steps of GWME method can be divided into three major parts: 1) Multiple Few-Shot Transfer Learning and predictions; 2) Calculate the contribution weights of each FSTL model to the target region; 3) Ensemble predictions from multiple FSTL models into the final prediction according to their weights. The process can be described as:

$$\mathbb{P} = \sum_{j=0}^{T} \mathcal{Q}(\mathbb{FSP}^{j}, \mathbf{w}^{j}, \mathbf{TH}). \tag{3.8}$$

where $\mathbb{P}$ represents the final prediction with test dataset $\mathbb{S}_{target}$ at target region A_{target}, $\mathcal{Q}$ specifically means a Weighted Boxes Fusion (WBF) (Solovyev et al., 2021) function for object detection fusion and **TH** represents the corresponding hyperparameters, such as the threshold of confidential scores. The general idea of the WBF is visualized in Figure 3.7 and the details will be introduced later.

FSTL Model Weighting

Now after multiple FSTL predictions were generated, the most important following step is to decide the weight of each individual FSTL model $\mathbb{M}_{fs}^{j}$. This thesis aims to consider both the vision information ($\mathbf{x}_i^j$) and geographic information ($\mathbf{g}_i^j$) between reference datasets $\{\mathbb{S}_{fs}^{j} = \{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j)\}_{i=1}^{n}\}_{j=1}^{T}$ and test dataset

$\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}_{i=1}^{M}$ with an explicit weighting algorithm $\mathcal{W} \in \mathbb{R}^{T \times M}$ in a tile-base manner. The weighting algorithm can determine how and how much the reference training samples can influence the final prediction at the target region. More specifically, this thesis proposes an unsupervised method to learn model ensemble weights by taking both image feature embedding and location embedding into account with a self-attention mechanism, which can be called self-attention weighting (shown in Figure 3.6), and additionally conducts three other weighting strategies (average weighting, image similarity weighting, and geographic distance weighting) for comparison. Four different weighting strategies were elaborated as follows.

Average Weighting (*average*) - In the simplest case, equal weights for all of FSTL models were considered. This approach aims for cases where the relationship between reference regions and the target region can not be obtained or evaluated. The average weighting can be the easiest way to make the ensemble model perform a bit better in prediction capability than a single FSTL model.

$$\{\mathbf{w}^j = 1\} \rightarrow \mathbb{FSIP}^j \tag{3.9}$$

Image Similarity Weighting (*similarity*) - For a geospatial object detection task, it is intuitive to think about considering the similarity of satellite images among the reference areas and the target area. Therefore, an average cosine similarity ("Cosine Similarity", 2023) was considered to determine the relationships between the histograms of satellite image pairs, namely $\{\mathbf{x}_i^j\}_{i=1}^{n} \in \mathbb{S}_{fs}^j$ and $\{\mathbf{x}_i\}_{i=1}^{M} \in \mathbb{S}_{target}$, as a proxy of their image similarity weights (see Equation 3.10). Herein, $HIS(\cdot)$ indicates a function to compute the image histograms for RGB channels. $COS(\cdot)$ indicates the average cosine similarity function for computing the correlation between two stacked vector data. n is the number of few-shot data samples, which may vary across different reference areas, used for training the FSTL model $\mathbb{M}_{fs}^j$ from $\mathbb{S}_{fs}^j$ in the reference area A_j.

$$\{\mathbf{w}_i^j = \frac{1}{n} \sum_{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j) \in \mathbb{S}_{fs}^j} COS(HIS(\mathbf{x}_i^j), HIS(\mathbf{x}_i)\} \rightarrow \mathbb{FSIP}^j \tag{3.10}$$

Geographic Distance Weighting (*distance*) - Given Tobler's First Law of Geography (D. Z. Sui, 2004), this approach expects a high spatial correlation between two objects to be observed if they are close to each other. In this case, the spatial correlation between the FSTL model trained in a reference region that is further away and the target region is considered to be lower, which means that

it should occupy a lower weight when ensemble models. To this end, an inverse distance weighting strategy was considered for the model ensemble, which is based on the prior knowledge of the geographic locations of the target dataset and reference datasets, specifically $\{\mathbf{g}_i\}_{i=1}^{M} \in \mathbb{S}_{target}$ and the geographic center of each reference region A_j. Equation 3.11 illustrates the general idea where $CEN(\cdot)$ indicates the geometric center of the study area and $DIS(\cdot)$ indicates a great circle distance, which represents the shortest distance between two points over the earth's surface ("Great-Circle Distance", 2023).

$$\{\mathbf{w}_i^j = DIS(\mathbf{g}_i, CEN(\mathbb{S}_{fs}^j))\} \rightarrow \mathbb{FSIP}^j \tag{3.11}$$

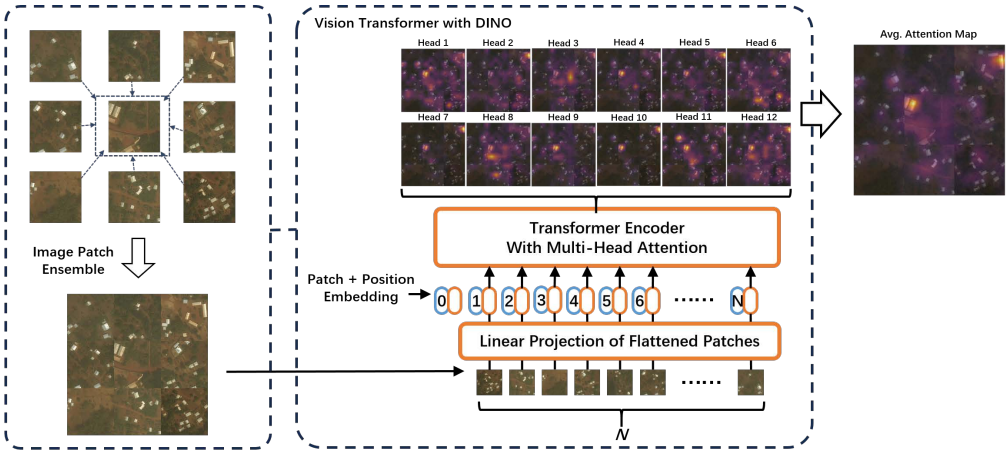


Figure 3.6.: The extraction of self-attention-based weights for the GWME using a pre-trained ViT with DINO.

Self-Attention Weighting (*attention*) - As the most interesting part, an unsupervised method was developed to learn self-attention weights from a pre-trained ViT model – the Self-Supervised ViT with DINO (Caron et al., 2021), which has been pre-trained on ImageNet (Deng et al., 2009). Leveraging the ability of ViT like models to capture both context embedding and position embedding (Dosovitskiy et al., 2021), this self-attention weighting approach aims to calculate weights by considering both image similarity and relative spatial relation among reference areas and target area, which are essential factors to FSTL models' geographic generalizability. To adopt DINO into GWME, an image patches ensemble approach was designed according to relative positions of reference regions and target region, where the central image patch was taken from the target area and the context image patches were taken from T reference areas $\{A_j\}_{j=1}^{T}$ as shown in Figure 3.1. In other words, unlike the original ViT which

splits one single image into different patches, this approach picks different image patches from different reference or target areas to form a merged image (see Figure 3.6) and uses relative position embedding of ViT to capture their relative spatial relations. The output of DINO was a multi-head average attention map, which illustrates the attention distribution over an image. The multi-head attention is used here to capture richer feature representations, and the overall attention distribution shows the significance of each part in the image. Further, the self-attention weights were summarized by splitting the average attention map back to patches with the same size as the input image patches, which can be formulated as Equation 3.12. Specifically, each $(\mathbf{x}_i, \mathbf{g}_i) \in \mathbb{S}_{target}$ can be merged to generate an attention map and then self-attention weights. In the end, these weights carry the image vision representations correlation and relative spatial relation together to support the FSTL models ensemble.

$$\{\mathbf{w}_i^j = subset(attention\_map((\mathbf{x}_i, \mathbf{g}_i), \{\mathbb{S}_{fs}^j\}_{j=1}^T))\} \rightarrow \mathbb{FSIP}^j \tag{3.12}$$

The advantage of this approach is twofold: first, the self-attention-based weighting can simultaneously consider the location (via position embedding) and image feature embedding (via patch image embeddings) for weighting; second, the extraction of self-attention relies only on pre-trained ViT and satellite image patches without any prior knowledge (e.g., geographical location, image source). Since self-attention can be calculated directly from a pre-trained model, the GWME is an unsupervised model ensemble method to improve the model's geographical generalizability for GeoAI applications.

Weighted Boxes Fusion

Given multiple Few-Shot Transfer Learning (FSTL) predictions and diverse model weights, a further step is to ensemble all of the predictions to generate more accurate and credible geospatial object detection predictions. As Equation 3.8 mentioned, $\mathcal{Q}$ uses a WBFmethod (Solovyev et al., 2021), which utilizes confidence scores of all proposed BBOX to construct the weighted averaged boxes.

The geospatial object detection task combines localization with classification for desired objects. Given a image, the FSTL building detection models usually return the predicted locations of the buildings with the image coordinates of BBOX and a confidence score. Since multiple FSTL were implemented with the same test dataset $\mathbb{S}_{target}$, the same buildings might be detected by more than one FSTL model as the Figure 3.7 (a) and (c) shown. These predictions may

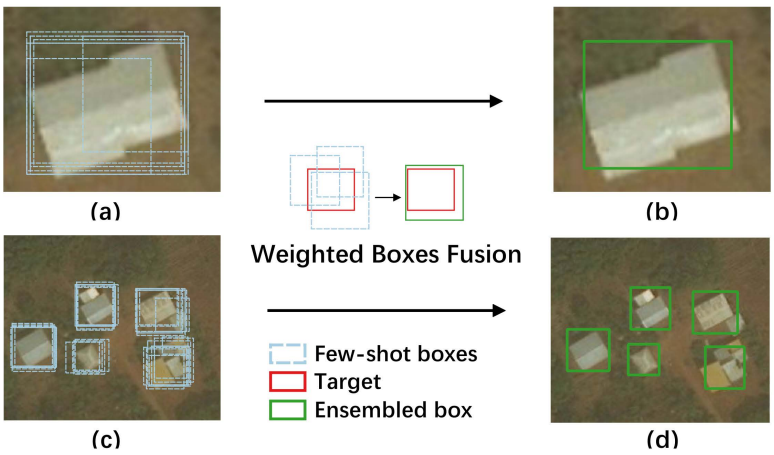


Figure 3.7.: An illustration of weighted boxes fusion. (a) and (c) are multiple predicted boxes from different FSTL models $\{\mathbb{M}_{fs}^{j}\}_{j=1}^{T}$; (b) and (d) are the ensembled boxes by WBF.

vary in terms of locations and confidence scores due to the diverse geographic generalizability of multiple FSTL models. The conventional solution to the problem of multiple BBOX overlapping each other uses non-maximum suppression (NMS) or soft-NMS (Bodla et al., 2017), which picks the highest confidence score by a ranking of confidence scores for all detection boxes and filter out other boxes. However, such methods work well for a single detection model, but they only select the boxes rather than produce an ensemble localization of different predictions from diverse detection models. Unlike NMS or soft-NMS methods that simply filter out part of predictions, the WBF method uses confidence scores of all prediction candidates to construct new weighted averaged boxes, which significantly improve the quality of the combined predicted BBOX (Solovyev et al., 2021). Given the self-attention weights from 3.12, Figure 3.7 illustrates that the WBF method conducts weighted detection boxes fusion for multiple building predictions yielded from reference FSTL models into the models ensembled predictions $\mathbb{P}$.

Put it All Together

To put everything together, the pseudo-code of the complete GWME process is presented in Algorithm 2, where it starts from multiple FSTL model predictions as well few-shot datasets $\{\mathbb{S}_{fs}^{j} = \{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j)\}\}_{j=1}^{T}$ from T reference areas, and ends with the ensemble predictions together with their confidential scores for OSM missing building detection task in the target test area.

Algorithm 2 Geographical Weighted Model Ensemble (GWME)

Input:
$\mathbf{M}_{ViT}$: the pre-trained ViT model;
$\mathbb{S}_{fs}^{j} = \{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j)\}$ with $i = 1, \ldots, n$ and $j = 1, \ldots, T$: FSTL training samples from reference areas;
$\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}$ with $i = 1, \ldots, M$: test dataset from the target area;
$\mathbb{FSIP}^j, j = 1, \ldots, T$: list of objects and scores predicted from different detection models;
TH: threshold of prediction score
$\mathbb{P}$: ensembled objects and scores;
Mode: weighting mode;
$\mathbf{w}^j$: corresponding weights for $\mathbb{M}_{fs}^{j}$.
Weights $\mathcal{W} \leftarrow []$;
for each $(\mathbf{x}_i, \mathbf{g}_i)$ in $\mathbb{S}_{target} = \{(\mathbf{x}_i, \mathbf{g}_i)\}_{i=1}^{M}$ **do**
    for dataset $\mathbb{S}_{fs}^{j}$ of each reference area A_j in $\{\mathbb{S}_{fs}^{j}\}_{j=1}^{T}$ **do**
        if Mode == "average" **then**
            average weights $\mathbf{w}_i^j = 1$;
        else if Mode == "similarity" **then**
            $\mathbf{w}_i^j = \frac{1}{n} \sum_{(\mathbf{x}_i^j, \mathbf{y}_i^j, \mathbf{g}_i^j) \in \mathbb{S}_{fs}^{j}} COS(HIS(\mathbf{x}_i^j), HIS(\mathbf{x}^j))$;
        else if Mode == "distance" **then**
            $\mathbf{w}_i^j = DIS(\mathbf{g}_i, CEN(\mathbb{S}_{fs}^{j}))$;
        else if Mode == "attention" **then**
            image patches **patch_list**$[] \leftarrow \{\mathbf{x}_i, \mathbf{g}_i\} \in \mathbb{S}_{target}$;
            patch_list.*append_patch*$(\{\mathbf{x}_i^j, \mathbf{g}_i^j\}), \{\mathbf{x}_i^j, \mathbf{g}_i^j\} \in \mathbb{S}_{fs}^{j}$;
            multi_heads_attentions $= \mathbf{M}_{ViT}($**patch_list**$)$;
            attention_map = *attention*(multi_heads_attentions);
            $\mathbf{w}_i^j = subset$(attention_map);
        end if
        $\mathbf{w}_i \leftarrow \mathbf{w}_i^j$
        prediction candidates $\mathbb{FSIP}_i \leftarrow \mathbb{FSIP}_i^j$;
    end for
    $\mathcal{W} \leftarrow normalize(\mathbf{w}_i)$;
    update $\mathbb{P}_i = \mathcal{Q}(\mathbb{FSIP}_i, \mathbf{w}_i,)$;
end for
Output:
$\mathbb{P}$: ensembled results and scores;

In conclusion, the GWME method can ensemble the detection performance of all of FSTL models according to their corresponding weights, which were learned by considering both image feature embedding and location embedding with a self-attention mechanism in an unsupervised way. It is still necessary to evaluate the geographic generalizability of proposed GWME methods with experiments of the case study of cross-country OSM missing building detection in Sub-Saharan Africa. The evaluation metrics for indicating the performance of object detection models will be presented in the following section.

3.5. Evaluation Metrics

In neural network training, besides the loss function used to determine whether a model is converging or not, some other metrics are still needed to quantitatively evaluate the performance of object detection models. In this study, some commonly used object detection metrics, such as the precision, recall, accuracy, and f1 score were selected to evaluate the OSM missing building detection model (Padilla et al., 2021).

Since this study conducts single-class object detection with the predicted BBOX and confidential scores, the locations of boxes were evaluated against the ground truth. Intersection Over Union (IOU) is a primary metric to determine if a detection is valid (True Positive) or not (False Positive). It requires a ground truth BBOX $\mathbf{B}_{truth}$ and a predicted BBOX $\mathbf{B}_{prediction}$ and evaluates the overlap between two boxes based on the Jaccard Index as shown in Figure 3.8. The calculation of IOU can be formulated as Equation 3.13.

$$IOU = \frac{area(\mathbf{B}_{prediction} \cap \mathbf{B}_{truth})}{area(\mathbf{B}_{prediction} \cup \mathbf{B}_{truth})} \tag{3.13}$$

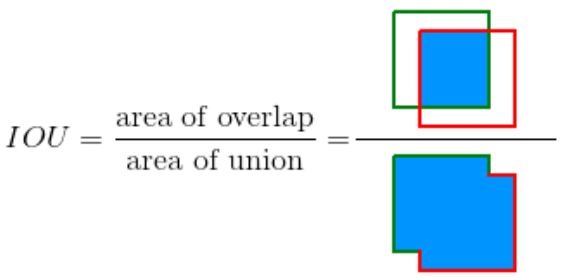


Figure 3.8.: An illustration of Intersection Over Union (IOU) (Padilla et al., 2021).

With IOU, it is possible to further count the confusion matrix (Table 3.2) for object detection. The basic metrics within the confusion matrix are as follows:

- **True Positive (TP) -** A correct prediction with $IOU \geq threshold$.
- **False Positive (FP) -** A wrong prediction with $IOU \leq threshold$.
- **False Negative (FN) -** A ground truth object is not predicted.
- **True Negative (TN) -** A correct false prediction. However, in object detection tasks, there are countless possible bounding boxes that should not be predicted within an image. Thus, TN is not used as a metric here.

Note, the *threshold* for IOU is usually set to 50%, 75%, 95%.

Table 3.2.: A confusion matrix for single-class object detection.

		Prediction		
		Positive	Negative	Total
Ground Truth	Positive	TP	FN	$TP + FN$
	Negative	FP	TN	$FP + TN$
	Total	$TP + FP$	$FN + TN$	N

Additionally, further metrics can be derived based on the confusion matrix:

- **Precision -** Precision shows the ability of a detection model to identify only the relevant objects. It is the percentage of true positive predictions among total positive predictions, which is formulated as:

$$Precison = \frac{TP}{TP + FP} = \frac{True\ Positive}{Total\ Positive\ Predctions} \tag{3.14}$$

- **Recall -** Recall is the ability of a detection model to find all ground truth bounding boxes. It is the percentage of true positive predictions among all relevant ground truths, which is formulated as:

$$Recall = \frac{TP}{TP + FN} = \frac{True\ Positive}{Total\ Ground\ Truths} \tag{3.15}$$

- **Accuracy -** Accuracy represents the number of correctly predicted data instances over the total number of data instances. Since TN is not available, the accuracy here only represents the percentage of true positives over all of the data instances. It is formulated as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} = \frac{Correct\ Predictions}{Total\ Instances} \tag{3.16}$$

- **F1 Score -** F1 Score is a metric that takes both precision and recall into account. It is typically used to seek a balance between precision and recall and is formulated as:

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{3.17}$$

The above statistical metrics were used throughout the study to evaluate the performance of geospatial object detection models, and the results are shown in chapter 5.

3.6. Web Mapping Infrastructure

In order to efficiently visualize the predicted building locations for OSM, a microservice-based web mapping application was designed to integrate GeoAI solutions. This section aims to introduce the potential of GeoAI as a containerized microservice (GeoAIaaS) to the GeoAI-based web mapping application, especially for geospatial object detection (J. Wang, n.d.).

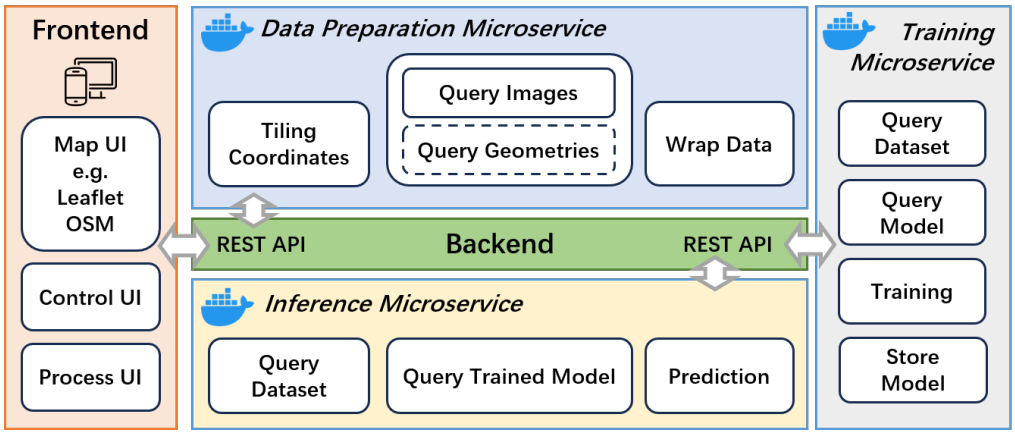


Figure 3.9.: The architecture of GeoAIaaS and a use case of geospatial object detection web application.

GeoAI as a Service

A GeoAI application is typically composed of three parts: frontend, backend, and microservices. This thesis depicts an easy-to-build architectural framework of a microservice-based GeoAI application and APIs between different parts (Figure 3.9).

Frontend - As the main entrance of the GeoAI application, the frontend should adhere to a user-centered design, aiming at providing a user-friendly and intuitive portal to let users interact with maps. Several tools like Leaflet and Cesium enable the integration of OGC WMS provided by various map providers like Bing Maps, OSM, and Google Maps.

Backend - Responsible for handling server-side functionalities that support data management, communication, and API Exposure. In the case of the GeoAI application, it provides REST APIs that enable the management of geospatial data, and AI models, and execute specific missions via microservices.

Microservice - It plays a significant role in GeoAI applications by providing modular, scalable, reusable, and distributed developable geospatial data analysis services. Each microservice can be tailored to handle specific missions via pre-defined recipes, such as processing remote sensing imagery, querying external geographic data, preparing training datasets, conducting object detection, performing semantic segmentation, etc. The key feature of GeoAIaaS is that it can hide the geographic-related process inside the microservice itself to lower the knowledge barrier for researchers. Additionally, it streamlines the intricacies inherent in programming workflows for processing big geospatial data. For example, if someone develops a microservice that splits remote sensing images into fix-sized image tiles with location embedding, given proper metadata, this microservice can be universally employed across various GeoAI applications requiring remote sensing image tiles.

The overall architecture of GeoAI applications may vary in demand, however, well-designed microservices can be easily reused and tailored through the containerization capability of Docker (Boettiger, 2015).

Geospatial Object Detection Case

Geospatial object detection, a prominent research area within GeoAI, has gained substantial development due to the growing availability of remote sensing imagery and advancements in AI techniques, showcasing the great potential for practical applications (H. Li, Herfort, et al., 2022; Sirko et al., 2021; Tiecke et al., 2017). This thesis conducts the full workflow of utilizing GeoAI methods to solve building detection from satellite imagery in three phases, data preparation, training models, and prediction (Figure 3.9). Each phase is wrapped into a standalone microservice which can be called via a REST API from the backend.

Data Preparation - This microservice can query and split remote sensing images sourced from Bing Imagery Service or other WMS providers into the

desired size without losing location information, and if needed, query geometries from OpenStreetMap. Furthermore, it can wrap image and geometry data into a specific data structure, like AtlasHDF (Werner & Li, 2022), which can directly be used for training, or inferencing.

Training - This microservice bridges datasets and DL models, providing the development environment and computation resource for training DL models. It can query a pre-trained object detection model from open sources, like Tensorflow Object Detection Model Zoo, and train it with OSM-labeled remote sensing images.

Inferencing - To efficiently use trained models in real-world scenarios, this microservice provides the interface for predicting buildings based on the geographic coordinates of a specific area. The predicted BBOX and confidential scores will be returned in the easy-to-transfer geospatial data structure, like GeoJSON, etc.

The frontend interface and more use cases of the above geospatial building detection web mapping application are presented in chapter 5.

In conclusion, the proposed GeoAIaaS was shown to be an efficient and easy-to-maintain framework for both GeoAI research and applications. It splits the whole workflow of the GeoAI solution into three major phases conducted by diverse microservices. Each microservice is standalone, distributed, developable, and simply deployable, aiming at lowering the geography knowledge barrier and improving the reusability of GeoAI applications. The experiment of geospatial object detection application shows the potential of building self-served spatial datasets, efficiently training processes, and locally runnable GeoAI services. However, since the complexity and high data volume of big geospatial data, the toleration of pre-defined microservices is still needed when facing different problems. Future work is to explore the standard of building a GeoAI microservice and leverage the community to support more use cases. The findings of this section offer insights into the establishment of a collaborative GeoAI workflow for small research groups or independent developers and shed inspiring light on possibilities for GeoAI applications across local workstations, edge computing, distributed development, etc.

4. Case Study

4.1. Dataset

For a case study of proposed GWME method, this thesis takes the open dataset collected in (H. Li, Herfort, et al., 2022), where training samples $\mathbb{S}_{bm}$ from a well-mapped area A_{bm} in Tanzania is used to train the base model $\mathbb{M}_{bm}$ and a geographically remote area in Cameroon is selected as the target area A_{target} who does not have any training samples. Eight reference areas $\{A_j\}_{j=1}^{8}$ were identified with few-shot training samples $\{\mathbb{S}_{fs}^{j}\}_{j=1}^{8}$ for the FSTL purpose surrounding the target area (as shown in Figure 4.1).

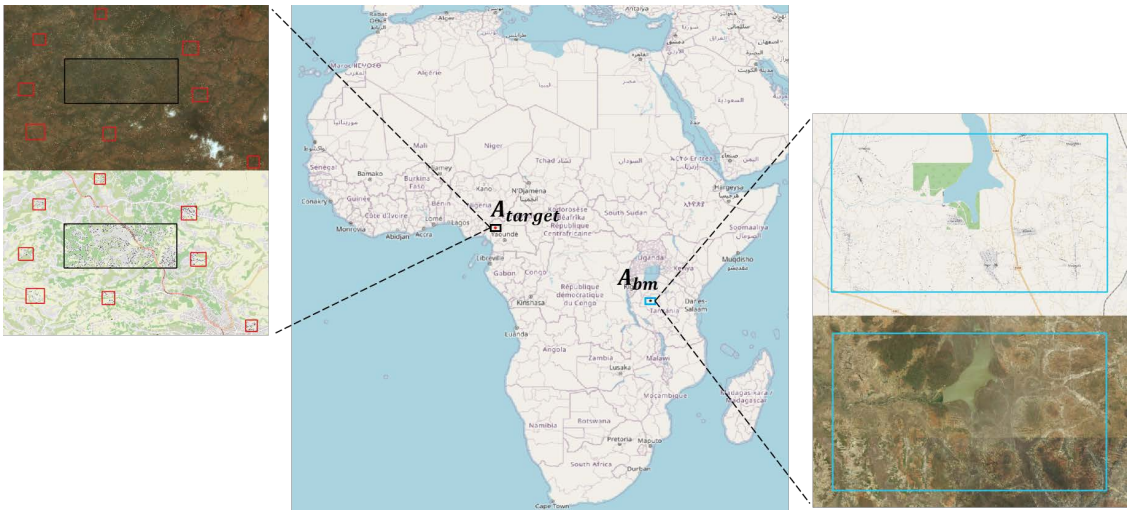


Figure 4.1.: The overview of datasets. A_{bm} refers to the source region that includes base model training dataset $\mathbb{S}_{bm}$ in the blue box. A_{target} refers to the target region, which includes test dataset $\mathbb{S}_{target}$ in the black box and T reference datasets $\{\mathbb{S}_{fs}^{j}\}_{j=1}^{T}$ in red boxes. The base map is from OpenStreetMap and the satellite imagery is from Bing Maps Aerial Imagery.

More specifically, OSM buildings within the training area A_{bm} in Tanzania

were fully mapped during a humanitarian mapping activity organized by the Humanitarian OpenStreetMap Team (HOT). For the target area A_{test}, since it is completely missing in OSM, an expert mapping campaign was organized by H. Li, Herfort, et al. (2022) and **1,811 buildings within an 8.57**km^2 **area** were digitized in total in Cameroon as the ground truth data. Table 4.1 gives the statistic of all datasets

Table 4.1.: Summary statistic of the datasets.

Counts	A_{bm}	A_{target}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Buildings	6,272	1,811	66	45	116	46	71	61	40	79
Areas (km^2)	232.50	8.57	0.35	0.16	0.35	0.22	0.34	0.44	0.25	0.20
Tiles n	1,744	343	5	5	9	7	9	9	7	7

4.2. Experiment Setup

To generate the training data, the ohsome2label package (Z. Wu et al., 2020) was used to combine OSM building geometries with Bing satellite imagery at a zoom level of 18 (i.e., a spatial resolution of 0.6m), which were then converted to training datasets for the TensorFlow Object Detection API [1]. For the SSD object detection model (W. Liu et al., 2016), the pre-trained parameters were downloaded from the TensorFlow Detection Model Zoo. With an initial learning rate of 0.0004, the training process for the base model in Tanzania was run for 50,000 epochs and the FSTL fine-tuning epochs were then set to 10,000 for all reference areas to ensure the training models were convergence. The algorithms were implemented using Python 3.10, TensorFlow 2.6, and TensorFlow object detection API on a Linux server with a GeForce RTX 3080Ti graphical processing unit (GPU) of 12 GB memory.

For evaluation, common evaluation metrics were used for a single-class object detection task, such as Precision, Recall, Accuracy, and F1-score. Specifically, a default IoU threshold of 0.5 was set as the criteria to decide whether a prediction bounding box refers to a building bounding box in the target data, which then distinguishes all predictions into False Negatives (FN), False Positives (FP), and True Positives (TP). There is no True Negative (TN) since detecting non-building objects is not reasonable.

[1] https://github.com/tensorflow/models/tree/master/research/object_detection

5. Experiments and Results

This chapter demonstrates the experiments and results on improving OSM missing buildings leveraging the proposed GWME method and showcases a minimalistic GeoAI web application for the building detection task. The OSM missing building detection task was divided into three main steps: 1) Multiple few-shot predictions by FSTL models. 2) Weight computing via diverse weighting strategies. 3) Weighted model ensemble and evaluation. Additionally, the hyperparameter setting within the model ensemble is also a noteworthy experiment.

5.1. Multiple Few Shot Model Predictions

First, a general one-stage object detection model SSD-ResNet101 pre-trained via Microsoft COCO dataset (Lin et al., 2015) was trained with the OSM-labeled training samples $\mathbb{S}_{bm}$ from the source region A_{bm}. The obtained building detection base model $\mathbb{M}_{bm}$ performed well with the validation set of the source area, with a series of promising evaluation results (91.04% precision, 51.69% recall, 49.18% accuracy, and 65.94% F1 score). Then few-shot training sets $\{\mathbb{S}_{fs}^{j}\}_{j=1}^{8}$ from 8 reference areas geographically surrounding the target area were used to fine-tune the base model $\mathbb{M}_{bm}$ and generated 8 FSTL models $\{\mathbb{M}_{fs}^{j}\}_{j=1}^{8}$, which were further used to predict missing buildings with the test dataset $\mathbb{S}_{target}$ respectively. The detection performance of the base model and FSTL models were shown in Table 5.1.

Herein, a finding from Table 5.1 shows an overall significant performance improvement of FSTL models over the base model, with so call $Mean(\{\mathbb{FSIP}^{j}\}_{j=1}^{8})$ over the base model. However, an interesting observation is that the individual FSTL model performance varies a lot, where $\mathbb{FSIP}^{4}$ leads to the biggest improvement and $\mathbb{FSIP}^{6}$ the lowest. Particularly, high precision means the ability of a model that correctly detect buildings among all detections, high accuracy means the ability to successfully detect buildings among all objects, high recall means the ability to completely detect buildings, and a high f1 score means that the

Table 5.1.: Evaluation metrics of predictions from the base model and single FSTL models on the test dataset. $BM\mathbb{P}$ and $\mathbb{FSP}^j$ indicate the model predictions of the base model $\mathbb{M}_{bm}$ as well as different FSTL models $\{\mathbb{M}_{fs}^j\}_{j=1}^T$.

Predictions	Precision (%)	Accuracy (%)	Recall (%)	F1
$BM\mathbb{P}$	97.66	13.71	13.75	0.2411
$\mathbb{FSP}^1$	99.00	60.90	61.27	0.7570
$\mathbb{FSP}^2$	96.94	**68.93**	**70.46**	**0.8160**
$\mathbb{FSP}^3$	98.18	53.06	53.58	0.6933
$\mathbb{FSP}^4$	98.22	49.06	49.50	0.6582
$\mathbb{FSP}^5$	98.44	61.27	61.87	0.7598
$\mathbb{FSP}^6$	84.65	40.90	44.18	0.5806
$\mathbb{FSP}^7$	**99.12**	52.66	52.91	0.6899
$\mathbb{FSP}^8$	98.73	52.60	52.96	0.6894
$Mean(\{\mathbb{FSP}^j\}_{j=1}^8)$	96.66	54.92	55.84	0.7055

model performs well in both detection precision and completeness. Such a distinct behavior implies the different levels of geographical generalizability among a set of FSTL models.

5.2. Weighting Result

To support the next model ensemble step, the tile-based weights matrix was computed via four different weighting strategies (*average*, *similarity*, *distance*, *attention*). These weights represent the importance of the neighboring reference areas to the target area, and also the degree of involvement of the FSTL models in the model ensemble. Specifically, a higher weight for an individual model represents a higher contribution of that FSTL model to the final ensemble prediction compared to other models. Therefore, in the experiments in this thesis, weights can be recognized as a concrete numerical representation of the geographical generalizability of a building detection model.

To better understand the relationships between weights and neighboring reference areas to the target area, Figure 5.1 illustrates the weight distribution of the distance weighting approach (*distance*). Obviously, the distance weighting

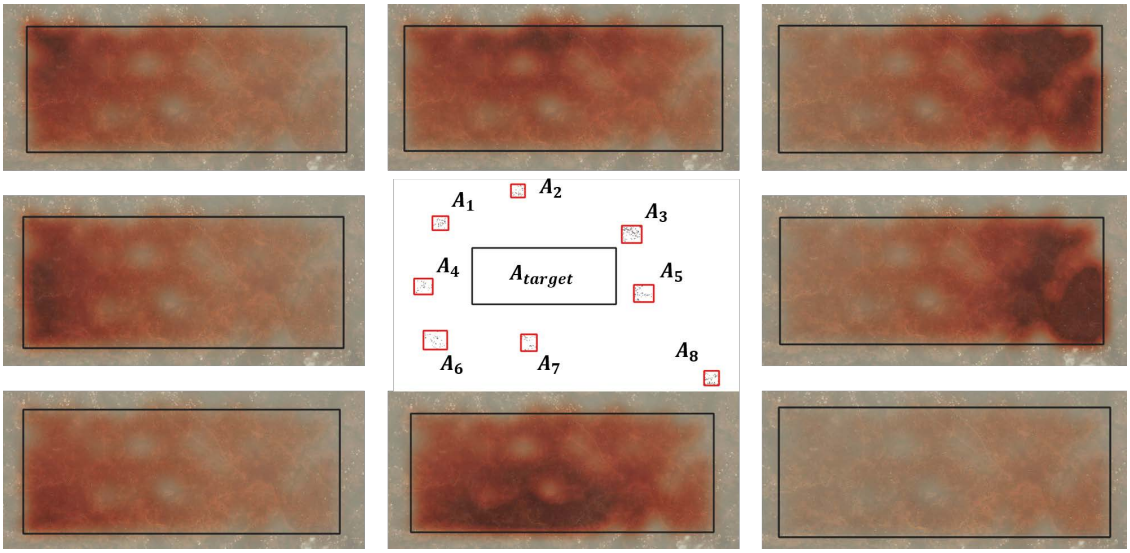


Figure 5.1.: The visualization of distance weighting results. The center is the geographic location of the reference areas and the target area. The 8-neighborhoods heatmaps represent the weight distribution by the distance weighting. The color of the heatmaps ranges from light white to dark red representing the distance weights from 0 to 1.

result fits Tobler's first law of geography very well, where the image tile is closer to the reference area, the distance weight is higher. To put it another way, each image tile is more susceptible to models that are geospatially close when ensemble models.

When comparing three weighting approaches of image similarity, distance, and self-attention, Figure 5.2 shows different weight distribution patterns. Specifically, the image-similarity-based weights vary due to the contextual difference of the images, which may also lead to similar images in different regions resulting in the disappearance of differentiation between diverse FSTL models. On the other hand, the distance-based weights exhibit strong spatial clustering, which may result in a single FSTL model contributing less to distant image tiles. However, the weights based on the self-attention mechanism, because it takes both the content correlation and the relative position correlation into account, show the weight difference caused by the distance of different FSTL models while ensuring that the image content is also considered.

Figure 5.3 illustrates the histogram distribution of self-attention-based weights attributed to the model ensemble predictions of 343 image tiles by 8 FSTL models. In conjunction with Table 5.1, a very interesting finding is that when using a

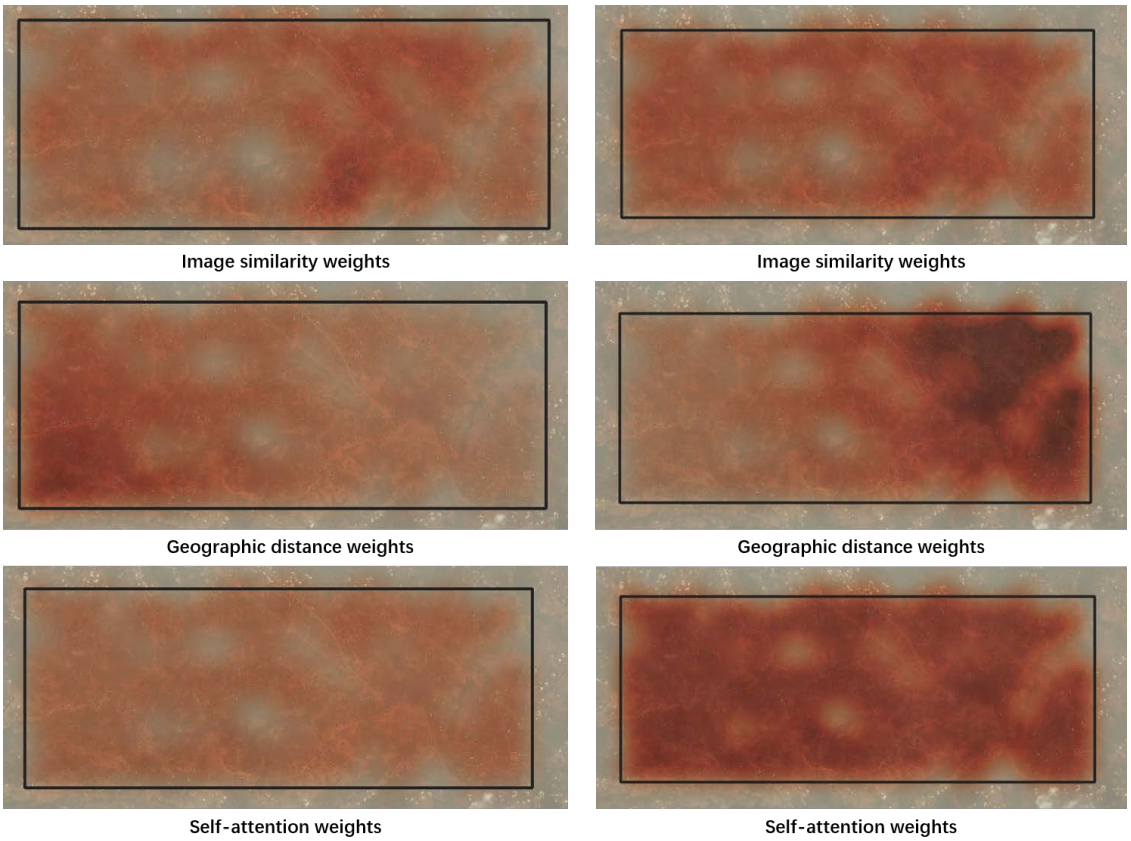


Figure 5.2.: The visual comparison of different weighting strategies. The left column is the weighting result of $\mathbb{M}_{fs}^{6}$, and the right column is $\mathbb{M}_{fs}^{3}$.

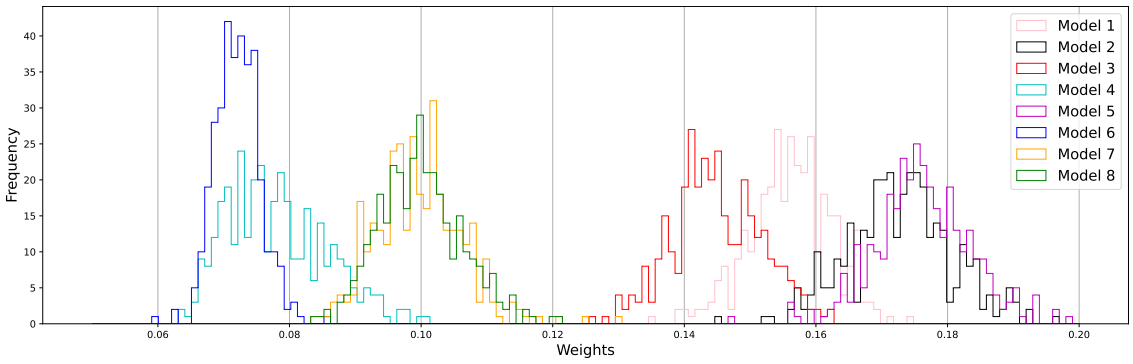


Figure 5.3.: The comparison of the self-attention weights histogram distribution of multiple FSTL models.

single FSTL model to predict buildings at the target area, FSTL models from reference area A_5, A_2, A_1, A_3, which account for larger self-attention weights, perform better in the single-FSTL-model prediction. In contrast, FSTL models from reference area A_7, A_8, A_4, A_6 share smaller self-attention weights while performing relatively poorly in the individual model prediction.

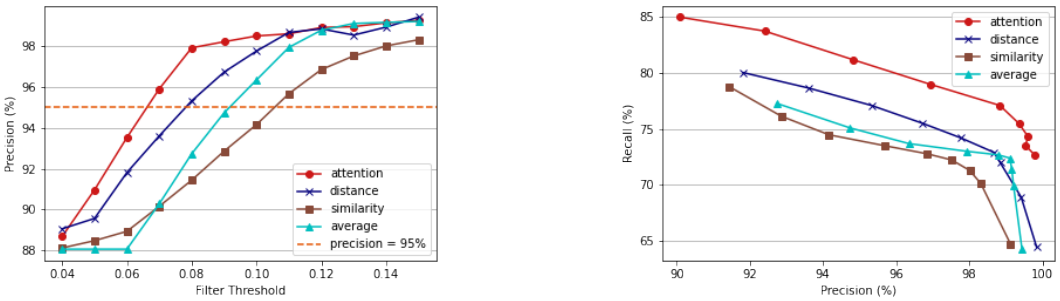
5.3. Predictions Ensemble Result

Carrying diverse weights, multiple WBF-based model ensemble experiments were conducted. Table 5.2 compares the performance of four different weighting strategies with the proposed GWME method, where a threshold of prediction scores (**TH**) is set to ones with *precision* $\geq$ 95% as shown in Figure 5.4. Several key findings can be observed. First, even with average weighting, the model ensemble leads to a significant improvement in the overall performances of single FSTL models, which proves the effectiveness of GWME compared to the baseline model. Second, although it assumes that image similarities play a role in the model's generalization, image similarity weighting (*similarity*) ends up with the least improvement in the model ensemble, while the inverse distance weighting gives a surprisingly better result. Last but foremost, the biggest performance improvement via the GWME method is with self-attention-based weighting (*attention*), which leads to more than 6% improvement in overall accuracy and the highest Recall of 78.99% in the target area A_{target}.

Table 5.2.: Evaluation metrics of predictions from ensembled results by different weighting modes.

GWME Weightings	Precision (%)	Accuracy (%)	Recall (%)	F1
average	96.35	71.70	73.70	0.8352
similarity	95.68	71.16	73.52	0.8315
distance	**97.76**	72.98	74.22	0.8438
attention	96.95	**77.07**	**78.99**	**0.8705**

The proposed GWME method can effectively improve the geographical generalizability of GeoAI models in an unsupervised manner. In Figure 5.5, multiple evaluation metrics were plotted between the baseline method (e.g., the base model $\mathbb{M}_{bm}$ and a single FSTL model $\mathbb{M}_{fs}^{4}$) and GWME results with different weighting strategies.



(a) The curve of precision and the threshold of prediction scores.

(b) Precision-Recall curve with the rising threshold.

Figure 5.4.: Evaluation Metrics

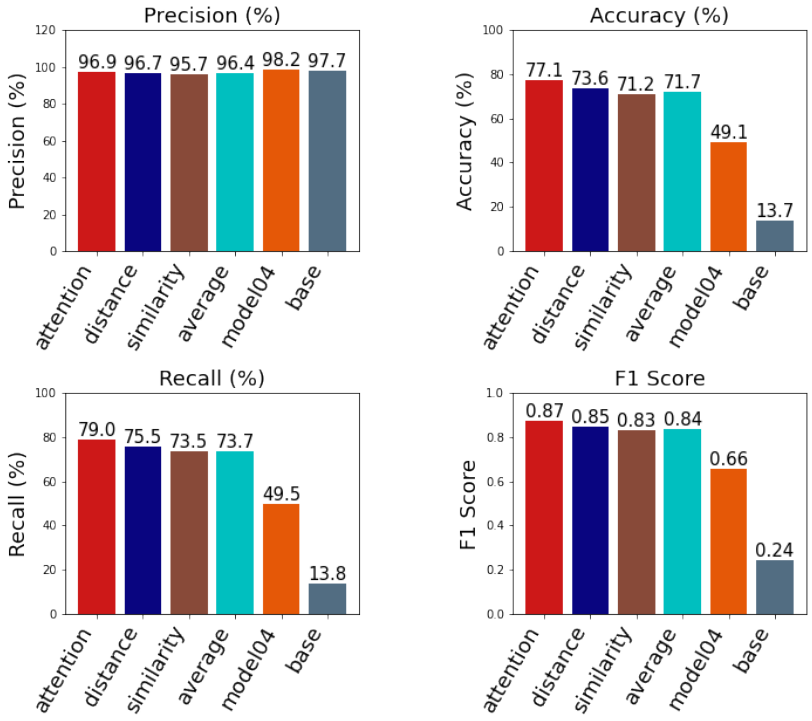


Figure 5.5.: Performance of GWME predictions (precision > 95%) using different weighting strategies.

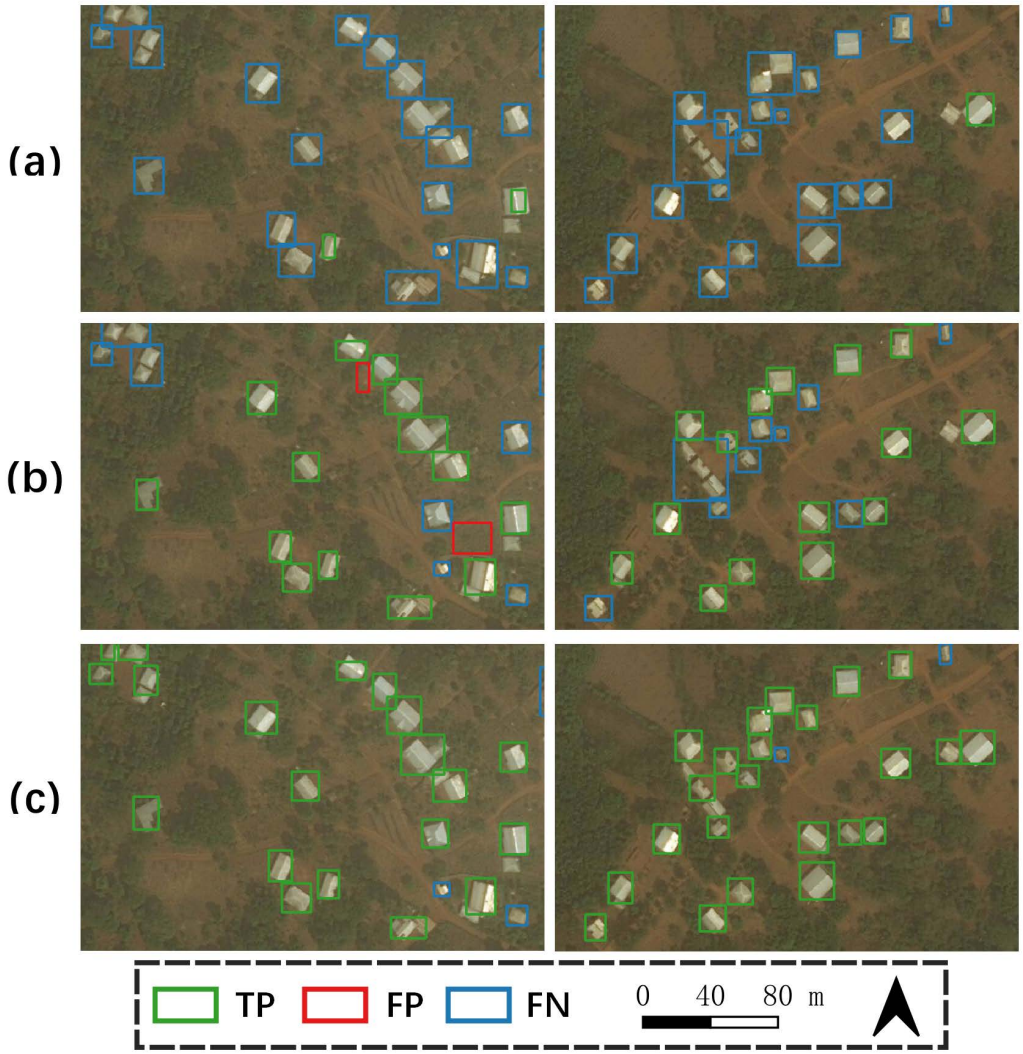


Figure 5.6.: The comparison map of prediction results. (a) the base model ($\mathbb{M}_{bm}$); (b) the single FSTL model (i.e., $\mathbb{M}^4_{fs}$); (c) the GWME result with self-attention-based weights *attention*.

To visually interpret the advantages of the proposed GWME method, Figure 5.6 compares the OSM missing building detection results of three different models: the base model $\mathbb{M}_{bm}$, the single FSTL model $\mathbb{M}^4_{fs}$, and the results from GWME method with self-attention-based weights (*attention*). Comparing Figure 5.6 (b) with (a), It's obvious that a significant decrease in FN, which originates from valid buildings that are overlooked by a model trained in geographically remote areas, confirms the assumption that few-shot learning is very effective in improving model performance in geographically remote areas. Comparing Figure 5.6 (c) with (b), the proposed GWME with self-attention weights (*attention*) further reduces the FN and FP. This confirms the effectiveness of GWME in achieving better geographical generalizability for GeoAI models, especially for geospatial object detection tasks.

5.4. Web Mapping Interface

To better visualize the machine-generated geographic results and explore the potential of the GeoAI as a Service (GeoAIaaS), a minimalistic building detection web mapping application was designed and implemented as Figure 5.7 shown. The basic development framework was orchestrated by Vue.js [1] as the frontend, GeoDjango [2] as the backend, and GeoAI based microservices. Specific application features are as follows:

Frontend - As shown in Figure 5.7, the frontend was mainly composed of two components, the map interface developed by Leaflet [3] and the control user interface (UI) implemented by modern web development techniques (HTML, CSS, JavaScript, etc.). Specifically, OSM was used as the base map layer, and Bing Maps Aerial and ERSI Imagery were added as overlay layers for comparison, which can be controlled by the layer and the opacity controllers at the top right corner of the interface. Furthermore, machine-predicted building instances can be added as new overlay layers in GeoJSON format or removed from current layer groups. With the drawing tool sitting at the left bottom corner, users can draw a rectangle region on the map or manually type coordinates into the panel to further request the training samples downloading process, which could be the data preparation mission provided by a microservice. More complex operations within the GeoAI solution (e.g. models training, inference, etc.) can also be provided with frontend entries for users depending on demands.

[1] https://vuejs.org/
[2] https://docs.djangoproject.com/en/4.2/ref/contrib/gis/
[3] https://leafletjs.com/

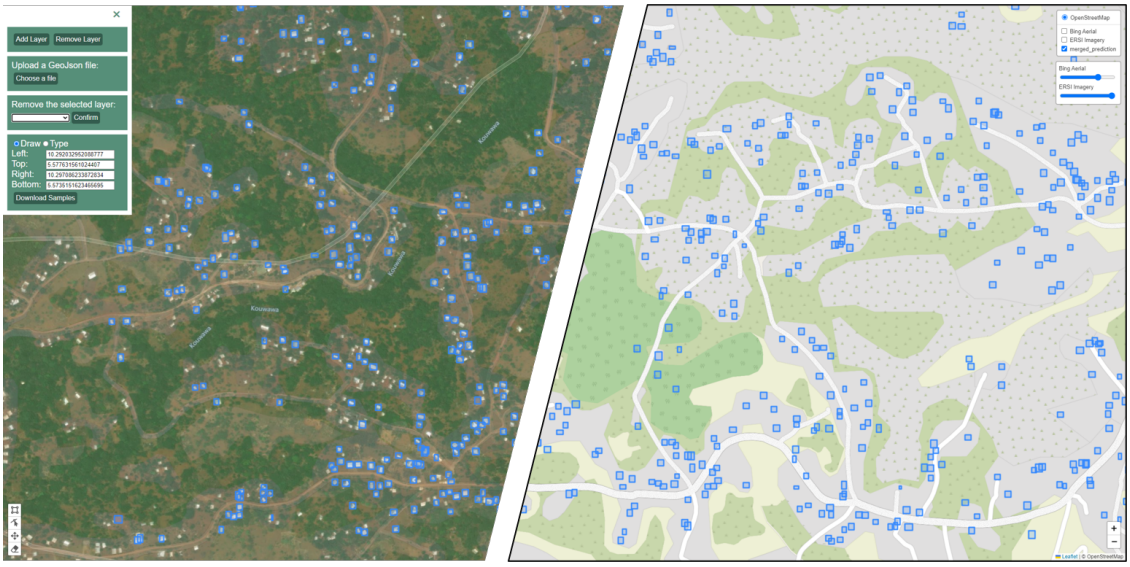


Figure 5.7.: The demo of a GeoAI web application for building detection. The left map is from Bing Maps Aerial, and the right map is from OSM. The blue rectangles displayed on the map represent the locations of predicted buildings.

Backend - Since this web application uses WMS from OSM, Bing Maps, and ESRI to provide the map tiling interface, the backend here only considers the storage and management of machine-generated geographic contents, like GeoJSON files, and the delivery of GeoAI mission requests between frontend and microservices. For instance, users can upload/delete GeoJSON format building prediction data to a PostGIS geospatial database by HTTP operations (like POST). In addition, the backend can also receive the type of GeoAI missions (like data preparation, inference, etc.) and the test area (coordinates) specified by users, and encapsulate these raw data into a standard HTTP request which can be received by microservices. Meanwhile, the backend also plays a role in accepting and parsing the response data from microservices.

Microservice - Leveraging the encapsulation capabilities and reusability of Docker (Boettiger, 2015), microservices can empower modern web mapping applications with more GeoAI solutions. Based on the building detection task in this thesis, two Flask-based microservices were designed to assist users in achieving more efficient model training and building detection. One is the data preparation microservice, which receives the HTTP request and based on the metadata and the required image data source can generate fixed-size remote sensing images and corresponding OSM geometries for the training purpose.

Another is a building prediction microservice, which can use a pre-trained building model to make predictions on the received images, and eventually return the prediction results in the form of GeoJSON. Furthermore, if possible, the model training process can also deployed on machines that carry sufficient computational resources or high-performance computing platforms and provide APIs for users to perform model training remotely.

6. Discussion

The overall research objective of this study is to research the geographical generalizability of GeoAI models across geospatial space with a case study of detecting OSM missing buildings across different countries in Africa and develop a web mapping application integrated with GeoAI solutions. Specifically, this was achieved by three parts: 1) investigated GeoAI trends and state-of-art technologies; 2) proposed a GWME method to improve the replicability and geographical generalizability of a trained building detection model from the source region to the target region; 3) explored GeoAI as a Service framework to empower web mapping applications.

First, Section 2.2 provides a comprehensive summary of the current development state of GeoAI and ideas in addressing real-world challenges, from definitions, popular models, and applications to transfer learning and spatial explicit AI. Second, Section 2.1, 3.2, 3.3, 3.4, 5.3 together present the existing challenge in the efficiency of OSM mapping activities and the difficulties of transferring building detection model across countries, and proposes a GWME method, which is based on FSTL and self-attention based weighted boxes fusion, greatly improved geographical generalizability of object detection models. Additionally, Section 2.3, 3.6 and 5.4 design and implement a visualization application for building detection results, and introduce an innovative attempt to integrate GeoAI solution into web mapping applications.

However, due to the complexity of the problem and the limitations of the experiments, there are still many pending challenges to be addressed in future research.

- Although the proposed GWME approach and GeoAIaaS application present a great improvement in transferring a pre-trained building detection model across geospatial space and high efficiency of visualizing machine-generated predictions with OSM map and Bing Maps Aerial Imagery, how to further extend such a model to more study areas and establish data connectivity with OSM mapping is still a future research direction. One possible solution is to design a system that maps the footprints of detected buildings and transfers these geographic data to the OSM mapping tools,

like Rapid Editor, to assist in manual mapping.

- This study uses ensemble weights, which were recognized as the importance of reference areas to the target area by considering both the spatial and image correlation between training samples, to represent the geographical generalizability of building detection models numerically. This representation is still limited to the quality of training samples and the number of reference regions, and in the future, correlations between different models can be considered from either the extraction of high-level features of the training process or transferability metrics for object detection (Fouquet et al., 2023), which may find the more important features of GeoAI models.
- The model ensemble in this study occurs at a prediction level, while a parameter-level model ensemble can be preferred by considering computational efficiency (Dong et al., 2020).
- The reference areas in the case study were picked in proximity to the target area, and future work could consider either a larger scale study area for more aggressive improvement of geographical generalizability, or the same study areas but at different times to study temporal generalizability.
- In the step of self-attention weighting, this study used the default position embedding from a pre-trained ViT mode, which only considers the relative spatial correlation across image patches. It would be interesting to integrate spatially explicit location embedding into the training process (Mai et al., 2020) or other advanced location encoding technologies (Mai, Janowicz, et al., 2022).
- Leveraging the great generalization capability of foundation models, more complex GeoAI solutions (such as multi-class geospatial object detection, semantic segmentation, satellite image retrieval, and forecasting) are yet to be explored(Mai, Cundy, et al., 2022; Mai, Huang, et al., 2023).
- The proposed GeoAIaaS pattern demonstrates the potential of GeoAI across distributed AI development, edge-computing, and advanced spatial data infrastructure (SDI), which could enhance automatic methods for improving the quality and sharing of geospatial data and metadata, for supporting reproducibility and replicability in GeoAI research (Janowicz et al., 2020). This also inspires future research possibilities into geospatial federated learning with multi-modal EO, mobile devices, and LBS.

7. Conclusion

The emergence of novel deep learning networks over the past few decades, together with multiple sourced big geographic data, has greatly contributed to the development of GeoAI. Aiming to improve mapping efficiency and reduce volunteer efforts for OSM building mapping, this study proposed a Geographical Weighted Model Ensemble (GWME) method to improve the geographical generalizability of GeoAI models (H. Li, Wang, et al., 2023). Leveraging the replicability of few-shot transfer learning (FSTL) (H. Li, Herfort, et al., 2022), this thesis conducts multiple FSTL with a SSD-ResNet101 based building detection model trained on the source region across several reference regions surrounding the target region, and develops a self-attention-based weighted boxes fusion approach by simultaneously considering the image and location correlation among diverse FSTL models. More importantly, compared with the other three weighting approaches (average weighting, image similarity weighting, and geographical distance weighting), self-attention weighting can intuitively learn both context and location information from a pre-trained ViT model without prior knowledge in a fully unsupervised manner. To evaluate the effectiveness of GWME, intensive experiments were conducted with a case study of OSM missing building detection in the African region, where the base model is trained in Tanzania, and the target test area is in Cameroon. Experimental results confirmed the capability of GWME with the self-attention-based weighting which can outperform both the base model and single FSTL model with overall performance improvement over the best single FSTL model. Future work is to explore a larger-scale GeoAI model on geographical generalization and temporal generalization.

Additionally, to explore the GeoAI-enhanced web mapping applications, this thesis demonstrates a GeoAI as a Service (GeoAIaaS) design pattern, which was shown to be an efficient and easy-to-maintain framework for both GeoAI research and applications. It splits the whole workflow of GeoAI into three major phases conducted by diverse microservices. Each microservice is standalone, distributed, developable, and simply deployable, aiming at lowering the geography knowledge barrier and improving the reusability of GeoAI solutions. The ex-

periment of a building detection application shows the potential of constructing self-served spatial datasets, efficiently training processes, and locally runnable GeoAI services. However, since the complexity and high data volume of big geospatial data, the toleration of pre-defined microservices is still needed when facing different problems. Future work is to explore the standard of building a GeoAI microservice and leverage the community to support more use cases. The findings of this thesis offer insights into the establishment of a collaborative GeoAI workflow for small research groups or independent developers

In short, this thesis inspires the general topic of the geographical generalizability of GeoAI models by the proposed GWME method and sheds light on possibilities for GeoAI-enhanced web mapping applications across local workstations, edge computing, and distributed development by the depicted GeoAIaaS.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2016, March 16). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. arXiv: 1603.04467 [cs]. https://doi.org/10.48550/arXiv.1603.04467

Agana, N. A., & Homaifar, A. (2018). EMD-Based Predictive Deep Belief Network for Time Series Prediction: An Application to Drought Forecasting. *Hydrology*, *5*(1), 18. https://doi.org/10.3390/hydrology5010018

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Amini Amirkolaee, H., & Arefi, H. (2019). CNN-based estimation of pre- and post-earthquake height models from single optical images for identification of collapsed buildings. *Remote Sensing Letters*, *10*(7), 679–688. https://doi.org/10.1080/2150704X.2019.1601277

Anselin, L. (1989). What is Special About Spatial Data? Alternative Perspectives on Spatial Data Analysis (89-4). Retrieved August 21, 2023, from https://www.semanticscholar.org/paper/What-is-Special-About-Spatial-Data-Alternative-on-Anselin/aef0b10c42f37ce266301a6a1f767a79251e10f3

Anthony C. Robinson. (n.d.). *Spatial is Special | Map MOOC*. Spatial is Special. Retrieved August 22, 2023, from https://www.e-education.psu.edu/maps/l2_p2.html

Ashley, H. (2009). *Change at Hand: Web 2.0 for Development*. IIED.

Bejiga, M. B., Zeggada, A., Nouffidj, A., & Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, *9*(2), 100.

Bi, F., Lei, M., Wang, Y., & Huang, D. (2019). Remote Sensing Target Tracking in UAV Aerial Video Based on Saliency Enhanced MDnet. *IEEE Access*, *7*, 76731–76740. https://doi.org/10.1109/ACCESS.2019.2921315

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017, August 8). *Soft-NMS – Improving Object Detection With One Line of Code*. arXiv: 1704.04503 [cs]. https://doi.org/10.48550/arXiv.1704.04503

Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, *49*(1), 71–79. https://doi.org/10.1145/2723872.2723882

Bousquet, O., & Elisseeff, A. (2002). Stability and Generalization. *Journal of Machine Learning Research*, *2*, 499–526. Retrieved August 23, 2023, from https://jmlr.org/papers/v2/bousquet02a.html

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021, May 24). *Emerging Properties in Self-Supervised Vision Transformers*. arXiv: 2104.14294 [cs]. https://doi.org/10.48550/arXiv.2104.14294

Carvalhais, N., Forkel, M., Khomik, M., Bellarby, J., Jung, M., Migliavacca, M., u, M., Saatchi, S., Santoro, M., Thurner, M., Weber, U., Ahrens, B., Beer, C., Cescatti, A., Randerson, J. T., & Reichstein, M. (2014). Global covariation of carbon turnover times with climate in terrestrial ecosystems. *Nature*, *514*(7521), 213–217. https://doi.org/10.1038/nature13731

Chen, H., Qi, Z., & Shi, Z. (2022). Remote Sensing Image Change Detection With Transformers. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–14. https://doi.org/10.1109/TGRS.2021.3095166

Chen, J., Zhou, Y., Zipf, A., & Fan, H. (2019). Deep Learning From Multiple Crowds: A Case Study of Humanitarian Mapping. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(3), 1713–1722. https://doi.org/10.1109/TGRS.2018.2868748

Chen, J., & Zipf, A. (2017). DeepVGI: Deep Learning with Volunteered Geographic Information. *Proceedings of the 26th International Conference on World Wide Web Companion*, 771–772. https://doi.org/10.1145/3041021.3054250

Cheng, G., Zhou, P., & Han, J. (2016). Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(12), 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, September 2). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv: 1406.1078 [cs, stat]. https://doi.org/10.48550/arXiv.1406.1078

Cosine similarity. (2023, August 9). In *Wikipedia*. Retrieved September 3, 2023, from https://en.wikipedia.org/w/index.php?title=Cosine_similarity&oldid=1169535260
Page Version ID: 1169535260

Dai, J., Li, Y., He, K., & Sun, J. (2016, June 21). *R-FCN: Object Detection via Region-based Fully Convolutional Networks*. arXiv: 1605.06409 [cs]. https://doi.org/10.48550/arXiv.1605.06409

Demiray, B. Z., Sit, M., & Demir, I. (2021, September 20). *DEM Super-Resolution with EfficientNetV2*. arXiv: 2109.09661 [cs, eess]. https://doi.org/10.48550/arXiv.2109.09661

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Ding, J., Xue, N., Xia, G.-S., Bai, X., Yang, W., Yang, M. Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., & Zhang, L. (2022). Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7778–7796. https://doi.org/10.1109/TPAMI.2021.3117983

Dittus, M., Quattrone, G., & Capra, L. (2016). Analysing volunteer engagement in humanitarian mapping: Building contributor communities at large scale. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 108–118.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z

Doshi, J., Garcia, D., Massey, C., Llueca, P., Borensztein, N., Baird, M., Cook, M., & Raj, D. (2019, October 14). *FireNet: Real-time Segmentation of Fire Perimeter from Aerial Video*. arXiv: 1910.06407 [cs, eess]. https://doi.org/10.48550/arXiv.1910.06407

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021, June 3). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv: 2010.11929 [cs]. https://doi.org/10.48550/arXiv.2010.11929

Du, B., Cai, S., & Wu, C. (2019). Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(8), 3043–3055. https://doi.org/10.1109/JSTARS.2019.2917703

Elmustafa, A., Rozi, E., He, Y., Mai, G., Ermon, S., Burke, M., & Lobell, D. (2022). Understanding economic development in rural Africa using satellite imagery, building footprints and deep models. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–4. https://doi.org/10.1145/3557915.3561025

Esch, T., Marconcini, M., Felbier, A., Roth, A., Heldens, W., Huber, M., Schwinger, M., Taubenböck, H., Müller, A., & Dech, S. (2013). Urban Footprint Processor—Fully Automated Processing Chain Generating Settlement Masks From Global Data of the TanDEM-X Mission. *IEEE Geoscience and Remote Sensing Letters*, *10*(6), 1617–1621. https://doi.org/10.1109/LGRS.2013.2272953

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Feng, Y., Thiemann, F., & Sester, M. (2019). Learning Cartographic Building Generalization with Deep Convolutional Neural Networks. *ISPRS International Journal of Geo-Information*, *8*(6), 258. https://doi.org/10.3390/ijgi8060258

Forkel, M., Migliavacca, M., Thonicke, K., Reichstein, M., Schaphoff, S., Weber, U., & Carvalhais, N. (2015). Codominant water control on global interannual variability and trends in land surface phenology and greenness. *Global Change Biology*, *21*(9), 3414–3435. https://doi.org/10.1111/gcb.12950

Fouquet, L., Maggio, S., & Dreyfus-Schmidt, L. (2023, June 27). *Transferability Metrics for Object Detection*. arXiv: 2306.15306 [cs]. https://doi.org/10.48550/arXiv.2306.15306

Fu, Y., Zhang, T., Zheng, Y., Zhang, D., & Huang, H. (2019). Hyperspectral Image Super-Resolution With Optimized RGB Guidance. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11653–11662. https://doi.org/10.1109/CVPR.2019.01193

GeoJSON. (2023, August 26). In *Wikipedia*. Retrieved August 26, 2023, from https://en.wikipedia.org/w/index.php?title=GeoJSON&oldid=1172309030 Page Version ID: 1172309030

Gómez-Barrón, J., Manso Callejo, M. Á., Alcarria, R., & Iturrioz, T. (2016). Volunteered Geographic Information System Design: Project and Participation Guidelines. *ISPRS International Journal of Geo-Information*, *5*, 108. https://doi.org/10.3390/ijgi5070108

Goodchild, M. (2001). Issues in spatially explicit modeling. *Agent-based models of land-use and land-cover change*, 13–17.

Goodchild, M. F. (2004). The Validity and Usefulness of Laws in Geographic Information Science and Geography. *Annals of the Association of American Geographers*, *94*(2), 300–303. https://doi.org/10.1111/j.1467-8306.2004.09402008.x

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*, *69*(4), 211–221. https://doi.org/10.1007/s10708-007-9111-y

Goodchild, M. F., & Li, W. (2021). Replication across space and time must be weak in the social and environmental sciences. *Proceedings of the National Academy of Sciences*, *118*(35), e2015759118. https://doi.org/10.1073/pnas.2015759118

Great-circle distance. (2023, August 5). In *Wikipedia*. Retrieved September 3, 2023, from https://en.wikipedia.org/w/index.php?title=Great-circle_distance&oldid=1168844391
Page Version ID: 1168844391

Grillo, A., Krylov, V. A., Moser, G., & Serpico, S. B. (2021). Road Extraction and Road Width Estimation Via Fusion of Aerial Optical Imagery, Geospatial Data, and Street-Level Images. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2413–2416. https://doi.org/10.1109/IGARSS47720.2021.9554540

Haklay, M., & Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, *7*(4), 12–18. https://doi.org/10.1109/MPRV.2008.80

Han, X.-H., Shi, B., & Zheng, Y. (2018). SSF-CNN: Spatial and Spectral Fusion with CNN for Hyperspectral Image Super-Resolution. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2506–2510. https://doi.org/10.1109/ICIP.2018.8451142

Han, X., Zhong, Y., & Zhang, L. (2017). An Efficient and Robust Integrated Geospatial Object Detection Framework for High Spatial Resolution Remote Sensing Imagery. *Remote Sensing*, *9*(7), 666. https://doi.org/10.3390/rs9070666

Haq, M. A., Rahaman, G., Baral, P., & Ghosh, A. (2021). Deep Learning Based Supervised Image Classification Using UAV Images for Forest Areas Classification. *Journal of the Indian Society of Remote Sensing*, *49*(3), 601–606. https://doi.org/10.1007/s12524-020-01231-3

Hardt, M., Recht, B., & Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. *Proceedings of The 33rd International Conference on Machine Learning*, 1225–1234. Retrieved August 23, 2023, from https://proceedings.mlr.press/v48/hardt16.html

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs]. https://doi.org/10.48550/arXiv.1512.03385

He, Q., Sun, X., Yan, Z., Li, B., & Fu, K. (2022). Multi-Object Tracking in Satellite Videos With Graph-Based Multitask Modeling. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–13. https://doi.org/10.1109/TGRS.2022.3152250

Herfort, B., Lautenbach, S., Porto de Albuquerque, J., Anderson, J., & Zipf, A. (2021). The evolution of humanitarian mapping within the OpenStreetMap community. *Scientific Reports*, *11*(1), 3037. https://doi.org/10.1038/s41598-021-82404-z

Herfort, B., Li, H., Fendrich, S., Lautenbach, S., & Zipf, A. (2019). Mapping Human Settlements with Higher Accuracy and Less Volunteer Efforts by Combining Crowdsourcing and Deep Learning. *Remote Sensing*, *11*(15), 1799. https://doi.org/10.3390/rs11151799

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hsu, C.-Y., & Li, W. (2021, March 5). *Learning from Counting: Leveraging Temporal Classification for Weakly Supervised Object Localization and Detection*. arXiv: 2103.04009 [cs]. https://doi.org/10.48550/arXiv.2103.04009

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017, April 24). *Speed/accuracy trade-offs for modern convolutional object detectors*. arXiv: 1611.10012 [cs]. https://doi.org/10.48550/arXiv.1611.10012

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (n.d.). *TensorFlow 2 Detection Model Zoo*. GitHub. Retrieved September 1, 2023, from https://github.com/tensorflow/models/tree/master/research/object_detection

Huang, X., Xu, D., Li, Z., & Wang, C. (2019, December 27). *Translating multispectral imagery to nighttime imagery via conditional generative adversarial networks*. arXiv: 2001.05848 [cs, eess, stat]. https://doi.org/10.48550/arXiv.2001.05848

Huck, J. J., Perkins, C., Haworth, B. T., Moro, E. B., & Nirmalan, M. (2021). Centaur VGI: A Hybrid Human–Machine Approach to Address Global Inequalities in Map Coverage. *Annals of the American Association of Geographers*, *111*(1), 231–251. https://doi.org/10.1080/24694452.2020.1768822

Import/Guidelines OSM Wiki. (2023, July 15). Retrieved July 31, 2023, from https://wiki.openstreetmap.org/wiki/Import/Guidelines

Janakiramaiah, B., Kalyani, G., Karuna, A., Prasad, L. V. N., & Krishna, M. (2023). Military object detection in defense using multi-level capsule networks. *Soft Computing*, *27*(2), 1045–1059. https://doi.org/10.1007/s00500-021-05912-0

Janowicz, K., Gao, S., McKenzie, G., Hu, Y., & Bhaduri, B. (2020). GeoAI: Spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *International Journal of Geographical Information Science*, *34*(4), 625–636. https://doi.org/10.1080/13658816.2019.1684500

Jiang, J., Sun, H., Liu, X., & Ma, J. (2020). Learning Spatial-Spectral Prior for Super-Resolution of Hyperspectral Imagery. *IEEE Transactions on Computational Imaging*, *6*, 1082–1096. https://doi.org/10.1109/TCI.2020.2996075

JSON. (2023, August 23). In *Wikipedia*. Retrieved August 26, 2023, from https://en.wikipedia.org/w/index.php?title=JSON&oldid=1171764242 Page Version ID: 1171764242

Kadow, C., Hall, D. M., & Ulbrich, U. (2020). Artificial intelligence reconstructs missing climate information. *Nature Geoscience*, *13*(6), 408–413. https://doi.org/10.1038/s41561-020-0582-5

Kaiser, P., Wegner, J. D., Lucchi, A., Jaggi, M., Hofmann, T., & Schindler, K. (2017). Learning Aerial Image Segmentation from Online Maps. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(11), 6054–6068. https://doi.org/10.1109/TGRS.2017.2719738

Kampffmeyer, M., Salberg, A.-B., & Jenssen, R. (2016). Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 680–688. https://doi.org/10.1109/CVPRW.2016.90

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., & Darrell, T. (2019, October 21). *Few-shot Object Detection via Feature Reweighting*. arXiv: 1812.01866 [cs]. https://doi.org/10.48550/arXiv.1812.01866

Kang, Y., Gao, S., & Roth, R. E. (2019). Transferring Multiscale Map Styles Using Generative Adversarial Networks. *International Journal of Cartography*, *5*(2-3), 115–141. https://doi.org/10.1080/23729333.2019.1615729

Khan, N., Chaudhuri, U., Banerjee, B., & Chaudhuri, S. (2019). Graph convolutional network for multi-label VHR remote sensing scene recognition. *Neurocomputing*, *357*, 36–46. https://doi.org/10.1016/j.neucom.2019.05.024

Klingner, M., Termöhlen, J.-A., Mikolajczyk, J., & Fingscheidt, T. (2020, July 21). *Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance*. arXiv: 2007.06936 [cs]. https://doi.org/10.48550/arXiv.2007.06936

Kumar, A., Abhishek, K., Kumar Singh, A., Nerurkar, P., Chandane, M., Bhirud, S., Patel, D., & Busnel, Y. (2021). Multilabel classification of remote sensed satellite imagery. *Transactions on Emerging Telecommunications Technologies*, *32*(7), e3988. https://doi.org/10.1002/ett.3988

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, H., & Zipf, A. (2022). A CONCEPTUAL MODEL FOR CONVERTING OPENSTREETMAP CONTRIBUTION TO GEOSPATIAL MACHINE LEARNING TRAINING DATA. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B4-2022*, 253–259. https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-253-2022

Li, H., Herfort, B., Huang, W., Zia, M., & Zipf, A. (2020). Exploration of OpenStreetMap missing built-up areas using twitter hierarchical clustering and deep learning in Mozambique. *ISPRS Journal of Photogrammetry and Remote Sensing*, *166*, 41–51.

Li, H., Herfort, B., Lautenbach, S., Chen, J., & Zipf, A. (2022). Improving OpenStreetMap missing building detection using few-shot transfer learning in sub-Saharan Africa. *Transactions in GIS*, *26*(8), 3125–3146. https://doi.org/10.1111/tgis.12941

Li, H., Wang, J., Zollner, J. M., & Mai, G. (2023). Rethink Geographical Generalizability with Unsupervised Self-Attention Model Ensemble: A Case Study of OpenStreetMap Missing Building Detection in Africa.

Li, H., Yuan, Z., Dax, G., Kong, G., Fan, H., Zipf, A., & Werner, M. (2023, July 5). *Semi-supervised Learning from Street-View Images and OpenStreetMap for Automatic Building Height Estimation*. arXiv: 2307.02574 [cs]. https://doi.org/10.48550/arXiv.2307.02574

Li, H., Zech, J., Hong, D., Ghamisi, P., Schultz, M., & Zipf, A. (2022). Leveraging OpenStreetMap and Multimodal Remote Sensing Data with Joint Deep Learning for Wastewater Treatment Plants Detection. *International Journal of Applied Earth Observation and Geoinformation*, *110*, 102804. https://doi.org/10.1016/j.jag.2022.102804

Li, J., Hong, D., Gao, L., Yao, J., Zheng, K., Zhang, B., & Chanussot, J. (2022). Deep learning in multimodal remote sensing data fusion: A comprehensive review. *International Journal of Applied Earth Observation and Geoinformation*, *112*, 102926. https://doi.org/10.1016/j.jag.2022.102926

Li, W. (2020). GeoAI: Where machine learning and big data converge in GIScience. *Journal of Spatial Information Science*, (20), 71–77. Retrieved August 1, 2023, from https://josis.org/index.php/josis/article/view/116

Li, W., & Hsu, C.-Y. (2020). Automated terrain feature identification from remote sensing imagery: A deep learning approach. *International Journal of Geographical Information Science*, *34*(4), 637–660. https://doi.org/10.1080/13658816.2018.1542697

Li, W., & Hsu, C.-Y. (2022). GeoAI for Large-Scale Image Analysis and Machine Vision: Recent Progress of Artificial Intelligence in Geography. *ISPRS International Journal of Geo-Information*, *11*(7), 385. https://doi.org/10.3390/ijgi11070385

Li, X., Deng, J., & Fang, Y. (2022). Few-Shot Object Detection on Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–14. https://doi.org/10.1109/TGRS.2021.3051383

Li, Y., Chen, R., Zhang, Y., Zhang, M., & Chen, L. (2020). Multi-Label Remote Sensing Image Scene Classification by Combining a Convolutional Neural Network and a Graph Neural Network. *Remote Sensing*, *12*(23), 4003. https://doi.org/10.3390/rs12234003

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017, April 19). *Feature Pyramid Networks for Object Detection*. arXiv: 1612.03144 [cs]. https://doi.org/10.48550/arXiv.1612.03144

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018, February 7). *Focal Loss for Dense Object Detection*. arXiv: 1708.02002 [cs]. https://doi.org/10.48550/arXiv.1708.02002

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015, February 20). *Microsoft COCO: Common Objects in Context*. arXiv: 1405.0312 [cs]. https://doi.org/10.48550/arXiv.1405.0312

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. https://doi.org/10.1007/978-3-319-46448-0_2

Liu, Y., Liu, X., Gao, S., Gong, L., Kang, C., Zhi, Y., Chi, G., & Shi, L. (2015). Social Sensing: A New Approach to Understanding Our Socioeconomic Environments. *Annals of the Association of American Geographers*, *105*(3), 512–530. https://doi.org/10.1080/00045608.2015.1018773

Mac Aodha, O., Cole, E., & Perona, P. (2019, October 28). *Presence-Only Geographical Priors for Fine-Grained Image Classification*. arXiv: 1906.05272 [cs]. https://doi.org/10.48550/arXiv.1906.05272

Mai, G., Cundy, C., Choi, K., Hu, Y., Lao, N., & Ermon, S. (2022). Towards a foundation model for geospatial artificial intelligence (vision paper). *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–4. https://doi.org/10.1145/3557915.3561043

Mai, G., Huang, W., Sun, J., Song, S., Mishra, D., Liu, N., Gao, S., Liu, T., Cong, G., Hu, Y., Cundy, C., Li, Z., Zhu, R., & Lao, N. (2023, April 13). *On the Opportunities and Challenges of Foundation Models for Geospatial Artificial Intelligence*. arXiv: 2304.06798 [cs]. https://doi.org/10.48550/arXiv.2304.06798

Mai, G., Janowicz, K., Hu, Y., Gao, S., Yan, B., Zhu, R., Cai, L., & Lao, N. (2022). A Review of Location Encoding for GeoAI: Methods and Applications. *International Journal of Geographical Information Science*, *36*(4), 639–673. https://doi.org/10.1080/13658816.2021.2004602

Mai, G., Janowicz, K., Yan, B., Zhu, R., Cai, L., & Lao, N. (2020, February 15). *Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells*. arXiv: 2003.00824 [cs, stat]. https://doi.org/10.48550/arXiv.2003.00824

Mai, G., Lao, N., He, Y., Song, J., & Ermon, S. (2023, May 8). *CSP: Self-Supervised Contrastive Spatial Pre-Training for Geospatial-Visual Representations*. arXiv: 2305.01118 [cs]. https://doi.org/10.48550/arXiv.2305.01118

Mao, H., Hu, Y., Kar, B., Gao, S., & McKenzie, G. (2018). GeoAI 2017 workshop report: The 1st ACM SIGSPATIAL International Workshop on GeoAI: @AI and Deep Learning for Geographic Knowledge Discovery: Redondo Beach, CA, USA - November 7, 2016. *SIGSPATIAL Special*, *9*(3), 25. https://doi.org/10.1145/3178392.3178408

Mercator projection. (2023, August 28). In *Wikipedia*. Retrieved August 31, 2023, from https://en.wikipedia.org/w/index.php?title=Mercator_projection&oldid=1172585098
Page Version ID: 1172585098

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., & Houlsby, N. (2022, July 20). *Simple Open-Vocabulary Object Detection with Vision Transformers*. arXiv: 2205.06230 [cs]. https://doi.org/10.48550/arXiv.2205.06230

Minghini, M., Coetzee, S., Grinberger, A. Y., Yeboah, G., Juhász, L., & Mooney, P. (2020). OpenStreetMap research in the COVID-19 era.

Minghini, M., & Frassinelli, F. (2019). OpenStreetMap history for intrinsic quality assessment: Is OSM up-to-date? *Open Geospatial Data, Software and Standards*, *4*(1), 9. https://doi.org/10.1186/s40965-019-0067-x

Mnih, V., & Hinton, G. (2012). Learning to label aerial images from noisy data. *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 203–210.

Mohajerani, S., & Saeedi, P. (2021). Cloud and Cloud Shadow Segmentation for Remote Sensing Imagery Via Filtered Jaccard Loss Function and Parametric Augmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 4254–4266. https://doi.org/10.1109/JSTARS.2021.3070786

Mohanrajan, S. N., & Loganathan, A. (2022). Novel Vision Transformer–Based Bi-LSTM Model for LU/LC Prediction—Javadi Hills, India. *Applied Sciences*, *12*(13), 6387. https://doi.org/10.3390/app12136387

Mou, L., & Zhu, X. X. (2018, February 27). *IM2HEIGHT: Height Estimation from Single Monocular Imagery via Fully Residual Convolutional-Deconvolutional*

Network. arXiv: 1802.10249 [cs]. https://doi.org/10.48550/arXiv.1802.10249

Neumann, A. (2008). Web Mapping and Web Cartography. In S. Shekhar & H. Xiong (Eds.), *Encyclopedia of GIS* (pp. 1261–1269). Springer US. https://doi.org/10.1007/978-0-387-35973-1_1485

Open Geospatial Consortium. (2023, August 9). In *Wikipedia*. Retrieved August 26, 2023, from https://en.wikipedia.org/w/index.php?title=Open_Geospatial_Consortium&oldid=1169451420
Page Version ID: 1169451420

OpenStreetMap Wiki. (2019, July 23). *Rapid - OpenStreetMap Wiki*. Retrieved July 31, 2023, from https://wiki.openstreetmap.org/wiki/Rapid

OpenStreetMap Wiki. (2023, July 4). *2023 Turkey Earthquakes - OpenStreetMap Wiki*. Retrieved July 25, 2023, from https://wiki.openstreetmap.org/wiki/2023_Turkey_Earthquakes

O'Reilly, T. (2005, September 30). *What Is Web 2.0*. Retrieved July 19, 2023, from https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

Osco, L. P., dos Santos de Arruda, M., Gonçalves, D. N., Dias, A., Batistoti, J., de Souza, M., Gomes, F. D. G., Ramos, A. P. M., de Castro Jorge, L. A., Liesenberg, V., Li, J., Ma, L., Marcato, J., & Gonçalves, W. N. (2021). A CNN approach to simultaneously count plants and detect plantation-rows from UAV imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *174*, 1–17. https://doi.org/10.1016/j.isprsjprs.2021.01.024

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & da Silva, E. A. B. (2021). A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics*, *10*(3), 279. https://doi.org/10.3390/electronics10030279

Pahl, M.-O., & Loipfinger, M. (2018). Machine learning as a reusable microservice. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 1–7. https://doi.org/10.1109/NOMS.2018.8406165

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Peng, D., Zhang, Y., & Guan, H. (2019). End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++. *Remote Sensing*, *11*(11), 1382. https://doi.org/10.3390/rs11111382

Pisl, J., Li, H., Lautenbach, S., Herfort, B., & Zipf, A. (2021). Detecting OpenStreetMap missing buildings by transferring pre-trained deep neural networks. *AGILE: GIScience Series*, *2*, 1–7. https://doi.org/10.5194/agile-giss-2-39-2021

Poornima, S., & Pushpalatha, M. (2019). Drought prediction based on SPI and SPEI with varying timescales using LSTM recurrent neural network. *Soft*

Computing, *23*(18), 8399–8412. https://doi.org/10.1007/s00500-019-04120-1

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017, June 7). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. arXiv: 1706.02413 [cs]. https://doi.org/10.48550/arXiv.1706.02413

Qin, M., Hu, L., Du, Z., Gao, Y., Qin, L., Zhang, F., & Liu, R. (2020). Achieving Higher Resolution Lake Area from Remote Sensing Images Through an Unsupervised Deep Learning Super-Resolution Method. *Remote Sensing*, *12*(12), 1937. https://doi.org/10.3390/rs12121937

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, February 26). *Learning Transferable Visual Models From Natural Language Supervision*. arXiv: 2103.00020 [cs]. https://doi.org/10.48550/arXiv.2103.00020

rbrundritt. (2022, June 8). *Bing Maps Tile System - Bing Maps*. Retrieved August 31, 2023, from https://learn.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, May 9). *You Only Look Once: Unified, Real-Time Object Detection*. arXiv: 1506.02640 [cs]. https://doi.org/10.48550/arXiv.1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016, January 6). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv: 1506.01497 [cs]. https://doi.org/10.48550/arXiv.1506.01497

Representational state transfer. (2023, August 26). In *Wikipedia*. Retrieved August 26, 2023, from https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=1172295675
Page Version ID: 1172295675

Ribeiro, M., Grolinger, K., & Capretz, M. A. (2015). MLaaS: Machine Learning as a Service. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 896–902. https://doi.org/10.1109/ICMLA.2015.152

Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.

Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020, April 28). *Meta-Learning for Few-Shot Land Cover Classification*. arXiv: 2004.13390 [cs, stat]. https://doi.org/10.48550/arXiv.2004.13390

Sefrin, O., Riese, F. M., & Keller, S. (2021). Deep Learning for Land Cover Change Detection. *Remote Sensing*, *13*(1), 78. https://doi.org/10.3390/rs13010078

Shao, J., Du, B., Wu, C., & Zhang, L. (2019). Tracking Objects From Satellite Videos: A Velocity Feature Based Correlation Filter. *IEEE Transactions on*

Geoscience and Remote Sensing, *57*(10), 7860–7871. https://doi.org/10.1109/TGRS.2019.2916953

Sherley, E. F., Kumar, A., Revathy, & Divyashree. (2021). Detection and Prediction of Land Use and Land Cover Changes Using Deep Learning. In S. C. Satapathy, V. Bhateja, M. Ramakrishna Murty, N. Gia Nhu, & Jayasri Kotti (Eds.), *Communication Software and Networks* (pp. 359–367). Springer. https://doi.org/10.1007/978-981-15-5397-4_37

Shi, Q., Liu, M., Li, S., Liu, X., Wang, F., & Zhang, L. (2022). A Deeply Supervised Attention Metric-Based Network and an Open Aerial Image Dataset for Remote Sensing Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–16. https://doi.org/10.1109/TGRS.2021.3085870

Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Sirko, W., Kashubin, S., Ritter, M., Annkah, A., Bouchareb, Y. S. E., Dauphin, Y., Keysers, D., Neumann, M., Cisse, M., & Quinn, J. (2021, July 29). *Continental-Scale Building Detection from High Resolution Satellite Imagery*. arXiv: 2107.12283 [cs]. https://doi.org/10.48550/arXiv.2107.12283

Soden, R., & Palen, L. (2014). From Crowdsourced Mapping to Community Mapping: The Post-earthquake Work of OpenStreetMap Haiti. In C. Rossitto, L. Ciolfi, D. Martin, & B. Conein (Eds.), *COOP 2014 - Proceedings of the 11th International Conference on the Design of Cooperative Systems, 27-30 May 2014, Nice (France)* (pp. 311–326). Springer International Publishing. https://doi.org/10.1007/978-3-319-06498-7_19

Solovyev, R., Wang, W., & Gabruseva, T. (2021). Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, *107*, 104117. https://doi.org/10.1016/j.imavis.2021.104117

Srivastava, S., Volpi, M., & Tuia, D. (2017). Joint height estimation and semantic labeling of monocular aerial images with CNNS. *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 5173–5176. https://doi.org/10.1109/IGARSS.2017.8128167

Stewart, A. J., Robinson, C., Corley, I. A., Ortiz, A., Ferres, J. M. L., & Banerjee, A. (2022). TorchGeo: Deep learning with geospatial data. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–12. https://doi.org/10.1145/3557915.3560953

Sui, D., Goodchild, M., & Elwood, S. (2013). Volunteered Geographic Information, the Exaflood, and the Growing Digital Divide. In D. Sui, S. Elwood, & M. Goodchild (Eds.), *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice* (pp. 1–12). Springer Netherlands. https://doi.org/10.1007/978-94-007-4587-2_1

Sui, D. Z. (2004). Tobler's First Law of Geography: A Big Idea for a Small World? *Annals of the Association of American Geographers*, *94*(2), 269–277. https://doi.org/10.1111/j.1467-8306.2004.09402003.x

Sun, X., Wang, P., Yan, Z., Xu, F., Wang, R., Diao, W., Chen, J., Li, J., Feng, Y., Xu, T., Weinmann, M., Hinz, S., Wang, C., & Fu, K. (2021, March 24). *FAIR1M: A Benchmark Dataset for Fine-grained Object Recognition in High-Resolution Remote Sensing Imagery*. arXiv: 2103.05569 [cs]. https://doi.org/10.48550/arXiv.2103.05569

Tiecke, T. G., Liu, X., Zhang, A., Gros, A., Li, N., Yetman, G., Kilic, T., Murray, S., Blankespoor, B., Prydz, E. B., & Dang, H.-A. H. (2017, December 15). *Mapping the world population one building at a time*. arXiv: 1712.05839 [cs]. https://doi.org/10.48550/arXiv.1712.05839

Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, *46*, 234–240. https://doi.org/10.2307/143141

Touya, G., Zhang, X., & Lokhat, I. (2019). Is deep learning the new agent for map generalization? *International Journal of Cartography*, *5*(2-3), 142–157. https://doi.org/10.1080/23729333.2019.1613071

Vargas-Muñoz, J. E., Lobry, S., Falcão, A. X., & Tuia, D. (2019). Correcting rural building annotations in OpenStreetMap using convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *147*, 283–293. https://doi.org/10.1016/j.isprsjprs.2018.11.010

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, December 5). *Attention Is All You Need* (5). arXiv: 1706.03762 [cs]. https://doi.org/10.48550/arXiv.1706.03762

Veenendaal, B., Brovelli, M. A., & Li, S. (2017). Review of Web Mapping: Eras, Trends and Directions. *ISPRS International Journal of Geo-Information*, *6*(10), 317. https://doi.org/10.3390/ijgi6100317

Wan, J., Liu, J., Ren, G., Guo, Y., Yu, D., & Hu, Q. (2016). Day-Ahead Prediction of Wind Speed with Deep Feature Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, *30*(05), 1650011. https://doi.org/10.1142/S0218001416500117

Wang, J. (n.d.). Towards GeoAI as a Containerized Microservice.

Wang, L., Geng, X., Ma, X., Liu, F., & Yang, Q. (2018, May 19). *Cross-City Transfer Learning for Deep Spatio-Temporal Prediction*. arXiv: 1802.00386 [cs]. https://doi.org/10.48550/arXiv.1802.00386

Wang, Q., Zhang, X., Chen, G., Dai, F., Gong, Y., & Zhu, K. (2018). Change detection based on Faster R-CNN for high-resolution remote sensing images. *Remote Sensing Letters*, *9*(10), 923–932. https://doi.org/10.1080/2150704X.2018.1492172

Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., & Yu, F. (2020, March 15). *Frustratingly Simple Few-Shot Object Detection*. arXiv: 2003.06957 [cs]. https://doi.org/10.48550/arXiv.2003.06957

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, *53*(3), 63:1–63:34. https://doi.org/10.1145/3386252

Werner, M., & Li, H. (2022). AtlasHDF: An efficient big data framework for GeoAI. *Proceedings of the 10th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, 1–7. https://doi.org/10.1145/3557917.3567615

Wu, M., Jin, X., Jiang, Q., Lee, S.-j., Liang, W., Lin, G., & Yao, S. (2021). Remote sensing image colorization using symmetrical multi-scale DCGAN in YUV color space. *The Visual Computer*, *37*(7), 1707–1729. https://doi.org/10.1007/s00371-020-01933-2

Wu, Z., Li, H., & Zipf, A. (2020). From Historical OpenStreetMap data to customized training samples for geospatial machine learning. *Proceedings of the Academic Track at the State of the Map 2020 Online Conference*.

Xie, Y., Cai, J., Bhojwani, R., Shekhar, S., & Knight, J. (2020). A locally-constrained YOLO framework for detecting small and densely-distributed building footprints. *International Journal of Geographical Information Science*, *34*(4), 777–801. https://doi.org/10.1080/13658816.2019.1624761

Xie, Y., He, E., Jia, X., Bao, H., Zhou, X., Ghosh, R., & Ravirathinam, P. (2021). A Statistically-Guided Deep Network Transformation and Moderation Framework for Data with Spatial Heterogeneity. *2021 IEEE International Conference on Data Mining (ICDM)*, 767–776. https://doi.org/10.1109/ICDM51629.2021.00088

Xu, Y., Piao, Z., & Gao, S. (2018). Encoding Crowd Interaction with Deep Neural Network for Pedestrian Trajectory Prediction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5275–5284. https://doi.org/10.1109/CVPR.2018.00553

Xuan, S., Li, S., Zhao, Z., Zhou, Z., Zhang, W., Tan, H., Xia, G., & Gu, Y. (2021). Rotation adaptive correlation filter for moving object tracking in satellite videos. *Neurocomputing*, *438*, 94–106. https://doi.org/10.1016/j.neucom.2021.01.058

Yan, X., Ai, T., Yang, M., & Tong, X. (2021). Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science*, *35*(3), 490–512. https://doi.org/10.1080/13658816.2020.1768260

Yang, W., Li, X., Yang, B., & Fu, Y. (2020). A Novel Stereo Matching Algorithm for Digital Surface Model (DSM) Generation in Water Areas. *Remote Sensing*, *12*(5), 870. https://doi.org/10.3390/rs12050870

Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 270–279. https://doi.org/10.1145/1869790.1869829

Yin, Y., Liu, Z., Zhang, Y., Wang, S., Shah, R. R., & Zimmermann, R. (2019). GPS2Vec: Towards Generating Worldwide GPS Embeddings. *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 416–419. https://doi.org/10.1145/3347146.3359067

Zamani Joharestani, M., Cao, C., Ni, X., Bashir, B., & Talebiesfandarani, S. (2019). PM2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data. *Atmosphere*, *10*(7), 373. https://doi.org/10.3390/atmos10070373

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115. https://doi.org/10.1145/3446776

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018, January 3). *Single-Shot Refinement Neural Network for Object Detection*. arXiv: 1711.06897 [cs]. https://doi.org/10.48550/arXiv.1711.06897

Zhang, Y., Jatowt, A., & Tanaka, K. (2017). Is Tofu the Cheese of Asia? Searching for Corresponding Objects across Geographical Areas. *Proceedings of the 26th International Conference on World Wide Web Companion*, 1033–1042. https://doi.org/10.1145/3041021.3055132

Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., & Ling, H. (2019, January 6). *M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network*. arXiv: 1811.04533 [cs]. https://doi.org/10.48550/arXiv.1811.04533

Zhou, L., Yan, H., Shan, Y., Zheng, C., Liu, Y., Zuo, X., & Qiao, B. (2021). Aircraft Detection for Remote Sensing Images Based on Deep Convolutional Neural Networks. *Journal of Electrical and Computer Engineering*, *2021*, e4685644. https://doi.org/10.1155/2021/4685644

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Zook, M., Graham, M., Shelton, T., & Gorman, S. (2010). Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake. *World Medical & Health Policy*, *2*(2), 7–33.

A. Appendix

All the code and test data used in this thesis are shown here:

The code repository of the full training workflow including data preparation, FSTL, and GWME can be accessed via https://github.com/Wjppppp/building-detection.git.

The code repository and test images of the proposed GWME method can be accessed via https://github.com/tum-bgd/GWME.git.

The code repository of the building detection web mapping application can be accessed via https://github.com/Wjppppp/missing-osm.git.