Universidade Estadual de Campinas
Instituto de Computação

John Edgar Vargas Muñoz

Machine learning applied to open geographical data

Aprendizado de máquina aplicado a dados geográficos abertos

CAMPINAS
2019

John Edgar Vargas Muñoz

Machine learning applied to open geographical data

Aprendizado de máquina aplicado a dados geográficos abertos

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Alexandre Xavier Falcão

Este exemplar corresponde à versão final da Tese defendida por John Edgar Vargas Muñoz e orientada pelo Prof. Dr. Alexandre Xavier Falcão.

CAMPINAS
2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

V426m

Vargas Muñoz, John Edgar, 1991-
Machine learning applied to open geographical data / John Edgar Vargas Muñoz. – Campinas, SP : [s.n.], 2019.

Orientador: Alexandre Xavier Falcão.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Sensoriamento remoto. 3. Reconhecimento de padrões. 4. Análise de imagens. 5. Processamento de imagens. I. Falcão, Alexandre Xavier, 1966-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Aprendizado de máquina aplicado a dados geográficos abertos
Palavras-chave em inglês:
Machine learning
Remote sensing
Pattern recognition
Image analysis
Image processing
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Alexandre Xavier Falcão [Orientador]
Raul Queiroz Feitosa
Clarimar José Coelho
Stanley Robson de Medeiros Oliveira
Rubens Augusto Camargo Lamparelli
Data de defesa: 04-10-2019
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-7483-8963
- Currículo Lattes do autor: http://lattes.cnpq.br/2188665254648890

Universidade Estadual de Campinas
Instituto de Computação

John Edgar Vargas Muñoz

Machine learning applied to open geographical data

Aprendizado de máquina aplicado a dados geográficos abertos

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão
  IC/Unicamp
- Prof. Dr. Raul Queiroz Feitosa
  DEE/PUC-Rio
- Prof. Dr. Clarimar José Coelho
  ECEC/PUC-Goiás
- Prof. Dr. Stanley Robson de Medeiros Oliveira
  Embrapa
- Prof. Dr. Rubens Augusto Camargo Lamparelli
  NIPE/Unicamp

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 04 de outubro de 2019

Acknowledgements

I am sincerely grateful to Prof. Alexandre Falcão and Prof. Devis Tuia, who advised me during my research work.

I thank my family, especially my father and my mother.

I thank my colleagues and friends from the University of Campinas and Wageningen University & Research.

I thank Shivangi Srivastava, Samuel Martins, Prof. Jefersson dos Santos, Diego Marcos, Prof. Ananda Chowdhury, Sylvain Lobry, and Alan Peixinho for their valuable collaborations which made me a better professional and person.

Finally, I thank the financial support provided by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, Processes #2016/14760-5, #2017/10086-0, and #2014/12236-1).

Resumo

Dados geográficos são utilizados em várias aplicações, tais como mapeamento, navegação e planificação urbana. Em particular, serviços de mapeamento são frequentemente utilizados e requerem informação geográfica atualizada. No entanto, devido a limitações orçamentárias, mapas oficiais (e.g. governamentias) sofrem de imprecisões temporais e de completude. Neste contexto projetos crowdsourcing, assim como os sistemas de informação geográfica voluntária, surgiram como uma alternativa para obter dados geográficos atualizados. OpenStreetMap (OSM) é um dos maiores projetos desse tipo com milhões de usuários (consumidores e produtores de informação) em todo o mundo e os dados coletados pelo OSM estão disponíveis gratuitamente. Uma desvantagem do OSM é o fato de poder ser editado por voluntários com diferentes habilidades de anotação, o que torna a qualidade das anotações heterogêneas em diferentes regiões geográficas. Apesar desse problema de qualidade, os dados do OSM têm sido amplamente utilizados em várias aplicações, como por exemplo no mapeamento de uso da terra. Por outro lado, é crucial melhorar a qualidade dos dados em OSM de forma que as aplicações que dependam de informações precisas, por exemplo, roteamento de carros, se tornem mais eficazes. Nesta tese, revisamos e propomos métodos baseados em aprendizado de máquina para melhorar a qualidade dos dados em OSM. Apresentamos métodos automáticos e interativos focados na melhoria dos dados em OSM para fins humanitários. Os métodos apresentados podem corrigir as anotações do OSM de edifícios em áreas rurais e permitem realizar a anotação eficiente de coqueiros a partir de imagens aéreas. O primeiro é útil na resposta a crises que afetam áreas vulneráveis, enquanto que o último é útil para monitoramento ambiental e avaliação pós-desastre. Nossa metodologia para correção automática das anotações de prédios rurais existentes em OSM consiste em três tarefas: correção de alinhamento, remoção de anotações incorretas e adição de anotações ausentes de construções. Esta metodologia obtém melhores resultados do que os métodos de segmentação semântica supervisionados e, mais importante, produz resultados vetoriais adequados para o processamento de dados geográficos. Dado que esta estratégia automática poderia não alcançar resultados precisos em algumas regiões, propomos uma abordagem interativa que reduz os esforços de humanos ao corrigir anotações de prédios rurais. Essa estratégia reduz drasticamente a quantidade de dados que os usuários precisam analisar, encontrando automaticamente a maioria dos erros de anotação existentes. A anotação de objetos de imagens aéreas é uma tarefa demorada, especialmente quando o número de objetos é grande. Assim, propomos uma metodologia na qual o processo de anotação é realizado em um espaço 2D, obtido da projeção do espaço de características das imagens. Esse método permite anotar com eficiência mais objetos do que o método tradicional de fotointerpretação, coletando amostras rotuladas mais eficazes para treinar um classificador para detecção de objetos.

Abstract

Geographical data are used in several applications, such as mapping, navigation, and urban planning. Particularly, mapping services are routinely used and require up-to-date geographical data. However, due to budget limitations, authoritative maps suffer from completeness and temporal inaccuracies. In this context, crowdsourcing projects, such as Volunteer Geographical Information (VGI) systems, have emerged as an alternative to obtain up-to-date geographical data. OpenStreetMap (OSM) is one of the largest VGI projects with millions of users (consumers and producers of information) around the world and the collected data in OSM are freely available. OSM is edited by volunteers with different annotation skills, which makes the annotation quality heterogeneous in different geographical regions. Despite these quality issues, OSM data have been extensively used in several applications (e.g., landuse mapping). On the other hand, it is crucial to improve the quality of the data in OSM such that applications that depend on accurate information become more effective (e.g., car routing). In this thesis, we review and propose methods based on machine learning to improve the quality of the data in OSM. We present automatic and interactive methods focused on improving OSM data for humanitarian purposes. The methods can correct the OSM annotations of building footprints in rural areas and can provide efficient annotation of coconut trees from aerial images. The former is helpful in the response to crises that affect vulnerable areas, while the later is useful for environmental monitoring and post-disaster assessment. Our methodology for automatic correction of the existing OSM annotations of rural buildings consists of three tasks: alignment correction, removal of incorrect annotations, and addition of missing building annotations. This methodology obtains better results than supervised semantic segmentation methods and, more importantly, it outputs vectorial footprints suitable for geographical data processing. Given that this automatic strategy could not attain accurate results in some regions, we propose an interactive approach which reduces the human efforts when correcting rural building annotations in OSM. This strategy drastically reduces the amount of data that the users need to analyze by automatically finding most of the existing annotation errors. The annotation of objects from aerial imagery is a time-consuming task, especially when the number of objects is high. Thus, we propose a methodology in which the annotation process is performed in a 2D space of projected image features. This method allows to efficiently annotate more objects than using traditional photointerpretation, collecting more effective labeled samples to train a classifier for object detection.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Geographical data describe objects in certain locations on the earth's surface and are usually represented in maps. Acquiring updated geographical data is crucial for several applications, such as mapping [26], urban planning [121], navigation [56], natural disaster management [85], and farming applications [103].

Keeping updated the data of mapping services is a challenging task. Frequently, authoritative maps suffer from temporal and completeness inaccuracies because of budget limitations. Thus, Volunteer Geographical Information (VGI) systems, like OpenStreetMap [1] (OSM), have emerged as an alternative approach collecting mapping information from volunteers around the world. However, the different expertise levels of the volunteers in annotating geographical data makes heterogeneus the quality of VGI systems. For instance, several research works have pointed out quality issues in OSM data, such as inaccurate positions of building footprints [141], missing buildings [54] and roads [36], and inaccurate tags of geographical objects [3].

In this thesis, we aim at better understanding how machine learning methods can be useful for improving the process of image annotation. We explore automatic and interactive approaches to efficiently create and correct image annotations. We focus our work on improving the annotations of geographical objects in OSM, by using machine learning methods applied to remote sensing imagery and OSM data.

Our proposed methodologies use information extracted automatically from aerial images (e.g., building segmentation) to identify and correct issues of the current annotations of geographical objects in OSM and also to perform more effective annotations of new objects. Thus, we propose techniques based on state-of-the-art computer vision methods, such as Convolutional Neural Networks (CNN).

We validated our proposed methods mainly on rural building annotations in OSM. Our motivation to work with OSM annotations of rural buildings is that a significant amount of buildings in rural areas are not mapped in any commercial mapping service (e.g., Google maps and Bing maps) and also not in open maps like OSM. One of the methods is validated for coconut tree annotations in OSM, which can be used in tasks like post-disaster assessment and environmental monitoring. However, most of the proposed methods in this thesis can also be applied to other types of geographical objects.

[1] https://www.openstreetmap.org

The thesis is divided into four parts. In the first part, we review recent works that use machine learning methods to improve OSM data quality. In this review, we also show how machine learning methods have used OSM data to solve problems from other domains. The second part presents a proposed methodology to automatically correct rural building annotations in OSM. This is done by correcting possible misalignments of building's annotations followed by the removal of wrong annotations and the addition of new annotations whenever needed. Methods for automatic annotation correction can not obtain the desired annotation quality level. Therefore, in the third part of the thesis, we present a interactive approach to correct rural building annotations in OSM. In this strategy, the computer selects a small set of regions in the image that most likely requires corrections, greatly reducing the amount of data that the user has to verify/correct. Finally, the fourth part of the thesis presents a strategy for interactive annotation of coconut trees in OSM, which can significantly reduce the human effort of traditional manual annotation over aerial imagery. This method extracts image features of the objects of interest and projects them into a 2D space for visualization. In 2D, several samples can be selected and annotated by a user at the same time, improving the efficiency of the image annotation system. The detailed structure of this thesis is presented in Section 1.5.

The next sections of this chapter are organized as follows. Section 1.1 defines the concept of Volunteer Geographical Information (VGI). Section 1.2 describes the OSM project and how geographical data are stored in OSM. Section 1.3 presents the types of quality issues in OSM data and gives examples of them. Section 1.4 presents the main research question of this thesis. Section 1.5 shows the structure of this thesis and Section 1.6 presents the accepted and submitted publications.

1.1 Volunteer Geographical Information

With the widespread use of web technologies, users became active producers of information. A good example of this phenomenon is the project Wikipedia, that has created a free encyclopedia with content edited by volunteers. Volunteer Geographical Information (VGI), term coined in [44], is a phenomenon in which geographical data are generated by users (e.g., OpenStreetMap and Wikimapia [2]). VGI projects have a great potential to obtain up-to-date information, because of their large number of users and the fact that several of them have local knowledge about the information registered in their databases. The data generated by VGI projects have been extensively used for applications such as routing and landuse mapping. Thus, VGI data quality became a very important problem and it has been studied in several works [39, 98].

1.2 OpenStreetMap

OpenStreetMap is a VGI project that started in 2004 and now it has more than 5 million registered users. Initially, most of the contributions were collected by GPS devices. Since 2007 satellite imagery, provided by Yahoo! and later by Bing maps, has been used to

[2] https://wikimapia.org/

collect new geographical data for OSM. Frequently, volunteers digitize buildings, roads, and other geographical objects guided by satellite imagery. After digitizing a geographical object, tags are added to each object with relevant information about it (e.g., address or landuse type).

In OSM, newly registered users can already add, remove, or modify geographical objects, and there are several web-based services that allow the download of raw and preprocessed OSM data [3]. Active OSM annotators participate in meetings called "Mapathons", where they annotate unmapped areas or improve the map in regions with existing annotations.

Geographical objects in OSM are represented by four types of data: nodes, ways, relations, and tags. Nodes represent particular locations in the earth's surface and are defined by latitude and longitude. For instance, nodes can represent bus stations, traffic signals or park benches. A way is a list of two or more nodes. It is used to represent objects that can be represented by polylines like rivers and roads. A way also represents the boundary of a polygonal object like buildings or parks. In this case, the last and first nodes of the list are the same. A relation is a data structure that stores the relationship among several OSM elements, for example, a set of roads can form an avenue or a set of polygons can define a multi-polygon that represents an area with holes. These basic three types of elements can have tags that give more information about the objects in OSM. A tag is a key-value pair that contains free format text. For example, a restaurant can be represented by a way with tag "amenity=restaurant" and a residential road can be represented by a way with tag "highway=residential". The keys can have any value, but there are conventions recommended by OSM [4]. This helps to standardize the annotation of common objects in OSM.

OSM data has been used by several companies, such as Apple, Facebook, and Foursquare, and several other mapping services, like Baidu maps (world map outside China), OpenTopoMap (topographical maps of Europe), and Mapbox.

1.3 Quality issues in OpenStreetMap

The applications that use OSM data in several domains depend on the quality of the data obtained from OSM. Thus, several works have studied the quality of OSM data [32, 113] by analyzing several aspects of the data. The International Organization for Standardization (ISO) recommend to use five quality measures for geographical data (standard 19138 created by the Technical Committee ISO/TC 211 [5]): completeness, consistency, positional accuracy, temporal accuracy, and thematic accuracy.

Completeness errors refer to the absence of data, for example, newly constructed buildings that are not present in the map. This measure can also refer to the excess of data as pointed out in [113]. For example, buildings that have been demolished should be removed from the map. Consistency refers to the coherence of the stored data, for example, similar polygons located very close to each other and representing the same

[3] https://export.hotosm.org/en/v3/
[4] https://wiki.openstreetmap.org/wiki/Map_Features/
[5] https://www.isotc211.org/

object is an example of consistency error. The positional accuracy is a measure of how close the location of an object is to the true location of the object. The thematic accuracy measures the correctness of the feature values of the geographical objects. For instance, a garden labeled as forest will represent a thematic classification error. Temporal accuracy refers to the correctness in the order of events related to a geographical object. For instance, if a building is extended and we have a database that stores all the versions of the polygons that delineate the buildings, then we expect to have an attribute that indicates which version of the polygon is the latest one.

Several works have pointed out that OSM present data quality issues [141, 32, 21, 36]. Most of these works have found positional, completeness, and thematic errors in common geographical objects like roads, buildings and other urban objects. For instance, positional and completeness inaccuracies of buildings are analyzed in [141] and [21], by comparing OSM data with governmental data. Completeness errors in road networks are analyzed in [36] and thematic classification errors of roads and other urban spaces are reported in several works [36, 31, 3].

1.4 Research question

In this thesis, we study mechanisms that use machine learning methods to make more efficient the process of image annotation. We focus our work in the task of obtaining up-to-date annotations of geographical objects, by processing remote sensing images. The main research question in this thesis is formulated as follow:

How can we improve the quality of open geographical databases with minimum human effort in data annotation?

This thesis has a particular focus on improving positional and completeness accuracy of OSM data by using machine learning methods and remote sensing imagery. Nevertheless, in Chapter 2 of this thesis, recent methods that propose solutions to different quality issues in OSM data are included in our literature review.

1.5 Structure of the thesis

This thesis is structured as follows:

In **Chapter 2**, we present a review of works that use machine learning methods to improve the quality of OSM data and also machine learning based techniques that use OSM data for applications from other domains. For instance, we review works that have been proposed to correct building annotations, find missing roads in road networks, autocomplete missing street names and correct wrong semantic tags. One of our works, presented in Chapter 3, is also briefly described in this review. We also present works that use machine learning and OSM data in other domains, such as landcover/landuse mapping and enable navigation/routing applications.

In **Chapter 3**, we present a methodology to correct rural building annotations in OSM, by analyzing remote sensing imagery [135]. We identified three main problems in OSM annotations of rural buildings: i) they are geometrically misaligned, ii) some annota-

tions do not correspond to buildings in the aerial imagery, and iii) some OSM annotations are missing for buildings that are present in the aerial imagery. We propose solutions for these three problems. The methodology starts by correcting alignment errors of the building annotations by using a Markov Random Field (MRF) method and removing polygons based on a building probability map obtained by a Convolutional Neural Network (CNN) model. The last step of the methodology is the prediction of new building annotations that are missing in OSM by using a CNN model that predicts building footprint with predefined shapes. Following the suggestions of the reviewers of the thesis we have better standardized the mathematical symbols used to describe the method proposed in Section 3.2.2. Since the content of Chapter 3 is already published we present the improved version of the aforementioned section in Appendix B.

Automatic approaches, including our method for automatic correction of rural building annotations can not obtain the desired quality level. Thus, in **Chapter 4** we present an interactive approach to correct annotations of rural buildings in OSM. It starts by correcting misalignment errors based on aerial imagery of the analyzed geographical locations. Then, a building probability map, as obtained by using an efficient building detection/segmentation method, is compared with the original OSM annotations to get a measure of annotation correctness. This measure is used to select just the regions in the map that require some correction. The user is asked to verify/correct the selected tiles during several iterations until a stopping criterion is met. In this process, the model is improved with the new annotated data. The experimental results show that the proposed method greatly reduces the amount of data that the volunteers of OSM need to verify/correct.

We also propose an alternative strategy to the traditional manual visual interpretation over a large remote sensing imagery, which can minimize the effort of human annotators. In **Chapter 5**, we present a general interactive method for the annotation of objects in aerial imagery [136]. In this approach, image features are extracted and projected onto a 2D space, by using the t-SNE (t-Distributed Stochastic Neighbor Embedding [78]) algorithm. Then, the samples are selected and annotated by a user in the 2D projection. This methodology facilitates the construction of effective training sets more efficiently than using the traditional manual annotation over the aerial imagery. The method is evaluated on annotations of coconut trees in OSM. Note that this method could be also applied to other geographical objects in OSM, as long as high spatial resolution imagery is available.

1.6 List of publications

This thesis includes the following submitted or published works.

- J. E. Vargas-Muñoz, S. Srivastava, and D. Tuia, A. X. Falcão OpenStreetMap, Machine learning and Remote sensing: challenges and opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 2019 (submitted)

- J. E. Vargas-Muñoz, S. Lobry, A. X. Falcão, and D. Tuia. Correcting rural building

annotations in OpenStreetMap using convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol 147, pages 283-293, 2019

- J. E. Vargas-Muñoz, and D. Tuia, A. X. Falcão Supporting digital humanitarians in OpenStreetMap: the role of deep learning and human-machine interaction. *International Journal of Geographical Information Science*, 2019 (submitted)

- J. E. Vargas-Muñoz, P. Zhou, A. X. Falcão, and Devis Tuia. Interactive coconut tree annotation using feature space projections. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019 (to appear)

During my doctoral studies, I have also worked in other projects that are not included in this thesis. I would like to mention especially two projects. In the first project, I worked on a framework for superpixel segmentation [133] that obtained state-of-the-art results. In the second project, I collaborated on a work to create a deep learning based method for multimodal landuse classification [119]. Below I list works that were published during my doctoral studies but were not included in this thesis.

- J. E. Vargas-Muñoz, A. S. Chowdhury, E. B. Alexandre, F. L. Galvão, P. A. V. Miranda, and A. X. Falcão. An iterative spanning forest framework for superpixel segmentation. *IEEE Transactions on Image Processing*, vol 28(7), pages 3477-3489, 2019

- S. Srivastava, J. E. Vargas-Muñoz, and D. Tuia. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. *Remote Sensing of Environment*, vol 228, pages 129-143, 2019

- S. Srivastava, J. E. Vargas Muñoz, S. Lobry, and D. Tuia. Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data. *International Journal of Geographical Information Science*, pages 1-20, 2018

- J. E. Vargas-Muñoz, D. Tuia, J. A. dos Santos, and A. X. Falcão. Post classification smoothing in sub-decimeter resolution images with semi-supervised label propagation. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3688-3691, 2017

Chapter 2

OpenStreetMap, Machine learning and Remote sensing: challenges and opportunities

Abstract

OpenStreetMap (OSM) is a community-based, freely available, editable map service that was created as an alternative to authoritative ones. Given that it is edited mainly by volunteers with different mapping skills, the completeness and quality of its annotations are heterogeneous across different geographical locations. Despite that, OSM has been widely used in several applications in Geoscience, Earth Observation, environmental sciences, and control. In this work, we present a review of recent methods based on machine learning to improve and use OSM data. Such methods aim either 1) at improving the coverage and quality of OSM layers, typically using GIS and remote sensing technologies, or 2) at using the existing OSM layers to train models based on image data to serve applications like navigation or landuse classification. We believe that OSM (as well as other sources of open land maps) can change the way we interpret remote sensing data and that the synergy with machine learning can scale participatory map making and its quality to the level needed to serve global and up-to-date land mapping.

2.1 Introduction

Mapping systems need to be reliable and frequently updated, which makes them costly to be maintained. Due to limited budget, authoritative maps are usually not fully updated at regular time intervals, and present temporal, spatial, and completeness inaccuracies. Recently, Volunteered Geographic Information (VGI) [44] has appeared as an alternative to authoritative map services. VGI collects mapping information from individuals, usually volunteers, and stores the information in a database which is often freely available. OpenStreetMap (OSM) is one of the most successful VGI projects. It started in 2004 and currently counts more than 5 million users [1] from different parts of the world. This gives

[1] https://osmstats.neis-one.org/

OSM the potential to provide updated mapping data at global scale.

OSM information is represented by four types of data: nodes, ways, relations, and tags, which are constantly edited by volunteers.

- A node is a location on the earth's surface, as determined by latitude and longitude coordinates. Points of Interests (POIs) such as bus stations can be represented by nodes.

- A way is a list of nodes forming polylines, that can represent road networks or areal objects (closed ways) like buildings.

- A relation represents the relationship among objects — e.g., a group of road segments can represent a bus route.

- A tag is a key-value pair that contains information of an object — e.g., a restaurant can be represented by a way with a tag "amenity=restaurant".

Although the OSM data is constantly under improvement, the completeness and quality of the annotations in different regions are affected by the number and mapping skills of the volunteers [94]. As reported in [49], the spatial coverage of OSM is heterogeneous in different geographical regions — i.e., urban areas are more regularly updated than rural areas. In road networks, missing roads are reported in [36] and inaccurate road tags are reported in [60]. The positional accuracy of building footprints in OSM sometimes requires corrections [141]. Several works in the literature have studied methods to assess the quality of OSM data by quantifying: data completeness [67], positional accuracy [32], semantic tag acccuracy [39], and topological consistency [98]. Some works focus on meta analysis of OSM, like the analysis of the contributors' activities [97, 6] and the quality assessment of the OSM data [113, 57].

Despite its completeness and quality issues, OSM has been widely used for several applications: e.g., validation of landcover maps [35]; landcover/landuse classification [117, 8, 119]; navigation and routing applications like traffic estimation [75] and pedestrian, bicycle, and wheelchair routing [112, 95]; detection of buildings and roads in aerial imagery [135, 93]; 3D city modelling [101]; indoor mapping [42]; and location-based map services [111].

In the recent years, the automation of tasks involving OSM data has received increasing attention: on one hand, research aiming at the improvement of the OSM layers has turned to Earth observation and machine learning algorithms as automatic ways to complete footprints in missing areas and verify specific annotations, mostly to ease and decrease the workload of volunteers. On the other hand, the rise of deep learning [145] has found in OSM a very valuable source of label information to train large models for image recognition from Earth Observation data, but also brought issues related to the quality, standardization, and completeness of the data used for training the models. In both cases, and in countless applications, the alliance of machine learning, Earth Observation, and OSM layers is proving to be an enabling factor for tackling global challenges in new ways.

To present the potential and opportunities of OSM for geoscience and remote sensing research, in this paper we present a review of methods based on machine learning to

improve and use OSM for applications in different domains. Note that, we do not intend to cover an exhaustive list of OSM applications, but we focus on the ones that involve machine learning techniques only.

In Section 2.2, we review methods based on machine learning to improve the completeness and quality of the OSM objects, such as building footprints, street networks, and points of interest. Section 2.3 reviews the works based on machine learning for applications like landcover/landuse classification, navigation, and fine-scale population estimation. Section 2.4 summarizes the discussion and draws promising future research areas at the interface of ML and VGI. Section 2.5 states the conclusion about this paper.

2.2 Improving OSM data with machine learning

The next sections describe methods based on machine learning to improve different types of OSM annotations: building footprints, street networks, semantic tags, and points of interest.

2.2.1 Building footprints

Detecting geometric mismatches

Buildings are one of the most widely annotated objects in OSM. Although the geometrical features and tags of the buildings in OSM are usually correct (especially in urban areas), there are cases where the building footprints are not accurately mapped by volunteers. Figure 2.1 presents examples of incomplete OSM building annotations in the cities of São Paulo and Amsterdam. An autoencoder neural network method is proposed in [141] to measure the accuracy of OSM building annotations with respect to official governmental data in the city of Toronto. The authors extract geometrical mismatch features to train an autoencoder neural network. Then, the reconstruction error predicted with the trained model is interpreted as a score that represents the quality of the annotation for a particular region. This method could be useful to import building footprints from other sources to OSM, since the proposed score could be used to identify where are the most mismatched regions that need to be carefully analyzed by annotators.

Detecting vandalism behavior

Some building annotations in OSM are intentionally edited with wrong geometries, such cases are known as digital vandalism. In order to identify vandalism in OSM data, the authors in [96] propose a rule-based system that analyzes temporal data of user annotations. For the case of buildings, in particular the authors in [124] propose a clustering-based method to detect vandalism of building annotations. This method extracts geometrical features from the OSM vectorial building data (e.g., perimeter, elongation, convexity, and compacity) and then finds groups in the feaure space to detect outliers, which are assumed to be possible vandalized building footprints.

(a) (b)

Figure 2.1: Examples of OSM building annotations (regions highlighted in blue) with completeness errors superimposed over Bing aerial imagery: a) incomplete annotation of buildings in the city of São Paulo, b) incomplete annotation of buildings in the city of Amsterdam.

Correct and create new annotations

In addition to the geometrical features of OSM building annotations, other works in the literature use aerial imagery to correct building annotations [135, 146]. In [135], the authors propose a methodology to correct rural building annotations in OSM. The paper points out three common problems in OSM building annotations in rural areas: they are geometrically misaligned (see Figure 2.2a), some annotations do not correspond to buildings in the updated aerial imagery (Figure 2.2b), and some OSM annotations are missing for buildings that are present in the updated aerial imagery (Figure 2.2c). The authors propose solutions for the three issues by using Markov Random Fields (MRF) to align annotations and remove annotations using a building probability map obtained by a Convolutional Neural Network (CNN). The last step of the method is the prediction of new building annotations that are missing by using a CNN model that predicts building footprint with predefined shape priors. The method in [146] aims at correcting OSM building annotations by using contour information from image segmentation of oblique images, acquired by Unmanned Aerial Vehicles (UAV). The paper uses contour information of multiple-view images and 3D building models to correct OSM building annotations. Some companies have also made great efforts to improve the geometrical completeness of OSM. Microsoft has used deep learning models to compute new building footprints by processing satellite imagery in the United States of America [2].

[2]https://blogs.bing.com/maps/2018-06/microsoft-releases-125-million-building-footprints-in-the-us-as-open-data

(a) (b) (c)

Figure 2.2: Examples of rural building annotations errors found in OSM superimposed over Bing aerial imagery (polygons in blue represent building annotations): a) misalignment annotation errors, b) annotations that do not match with any building, c) missing building annotations.

2.2.2 Street Network

The quality of street network information in OSM is crucial for several applications. The quality of street type tags and the completeness of the road network data are critical for route planning, while the street names are important to perform queries on the OSM map. As reported in [98, 11], road networks in OSM present heterogeneous quality and some completeness errors even in urban areas.

Correcting topology

In [36] the authors propose a method to improve the completeness of road networks. Specifically, they present a method based on machine learning to identify missing roads between candidate locations (two nodes of the OSM road network). The method extracts several features from each pair of candidates from OSM data such as connectivity, street type, and node degree in the OSM road network. The work shows empirically some evidence that the shortest path distance between two nodes in an OSM road network is correlated with the straight line distance. The extracted features are then used to train a Logistic Regression classifier to predict missing roads. The last step involves the pruning of some predictions to increase their precision.

Extracting roads from aerial images

The automatic extraction of road networks has also been attempted by analyzing remote sensing imagery. The work proposed in [24] performs per pixel classification using a CNN-based method and later obtain the centerline of the roads. The authors of [90] also use CNNs and centerline computation, but additionally they correct some gaps in the extracted road network by generating several possible missing road candidates and selecting some of them with the help of another CNN. Recently, more accurate results have been obtained by some methods [16, 137] that iteratively construct the road network

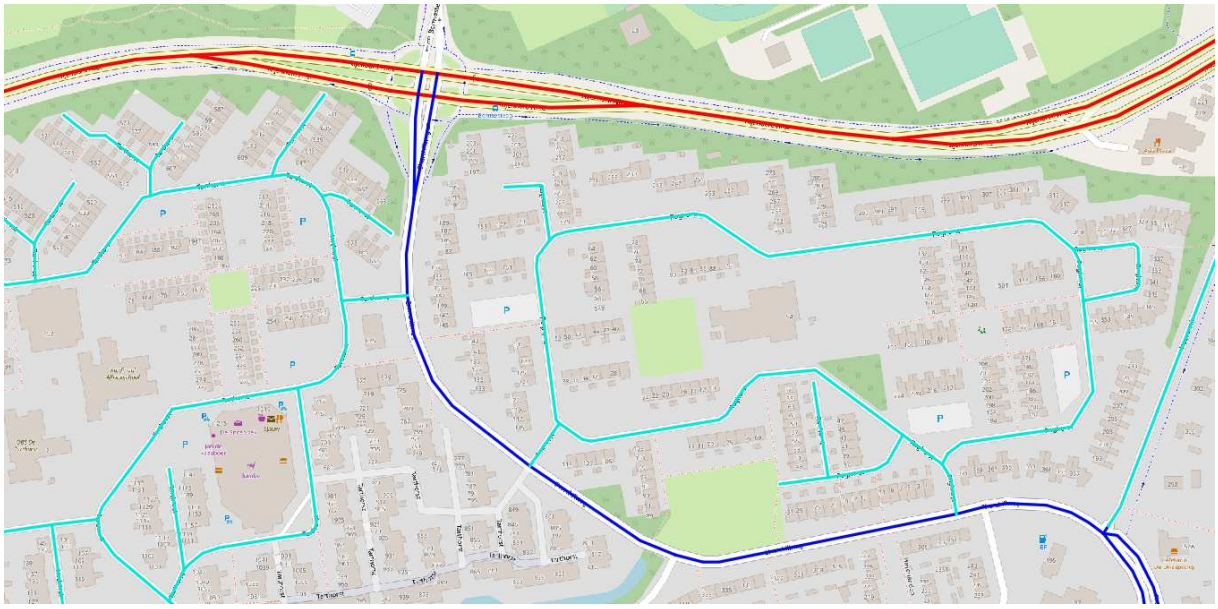


Figure 2.3: Three different street types (secondary, tertiary and residential colored in red, blue, and cyan, respectively) over the OSM map in the city of Wageningen, the Netherlands.

graph, by adding new edges to the graph. The authors in [15] propose a method that is as accurate as [16] but much more efficient. This CNN-based method output road directions for each pixel to create a road network.

The authors of [92] propose a CNN-based method that uses aerial imagery as well as ground-based pictures, in the city of Karlsruhe, Germany, to extract road networks and other objects such as parking spots and sidewalks that could be integrated into the OSM database. Facebook has also implemented deep learning methods to analyze satellite imagery but for the detection of new road networks in developing countries in the OSM map [3]

Assigning attributes: road types

Topological and geometrical features extracted from road networks can be very useful to predict street types. Figure 2.3 depicts three types of streets (i.e., secondary, tertiary, and residential streets) over the OSM map. It can be observed that residential street segments are small in length (distinctive geometrical feature) and that tertiary roads are connected with several residential streets (distinctive topological feature). A solution to fix incorrect street type tags in OSM is then presented in [58]. The authors extract topological features from OSM road network data to train a neural network classifier that predicts if a street is of type residential or pedestrian. This classifier can be useful to find inconsistent street type tags in OSM.

A multi-granular graph representation of street networks is proposed in [59]. This structure combines the primal (where nodes are road intersections) and dual representations (where nodes are fragments of roads) of road networks. This multi-granular rep-

[3] https://wiki.openstreetmap.org/wiki/AI-Assisted_Road_Tracing

resentation is used in [60] to extract features and train a Random Forest classifier that is able to classify streets to 21 different street categories in OSM. The method uses Bag of Words computed over geometrical and topological features of the analyzed streets and their neighbors. The method in [61] uses graphical models with geometrical and spatial features, such that the parameters of the model are learned by Structured Support Vector Machines (SSVM) [125]. More recently, the authors in [62] propose a multi-layer CRF (Conditional Random Field) model to perform hierarchical classification of street types into coarse and fine-grained classes.

Assigning attributes: structural

Detecting multilane roads is important to model traffic in urban areas. However, the tag 'lanes' that is used to specify how many traffic lanes a road has is usually empty in OSM. Therefore, some works [73, 142] have developed methods to detect multilane roads in OSM data by analyzing the polygons formed by the road network. This is possible because frequently multilane roads are digitized as multiple parallel roads with terminals in road crossings. In [73] the authors proposes to extract geometrical features (e.g., area, perimeter, and compactness) from polygons obtained from the road network and train SVM classifier to predict if a road has multiple lanes. After that a postprocessing step is performed, by using a region growing algorithm, to analyze if roads connected to the predicted multilane roads are also multilane roads. The method proposed in [142] also uses geometrical features to train a classifier, a Random Forest in this case, but the predictions are used to train a second Random Forest classifier that uses geometrical and topological features, such as the percentage of neighboring roads that were classified as a multilane road by the first classifier. In [91] the authors use an MRF and data extracted from remote sensing images (e.g., edge information, cars detected, and contextual information) to correct OSM road centerline locations and estimate the width of OSM roads.

Extracting road data from GPS locations

Sequences of GPS positions (also called tracking data) of users can be used to enrich OSM data. This information can be obtained by GPS locations of cars or applications installed in the volunteers' mobile devices. In [14], the authors propose to find errors in OSM data by analyzing patterns extracted from GPS positions and OSM mapping information. For instance, indoor corridors wrongly labeled as tunnels in OSM can be detected by verifying if the trajectory data comes from a pedestrian or a car.

In [13], the authors propose to use GPS positions and machine learning models for recomending the addition of new objects to OSM. For instance, GPS positions can help us predict a missing street in OSM by observing a linear shaped agglomeration of points at some location, where there is not a street in OSM but it is close to the road network. Analyzing the spatiotemporal GPS positions one can also identify, for example, that a road in OSM is a motorway because of the high velocity of objects derived from the GPS information data. The authors in [13] extract several features from GPS information, such as the density of nodes in the trajectory and speed of movement. Two types of classifiers

are trained: one to predict the geometry and the other to predict object attributes, such as motorway, bicycle lanes, one or two-way street. For geometry classification, the KNN classifier performed better than other algorithms like Logistic Regression and Random Forest. It is observed that the KNN model obtains poor results when detecting polygonal geometries, because of the lack of data along the boundaries of polygonal objects. For the classification of geographical object attributes, the Random Forest classifier outperforms the compared traditional machine learning methods.

In [69], the authors utilize GPS information for the reconstruction of road network geometries. The extracted road networks from updated GPS locations can be useful to improve the OSM map. One issue is obtain accurate road geometries since the GPS locations present errors in the range of 5-20 meters. This shortcoming can be mitigated by using multiple trajectories obtained from the same road segment. The authors observe that the accuracy of geometries increase with the number of GPS samples for each road segment.

2.2.3 Semantic tags

The annotation of a geographical object in OSM consists in the digitization of the object geometry (e.g., polygons, lines, or points) and also the attribution of a tags to it. OSM does not provide a rigorous classification system of the geographical objects. It just gives some recommendations and a set of predefined tags that can be used to annotate objects. Thus, the final label attributed to the OSM objects is defined by the volunteers based on their knowledge about the objects under annotation. This can lead to incorrect tag annotations since sometimes it is difficult for inexperienced users to differentiate between similar classes. The decision if a water body is a lake or a pond will depend on the knowledge of the volunteer and on his/her analysis of the aerial imagery or in-situ information.

Recommendation systems

OSM also allows the assignment of tag values that are not in the set of recommended OSM tag values, which detriments standardization of OSM data. The authors in [83] propose a method to identify recommended OSM tags that are equivalent to new tags created by annotators. The paper proposes an unsupervised method that uses tag usage statistics and geometry type information to compute a similarity measure between a given tag value and a set of common tag values recommended by OSM. This approach just uses OSM data, in contrast with the method proposed in [10], which uses external data, such as information from the OSM wiki website. Other works [65, 131] have implemented tag recommendation tools as plugins of JOSM [4], a widely used editor of OSM data. In [65], the authors proposed the tool called OSMRec that uses geometrical and textual features to train a Support Vector Machine classifier that is used to recommend a set of tags for new objects that are being digitized by annotators. The tool OSMantic is proposed in [131] and uses semantic similarity and tag frequency to recommend tags.

[4] https://josm.openstreetmap.de/

Tags verification

As shown in the previous section, several methods propose machine learning models based on properties of OSM objects. Then, the trained classifiers can be applied to another set of OSM objects to find possible annotation errors. In [3], the authors present three strategies of how such learned classifiers can be applied (see an illustration of them in Figure 2.4).

1. *Consistency checking*, where the classifier is applied while the user is editing and assigning tags to OSM objects. In this case, the editing tool can, for example, inform the volunteer that the assigned tag value is inconsistent with what the classifier predicted. Then, the annotator can modify the annotation if required, by taking into consideration the classifier's recommendation

2. *Manual checking*, in which the classifier is applied over a selected set of objects already registered in OSM. Then, the objects whose tags present inconsistencies with the predictions of the classifier are manually validated by the users

3. *Automatic checking*, in which a classifier is used to automatically correct annotations based on its predictions without human verification.

The method proposed in [3] aims to find errors in tags used for annotating green area objects (i.e., meadow, garden, grass and park). The authors observed that these four types of green area objects are some times mislabeled by OSM annotators. Figures 2.5a-b illustrate a case where a grassland area in the center of a roundabout was wrongly labeled as a park. Figures 2.5c-d depict a case where a grassland area with some trees is wrongly labeled as a forest. The technique proposed in [3] extracts geometrical, topological, and contextual properties (e.g., object area and features based on the 9-Intersection model [30]) and trains a K-Nearest Neighbours (KNN) classifier to analyze the labels of the four types of green area objects in OSM. The authors in [3] perform an experiment that consists in asking users to manually verify/correct objects with possible erroneous labels. These objects are detected by a classifier, and the experiment shows the effectiveness of the approach to detect mislabeled green areas. Another approach proposed in [2] tries solving this problem (disambiguation of green areas with the same four classes) by extracting rules from the OSM dataset using the algorithm proposed in [1]. These rules are extracted based on topological relations between geographical objects.

2.2.4 Points of Interest

Points of Interest (POIs) are key elements in OSM. They indicate the location of geographical objects that are commonly used in the city, such as bus stations, cafes, restaurants, ATMs, etc. Thus, the quality control of new added POIs to the OSM database is very important. Some OSM editors, like JOSM implement basic rules to avoid errors (e.g., duplicate elements) while editing objects in OSM. However, this type of topology quality control verifications does not take into account the spatial relationship between a new POI and neighboring geographical objects in OSM.

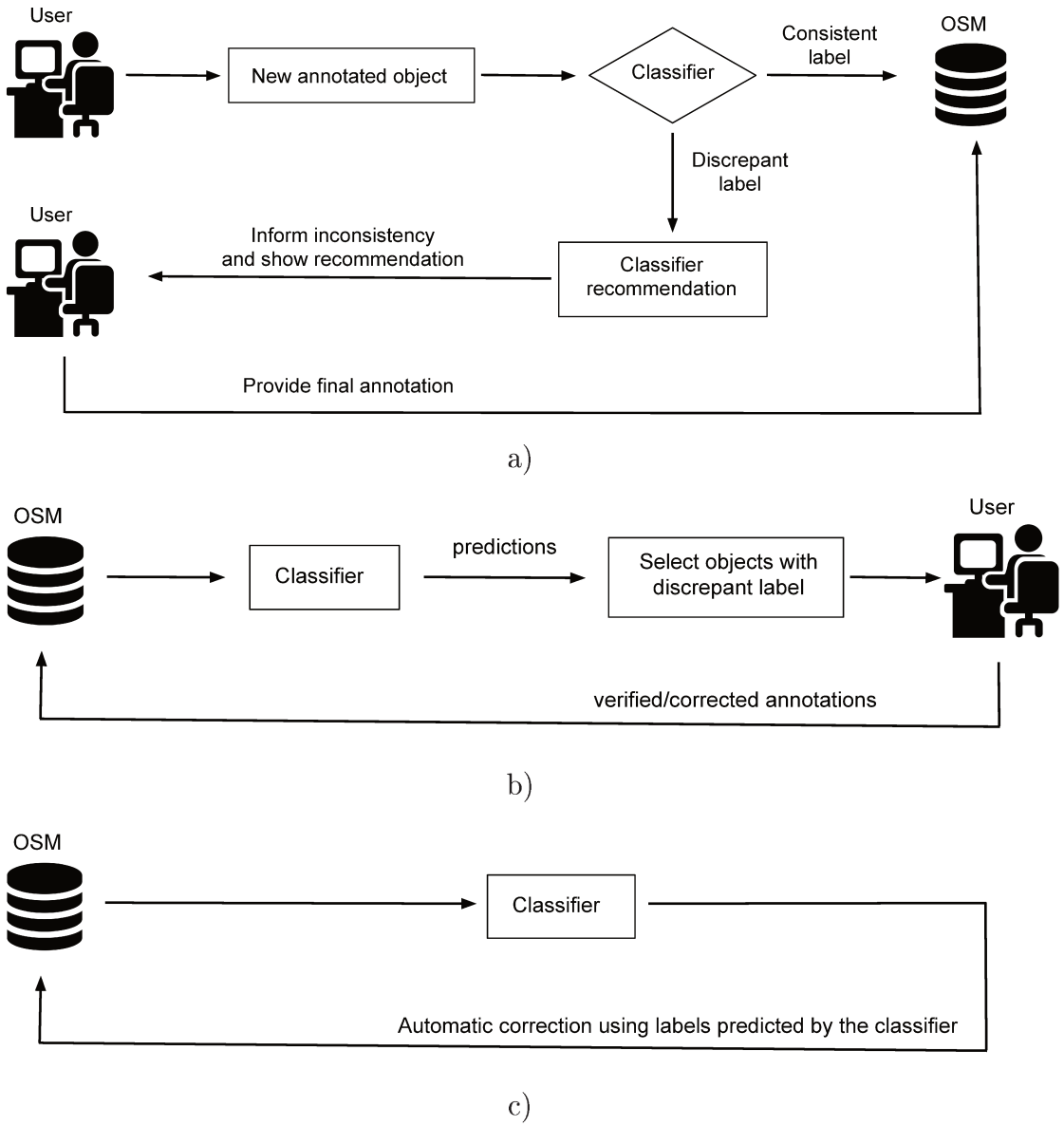


Figure 2.4: Three strategies used with a classifier to improve OSM data, according to [3]. a) *contribution checking*, b) *manual checking* and c) *automatic checking*

Plausibility of new additions

In [66], the authors propose a recommendation tool that evaluates the positional plausibility of a new registered POI with a certain category label. That work proposes to use spatial co-existence patterns for computing a plausibility score. Confidence scores are computed based on the frequency of occurrence of each pair of POI categories (i.e., {ATM, Bar} or {Bank, supermarket}). Then, the similarity score of two POIs is defined based on the confidence values of their POI categories. Finally, the plausibility score of a new POI is computed as the sum of the similarity score of the new POI and its neighbors. In order to compute the confidence values of pairs of POI categories, the authors recommend using POIs of the same city where the tool has to be validated. This is because different

Figure 2.5: Misannotated green area objects in the OSM map alongside with the Bing imagery of the corresponding location: a-b) A grassland area at the center of a roundabout mislabeled as a park in the city of Campinas, Brazil, c-d) A grassland area with some trees (located at the bottom right of the image) is mislabeled as a forest in the city of São Gonçalo, Brazil.

cities may have different patterns.

In [66], a case study is shown by evaluating the plausibility of a new ATM being added to four locations in Paris (besides a river and bridge, Paris downtown, middle of a park, and outside the city). The plausibility values obtained are coherent with what is expected – e.g., the plausibility score of a new ATM located in Paris downtown is much larger than the other alternatives.

Tags prediction

In [37] the authors propose a method that can predict tags of Points of Interest (POIs) based on their names. This method can be useful to extract tag information for POIs that lack of tagging information. For instance, a POI with name "Chicken Palace" probably should have the tag “amenity=restaurant". This work used the number of occurences of k-grams, substrings of a given size, extracted from the POI names to create feature vectors. Then, a Random Forest classifier was trained with OSM reference data to food, shop and tourism related POIs in OSM obtaining accurate prediction for some food related classes.

2.3 Using OSM data with machine learning algorithms

This section presents methods based on machine learning to use OSM in other applications, namely landuse/landcover classification, building detection and segmentation, navigation, traffic estimation, and fine-scale population estimation.

2.3.1 Landuse and lancover

Landcover/landuse mapping has been attempted by governmental organizations (e.g., Urban Atlas [5]), commercial services (e.g., Google maps) and crowdsourced projects (e.g., OpenStreetMap). Several governmental surveys are freely available. However, the quality of the landuse maps depends on the city and country and also this data is of few use when is not updated frequently. Commercial services like Google maps are more frequently updated but great part of the geographical information in such services are not openly available. In contrast, crowdsourced projects, like OSM provide access to all the collected geographical information and they are regularly updated in several cities.

The data quality of some landuse types in OSM is comparable to governmental surveys. In [7], the authors compare the accuracy of OSM data for landuse mapping in seven large European metropolitan regions. The thematic accuracy and degree of completeness of OSM data are compared to the available Global Monitoring for Environment and Security Urban Atlas (GMESUA) datasets. Evaluation of several land use types suggests that some OSM classes have good quality, such as forest, water bodies, and agricultural areas, and could be used for landuse planning.

Several works have proposed methods to predict landcover/landuse labels by using remote sensing imagery and OSM landuse labels as reference data to train a classifier. The authors in [63] use time-series Landsat imagery and OSM annotations (i.e., object boundary delineations and landuse labels) to train and evaluate several supervised methods for landcover classification, considering six classes (e.g., impervious, farm, forest, grass, orchard, and water). The authors in [64] use aerial imagery and a large amount of building and road annotations from OSM as training data for supervised classification. The collected annotations are selected without any quality verification and thus the authors observe several cases of low-quality annotations. The authors show that Convo-

[5] https://land.copernicus.eu/local/urban-atlas

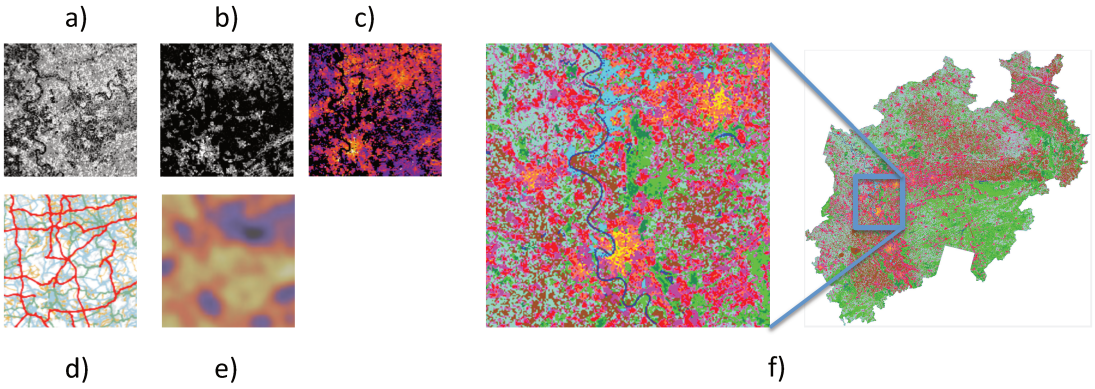


Figure 2.6: Data sources and local climate zones prediction map in the region of North Rhine Westfalia [126]. All the subfigures correspond to the region in the blue square in the right most map.

lutional Neural Network (CNN) methods trained with this type of data can achieve high accuracies as compared with methods that use a relatively small amount of good quality data. The authors in [8] propose a CNN based method that combines aerial imagery and rasterized OSM data to predict per-pixel landuse labels. In [126], The authors propose a method for landuse mapping at large scale (approx. 34000 km^2 in the region of NorthRhine-Westfalia, Germany). Each spatial unit is a 200m x 200m cell characterized by multimodal features: remote sensing data (RapidEye bands and texture features, see Fig. 2.6a-b), 3D models (see Fig. 2.6c) and OSM data (Point of interests location and road networks (see Fig. 2.6d), both encoded as street densities at the cell level (see Fig. 2.6e)).

Other works use OSM data and ground-based pictures of a set of OSM objects, to teach a model how to predict the landuse (e.g., museums, parks, educational institutions, sports centers, and hotels) of other OSM objects [118, 117, 119]. The method proposed in [118] use pictures, obtained from Google Street View (GSV), which capture multiple viewpoints of OSM objects (see the last three columns in Figure 2.7) and use a pre-trained CNN model to extract features and perform label prediction of 13 landuse types. The method proposed in [117] presents an extension of [118] and considerably improves the prediction accuracy. That method train end-to-end a CNN model that take as input a variable number of ground-based pictures and extract and combine features from the multiple pictures learning to identify the proper landuse class (among 16 landuse categories) of an OSM object.

More recently, the authors in [119] propose a CNN-based method that combines aerial imagery and ground-based pictures information to perform landuse prediction, using OSM as reference data. That method greatly improves the accuracy obtained by the method in [117] which uses only ground-based pictures. Additionally, that work also proposes a strategy to deal with the cases when ground-based pictures are missing for an OSM object. Figure 2.7 illustrates aerial and ground-based images corresponding to two OSM objects in the city of Paris. The first row shows the images of a church. If we just observe the aerial imagery, it is difficult to be confident in predicting the object as a church. Ground-based pictures can give additional visual features to predict the correct landuse

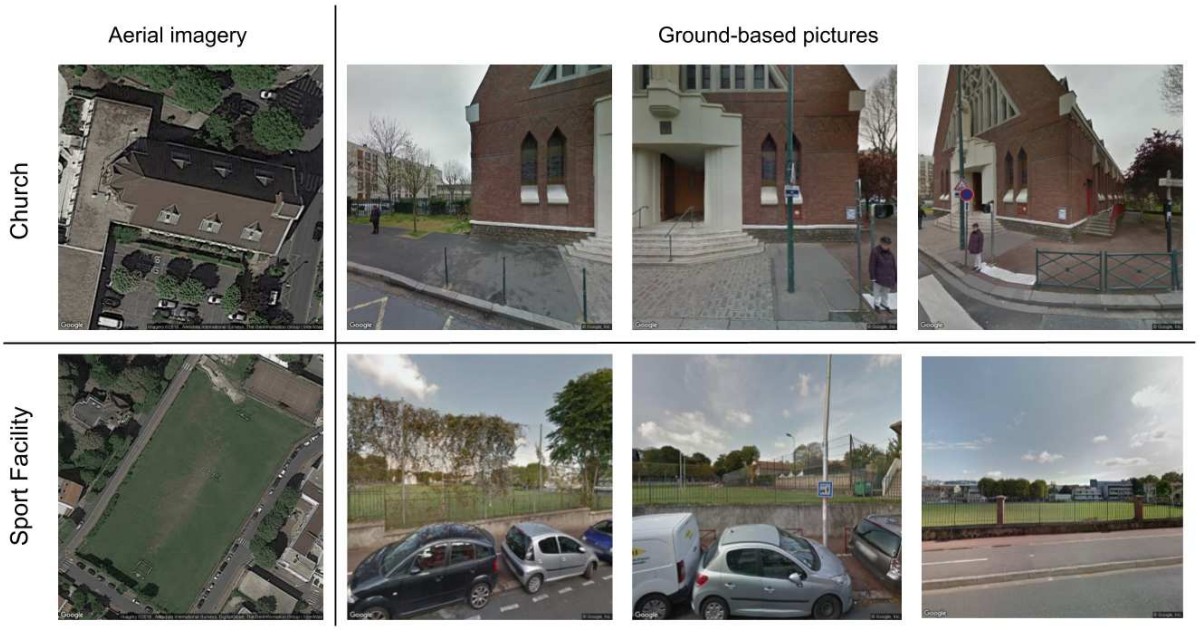


Figure 2.7: Aerial imagery, from Google maps, and ground-based pictures, from Google Street View, for two different OSM objects in the city of Paris. The pictures in the first row correspond to a church. The pictures in the second row correspond to sport facility.

label. The second row shows images of a sport facility in OSM. Because of the fences around the building, sometimes it is difficult to recognize a sport facility. In this case, aerial imagery is usually more valuable to predict the correct landuse label.

2.3.2 Building detection and segmentation

Building annotations are widely available in OSM and they are usually of considerable quality in several urban areas. Thus, OSM building annotations have been used as reference data to train CNN-based building segmentation methods [81, 93, 144]. The building segmentation maps produced by these methods are not directly usable for Geographical Information Systems (GIS) because they are raster images. However, some methods [87, 122] have been recently proposed to output vectorial building polygons suitable for GIS software. The method proposes in [87] uses Active Contour Models with its parameters learned by a CNN to output vectorial footprints. The authors of [82] and [122] propose to convert binary building classification maps into vectorial outputs by using a mesh-based approximation method.

In contrast to urban buildings, rural buildings are sparsely located in large geographical areas and their annotations in OSM are less frequent and of lower quality as compared to buildings in urban areas. Some techniques have proposed to improve the detection of locations of rural buildings in aerial imagery [23, 22]. In [23], the authors propose a CNN-based method to detect buildings in aerial imagery using an iterative process, in which new samples are selected for annotation by an active learning method and the model is retrained. In [22] the authors use multiple sources of crowdsourced geographical data

(namely OSM, MapSwipe [6], and OsmAnd [7]) and an active learning strategy to train a CNN model that detects image patches with buildings. Furthermore, the authors perform an experiment in MapSwipe (smartphone-based application for humanitarian mapping) asking the volunteers to just verify the tiles that are selected by a trained classifier, saving considerably the user effort and obtaining an accurate classifier.

2.3.3 Navigation

In several urban areas, building and road network data in OSM have similar quality as commercial map services. Thus, OSM has been used for navigation/routing applications [77, 45, 48]. In [77], the authors show that routing services based on OSM data can attain real-time shortest path computation in large areas for web-based applications as well as in hand-held devices. In [45], the authors show that OSM data can be used for pedestrian routing. The paper proposes a solution to efficient routing in open spaces (e.g., squares, parks, and plazas). A more recent and extensive analysis of different strategies to deal with routing that consider open spaces is presented in [48].

In [34], the authors propose a method for accurate global vehicle localization. That work uses visual odometry and OSM data obtaining better localization results than methods that just rely on visual odometry approaches. A probabilistic model for autonomous robot navigation is proposed in [120]. It uses a Markov-Chain Monte-Carlo method to combine semantic terrain information extracted from 3D-LiDAR data and OSM information.

2.3.4 Traffic estimation

Traffic prediction is a challenging task that can be very useful for congestion management and car navigation. The authors in [140] propose a method to predict four classes of traffic (i.e., good, slow, congested, and extremely congested) in four cities in China. That work uses data obtained from POIs of Baidu maps and geographical objects from OSM (the number of POIs in OSM is limited in China). This geographical data together with other features, like weather, temperature, and house pricing are used to train a machine learning model for traffic prediction. The authors use traffic data from Baidu maps as reference data to train a Support Vector Machine (SVM) classifier. In that work, the authors observe that traffic congestion data is very unbalanced, because most of the time the traffic is good considering all the time intervals in one day (with exception of the rush hours 9h and 17h). Thus, the authors assign higher weights to the classes with less number of samples. The results show that even using class weighting, the accuracy of the model is high for the class good and poor for the other classes. Additionally, that method is compared to the traffic prediction system of Baidu maps outperforming it in some time intervals of the day and obtaining reasonably good performances when the model trained in one city is used for prediction in another city.

[6] https://mapswipe.org/
[7] https://osmand.net/

In [75], the authors deal with a similar problem, predicting traffic speed using a regression method. A public dataset named UIUC New York City Traffic Estimates [8] is used for their experiments. This dataset contains hourly average traffic speed measurements in the New York City road network, obtained from car trips. Several geographical features are obtained from OSM, such as road length, number of roads connected to the analyzed road, number of neighbor nodes and roads in the area, and also temporal features (e.g., time and whether it is a workday). In order to predict traffic speed in the target areas, the authors of that work propose the method Cluster-based Transfer Model for Prediction (CTMP), which first clusters the road features of the source and target areas. Then, the traffic speed of the target area is computed based on the nearest neighbor roads data of the source area, which contains traffic speed information. CTMP shows better results than other baseline methods, such as Neural Networks and Support Vector Regression. More recently in [106], the authors propose a deep learning method that models the road network topology to predict traffic flow in the city of Chengdu, China. The authors use neural networks to model road network topology and residual learning [53] to model spatio-temporal dependencies. One limitation of that work is that it requires traffic flow historic reference data of the target location to be able to predict the traffic flow in a different time interval.

2.3.5 Fine-scale population estimation

Population distribution at the building level is important for several tasks, like urban planning and business development. Population estimation at the building level scale can be obtained by areal interpolation. Although this technique usually requires 3D building models, obtained by LiDAR data, the authors in [9] propose to use building footprints and POIs from OSM to predict population distribution by using areal interpolation. The authors in [143] use a Random Forest model to predict population at the grid level (i.e., the area of study is divided into grid cells) by using Baidu POIs, mobile user density data, and road networks from OSM. Then, the grid level estimations are transformed to building level estimations.

In [38], the authors propose a CNN-based method to perform population density estimation at the grid level. First, the area of study is divided into grid cells of size 200×200 meters. Then, for each cell, they compute several urban features, such as building area, number of buildings, and number of POIs. Finally, the authors use a fully convolutional neural network, applied over the urban features of the grid cells, to obtain the population estimation of the corresponding grid cells. The experimental results show that by training the model with data from 14 French cities, the model attains low error rates in the validation data extracted from the city of Lyon.

[8] https://lab-work.github.io/data/

2.4 Discussion and future works

In this section, we discuss potential promising research avenues at the interface of OSM and machine learning.

Going multimodal

Although the use of several sources of data have proven to be beneficial for solving several problems it has been only applied in relatively few works described in this manuscript, such as in [119]. We believe that the performance of supervised methods used to improve OSM data can be greatly improved with the use of several data sources, such as images, tracking data, and social media data. This has been also pointed out in [123] for the particular case of POIs, where the authors recommend the use of several data sources, like OSM geographical data, ground-based pictures, and historical data to create more accurate models for POI label prediction. Among these external data sources, ground-based pictures obtained from Google Street View (GSV) have found to be particularly useful to enrich OSM data, for example, for landuse prediction [117] and crosswalk localization [19]. Recently, crowdsourced ground-based images collected by the TeleNav's project, called OpenStreetCam [9], has also been used to improve OSM. For instance, pictures obtained from OpenStreetCam have been used to detect traffic signs [10]. Although nowadays the coverage of crowdsourced ground-based pictures platforms is not as complete as GSV, they have the potential to obtain more updated data and to be available for everyone at no cost. This data can then be used to obtain up-to-date OSM data of objects like traffic lights and road signs, which can greatly benefit navigation applications.

Supporting users via interaction and skills estimation

Mapping information obtained by machine learning models applied with Earth Observation data (e.g., building footprints, and road networks) could contain some errors. Thus, an alternative to performing automatic updates in OSM is to use a human-computer interactive approach. In this strategy, the machine learning model is used to minimize the effort of the users during the annotation process. This strategy has been already applied in [15] to improve road network completeness. In that work, an automatic method is used to extract major roads (in places with a few road annotations) and missing roads (in places where major roads are already annotated) and the user is asked to verify or correct if needed the extracted roads. This work shows experimentally that such an interactive approach is more efficient than traditional manual annotation. A related and also effective approach that involves the user in the process is active learning [114, 27]. Active learning is an iterative process that consists of intelligently selecting a small number of samples for user annotation that allow training an effective classifier. We think that these approaches that involve the interaction between a machine learning model and the annotator could be applied to improve other aspects of OSM data with fewer annotators'

[9] https://openstreetcam.org

[10] https://blog.improveosm.org/en/2018/02/detecting-traffic-signs-in-openstreetcam/

effort, especially if user's skills are involved in the process, as shown in concurrent research in crowdsourcing [43, 127].

Between January and May of 2019, the number of active contributors per month in OSM was less than 1% of the total registered OSM users. However, a few efforts have been made to encourage volunteers to frequently edit data. Gamification strategies could be applied to solve the problem, by assigning annotation tasks to volunteers with a game-like interface and scoring systems. Some gamification projects have been listed in the OSM wiki [11] but they are not of widespread use.

Contributor analysis in OSM have been studied in several works [99]. It has been observed that volunteers' experience and familiarity with the area edited in the map are good proxies to estimate the quality of their annotations in OSM. The authors in [115] propose a measure for estimating annotation trust, by using annotation statistics obtained from volunteers' activities, object geometries, and temporal data. A trust index could be also learned with machine learning methods using OSM data statistics with some reference data. Such trust index could be used to improve methods that are created to verify the correctness of labels of objects registered in OSM [3, 2].

New ways of searching and interacting with OSM

Search tools are important features of mapping services. However, the search tool provided by the OSM website has limited capabilities, basically just trying to find the location of a given place name or address. In [70], the authors propose NLmaps [12], a natural language interface to query data in OSM. This service can answer textual questions about geographical facts in OSM. The response is a text and a map with geographical objects of interest highlighted. For instance, NLmaps can answer the following question: "What is the closest supermarket from the Royal Bank of Scotland in Edinburgh?". Figure 2.8 illustrates the search results in NLmaps for the question: "Which museums are there in Heidelberg?". First, the names of the museums found are shown and then their locations are shown with markers in the OSM map of the city of Heidelberg.

This work uses a semantic parser for the OSM data, proposed in [47] and extended in [71], to transform natural language text to a Machine Readable Language formula (MRL) [5]. This representation is used to create a structured query and retrieve OSM information using an extension of the Overpass API, called Overpass MLmaps [13]. Although NLmaps can handle several common questions, it is not able to answer complex questions like "Where are 4 star hotels in Berlin?". The authors of [71] also propose to improve NLmaps performance by using user feedback. A simple example of feedback is to ask the user if the result of the query is helpful or not. More complex feedback can be obtained from expert users, with knowledge of OSM and the Overpass API. For instance, one can ask the expert user if the intermediate results of the query processing pipeline are correct or not. We believe that the implementation of a more intelligent tool for answering natural language queries in the OSM website could potentially attract more users.

[11] https://wiki.openstreetmap.org/wiki/Gamification
[12] https://nlmaps.cl.uni-heidelberg.de/index.html
[13] https://github.com/carhaas/overpass-nlmaps

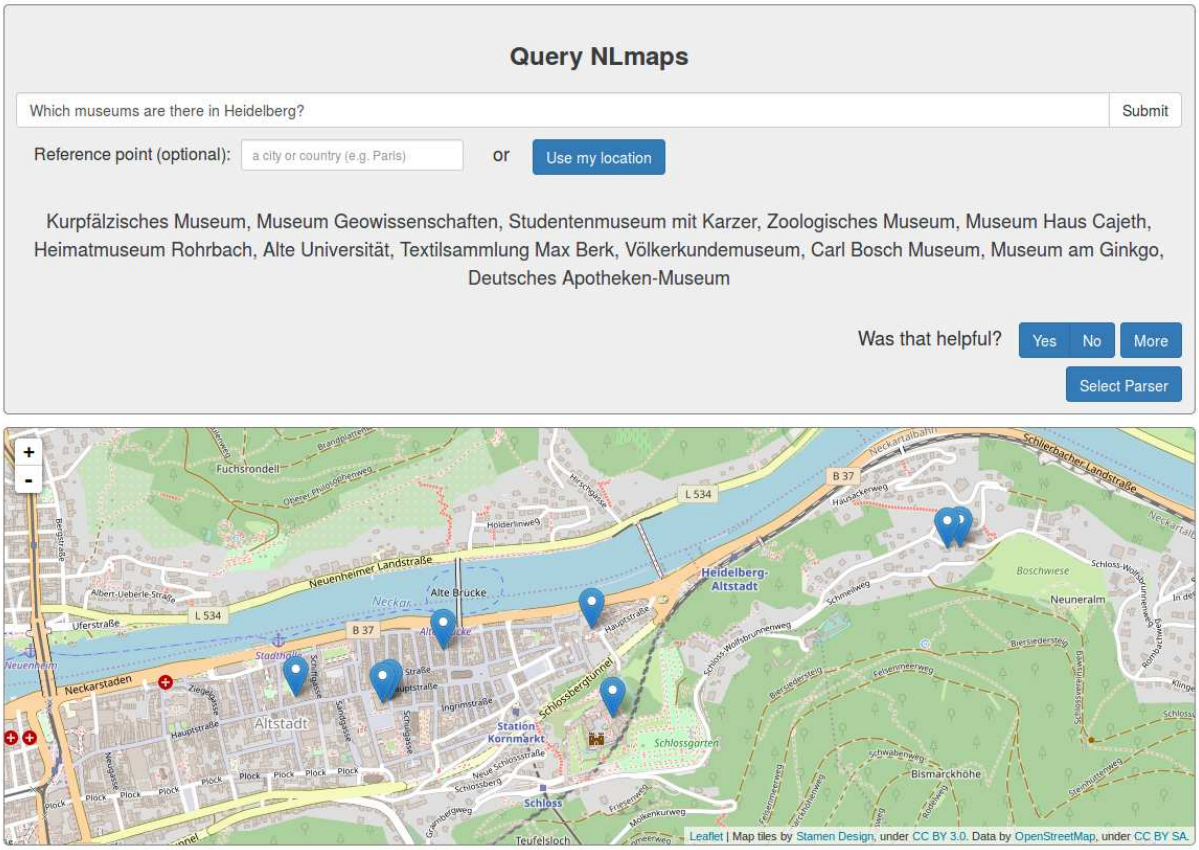


Figure 2.8: NLmaps search results for the query: Which museums are there in Heildelberg?.

Recently, the authors of [76] use OSM data for the Visual Question Answering (VQA) problem, a task that consists in providing a natural language answer for a given image and a natural language question about the image. The authors create a dataset using OSM data and Sentinel-2 satellite imagery for training a VQA model. VQA is a challenging problem and its application for remote sensing imagery is still in its infancy. VQA can enable to perform some queries in places where there is incomplete mapping data, but aerial imagery is available.

OSM-enabled augmented reality

Recent applications of OSM data can be further improved by using machine learning methods. For instance, augmented reality (AR) has great potential to improve the way we experience the cities, especially for tourism. In [109], an augmented reality tool is presented to recommend Points Of Interest (POIs) in the city by taking into account the profile of the users. The authors propose a mobile application that shows POIs registered in OSM in real time with the mobile camera view as background. A case study in the city of Trani, Italy, shows how a tourist interested in local architectural work can visualize in the mobile application the POIs, within a distance radius, marked as colored circles with labels superimposed over the mobile camera view. In [109], a hand engineered rule is proposed to define which POIs to show to the user. For this type of AR applications,

reference data could be used by asking users' feedback (e.g., asking to the user to add POIs visited in a favorites list). Thus, obtaining reference data can enable the use of more effective supervised learning methods to predict adequate POIs for new users.

2.5 Conclusion

In this paper, we reviewed synergetic efforts involving OpenStreetMap and machine learning. In the first part, we review works that use machine learning to improve OSM data. These methods deal with the three main object geometric types in OSM: points (e.g., ATMs), lines (e.g., roads) and polygons (e.g., buildings). The reviewed works use frequently traditional machine learning methods (e.g., Support Vector Machines and Random Forest), but also several of them have used state-of-the-art methods, such as deep learning based techniques, especially when dealing with image data. Although several methods could be integrated into OSM editors (e.g., iD editor [14] and JOSM) just a few works [65, 131] have implemented their methods in such tools.

The second part of the manuscript reviewed works that have used machine learning based techniques to use OSM data for applications in other domains. We identified two groups of works. The first group uses OSM data as reference data to train machine learning models, for examples several works that perform landuse classification [117, 119] and building segmentation [81, 93]. The second group uses OSM data to extract features for training the machine learning model (e.g., fine-scale population estimation [38]).

We believe that a mixture of automatic and human-interactive approaches could lead to obtaining accurate data for OSM with efficient use of the annotators' labor. The strong links with machine learning and the ever increasing availability of up-to-date remote sensing data open countless opportunities for research in this exciting interface among disciplines.

Acknowledgment

This research was funded by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, grant 2016/14760-5 and 2014/12236-1), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, grant 303808/2018-7) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES, finance code 001).

[14] http://ideditor.com/

Chapter 3

Correcting rural building annotations in OpenStreetMap using convolutional neural networks

Abstract

Rural building mapping is paramount to support demographic studies and plan actions in response to crisis that affect those areas. Rural building annotations exist in OpenStreetMap (OSM), but their quality and quantity are not sufficient for training models that can create accurate rural building maps. The problems with these annotations essentially fall into three categories: (i) most commonly, many annotations are geometrically misaligned with the updated imagery; (ii) some annotations do not correspond to buildings in the images (they are misannotations or the buildings have been destroyed); and (iii) some annotations are missing for buildings in the images (the buildings were never annotated or were built between subsequent image acquisitions). First, we propose a method based on Markov Random Field (MRF) to align the buildings with their annotations. The method maximizes the correlation between annotations and a building probability map while enforcing that nearby buildings have similar alignment vectors. Second, the annotations with no evidence in the building probability map are removed. Third, we present a method to detect non-annotated buildings with predefined shapes and add their annotation. The proposed methodology shows considerable improvement in accuracy of the OSM annotations for two regions of Tanzania and Zimbabwe, being more accurate than state-of-the-art baselines.

3.1 Introduction

The amount of publicly available mapping information in web services, like Google Maps and OpenStreetMap (OSM), is large, covering great part of the existing human settlements in the world. Although mapping information of buildings and several other man-made structures are largely available for urban areas, a significant amount of rural buildings is not mapped in any of the aforementioned systems. Rural building mapping information

is important to assist demographic studies and help Non-Governmental Organizations to plan actions in response to crises [1]. There is therefore a need for creating (or at least updating) urban footprint vector databases in rural areas.

Several works in the literature have approached this problem as the one of detecting buildings in remote sensing images using shape, color, edge, and texture knowledge-based features [116, 17]. More recently, Convolutional Neural Networks (CNNs, for a review in remote sensing see [145]) in combination with other image processing methods have been used to detect and delineate buildings in urban areas with successful results [93, 81, 110]. Most commonly, the pixel (or region) level detections are merged into vectorial shapes in a post-processing step. In [87], a CNN model was proposed to avoid this postprocessing step: vector footprints of buildings are learned directly, by defining the building outline definition as an active contour model, whose parameters are learned with a CNN. The investigation of building detection using deep learning is a field of growing interest, also supported by recent data processing competitions in this direction, e.g. DeepGlobe [29].

Irrespectively of the strategy chosen, the main drawback of using CNN methods in remote sensing is the need of large amount of labeled data for training. In recent research, OSM annotations have been used as repositories of large labeled data collections. In GIScience, this source of data has proven to be very powerful, and several works have proposed methods to automatically predict attributes of OSM objects. For example in [61], the authors proposed a methodology for automatic prediction of street labels (e.g., motorway and residential). In [32], authors proposed a method using geometrical properties of the OSM annotation polygons to predict the types of buildings (e.g., residential, industrial and commercial). In [33], OSM data was used to improve robot navigation for autonomous driving and in [139] OSM data was used for 3D building modeling, allowing visualization of indoor and outdoor environments in 3D maps. Authors in [117] use Google Street View pictures to predict the landuse of the footprints. They use OSM annotations as labels to train a deep learning model. Within the remote sensing building segmentation field, OSM annotations of urban areas have been recently used in [8] and [64] as label information to perform semantic segmentation of buildings and roads. The INRIA building detection challenge uses corrected OSM footprints as labels [80].

Despite the appeal of using OSM data for training deep learning models, the quality of these data is uneven. Usually CNNs trained with this type of reference data can learn to predict the location of the object but not the exact object extent [93]. Several works proposed methods that can be useful to improve the quality of the OSM data, both for attribute classification and positional inaccuracies. Authors in [14] detect errors in OSM annotations of roads using patterns extracted from GPS tracking data. For instance, indoor corridors wrongly classified as tunnels can be detected using tracked trajectories of cars and pedestrians. In [52], distance, directional, and topological relationship of OSM objects are used to detect inconsistencies.

OSM has gathered and made publicly available large amounts of building annotation data. But if the quality of OSM data has been judged sufficient for urban areas [31], the same does not hold in rural areas, especially because of the lower update rate and the drop in the number of volunteers out of cities. By analyzing available OSM data in

[1] `https://www.missingmaps.org/`

(a) Urban buildings (b) Rural buildings

Figure 3.1: Misaligned OSM building annotations (in orange) superimposed on the imagery obtained from Bing maps: a) For urban building misaligned annotations, there is a considerable overlap with the object in the imagery; b) For the case of rural building misaligned annotations, some buildings in the imagery and their corresponding annotations do not overlap.

rural areas, we observed that the annotations performed by the volunteers suffer from three main issues, mostly due to infrequent imagery updates and incomplete/inaccurate volunteer annotations [14, 12]:

(a) the locations of building annotations might be inaccurate: building footprints are often present, but displaced on the image plane by up to 9 meters. These displacements are often due to the fact that the image used to digitize the footprint does not correspond to the image being used for analysis. Two examples of misalignments are given in Figure 3.1;

(b) some annotations do not correspond to buildings in the imagery: in this case, some buildings might have been demolished, or simply the annotations by the volunteers are erroneous [35];

(c) some objects that appear in the imagery are not present in the annotation dataset: in this case, some buildings might have been missed by the volunteers or new buildings might have been built in between the two image acquisitions.

In order to deal with inaccurate reference building data, the authors in [93] propose a loss function to reduce the effect of this problem, while the authors in [79] use a Recurrent Neural Network to improve the classification maps with a small set of perfectly and manually annotated data. However, as mentioned above, for rural buildings the problem of inaccurate annotations is more severe, since buildings are smaller and scarcer than urban buildings in OSM [23]. As one can see in Figure 3.1, there exists considerable overlapping areas between urban buildings and the misaligned OSM annotations, while some rural buildings in the image and the OSM annotations do not overlap.

In this work, we propose a methodology to correct OSM rural building annotations. We tackle the three problems above simultaneously, with a three-stage strategy based on the predictions of a fully convolutional deep learning model that estimates the likelihood of presence of buildings.

(i) First, we propose a method to align buildings and their annotations based on Markov Random Field (MRF) [20]. We make the hypothesis that alignment errors can be fixed by simply translating the annotations themselves, since we observed that this type of error is the most frequent in rural areas (see Figure 3.1(b)). MRFs have been successfully applied to solve registration problems in several image domains [41, 86, 130]. Our MRF-based method maximizes the correlation between OSM annotations and a predicted building probability map, while enforcing that nearby buildings have similar alignment vectors (shift correction vectors). Usually, rural buildings appear in small groups with the same alignment errors (as a given area is annotated on one image by the same volunteer, this whole area will present a similar misalignment when the imagery is updated). For this reason we use nearby rural buildings as nodes of a small MRF graph. The method then computes a single alignment vector for all the buildings in each small group of rural buildings.

(ii) Second, the OSM annotations with no evidence in the previously computed building probability map are removed. For each OSM annotation, we compute the mean building probability value of the pixels contained in the aligned annotations. If the computed values are smaller than a threshold [40], we remove the OSM annotations.

(iii) Third, we present a CNN-based method for adding new building annotations. Since the variety of rural building shapes and sizes is very small as compared to the ones of urban buildings, the CNN estimates one of 18 commonly appearing rural building shapes for each non-annotated building.

In Section 3.2 we present the proposed methodology to correct OSM rural building annotations. Section 3.3 shows the dataset and the setup of our experiments and Section 3.4 compares the results of our proposed method with other baseline methods. Section 3.5 concludes the paper.

3.2 Methodology

Our methodology to correct OSM annotations of rural buildings requires a fully convolutional neural network (CNN) model trained to generate a building probability map for the overhead image (Figure 3.2a): this method is detailed in Section 3.2.1. Once this classifier is trained, the building correction module consists of three main tasks, as described in Section 3.1. Figures 3.2b-d illustrate them, from top to bottom. In sections 3.2.2 to 3.2.4 we detail these methods.

3.2.1 Computing building probability maps

In order to correct OSM rural building annotations, we use a building probability map obtained by a CNN model that performs pixel classification. In this work we use a CNN model based on [138] that is trained on a small set of manually verified/corrected rural building OSM annotations. The CNN model performs four convolutional blocks (convolution followed by spatial pooling, non-linear activation and batch normalization

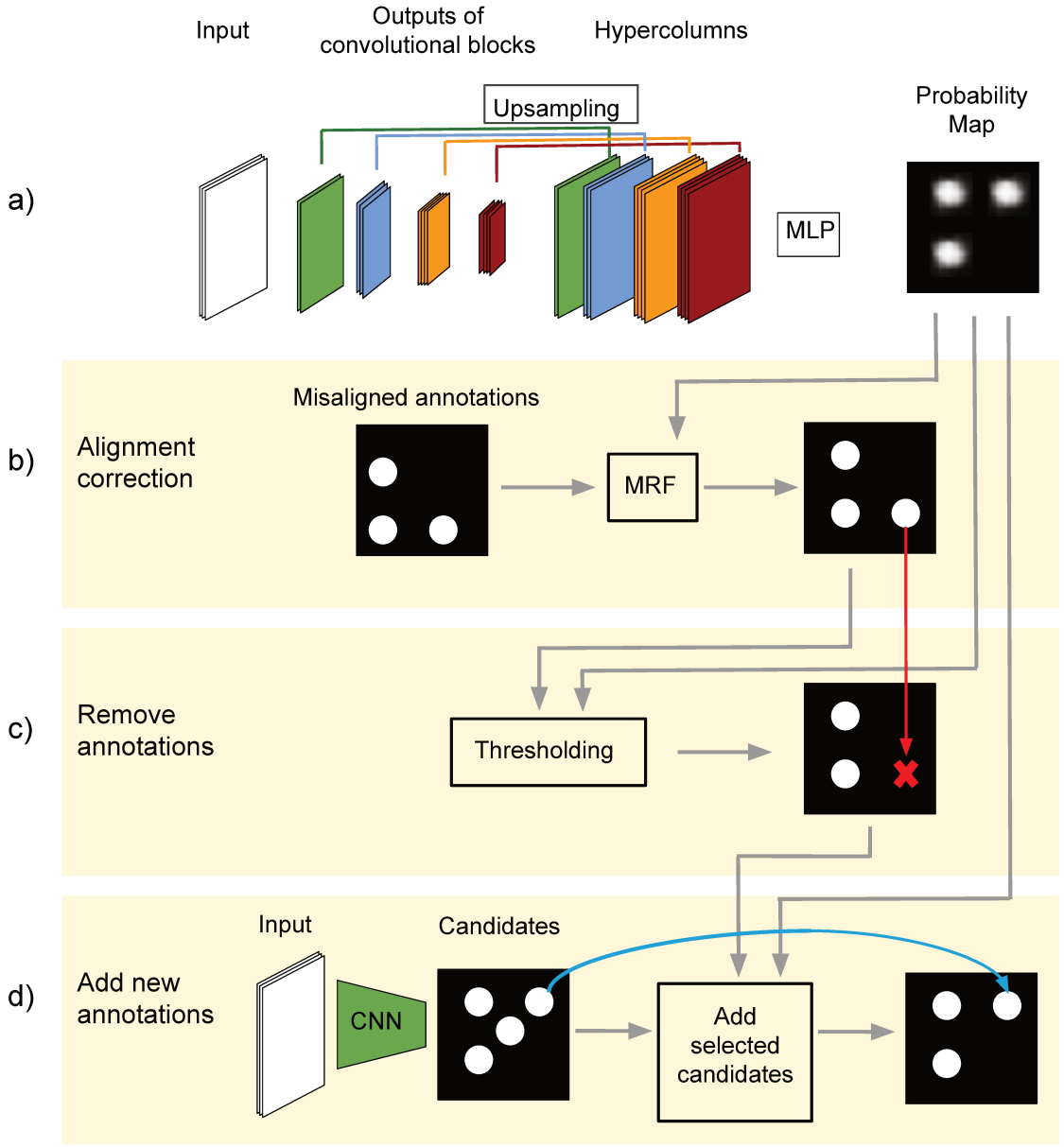


Figure 3.2: Proposed methodology to correct OSM rural building annotations: a) predict a building probability map from an aerial image using a CNN trained for per-pixel classification; b) correct alignment errors in the OSM annotations using a MRF-based method and a building probability map; c) remove OSM annotations based on the aligned annotations, a building probability map, and a thresholding method; d) add new annotations selected from a set of candidates obtained by a CNN that predicts rural buildings with predefined shapes.

operations) but, differently from [138] that uses deconvolutions to upsample the feature map, we apply the concept of hypercolumns [51] to perform pixel classification. We modified the original hypercolumn model in the same way as for the baselines of [88]: the hypercolumns are obtained by upsampling the outputs of previous convolutions to the size of the input image using bilinear interpolation. This makes the training of the CNN more efficient and with similar performance. These activations are then stacked to

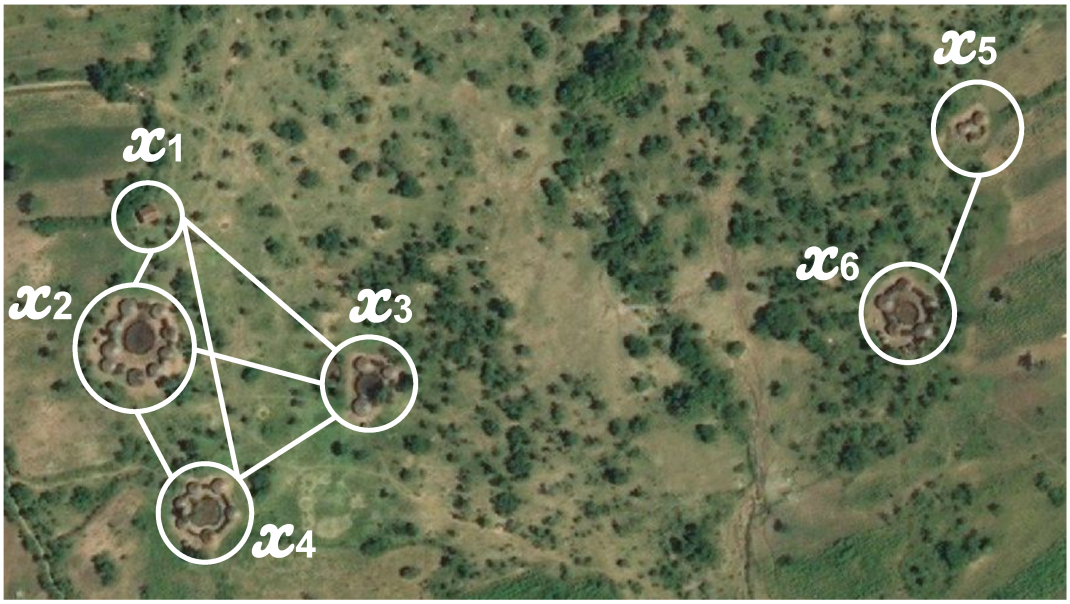


Figure 3.3: Neighboring system of the proposed MRF method. Groups of rural buildings are used as nodes of the MRF graph.

a single tensor which is used to train a Multi-layer Perceptron classifier to perform pixel classification. The architecture of the described CNN is presented in Figure 3.2a, while the details of the specific architecture are presented in Section 3.3.

3.2.2 Aligning OSM rural building annotations

The building registration problem is considered as the problem of aligning the vector shapes from OSM to the predictions of the CNN (Figure 3.2b). Such alignment is performed by estimating alignment vectors, basically shifting every OSM polygon to an area of high building probability in the CNN map.

In order to compute these alignment vectors, we need to measure how well a given shift performs. To this end, we use the correlation between the aligned annotations and the building probability map obtained previously using the image on which the annotations need to be registered. Making the hypothesis that rural buildings are gathered in small groups where each building has the same misalignment error, we align groups of buildings instead of individual buildings. This reduces greatly the computational load and is numerically more effective (see the results Section 3.4). Moreover, using groups of buildings instead of single ones makes the results less dependent on the quality of the building probability map.

Additionally, we observed that nearby groups of buildings have similar registration errors. Based on this observation, we build our building registration module on a MRF model using this prior together with the evidence provided by the building probability map. Our method aims at finding the alignment vectors $\mathbf{d} = \{d_0, d_1, \ldots, d_n\}$ that need to be applied to the annotation locations $\mathbf{x}$ based on the a probability map $\mathbf{y}$. Groups

Algorithm 1 – MRF-BASED ALIGNMENT ALGORITHM

INPUT: Original OSM annotations M and building probability map $\mathbf{y}$
OUTPUT: Alignment vectors $\mathbf{d}$

1. *Group the original rural building annotations M according to their spatial distance from each other, obtaining the set of building groups* $\mathbf{x}$.
2. *Define the neighbors N_i of each site i as spatially close sites.*
3. *Initialize each d_i to* $\arg\max_{d \in \mathcal{D}} C(d_i(x_i), y_i)$
4. *Run Iterated Conditional Modes (ICM) for $MaxIters$ iterations*
5. ***For*** $t \leftarrow 1...MaxIters$, ***do***
6. ***For each*** $x_i \in \mathbf{x}$, ***do***
7. ***For each*** $D \in \mathcal{D}$, ***do***
8. *Compute energy $U(D|x_i, y_i)$, equation (3.1)*
9. ***If*** $U(D|x_i, y_i) < U(d_i|x_i, y_i)$, ***then***
10. $d_i \leftarrow D$
11. *Return* $\mathbf{d}$

of buildings, or *sites*, are used as nodes of the MRF graph (See Figure 3.3), where sites i and j are neighbors (i.e., $j \in N_i$) in the graph if they are spatially close (see Section 3.3.2 for more details on the MRF graph definition).

In our MRF formulation, the unary term is obtained by using the normalized correlation $C(d_i(x_i), y_i)$ between the annotation after alignment $d_i(x_i)$ and the building probability map y_i. This term is equal to the average of the predicted probability values y_i of the pixels contained in the aligned annotation $d_i(x_i)$. The pairwise term is defined by the dissimilarity (vector norm of the difference of two vectors) between the alignment vector d_i of the annotation i and the alignment vectors d_j of neighboring annotations $j \in N_i$ [86]. The optimal set of alignment vectors $\hat{\mathbf{d}}$ for the annotations is defined by:

$$\begin{aligned} \hat{\mathbf{d}} &= \arg\min_{\mathbf{d} \in \mathcal{D}^{\mathcal{N}}} \sum_i U(d_i|x_i, y_i) \\ &= \arg\min_{\mathbf{d} \in \mathcal{D}^{\mathcal{N}}} \sum_i -\log C(d_i(x_i), y_i) + \beta \sum_{j \in N_i} \frac{1}{Z} ||d_i - d_j||_2, \end{aligned} \tag{3.1}$$

where $\mathcal{D} = \{D_1, D_2, \ldots, D_m\}$ is the set of all possible m alignment vectors, β is the spatial regularization parameter and Z is a normalization factor, defined as the maximum possible distance between two alignment vectors in $\mathcal{D}$. To compute the optimal $\hat{\mathbf{d}}$ by minimizing the energy function U, we use the Iterative Conditional Modes (ICM) [20] algorithm initialized with $d_i = \arg\max_{d \in \mathcal{D}} C(d(x_i), y_i)$. As this initialization is already a good heuristic (see Section 3.4), the ICM algorithm allows to obtain a good solution in a few iterations. The inclusion of a distance-based weight in the pairwise term does not lead to better performances, so it is omitted for clarity. We presented preliminary results of our proposed method for alignment of OSM annotations in the conference paper [134]. Algorithm 1 summarizes the proposed method for aligning OSM annotations.

3.2.3 Removing incorrect building annotations

In order to remove OSM annotations that no longer exist in the updated imagery (Figure 3.2c), we compute the mean building probability value of the pixels contained in the aligned annotations. We observe that the histogram of these average probability values roughly follows a bimodal distribution. The group of annotations close to the first local maximum corresponds to some of the few annotations that have average probability values close to zero (showing high evidence that there is no longer a building in that location of the imagery) while the other group of annotations gathered around the second and most prominent local maximum corresponds to the majority of the aligned annotations that have higher average probability values. Since Otsu's thresholding method [100] is known not to perform well for unbalanced distributions [72] we use the Minimum threshold method [40]. This method iteratively smooths the histogram until only two local maxima are found. After that, the minimum value between the two local maxima is selected as the threshold. We then remove annotations, which have an average probability value below this threshold.

3.2.4 Add new building annotations

The last task is the addition of new building footprints (Figure 3.2d). We observed that rural buildings appear with very few different shapes in the imagery (e.g., circles and rectangles), as compared to urban buildings. Therefore, we make the hypothesis that a restricted number of shapes is sufficient to represent most buildings in rural areas. Inspired by this, we compile a set of 18 commonly appearing shapes and propose a CNN model that predicts if a building with one of these predefined shapes is present in a particular location of the imagery (see Figure 3.4). Based on our observations, we select 6 basic geometrical shapes: a circle of radius 3.3 meters, a square of side 4.8 meters, a rectangle of sides 3.6 and 6 meters, and the same rectangle rotated by $45°, 90°$ and $135°$. Furthermore, for each base shape we generate two more scaled versions, by approximately increasing its area by a factor of 2 and 4, resulting a total of 18 considered shapes (see Figure 3.4).

The architecture of the proposed CNN model is depicted in Figure 3.4: we apply two convolutional blocks followed by one convolutional layer to the input image of size 256×256, leading to a 61×61 feature map with 512 activations per location (details of the specific architecture are presented in Section 3.3). Afterwards, we apply a 1×1 convolutional layer that outputs a matrix of size 61×61 and 36 bands. This operation is performed to compute scores for the two classes of interest (presence or absence of buildings) with the 18 different shapes in each location of the 61×61 grid. This means that we have a different classifier for every building shape. Every pixel in the 61×61 grid corresponds to one location in the original 256×256 input image. Therefore, the location of our building predictions will have an additional approximation error of less than four pixels.

For training the CNN model, we use a cross entropy loss on the sum of the binary shape classification problems. We consider as positive samples of a given building shape, rural buildings with more than 0.75 Intersection over Union (IoU) value with a shape mask. The rural buildings with less than 0.30 IoU value with a shape mask are considered as negative

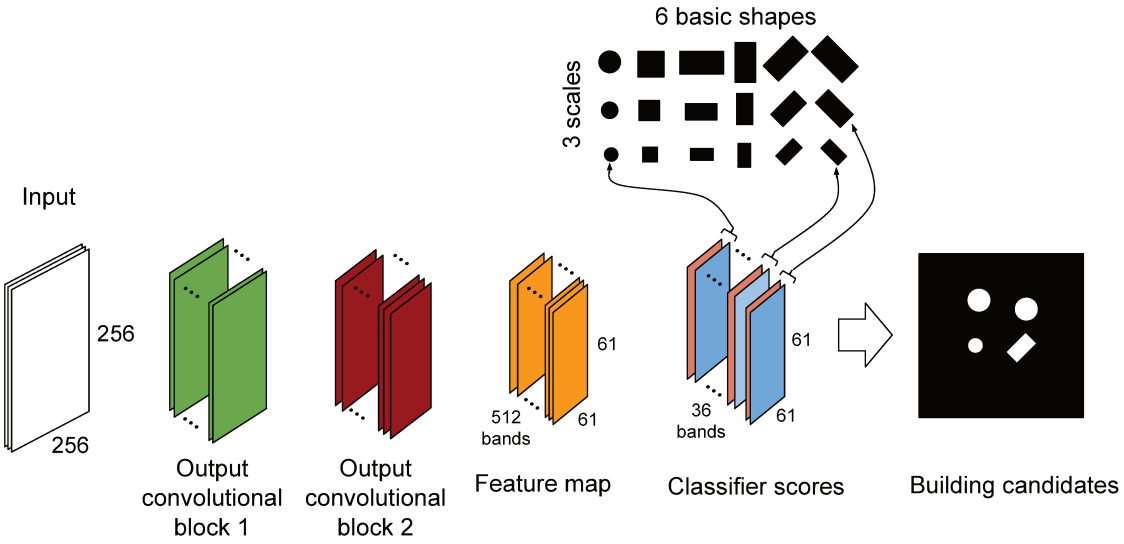


Figure 3.4: CNN model for adding new annotations of buildings that appear for the first time in the updated imagery.

samples for that particular building shape. The threshold values are chosen empirically based on the object detection method presented in [105]. Note that if we choose a higher value for the positive sample's threshold, we might ignore some buildings that have very similar desired shape and if we use lower values for that threshold, we would take the risk of including buildings whose shape does not fit with the desired building shape.

The output of this CNN model is a set of rural building candidates that have predefined shapes. We select a subset of these candidates based on the building probability map and the aligned building annotations, obtained after the annotation removal process. We filter out all the candidates that have average probability values (as obtained by the CNN model that performs per-pixel classification) and detection probability values (obtained by the CNN model described in this section) lower than a certain threshold t. In case of overlapping candidates, we select the one with the highest sum of average probability and detection probability values.

3.3 Data and experimental setup

3.3.1 Datasets

We evaluate our method with OSM rural building data from two different countries, namely the United Republic of Tanzania and the Republic of Zimbabwe. The evaluation data collected from these two countries have different characteristics: while the Tanzania's evaluation region contains severe misaligned and incomplete annotations, the evaluation region in Zimbabwe contains more accurate annotations. The Bing imagery utilized for the two datasets were acquired between 2004 and 2014, while the annotations obtained from OSM were performed by volunteers between 2013 and 2018. Bing maps provides an API to obtain aerial imagery (red, green and blue channels) at different spatial resolutions

(e.g., 119 cm, 60 cm, 30 cm). In this work, for the training and testing datasets, we use Bing maps imagery of 30 cm spatial resolution since we wanted to obtain accurate building classification maps with the CNN. The lower the spatial resolution, the higher are the chances to obtain inaccurate building classification maps, with missing buildings and false positives. Therefore, we recommend the use of imagery with 60 cm or higher spatial resolution that can be obtained from pansharpened images of satellites such as QuickBird, GeoEye, Pléiades, WorldView-2, WorldView-3, and WorldView-4.

In order to train the CNN model that predicts the building probability maps (Section 3.2.1), we use 3134 OSM rural buildings annotations. These OSM annotations were manually verified/corrected on a set of Bing aerial images, that cover 23.75 km^2, acquired over the Geita, Singida, Mara, Mtwara, and Manyara regions of Tanzania. In order to obtain the building probability maps for the Zimbabwe dataset, we finetune the CNN model trained on Tanzania's annotations with a small dataset of 559 building annotations obtained from the region of Matabeleland North in Zimbabwe.

In order to evaluate our methodology, we create validation datasets spatially disconnected from the training regions. The first one is composed of 1094 manually corrected misaligned building annotations located close to the city of Mugumu in Tanzania, where we found OSM annotations with different misalignment orientations. The second dataset is composed of 811 manually corrected misaligned annotations located in the region of Midlands in Zimbabwe. The validation dataset from Tanzania consists of three rural areas, for which we obtained Bing images of sizes (in pixels) 7936×8192, 8192×8192 and 7168×3840, respectively. The validation dataset from Zimbabwe consists of four rural areas that were covered by Bing images of sizes 4096×3328, 4096×3584, 5120×4352 and 5120×4352 pixels, respectively.

3.3.2 Model setup and evaluation procedures

- **Building probability CNN.** For the CNN model that obtains the building probability maps, the numbers of filters in the convolutional layers are 32, 64, 128, and 128, with corresponding kernel sizes of 7×7, 5×5, 5×5 and 3×3. We apply max-pooling (with stride 2 and kernel size 3×3), Rectified Linear Unit (ReLU) as activation function and batch normalization after every convolutional layer. We use 90% rate dropout after on the final fully connected layer. We train the model for 5000 stochastic gradient descent iterations using a learning rate of 0.001 and other 5000 iterations using a learning rate of 0.0001.

- **MRF graph.** As mentioned in Section 3.2.2, we use groups of buildings as nodes of the MRF graph. A building belongs to a group if its center is less than 21 meters away from the center of any of the buildings in this group. In the MRF graph, every group of buildings is then connected to the 5 closest groups of buildings. Both parameters (minimum distance to single buildings for inclusion and number of closest groups) have been set empirically.

- **Alignment with MRF.** The alignment vectors $\mathcal{D} = \{(x, y), x \in \mathcal{D}_x, y \in \mathcal{D}_y\}$ are

defined with $\mathcal{D}_x = \mathcal{D}_y = \{-30, -29, ...0, ..., 29, 30\}$ (values in pixels) based on the maximum expected misalignment. We set the MRF spatial regularization parameter $\beta = 2$ and the maximum number of iterations of the ICM to 10, experimentally. The ICM algorithm has converged before the tenth iteration in all the datasets.

- **Building generation by CNN.** For the CNN model that detects buildings with predefined shapes, the number of filters in the convolutional layers were, 32, 128, 512 and 16 with corresponding kernel sizes of 5×5, 3×3, 3×3 and 1×1. We apply max-pooling (with stride 2 and kernel size 3×3), ReLU as activation function and batch normalization after the first two convolutional layers. In order to select the predicted building candidates to be added to the OSM annotations, we use a threshold value $t = 0.80$, that was found experimentally. This high threshold value is selected to decrease the number of false positives.

We evaluated the performance of the proposed method using the Precision, Recall and F-score metrics with a pixel-level evaluation of the predictions.

3.4 Results

We compare the proposed method for alignment of OSM annotations (`MRFGroups`) with the original annotations (‘`without alignment`’) and the following competitors from the literature:

- ‘`DeformableReg`’, a deformable registration method trained using an unsupervised approach that optimizes a similarity metric between pairs of images [28]. `DeformableReg` analyzes pairs of image patches extracted from building classification maps and OSM annotation maps (of the training set) to generate a displacement vector field. During the inference phase, these vectors allow to perform the registration of the OSM annotation maps into the building classification maps for the test dataset. `DeformableReg` is trained for 10000 stochastic gradient descent iterations using a learning rate of 0.0001.

- ‘`Semantic segmentation`’, the fully convolutional CNN-based segmentation model in [81]. The CNN architecture is composed of several convolutional layers that extract features followed by a deconvolutional layer that output the final per-pixel classification. The model is trained for 5000 stochastic gradient descent iterations using a learning rate of 0.001 followed by 5000 iterations using a learning rate of 0.0001.

In addition to the competitors from the literature, we report results obtained by our model in varying conditions:

- ‘`CorrBuildings`’. When selecting the alignment vectors that maximize the correlation between individual building annotations and a building probability map (equivalent to use our MRF alignment model with $\beta = 0$).

Table 3.1: Pixel-based performance of alignment correction methods for the Tanzania evaluation dataset.

Methods	Precision	Recall	F-score	Time (sec)
`Without alignment`	0.108	0.115	0.111	0
`CorrBuildings`	0.565	0.460	0.507	141.7
`CorrGroups`	0.620	0.658	0.639	102.3
`MRFBuildings`	0.644	0.644	0.644	147.8
`AbsDifference`	0.303	0.322	0.312	41.2
`MutualInfo`	0.570	0.606	0.587	520.6
`MRFGroups` (proposed method)	**0.638**	**0.677**	**0.657**	103.0
`DeformableReg` [28]	0.380	0.500	0.430	29.3

Table 3.2: Pixel-based performance of alignment correction methods for the Zimbabwe evaluation dataset.

Methods	Precision	Recall	F-score	Time (sec)
`Without alignment`	0.526	0.519	0.523	0
`CorrBuildings`	0.793	0.663	0.723	84.9
`CorrGroups`	0.821	0.810	0.816	62.0
`MRFBuildings`	0.832	0.800	0.816	90.1
`AbsDifference`	0.806	0.795	0.800	36.7
`MutualInfo`	0.815	0.804	0.809	428.2
`MRFGroups` (proposed method)	**0.830**	**0.819**	**0.825**	63.9
`DeformableReg` [28]	0.700	0.735	0.717	20.8

- `'CorrGroups'`. When obtaining the alignment vectors that maximize the correlation between groups of buildings and the building probability map.

- `'MRFBuildings'`. When performing the alignment with the proposed MRF formulation, but using individual buildings as nodes of the MRF graph. It is mostly meant to assess computational speedups and the loss of precision when using individual of buildings.

- `'AbsDifference'`. When obtaining the alignment vectors that minimize the sum of absolute difference between groups of buildings and the building probability map.

- `'MutualInfo'`. When obtaining the alignment vectors that maximize the mutual information between groups of buildings and the building probability map.

Table 3.3: Pixel-based and object-based performance of the removal and building addition methods for the Tanzania evaluation dataset.

Methods	Precision	Recall	F-score (pixel)	F-score (object)	Time (sec)
`Semantic segmentation` [81]	0.548	0.819	0.657	0.518	80.0
`MRFGroups`					
+ remove	0.763	0.673	0.715	0.743	103.3
+ remove, then add (by shape priors)	0.727	0.724	0.725	0.690	284.5
+ remove, then add (by semantic segmentation)	0.649	0.776	0.707	0.719	183.8

Table 3.4: Pixel-based and object-based performance of the removal and building addition methods for the Zimbabwe evaluation dataset.

Methods	Precision	Recall	F-score (pixel)	F-score (object)	Time (sec)
`Semantic segmentation` [81]	0.653	0.782	0.712	0.519	41.0
`MRFGroups`					
+ remove	0.837	0.814	0.825	0.846	64.2
+ remove, then add (by shape priors)	0.833	0.817	0.825	0.841	180.1
+ remove, then add (by semantic segmentation)	0.843	0.816	0.829	0.802	105.6

3.4.1 Numerical results

Tables 3.1 and 3.2 present the performances and processing times of several alignment methods for the Tanzania and Zimbabwe evaluation datasets respectively. For the Tanzania dataset (Table 3.1), we can observe that the original misaligned annotations poorly match the actual building footprints visible in the image. All the alignment methods drastically improve the performance of the misaligned annotations. MRF-based methods show better performances than methods based only on correlation. This shows that adding the prior knowledge of smoothness of the alignment vectors helps to improve the results. We can also observe that the alignment methods based on groups of buildings are more effective and efficient than the ones based on individual buildings. For the case of the Zimbabwe dataset (Table 3.2), the performances of the original misaligned annotations are considerably better than the ones of the Tanzania dataset. As in the Tanzania dataset, all the alignment methods considerably improve the performances of the misaligned annotations and the proposed method based on MRF spatial logic applied on groups of buildings outperforms the other baseline alignment methods, as well as the

state-of-art semantic segmentation approach in terms of precision and recall.

Tables 3.3 and 3.4 show the performance of the proposed methods for the removal of incorrect annotations and the addition of new annotations in the two datasets. As a starting point, they use the proposed `MRFGroups`. In order to evaluate the performance of the methods at the object level we consider that a building is detected if its IoU (Intersection over Union) with the ground truth is greater than 0.5. This value corresponds to a misalignment of 2 pixels (60 cm) in both axes when considering the smallest shape (circle) in our dataset.

In the Tanzania dataset, the removal of incorrect annotations considerably improves the precision of the method while maintaining the recall. When the method that used shape priors for adding new buildings annotations is applied, the recall considerably increases. This is at the cost of a slight decrease in precision because of some false positive predictions. However, the gain in recall is larger in the pixel-level evaluation, which is reflected in the improvement of the F-score. Overall in the Zimbabwe dataset the results of the aligned polygons and the result of removing and adding new polygons to the aligned polygons are equivalent. This happens because most of the buildings in the imagery are already well detected and considerably well delineated by the aligned annotations. Thus, few candidates are removed and new building candidates, as predicted by the proposed CNN, are already at their pre-annotated locations. Therefore, very few new candidate buildings are added.

3.4.2 Analysis of shape priors

In Tables 3.3 and 3.4 we also compare our proposed methods with the fully convolutional semantic segmentation approach proposed in [81] (line `'Semantic segmentation'`). As it can be observed, in both datasets the proposed methods achieve better performances than this baseline. Alternatively, one could also use a semantic segmentation method (e.g. [81]) to add new building footprints after running `MRFGroups` and removing incorrect footprints: this result is reported in the last line of both tables (see `'+ remove, then add (by semantic segmentation)'`). In this case, we observe similar numerical performances to our proposed method in terms of F-score. Our proposed method is more precise, while this baseline obtains higher recall values (possibly related to oversegmentation). However, our method has the advantage of returning an output that can be easily converted into vectorial data. As it can be observed in the visual comparisons in Section 3.4.3 (Figure 3.9), our method obtains building predictions with shapes that fit better to the ground truth, not oversegmenting. Also, in cases of objects with shared or very close boundaries, the buildings outlines are easily disentangled, while they cannot be recovered from the semantic segmentation results, since both objects are included in a single blob.

We also evaluate how accurate our method based on shape priors is in differentiating building shapes. To do so, we consider all the newly added buildings showing a considerable overlap (IoU > 0.3) with a building in the ground-truth map. Considering as classes the six basic primitive shapes, the predicted shapes obtains an accuracy of 90.0 %. If we consider as classes the 18 shapes (therefore shape *and* size of the object) an accuracy is

38.3 % is reached. Most common errors are cases where the correct shape is predicted, but not the correct size.

For the evaluation of the geometrical accuracy of the new buildings, we use the average symmetric surface distance metric (ASSD). This metric computes the average distance between all the pixels in the boundary of the predicted object to the closest pixel in the boundary of the ground-truth object. A perfect building prediction will obtain an ASSD value of 0 (the lower the value the better it is). We have computed this metric for all the building predictions that have some overlap with the ground-truth. The average ASSD value for the predictions of the proposed method is 2.54 in the Tanzania dataset, while the method that add buildings based on semantic segmentation obtains an average ASSD value of 2.56.

3.4.3 Visual comparisons

Alignment of footprints

Figure 3.5 presents five examples of groups of rural buildings from the Tanzania dataset. For each example, we show the image, the building probability maps obtained by the hypercolumn model, the original OSM annotations (in yellow) and the aligned annotations obtained by different methods (in other colors). For the proposed method, `MRFGroups`, only the alignment is performed and no removal / addition component is considered in the figure.

- Example 1 (first row). Figure 3.5c shows in green circles the aligned annotations obtained by `CorrGroups`. The alignment results obtained by our proposed method (`MRFGroups`), blue circles in Figure 3.5d, are more accurate, despite missing the bottom building, since the component that adds new building footprints was not used in this case.

- Example 2 (second row). Figure 3.5g shows the alignment results obtained by the MRF applied on individual buildings (`MRFBuildings`, orange circles). One of the buildings was moved to an incorrect location because the values of the probability map are high in a location where there are no buildings. This does not happen in the case of the MRF applied over groups of buildings (`MRFGroups`, blue circles in Figure 3.5h) because we applied the prior knowledge that buildings that are spatially close should be registered with the same alignment vector.

- Example 3 (third row). Figure 3.5k presents the results obtained by the alignment method that uses the correlation on individual buildings (`CorrBuildings`, red circles). We can observe that some building annotations are moved to the same location of high building probability values. The proposed `MRFGroups`, denoted by blue circles in Figure 3.5l, obtains a more desirable alignment, but still an inaccurate one (annotations are shifted to the left side of the buildings) because the building probability map itself is not accurate enough.

- Example 4 (forth row). Figures 3.5o and 3.5p present the results obtained by `AbsDifference` and `MutualInfo`, respectively. We can observe that for both meth-

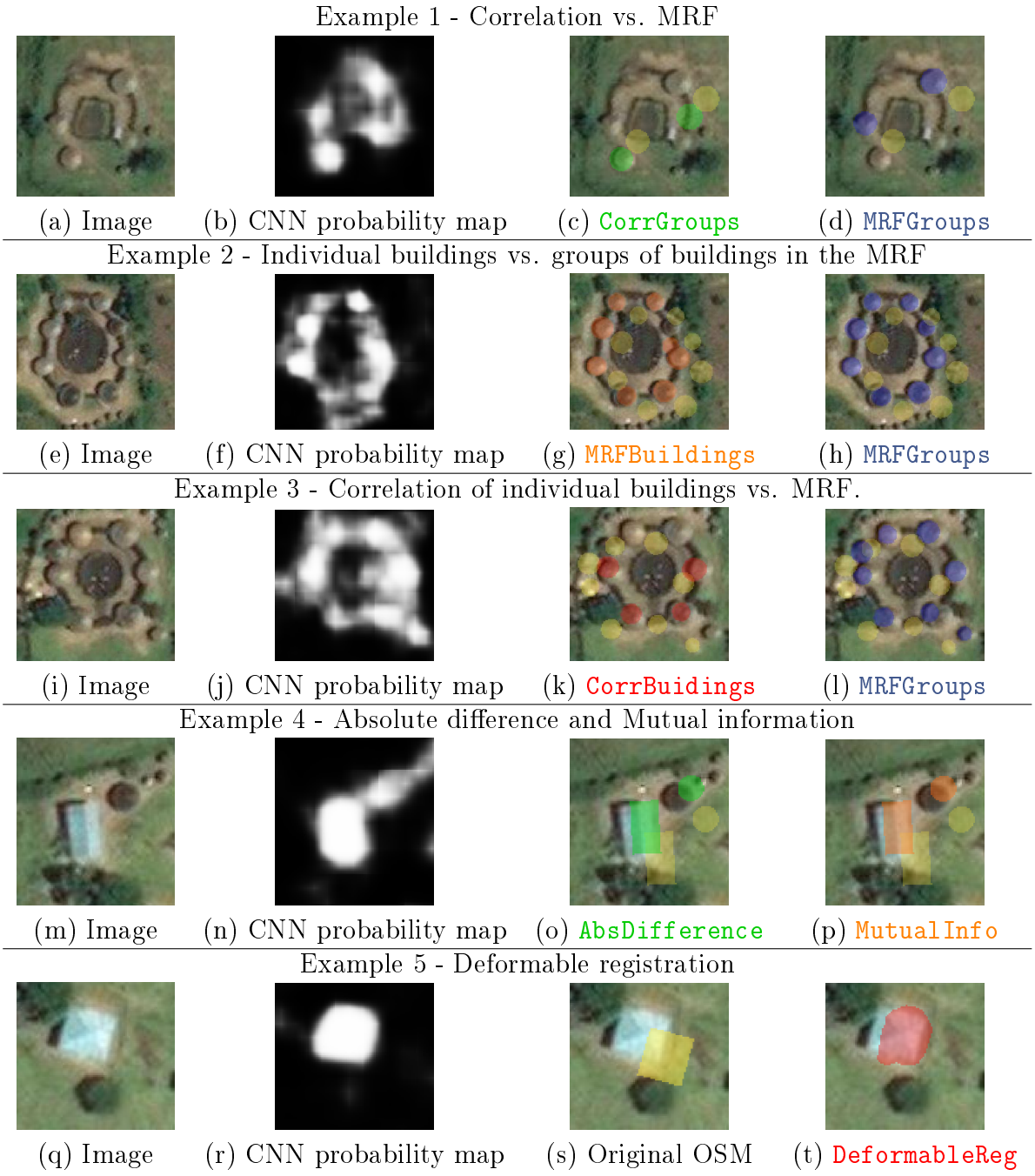


Figure 3.5: Examples of alignment results (the original misaligned annotations are presented in yellow) from the Tanzania dataset.

ods the two building annotations are not well aligned with the objects in the imagery. As for the previous example, this happens mainly because of inaccurate probability maps.

- Example 5 (fifth row). Figure 3.5t presents the result obtained by `DeformableReg` applied to correct the OSM annotation presented in Figure 3.5s. We can observe that the shape of the resulting annotation is inaccurate since it is registered to an object with an inaccurate shape in the building classification map.

Figure 3.6: Examples of alignment results in the Zimbabwe dataset using `MRFGroups`.

Although the proposed MRF based method is more robust to inaccurate building probability maps than the other alignment methods, the quality of the building probability map remains the main factor to compute accurate alignment vectors.

Figure 3.6 illustrates the alignment results for the proposed `MRFGroups` in three examples. In the first case, no alignment is necessary, and `MRFGroups` result is equivalent to the original labels. In the two other cases, `MRFGroups` aligns the buildings correctly, and the removal and addition of footprints is not necessary. This is in line with expectations from this dataset, as we observe that the Zimbabwe dataset has better quality OSM annotations, only requiring geometric alignment. Missing building annotations or incorrect annotations after alignment are rare. This is also reflected in Table 3.2, in which the alignment of the original annotations considerably improved the performance, but the removal and addition of building annotations did not improve the final performance.

Including footprint removals and additions

Figure 3.7 presents results of the methods for alignment (orange), removal of incorrect annotations (green) and addition of new annotations (blue) in the Tanzania and Zimbabwe datasets. For Tanzania dataset example, on the top row, an incomplete set of annotations (Figure 3.7b) is first geometrically aligned so that the large buildings correspond to structures in the image (Figure 3.7c); then, the small structure at the bottom is removed, since there is no evidence that a small building would be located there (Figure 3.7d). One could argue that the removed building corresponds to a small structure at the bottom, but given the relative configuration of the annotations, this is against the image evidence learned by the CNN model. Finally, the second CNN adding new footprints succeeds in adding the two missing large buildings in the right side (Figure 3.7e). For the example from the Zimbabwe dataset (Figure 3.7f), the original OSM annotations (Figure 3.7g) are already well aligned. As a consequence, the alignment correction (Figure 3.7h) and the removal of incorrect annotations (Figure 3.7i) do not change the location of the original annotations. However, two new footprints of missing buildings are correctly added using the second CNN (Figure 3.7j).

Figure 3.8 compares the results obtained by our proposed method (`MRFGroups` followed by the removal and addition of building annotations) with the result of a CNN-based method trained for building segmentation [81]. We can observe that, despite detecting most buildings, the prediction of the CNN segmentation model is not precise, containing

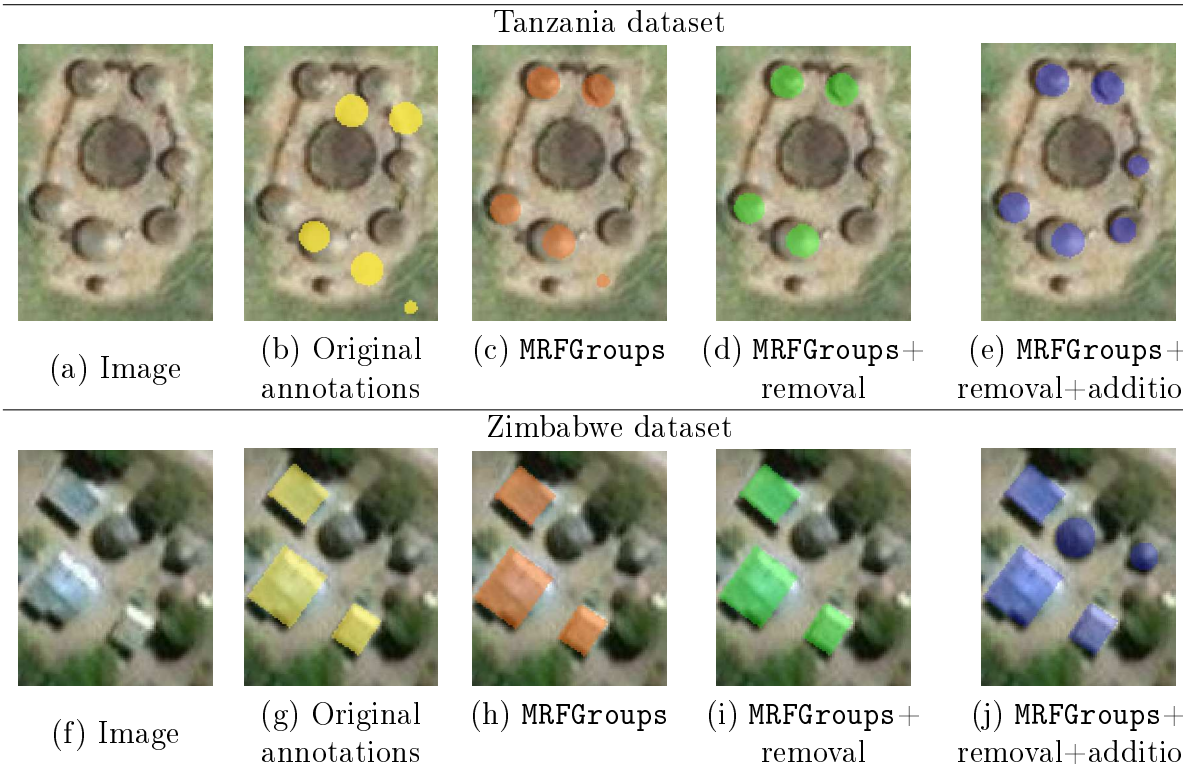


Figure 3.7: Results of our method (the original misaligned annotations are presented in yellow) for the Tanzania and Zimbabwe dataset.

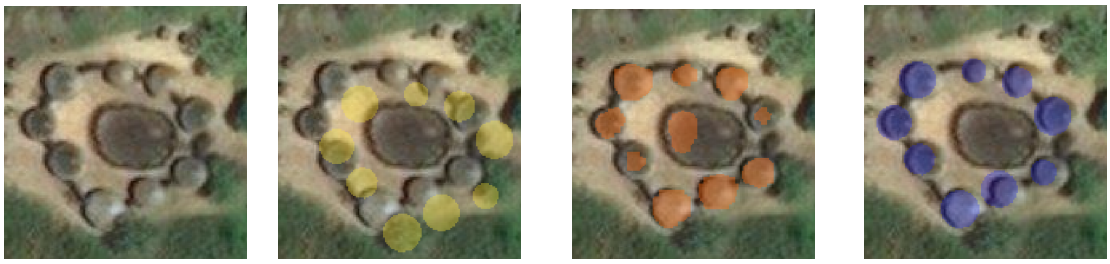
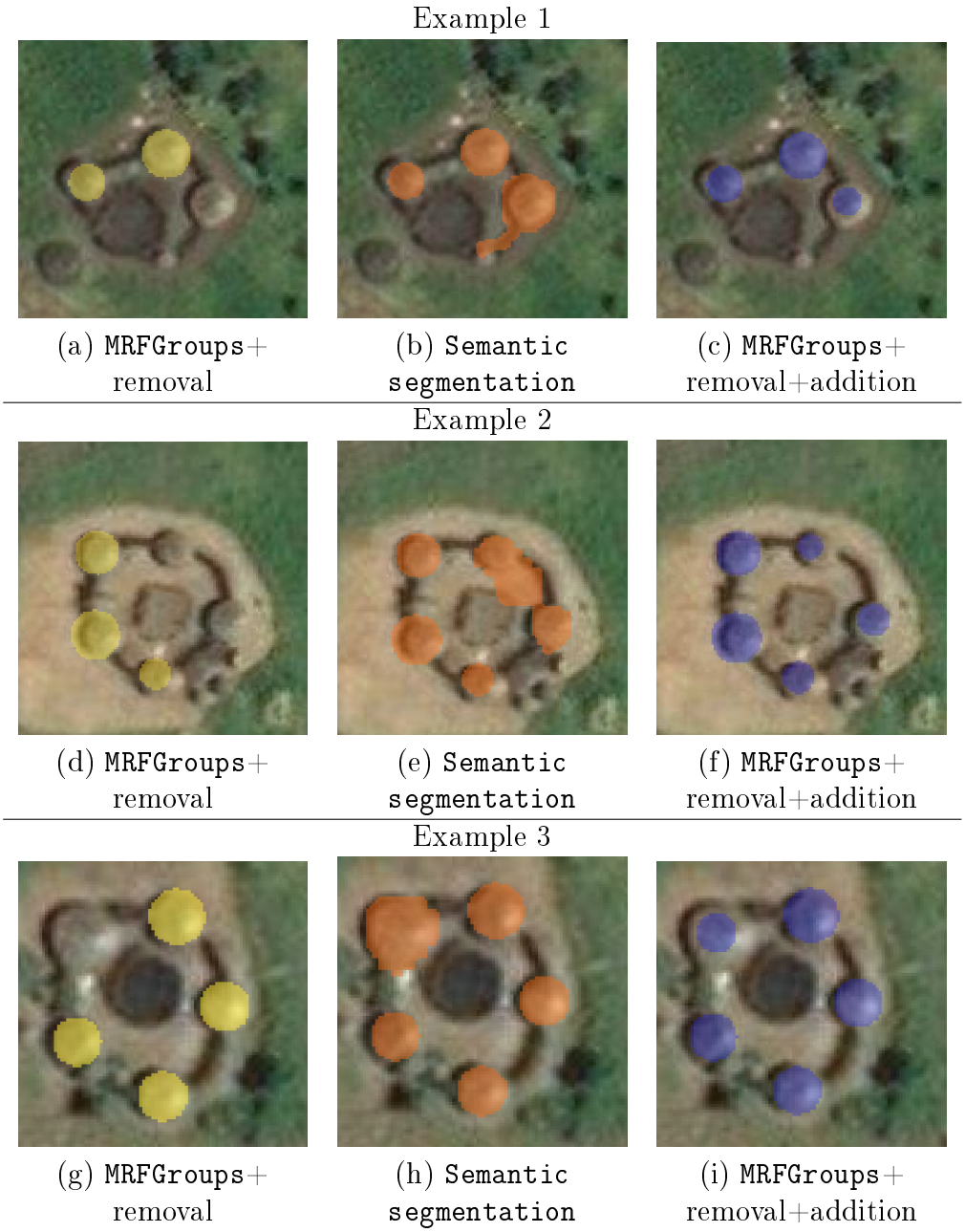


Figure 3.8: Results of our method compared with semantic segmentation [81]: a) Imagery of groups of buildings b) Original OSM annotations (yellow circles) c) Results obtained by using a CNN model trained for building segmentation (orange circles) and d) Annotations, in blue circles, obtained using the propose method (MRF alignment followed by removal and addition of annotations)

several false positive pixels, while our proposed method obtains a better result, more coherent with the shapes of the buildings to be detected.

Figure 3.9 shows three examples of comparisons of the results of adding buildings using a semantic segmentation method [81] and our proposed method for adding building annotations, based on shape priors. The shape of the output of the semantic segmentation method can be very irregular, while our proposed methods obtain predictions that fits



(a) `MRFGroups`+ removal (b) `Semantic segmentation` (c) `MRFGroups`+ removal+addition

(d) `MRFGroups`+ removal (e) `Semantic segmentation` (f) `MRFGroups`+ removal+addition

(g) `MRFGroups`+ removal (h) `Semantic segmentation` (i) `MRFGroups`+ removal+addition

Figure 3.9: Visual comparison of two methods for adding new building annotations, after the alignment and removal of annotations. 1) Add new buildings using the semantic segmentation method proposed in [81] and 2) the proposed method based on shape priors.

better to the ground truth (see examples 1 and 2). In some cases, the prediction of the semantic segmentation method can obtain higher values of IoU with the ground truth than our proposed method since it tends to predict more pixels as buildings (oversegmentation). However, it can also obtain some undesirable results like in Figure 3.9e. Overall, the proposed method leads to a more precise outlining of buildings, easily exportable to vector footprints, and also can disambiguate effectively with polygons with very close boundaries.

3.5 Conclusion

We presented a methodology for correcting rural building annotations in OpenStreetMap. Our methodology consists of three steps: alignment of the original annotations, removal of incorrect annotations, and addition of new annotations of buildings that appear for the first time in the updated imagery. In order to solve the problem of misaligned OSM annotations, we proposed an MRF-based method that encodes the dependency of the alignment vectors of neighboring buildings and maximizes the correlation of aligned annotations and a building probability map learned by a fully convolutional neural network. We used the evidence provided by a building probability map to remove annotations of buildings that no longer exist in the updated imagery. In order to add new building annotations, we learn a second CNN model that predicts building annotations with predefined shapes candidates. We evaluated our methodology in a region of Tanzania that contains misaligned and incomplete/inaccurate annotations and in a region in Zimbabwe that contains mostly misaligned annotations. We observed that the alignment process drastically improves the accuracy of the annotations in the two evaluated datasets. We observed, specially in the Tanzania dataset, that the proposed method for the removal of annotations improves the precision of the annotations and the proposed method for the addition of new annotations considerably improves the recall of the annotations. The proposed methodology will be helpful to reduce the large human effort required to correct existing rural building OSM annotations. As future work, we plan to improve the building delineation results by combining building probability maps learned by CNNs, graph-based segmentation methods, and shape priors.

Acknowledgment

This research was funded by FAPESP (grant 2016/14760-5, 2017/10086-0 and 2014/12236-1), the CNPq (grant 302970/2014-2) and by the Swiss National Science Foundation (grant PP00P2-150593).

Chapter 4

Supporting digital humanitarians in OpenStreetMap: the role of deep learning and human-machine interaction

Abstract

Locating populations in rural areas has attracted the attention of humanitarian mapping projects since it is important to plan actions that affect vulnerable areas. Recent efforts have tackled this problem as the detection of buildings in aerial images. However, the quality and the amount of rural building annotated data in open mapping services like OpenStreetMap (OSM) is not sufficient for training accurate models for such detection. Although these methods have the potential of aiding in the update of rural building information, they are not accurate enough to automatically update the rural building maps. In this paper, we explore a human-computer interaction approach and propose an interactive method to support and optimize the work of volunteers in OSM. It starts by correcting misalignment errors with an aerial image and then uses an efficient building detection/segmentation method to select just the regions in the map that require some correction. The user is asked to verify/correct the selected tiles during several iterations and therefore improving the model with the new annotated data. The experimental results, with simulated and real user annotation corrections, show that the proposed method greatly reduces the amount of data that the volunteers of OSM need to verify/correct.

4.1 Introduction

A large amount of mapping information of buildings has been collected in open and commercial mapping services like OpenStreetMap and Google maps. However, the building mapping data is concentrated mainly in urban areas. Recently, humanitarian organizations like, the Humanitarian OSM Team (HOT) and the Red Cross have created projects to map buildings in rural areas, in order that Non-Governmental Organizations (NGOs)

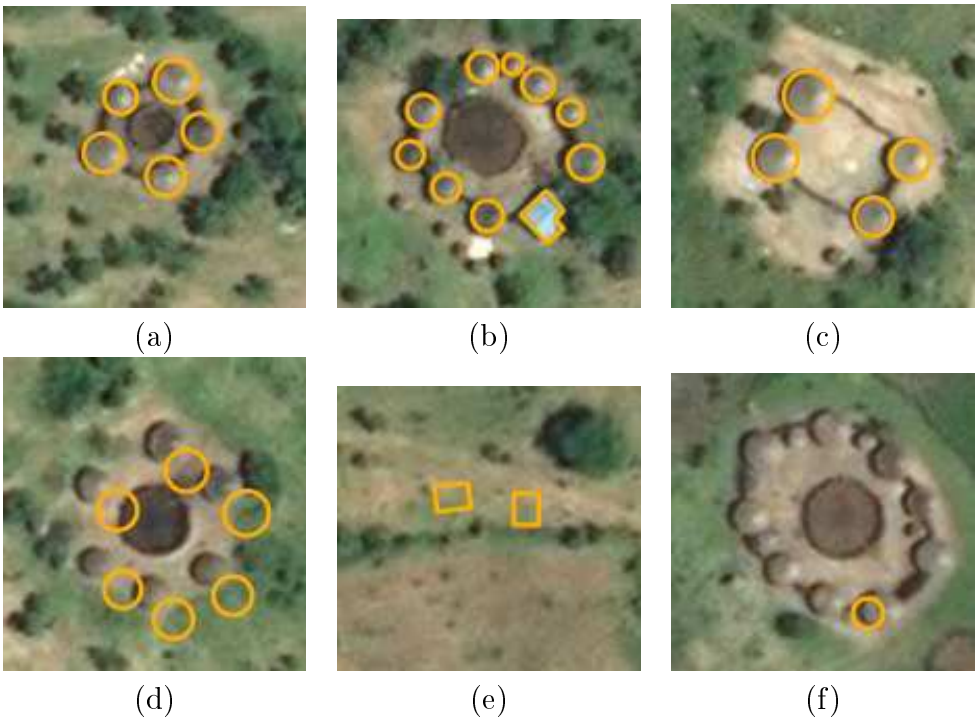


Figure 4.1: Examples of rural building annotations (orange circles), performed by volunteers in OpenStreetMap, superimposed over Bing imagery. a-c) correct annotation, d) annotations with misalignment errors, e) annotations that do not correspond to any building in the aerial images, f) incomplete annotations.

can use the maps to plan actions in response to crises that affect those areas. For instance, the HOT Task manager project [1] provides a platform that includes a web-based map editor in which volunteers manually annotate buildings and other classes of interest in some predefined areas in OSM. The volunteers verify/annotate all the tiles in a selected geographical area by observing aerial imagery. This is a time-consuming task since most of the tiles do not contain the objects of interest in rural areas. Figure 4.1 shows rural building annotations in OSM that were manually digitized by volunteers. Although some annotations are of good quality, several others have issues of misalignment or incompleteness with respect to aerial imagery.

Several deep learning methods have been proposed for segmenting buildings in aerial images [81, 110, 138], with accurate results in urban areas [50]. However, the main drawback of deep learning methods is the requirement of a large amount of labeled data for training. OSM data have been used for several applications including real-time routing [77], autonomous driving [33], 3D building modeling [139] and landuse classification [117, 119]. The authors in [8] and [64] use aerial images and OSM footprint annotations to train deep learning-based methods to perform semantic segmentation of buildings and road networks. More recently, the authors in [22] propose a deep learning-based building detection method that uses data from three open crowdsourced geographic

[1] https://tasks.hotosm.org/

systems: OSM, MapSwipe [2], and OsmAnd [3]. However, the two last resources just provide annotations at the tile level, useful for building patch detection but not for building segmentation. Active learning methods were proposed in [22] and [23] to select better image patches for annotation and then train an effective classifier from those patches. However, these works propose solutions for patch-based building detection which cannot be directly used to automatically update maps of open mapping services, like OSM and Wikimapia [4], because those maps store vectorial footprints that delineate individual buildings.

Although OSM data has been used to train machine learning methods for some applications, the OSM annotations are very often of not good quality [93, 60, 3, 135]. In [60], the authors observed that there exist errors in the tags associated with different types of road networks in OSM data. Other works [3, 2] found several green areas of different classes (e.g., meadows, grass, parks and gardens) that have incorrect labels in OSM. In the case of building annotations, the authors in [135] identified three main issues in building annotations of OSM: i) location inaccuracies (see an example in Figure 4.1d), ii) annotations not related to a building (see Figure 4.1e) and iii) missing annotations (see Figure 4.1f).

In addition to the problem of low data quality in OSM, the amount of available labeled data is small in rural areas as compared to urban regions. This happens because the number of volunteers that update OSM data drops outside cities [97]. This makes more difficult to train accurate models for the segmentation of buildings in rural areas. The authors in [135] propose a methodology to automatically correct rural building annotations with the aforementioned problems in OSM. However, the accuracies attained by this method are not sufficient for production-ready automatic updates of OSM.

An additional problem for the methods that try to automate the process of updating automatically OSM building data is that the output of such methods should be vectorial building footprints. Recently, the authors in [87] proposed a method that outputs vectorial building delineations using a method based on Active Contour Models (ACM), which uses a Convolutional Neural Network (CNN) to learn the parameters of the ACM model. A different approach is proposed in [122] via a mesh-based approximation method that converts binary building classification maps into polygonized buildings. Authors in [135] propose a CNN that detects rural buildings of predefined shapes, which can be easily exported to vectorial format. All these methods strongly depend on the accuracy of the previously computed building map.

In this work, we propose a methodology for interactive correction of rural buildings in OSM. Our proposed method starts by correcting misalignment errors of the annotations, using a Markov Random Field (MRF) [20] approach. From that moment on, the model interacts with a human operator for digitizing objects in OSM. The model selects the image locations that maximize the chances of editing corrections.

In this way, the user just analyzes a small set of selected regions of a large geographical area. Our aim is to reduce the effort of the user for mapping buildings in a certain geographical area, by intelligently selecting just the regions that require user cor-

[2] https://mapswipe.org/
[3] https://osmand.net/
[4] http://wikimapia.org

rections/annotations. The selection of these regions is determined by analyzing a building probability map obtained by a CNN method on aerial images and the current OSM annotations. Given that large areas need to be analyzed to select the most interesting regions for annotation, we propose an efficient CNN model for building segmentation that performs fast inference. To do so, we enrich our CNN with a branch for early stopping when there are no buildings in the analyzed image patch. Whenever a considerable amount of new labeled data is available, the CNN model is retrained to improve the accuracy of the building predictor, and so the accuracy of the selection of tiles that need annotations/corrections. Experiments performed by simulation and real human annotations show that the proposed approach effectively reduces the number of tiles that need to be screened and therefore the human workload.

In Section 4.2, we present the proposed methodology for interactive annotation/correction of OSM rural building annotations. Section 4.3 presents the data set and the setup used in our experiments, and Section 4.4 shows the experimental results. Finally, Section 4.5 presents the conclusions of the paper.

4.2 Methodology

Our proposed methodology is illustrated in Figure 4.2. First, we train a CNN building segmentation model to obtain a building probability map, which is used to perform the alignment of the original OSM polygons. Then, the image being analyzed is split into tiles, in order to select a small number of regions that need manual editing in OSM. This is done based on the probability map and the currently aligned polygons. After the annotation of the selected samples, a stopping criterion is verified to finish the process of annotation. If this criterion is not met, the new building annotations are used to improve the building classification CNN, which in turn improves the process of selection of tiles in the next iteration. The next sections explain in detail the steps of the proposed methodology.

4.2.1 Computation of the building probability map

Given that our model is applied to large areas where rural buildings are sparsely located, we propose an efficient CNN-based method for building segmentation. Our proposed model is based on the U-Net architecture proposed in [108]. U-Net has shown good performance in several applications and recently an ensemble of U-Net models has obtained the best performance in a building segmentation competition, the DeepGlobe challenge [29]. The U-Net model (illustrated in the gray boxes of Figure 4.3) consists of several operations that extract image features and capture contextual information, alongside a symmetric set of operations that upsample the feature maps, therefore enabling precise pixel-level semantic labeling. We extend the U-Net model by adding an extra branch after the third set of convolution groups (see Figure 4.3a). This branch of the model considers that the image features extracted until that point are sufficient to identify whether the image contains buildings. A fully connected layer, denoted as FC, is applied over the aforementioned features to perform binary classification (i.e., there are buildings in the patch or not). In order to train the model with the additional branch, we define a new loss

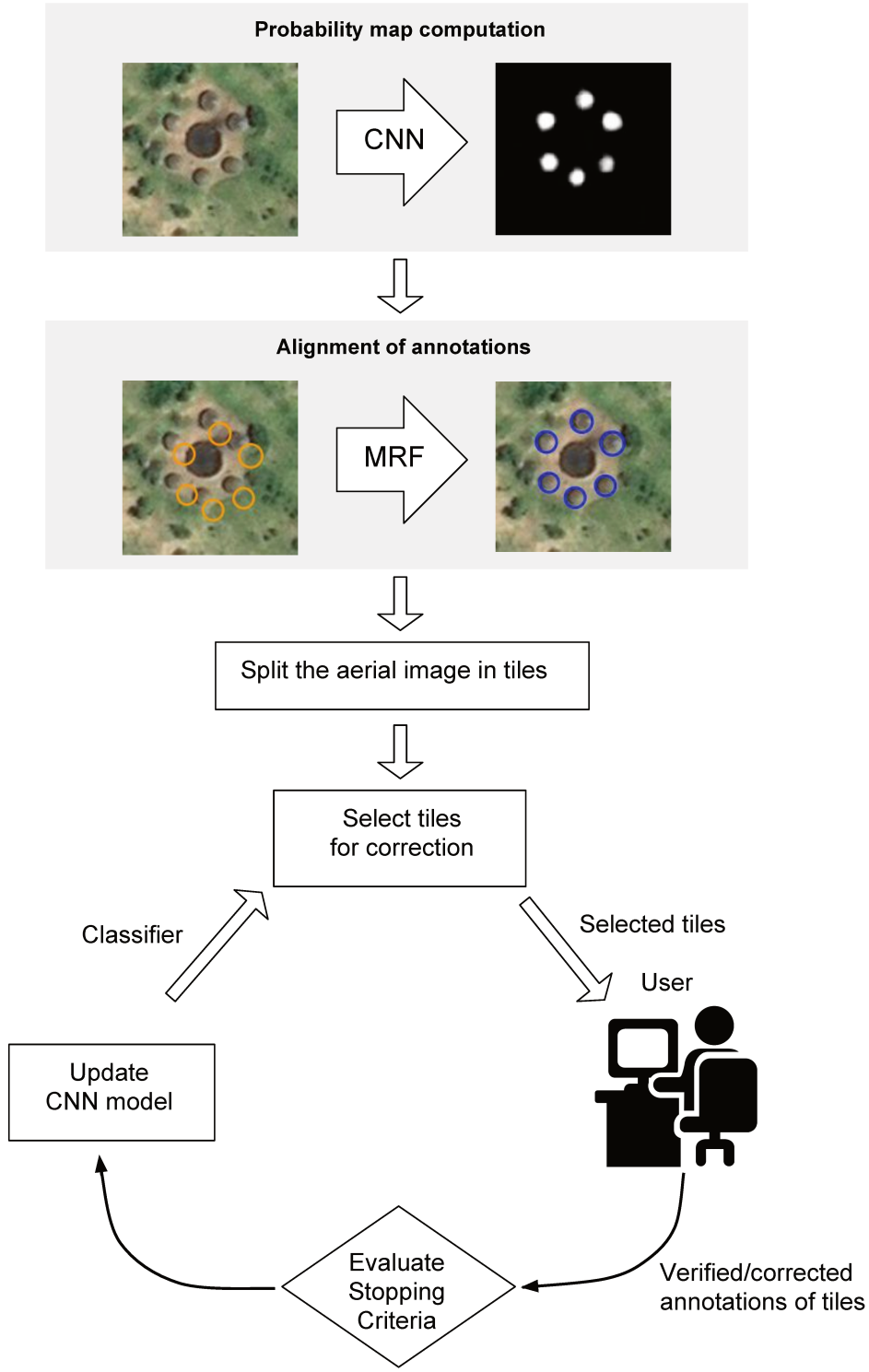


Figure 4.2: The proposed methodology for interactive correct OSM rural building annotations. The orange circles represent misaligned OSM building annotations while the blue circles represent the annotations after the misalignment correction.

function illustrated in Figure 4.3b. Our loss function is defined as the sum of a detection and segmentation loss, both computed using binary cross entropy:

$$L_{seg}(\hat{y}^s, y^s) = -\frac{1}{N}\sum_{i}^{N} y_i^s \log(\sigma(\hat{y}_i^s)) + (1 - y_i^s)\log(1 - \sigma(\hat{y}_i^s)) \quad (4.1)$$

$$L_{det}(\hat{y}^d, y^d) = -\frac{1}{M}\sum_{j}^{M} y_j^d \log(\sigma(\hat{y}_j^d)) + (1 - y_j^d)\log(1 - \sigma(\hat{y}_j^d)) \quad (4.2)$$

$$L(\hat{y}^s, \hat{y}^d, y^s, y^d) = L_{seg}(\hat{y}^s, y^s) + L_{det}(\hat{y}^d, y^d), \quad (4.3)$$

where σ is the sigmoid function, $\hat{y}^s$ is the output of the last convolutional layer of the U-Net model and y^s is the segmentation ground truth map, $\hat{y}^d$ is the detection prediction obtained from FC and y^d is the detection ground truth, N is the number of samples in the processed batch of pixels (segmentation output) and M is the number of samples in the processed batch of images.

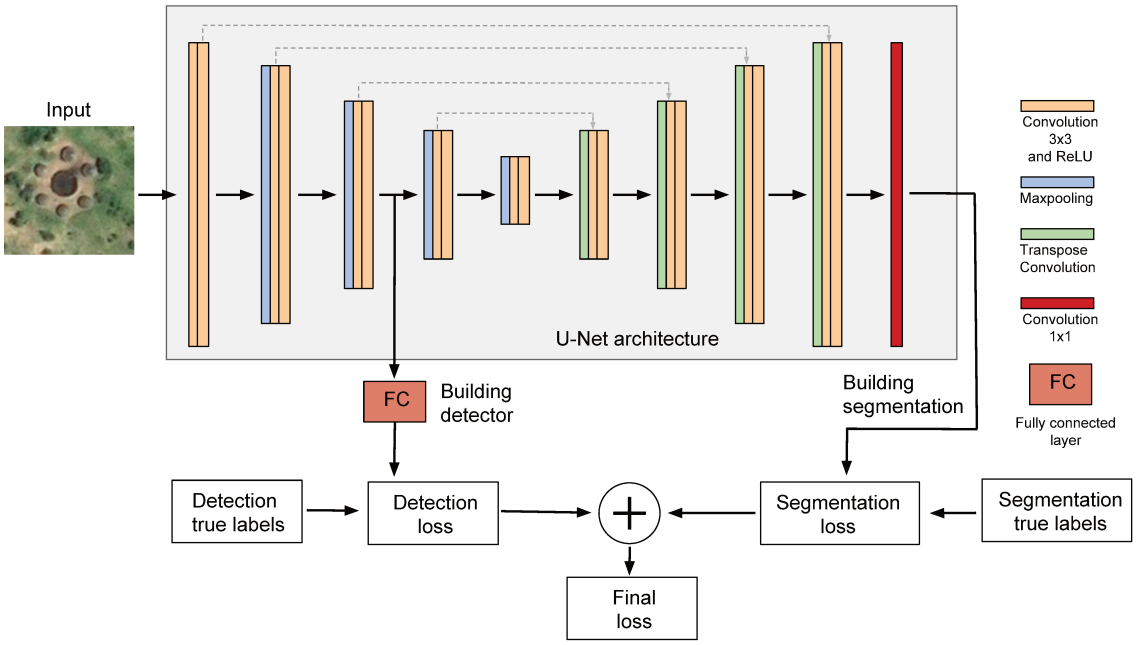
During inference, the proposed model computes sequentially the first three groups of convolutions followed by a fully connected layer and a sigmoid function that outputs the building presence probability value. If this value is higher than a certain threshold θ, then we consider that there exist buildings in the analyzed patch and the rest of the U-Net model is executed to obtain a building segmentation map. Otherwise, we consider that there are no buildings in the image patch and the model outputs an empty probability map. In this way, the inference time is reduced.

4.2.2 Correction of misalignment errors

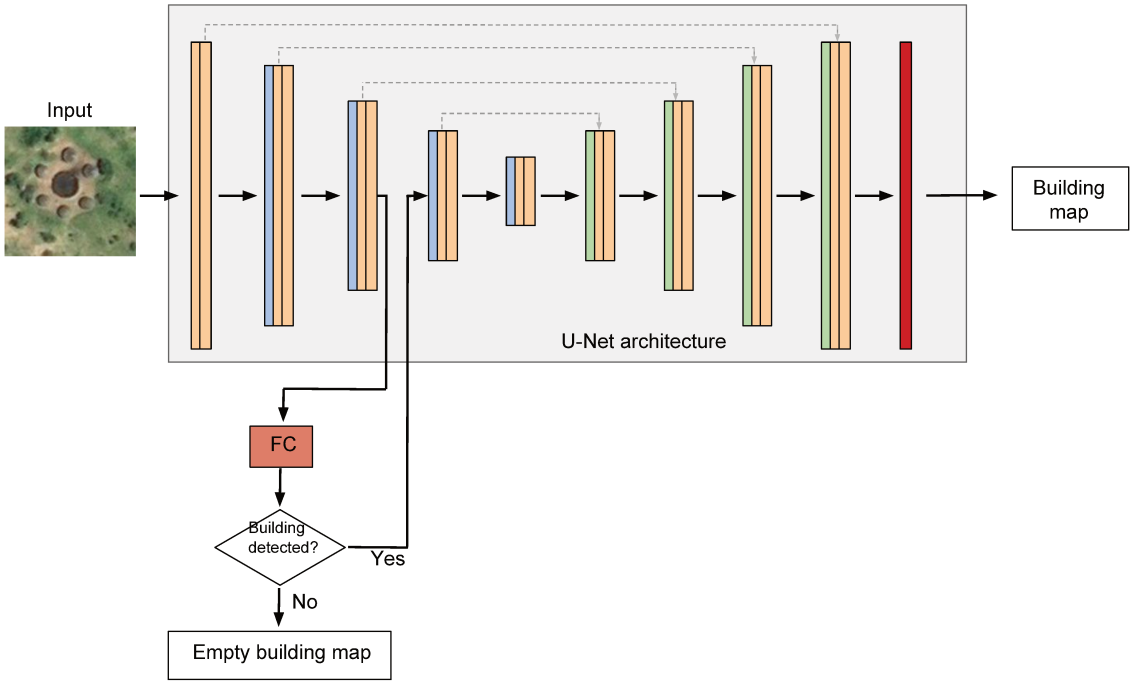
In order to correct alignment errors in OSM, we use the approach proposed in [135] that showed good performances for the alignment of rural building annotations. In that work, the OSM polygons of rural buildings are grouped based on spatial proximity. Then, groups of buildings are aligned with a single shift vector. Aligning groups of polygons makes the method more robust to building probability maps of bad quality. The correlation of the aligned annotations and building probability maps is used to measure the performance of a given alignment vector. Given that nearby groups of buildings have similar registration errors, authors in [135] used MRF to find the alignment vectors $\mathbf{d} = \{d_0, d_1, \ldots, d_n\}$ that can correct the current annotation locations $\mathbf{x}$ using a building probability map $\mathbf{y}$. The optimum alignment vectors minimize the following energy function:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}\in\mathbf{v}^{\mathcal{N}}} \sum_{i} -\log C(A(x_i, d_i), y_i) + \beta \sum_{j\in\mathbf{h}_i} \frac{1}{\mathcal{Z}} ||d_i - d_j||_2, \quad (4.4)$$

where the unary term, that measures the matching between the annotations and probability map, is computed by using the normalized dot product $C(A(x_i, d_i), y_i)$ between the aligned annotations $A(x_i, d_i)$ (A is a function that shifts the position of the annotation x_i using the alignment vector d_i) and the probability map y_i, while the pairwise term is obtained by computing the vector norm of the difference of the alignment vector d_i



(a) Loss computation



(b) Proposed strategy for fast inference

Figure 4.3: Proposed method for building segmentation: a) Loss computation based on a detection and segmentation loss b) Fast inference by using a early exiting brach.

and the neighboring alignment vectors d_j ($j \in \mathbf{h}_i$). In equation 4.4 $\mathbf{v} = \{v_1, v_2, \ldots, v_p\}$ represents the set of all the alignment vectors, $\mathcal{Z}$ is a normalization term computed as the maximum possible distance between two alignment vectors in $\mathbf{v}$ and β is a regularization

parameter. We use the Iterative Conditional Modes (ICM) algorithm to find the solution of equation 4.4.

4.2.3 Selection of tiles for correction/verification

Once the building probability map has been generated and the existing OSM annotations have been aligned to it, volunteers can start the editing process in OSM. The aim of this section is to define a criterion to focus the volunteers' efforts where it really matters (regions with wrong or missing annotations). In this respect, it is crucial to obtain a measure of how incorrect an annotation is, based on the current OSM annotations and a building probability map. Such measure is used to create an ordered list of tiles L, where those whose annotations have more chances to be incorrect are ranked higher. Then, the top t tiles from L are selected for correction/verification by a user. We convert the OSM vectorial polygons to images with value one in the positions that are inside a building annotation, otherwise, zero everywhere else. This is done in order to be able to compare the OSM annotations with the building probability map. In the next sections, we describe the measures proposed.

Mutual information

Mutual information (MI) has been used as an effective similarity metric to compare images [102]. The MI of two variables measures the amount of information that one variable carries about the other. The mutual information of two images A and B can be defined as follows:

$$MI(A;B) = \sum_{b\in B}\sum_{a\in A} p(a,b)\log\left(\frac{p(a,b)}{p(a)\,p(b)}\right), \tag{4.5}$$

where $p(a,b)$ is the joint probability distribution of the pixels that correspond to A and B and $p(a)$ and $p(b)$ are the marginal probability distributions of A and B, respectively. For the MI metric, the lower the value the higher the priority to be selected for verification/correction.

Normalized dot product

In [135] authors used the normalized dot product (NDP) to measure the degree of matching between the building annotations and the building probability map. This measure is defined as follows:

$$NDP(A;B) = \frac{A\cdot B}{size(A)}, \tag{4.6}$$

where A and B are images of the same size, $(\cdot)$ represents the dot product operation and $size(A)$ is the number of pixels in image A. For the NDP metric, the lower the value the higher the priority to be selected for verification/correction.

Sum of absolute differences

The sum of absolute differences (SAD) of the pixel values of two images is an efficient and effective way to measure the degree of matching of two images, as shown in [4]. For

the SAD metric, the higher the value the higher the priority to be selected for verification/correction.

4.2.4 User annotation and evaluation of the stopping criteria

The user verifies/corrects the annotations in the selected tiles. In order to correct annotations, the user applies three types of manual operations that are available in a Graphical User Interface (GUI), for example, the web-based iD editor [5] of OSM: align (drag and drop annotations into the right position), remove and add (digitize manually new polygons). The user performs these operations by visually inspecting the original OSM annotations superimposed over the aerial imagery.

After the user verification/correction of the selected tiles, a stopping criterion is evaluated. We propose as criterion the percentage of tiles that required correction since the last k analyzed tiles, denoted as p_k. This measure is updated after every tile is analyzed by the user. Note that the tiles are selected from a list L, sorted in ascending order in terms of expected correctness of the annotations available, estimated with measures presented in Section 4.2.3. Thus, the more tiles are analyzed by the user, the lower the value of p_k. This happens because, eventually, the majority of the tiles with incorrect annotations will be already analyzed and the majority of the remaining tiles will not need any correction. Once p_k is lower than a given threshold r_k the process of user verification/correction ends.

After a considerable amount of tiles are verified/annotated by the user, the proposed CNN building segmentation model is fine-tuned with the new annotated data. This updated model is then used to improve the building probability maps, which in turn improves the performance of the annotation correctness measures described in Section 4.2.3.

4.3 Data and experimental setup

4.3.1 Datasets

For the validation of our proposed methodology, we used data obtained from the countries of the United Republic of Tanzania and the Republic of Zimbabwe. The CNN building segmentation model was trained with 3134 OSM rural buildings annotations from several regions of Tanzania. These OSM annotations for these regions were manually verified/corrected on a set of Bing aerial images, that cover 23.75 km^2, with 30cm spatial resolution, acquired over the Geita, Singida, Mara, Mtwara, and Manyara regions of Tanzania.

To evaluate our proposed methodology with simulations of user annotation corrections we collected two test datasets, spatially disjoint with respect to the training set. The first dataset is collected in the country of Tanzania and is composed of 1267 OSM building footprints that contain annotation errors. These annotations were located close to the region of Mugumu in Tanzania, this data is called as Tanzania dataset in Section 4.4. The second dataset contains 1392 OSM annotations, that also contains errors, is collected in the region of Midlands in Zimbabwe, called Zimbabwe dataset in Section 4.4. In order to

[5] http://ideditor.com/

perform simulations of user annotations, we manually corrected the annotations in these two datasets.

In order to perform experiments with real user interactions, we used OSM annotations from two regions. The first one located close to Mugumu region in Tanzania, denominated as Mugumu dataset. The second evaluation area is located in the Gweru region, in Zimbabwe, denominated as Gweru dataset.

4.3.2 Model setup

The CNN model was trained for 20 epochs with an initial learning rate of 0.001 decreased by a factor of 0.1 after every 10 epochs. During the interactive annotation process, we finetuned the CNN model for 10 additional epochs and a learning rate of 0.0001 with the new annotated data. For our proposed CNN model, we used $\theta = 0.1$ to predict if an image patch contains or not buildings. As in [135], we used $\beta = 2.0$ for the alingment correction process. We split the images corresponding to the selected geographical regions in tiles of size 256×256 pixels. This tile size was chosen because we observed that it is large enough to cover groups of rural buildings with little background information. In order to evaluate the proposed stopping criteria, we fixed the parameter value $k = 100$ and evaluated the performance of different values $r_k \in \{0.02, 0.05, 0.10\}$. For the experiments performed to choose the tile selection strategy, we took into consideration a common tricky case that affects the metrics MI and NDP. In the case where there are no annotations in a tile but there are a small amount of false positive predictions, the aforementioned metrics will output zero. Then, the method will assign a high priority to the tiles to be verified/corrected by a user, which is not correct since there is nothing to edit in that tile. Thus, for MI and NDP, we verified if the tile does not contain OSM annotations, and if the CNN prediction has a building with size shorter than a very small building in our dataset (20 pixels of size). If that is the case we output the value one which will assign a low selection priority to the tile.

4.3.3 Setup for experiments with real user annotation corrections

So far, the proposed methodology has considered simulated annotation corrections to obtain perfect annotations. In order to evaluate our proposed methodology in a realistic scenario, we also performed a set of experiments with real volunteers performing annotations. To run those tests we deloped a web application that allows several users to correct annotations in OpenStreetMap, by using the iD editor [6] API. Given a geographical area where we want to correct the rural building annotations, our method selects the tiles that need to be verified/corrected by the user with higher priority. When a user asks for a tile for analysis, the web application loads a tile highlighted with a bounding box in magenta in the iD editor. Figure 4.4a depicts the bounding box of the selected tile superimposed over aerial imagery and the current rural building annotation at that location. The user is asked to correct/verify existing annotations, and also to add new building annotations if needed. For instance Figure 4.4b shows the annotations after user corrections, where

[6] http://ideditor.com/

(a) (b)

Figure 4.4: Gaphical User Interface of the iD editor for editing OpenStreetMap annotations: a) Tile (deliminted by the bounding box in magenta) loaded for verification/correction of rural building annotations, b) OSM map after the annotation of two new buildings in the bottom left part of the tile.

two missing buildings were added on the bottom left of the bounding box. For these experiments, 9 volunteers used the web application to interactively correct OSM rural building annotations.

4.4 Results

In this section, we present the results obtained by the proposed methodology. Section 4.4.1 compare the proposed CNN model with the standard U-Net model. Using the computed probability maps obtained by our proposed CNN method we correct alignment errors, as explained in Section 4.2.2. Section 4.4.2 evaluates the alignment results. Then, after correcting existing alignment errors in Section 4.4.3 we compare several approaches to select the tiles that require correction. Section 4.4.4 evaluates the proposed stopping criteria by analyzing the percentage of wrong annotations corrected and the percentage of analyzed tiles when the stopping criteria is met. In Section 4.4.5, we report the results of the experiments including real annotation corrections.

Table 4.1: Performance of the proposed CNN method for building detection and segmentation as compared to the standard U-Net model.

Methods	Tanzania dataset				
	Precision	Recall	F1-score	Time(sec)	Detection recall
Proposed	0.47	0.84	0.60	90.0	0.99
U-Net	0.43	0.72	0.54	157.0	-

4.4.1 Evaluation of the proposed CNN method to compute a building probability map

We compared the efficiency and efficacy of the proposed CNN method with the standard U-Net model, in the Tanzania dataset. The computed probability maps were thresholded to obtain the pixel-based binary classification map (i.e., greater than 0.5 building pixel otherwise background pixel). For the comparison, we used precision, recall, and F1-score at the object-level, as in [135]. Both models were trained with the configuration explained in Section 4.3.2. Table 4.1 shows that the proposed method attains better performance than U-Net, in the three metrics. It is also more than 70% faster than the U-Net model in the evaluated dataset. Our proposed method performs first detection (classifying if an image patch contains buildings or not) and performs semantic segmentation (pixel-based classification) just if needed. It might happen that some image patches classified as not containing buildings actually contain buildings and semantic segmentation should be applied to them. Thus, we measure the detection recall to verify that our model is not having many errors of this type. Table 4.1 shows the detection recall of our proposed CNN method, which attains a very high value.

4.4.2 Evaluation of the alignment method

We applied the alignment method, presented in Section 4.2.2, to the original OSM annotations of the Tanzania dataset. In order to measure how well this method performs we computed the object-level accuracy of overlapping annotations with the ground-truth. This is computed as the number of annotations that have a strong overlap with a ground-truth annotation, Intersection over Union (IoU) greater than 0.5 as in [135], divided by the number of annotations that have at least a very small overlap with a ground-truth annotation (IoU greater than 0.05). The object-level overlapping accuracy of the aligned annotations is 93.9%, which is much better than 12.5% obtained by the original OSM annotations (without alignment). Figure 4.5 presents visual examples of OSM annotations before and after alignment. The first two rows in Figure 4.5 show examples that by aligning the annotations we obtain a correct set of annotations in those locations. The third row shows an example where there is just one OSM annotation when there are many buildings in the aerial imagery. Thus, after applying alignment there are still buildings that need to be digitized by annotators. The fourth row presents an example where all the buildings seem to have a circular shape but the leftmost annotation have a rectangular shape. Thus, after performing alignment the annotator will still need to remove the polygon with wrong shape and digitize a new building annotation.

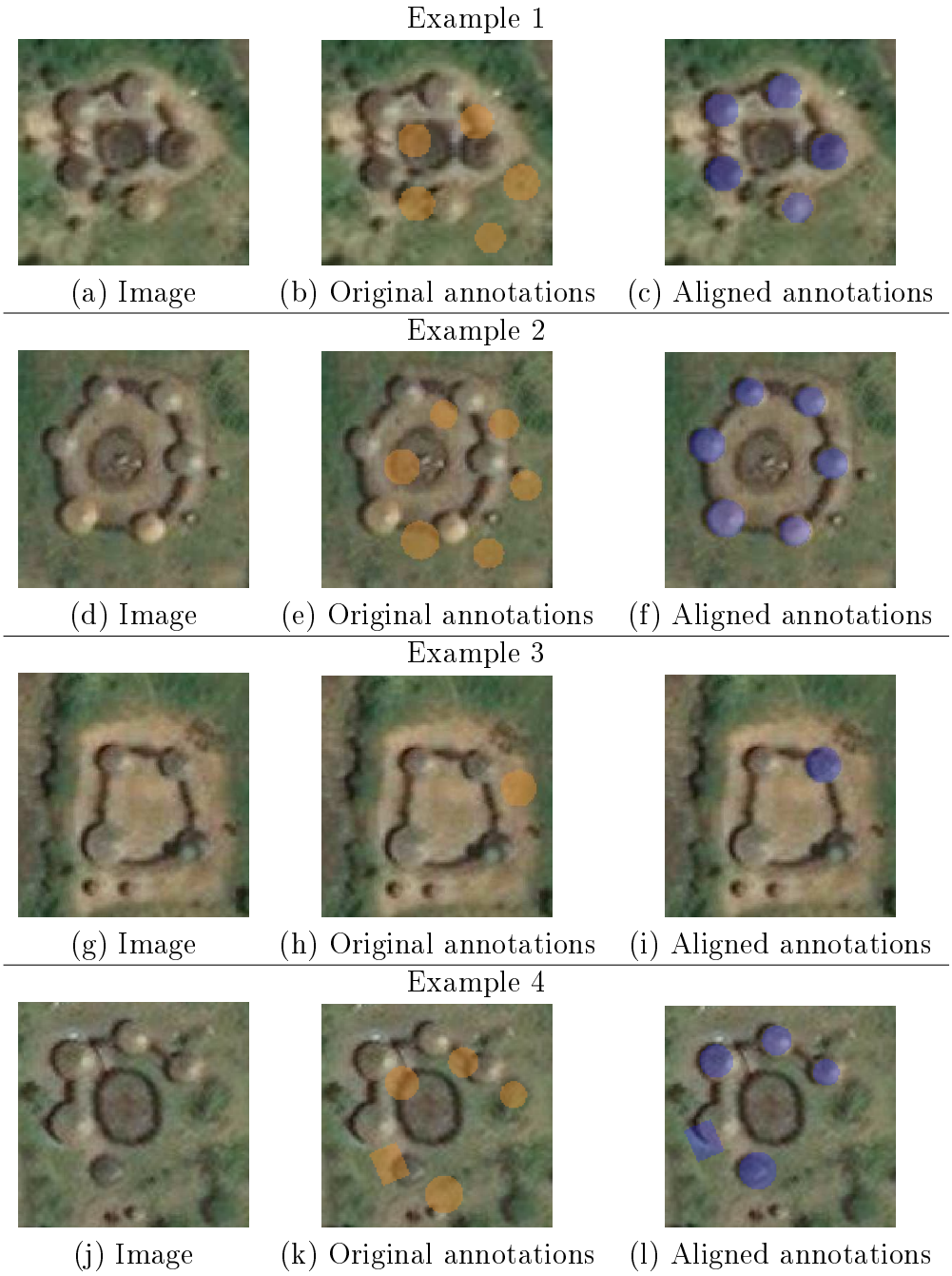


Figure 4.5: Examples of alignment results in the Tanzania dataset. The original misaligned annotations are presented in orange and the aligned annotations in blue.

4.4.3 Tile selection strategy

After correcting the alignment of the original OSM annotations, we evaluated the different selection strategies presented in Section 4.2.3, to find which approach can better rank the tiles that require corrections, and therefore minimize the effort of the user to verify/correct rural building annotations in OSM. We included as a baseline a method that randomly selects tiles to be annotated by the user and also show an upper bound that uses the ground-truth data to select the tiles, denoted as GT. Figure 4.6 shows the

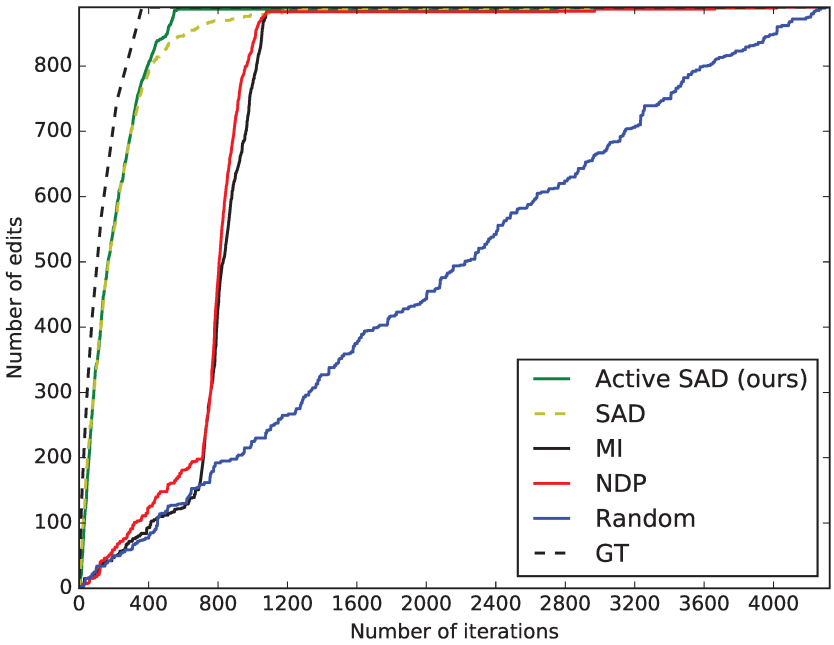


Figure 4.6: Results on the Tanzania dataset of our proposed methodology for interactive annotation of rural buildings in OSM using several measures for error annotation detection. The vertical axis of the plot shows the number of annotation errors detected by the evaluated strategy and the horizontal axis represents the number of tiles analyzed by the user.

results of our proposed methodology in the Tanzania dataset using different strategies to select tiles for annotation: Mutual information (MI), Normalized dot product (NDP) and Sum of absolute differences (SAD). We also show a strategy, called Active SAD, that uses the SAD to select tiles and then updates the CNN segmentation model with the new verified/annotated data (i.e., the method using the complete interactive pipeline in Figure 4.2). We can observe that SAD performs considerably better than MI and NDP and Active SAD performs better than SAD in the last iterations when a great part of the tiles with errors have been discovered and corrected. Figure 4.7 shows the results of the evaluated methods in the Zimbabwe dataset. The same trend can be observed in this dataset, SAD performs better than the other selection methods. However, in this dataset Active SAD is considerably better than SAD. This happens because the building probability maps of the Zimbabwe dataset are less accurate than the probability maps obtained for the Tanzania dataset. Therefore updating the building segmentation CNN model has more impact on the improvement of the selection strategy. Retraining the model with the new corrected annotations adapts the model to the dataset and reduces domain adaptation problems [89].

4.4.4 Stopping criteria

We evaluated the stopping criteria in the Tanzania and Zimbabwe datasets. For the experiments, we used the method Active SAD, since it showed the best performance among

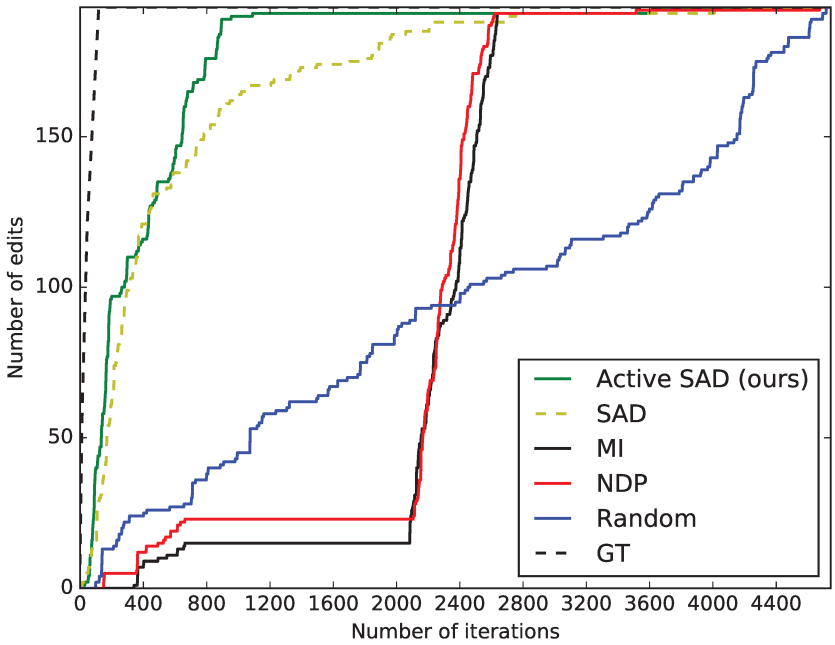


Figure 4.7: Results on the Zimbabwe dataset of our proposed methodology for interactive annotation of rural buildings in OSM using several measures for error annotation detection. The vertical axis of the plot shows the number of annotation errors detected by the evaluated strategy and the horizontal axis represents the number of tiles analyzed by the user.

Table 4.2: Percentage of wrong annotations corrected (% corrected) and percentage of tiles analyzed for three different values of r_k (stopping criteria parameter) in the Tanzania and Zimbabwe datasets.

r_k (%)	Tanzania dataset		Zimbabwe dataset	
	% corrected	% tiles analyzed	% corrected	% tiles analyzed
10	99.7	14.8	52.8	6.0
5	99.7	14.9	87.6	16.3
2	99.7	15.0	98.4	20.9

the other analyzed tile selection approaches (see Section 4.4.3). Table 4.2 shows the percentage of wrong annotations corrected and the percentage of tiles analyzed when the stopping criterion is met. Note that % corrected is computed as the number of wrong annotations shown for user correction divided by the total number of wrong annotations. We can observe that in the Tanzania dataset, when the annotation process stops, most of the wrong annotations were already corrected and around 15% of the tiles were analyzed for all the three values of r_k. In the Zimbabwe dataset $r_k = 10\%$ stops too early the annotation process, and leads to the correction of just around half of the wrong annotations. Using $r_k = 2\%$, almost all the wrong annotations are detected but 20.9% of the tiles are analyzed. The difference in the results in the two datasets is mainly due to the difference in the accuracy of the respective building probability maps.

We also evaluated the behavior of the stopping criteria in datasets that have annota-

tions of different quality levels. To do that we validated the performance of the proposed heuristic in several datasets obtained by simulations of removal, addition and random small shifts of buildings annotations. For a given initial set of annotations, we randomly add and remove a certain number of annotations and then randomly shift the annotations in the horizontal and vertical axis in the range of 0 to 2 pixels. We performed these operations over the Tanzania and Zimbabwe datasets by adding and removing a percentage of the initial number of annotations.

Figure 4.8 shows the results of our proposed method with different stopping criteria parameter values on the Tanzania and Zimbabwe datasets with different quality levels of simulated wrong annotations (different percentages of simulated additions and removals). In the Tanzania dataset (see Figures 4.8a-b)), we can observe that even in the case of very low quality annotations our method can find almost all the wrong annotations for the three differents r_k values (stopping criteria). This at the cost of analyzing around 14-17% of the tiles in the Tanzania dataset.

In the Zimbabwe dataset (see Figures 4.8c-d)), we can observe that the percentage of wrong annotations corrected varies with the quality level of the annotations and the r_k values. When the quality of the annotations is reasonably good (10% of simulated additions and removals) the annotation process stops too early because the probability maps are not of good quality in this dataset and it is more difficult to find tiles with wrong annotations. Thus, the percentage of wrong annotations corrected does not attain very high values. When the annotations are of bad quality (e.g., 40% of simulated additions and removals) it is easier to find the tiles with wrong annotations and the percentage of wrong annotations corrected is very high, at the cost of analyzing around 20% of the tiles in the Zimbabwe dataset.

In general, when the probability maps are accurate higher values of r_k can be used and therefore less amount of tiles will be analyzed by user annotators. On the other hand, when the probability maps are not reliable lower values of r_k should be used to find most of the tiles with wrong annotations at the cost of correcting/verifying more tiles.

4.4.5 Experiments with real users

Table 4.3 presents the results showing the percentage of wrong annotations corrected and percentages of tiles analyzed for different values of r_k (stopping criteria parameter) in the two analyzed datasets. For the Mugumu dataset, we can observe that almost all the wrong annotations were corrected when the annotation process stops using the three values of r_k. The results in the Gweru are similar, but considerably more samples are analyzed when $r_k = 0.02$ than when $r_k = 0.10$. These results using real user annotation corrections confirm the effectiveness of the proposed method in a more realistic scenario. During the experiment, the volunteers commented that at the beginning of the experiment most of the tiles require some annotation correction and the need for correction decreased over time.

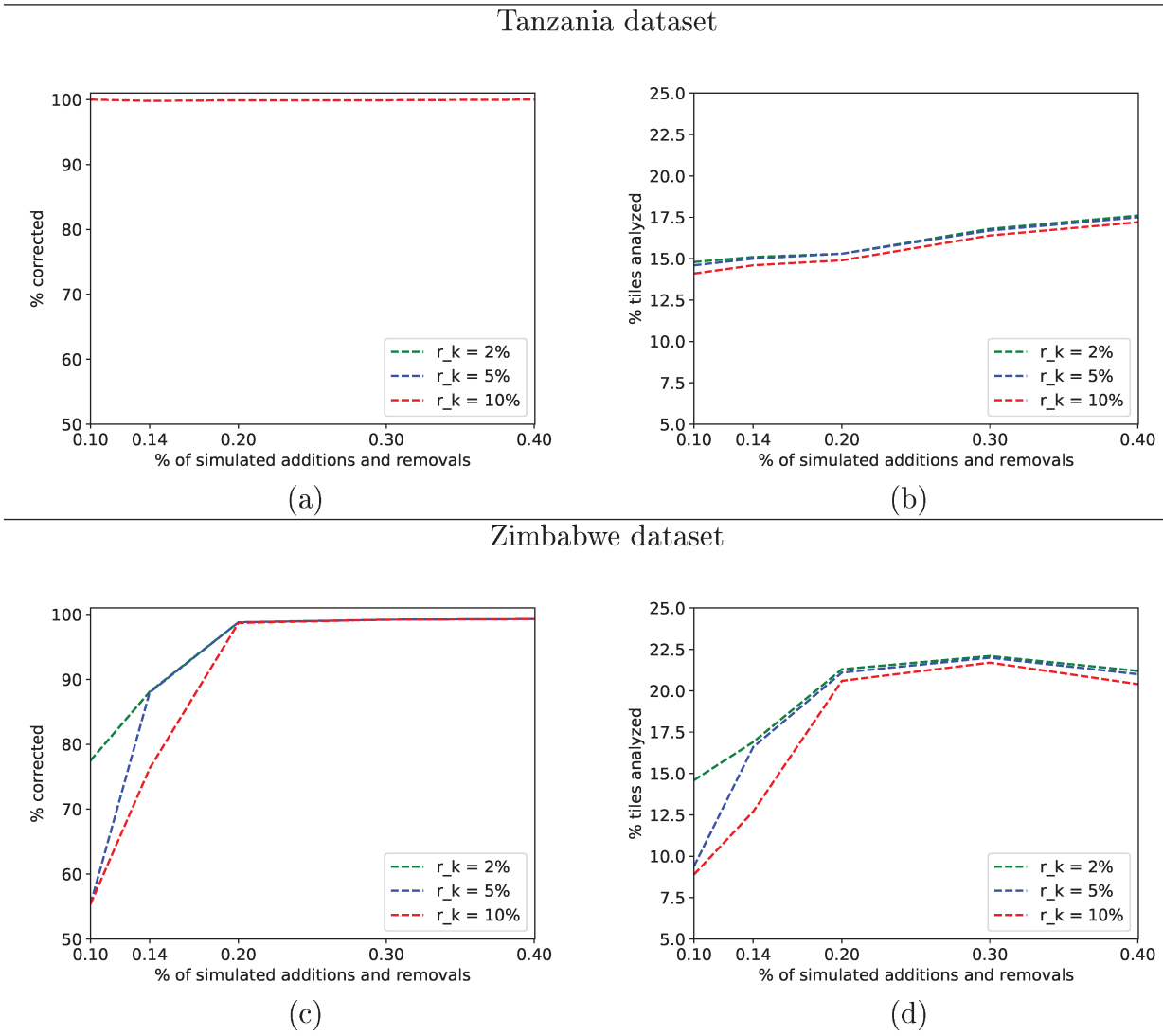


Figure 4.8: Stopping criteria performance using simulated addition, removals and small shifts.

Table 4.3: Results of the experiments with real user interactions. We report the percentage of wrong annotations corrected (% corrected) and percentages of tiles analyzed for three different values of r_k (stopping criteria parameter) in the Mugumu and Gweru datasets.

Experiments with real user interactions				
r_k (%)	Mugumu dataset (Tanzania)		Gweru dataset (Zimbabwe)	
	% corrected	% tiles analyzed	% corrected	% tiles analyzed
10	99.0	22.7	97.9	23.7
5	99.0	23.6	98.6	24.8
2	99.0	24.6	98.6	28.7

4.5 Conclusions

In this work, we proposed a methodology for the interactive correction/verification of rural building annotations in OpenStreetMap. The proposed methodology aims to quickly find a small number of regions that need to be verified by a user, avoiding the task of

exhaustively verifying the large imagery obtained from the analyzed geographical area. In order to analyze such large images, we proposed an efficient CNN building segmentation method to obtain a building probability map. We evaluated several strategies to measure how wrong the current annotations are, based on both the current annotations and the generated building probability map. The sum of absolute differences between the original annotations and the probability map leads to better performances, among the evaluated tile selection strategies. We also observed that the approach of retraining the CNN building segmentation model with the newly annotated/verified data considerably improves the accuracy of the method. The experiments that involve real user annotations show that the proposed stopping criterion allows our method to analyze less than a quarter of the total number of tiles obtained from the image and correct more than 98% of the annotation errors. As future work, we will propose a model that can estimate the number of edits needed to correct a group of annotations and also analyze the annotation skill of different users.

Acknowledgment

The authors would like to thank Bing maps and OpenStreetMap for the access to the imagery and geographical objects' footprints respectively through their APIs. This research was funded by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, grant 2016/14760-5 and 2014/12236-1), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, grant 303808/2018-7), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES, finance code 001) and by the Swiss National Science Foundation (grant PP00P2-150593).

Chapter 5

Interactive coconut tree annotation using feature space projections

Abstract

The detection and counting of coconut trees in aerial images are important tasks for environment monitoring and post-disaster assessment. Recent deep-learning-based methods can attain accurate results, but they require a reasonably high number of annotated training samples. In order to obtain such large training sets with considerably reduced human effort, we present a semi-automatic sample annotation method based on the 2D t-SNE projection of the sample feature space. The proposed approach can facilitate the construction of effective training sets more efficiently than using the traditional manual annotation, as shown in our experimental results with VHR images from the Kingdom of Tonga.

5.1 Introduction

Coconut trees are the primary food and income source for people in several South Pacific countries, especially the developing ones, which makes their detection crucial for environment monitoring and post-disaster assesment. Chong et al. [25] review existing techniques that can be used to automate coconut tree detection in aerial imagery. Most techniques rely on effective image feature extraction (e.g., by scale-invariant feature transform) followed by pattern classification (e.g., by support vector machines or random forest) [84]. More recently, deep-learning-based methods [145] have shown accurate results for tree detection [74, 46]. However, these methods require a high number of annotated training samples, even for fine-tuning a pre-trained model.

Generally speaking, the accuracy of deep learning and other machine learning methods strongly depends on the informativeness, diversity, and number of annotated training samples. The construction of such training sets is not simple, mainly under a limited budget for manual annotation. Active learning methods can assist the user to focus on the manual annotation (label supervision) of a small and yet effective set of samples [128]. These methods usually involve multiple iterations of sample selection based on a previous

instance of a classifier, label verification/correction of the selected samples by the user, and training of a next instance of the classifier with a larger set of annotated samples. It is expected that the classifier becomes effective within a few iterations of active learning. Sample selection usually exploit uncertainty in classification, but it can also add extra criteria to ensure diversity within a single batch or along the iterations [129]. In this work, we propose an interactive approach based on 2D projections for the annotation of coconut trees in aerial images. We use the t-SNE algorithm (t-Distributed Stochastic Neighbor Embedding [78]) to project from the original high dimensional feature space to a 2D coordinate space, for sample annotation. Through navigation in a Graphic User Interface [18], the user has total freedom to chose the area of the feature space to sample from and to priorize exploration versus exploitation of the current model.

Experimental results for coconut detection from VHR drone images of the Tongatapu islands in the Kingdom of Tonga show that more effective classifiers can be obtained by the proposed approach with considerably less human effort than using manual annotation.

5.2 Methodology

Our methodology for coconut tree annotation consists of six steps: i) annotation of a small initial training set, ii) feature extraction, iii) candidate selection, iv) feature projection, v) interactive annotation in the projected space and vi) classification. These steps are detailed in the this section.

i) Annotation of a small initial training set. Initially, the user manually annotates a small balanced set of training samples (a few tens of image patches) containing examples of trees and non-trees (referred to as 'background' hereafter). Figure 5.1 illustrates the process of manual annotation of samples in the aerial imagery. The user clicks at the center of a coconut tree and a bounding box of fixed size is drawn around the tree (red squares in Figure 5.1). Similarly, the user is asked to annotate some background examples (blue squares in Figure 5.1). The image patches (samples) represented by the bounding boxes are used later for feature extraction.

ii) Feature extraction. In order to train an initial classifier, feature vectors must be extracted from the annotated image patches. We use Bag of visual Words (BoW) for feature extraction, since it has shown good performances in remote sensing data [132], in particular for the problem of object detection in submetric drone images [107]. To compute BoW features, we use dense sampling inside the image patch and Scale-invariant feature transform (SIFT) as local region descriptors.

iii) Candidate selection. We train an initial Linear Support Vector Machine (SVM) classifier with the small set of manually annotated samples at hand. In order to generate candidates (unlabeled image patches) for further annotation, we use the sliding-window approach with a small stride to increase the chances of accurately selecting candidates centered at coconut trees. Then, we extract the BoW features from all candidates and classify them with the pre-trained SVM classifier to obtain a probability of containing a coconut tree.

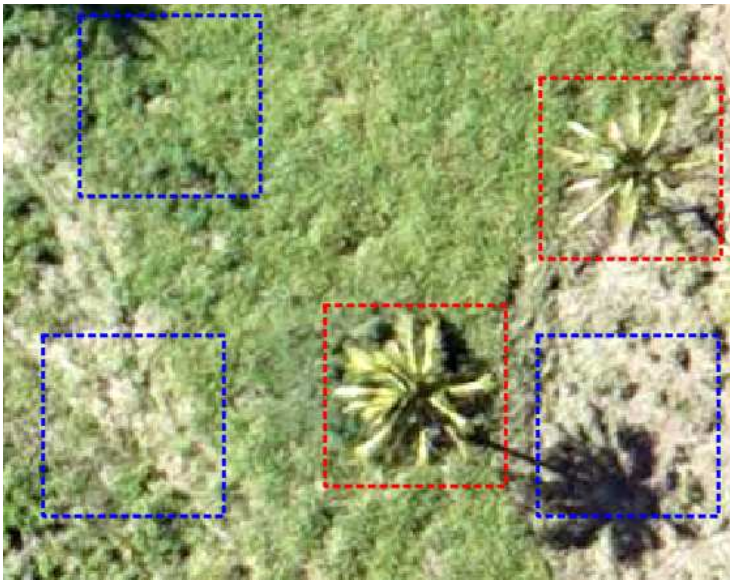


Figure 5.1: Manual annotation of coconut trees (red squares) and background (blue squares) samples.

Given that the number of candidates obtained by the sliding-window approach is very large, we select a relatively small number of coconut tree candidates, by choosing only those with probability greater than θ_{coco} (as given by the SVM posterior probability score) and removing overlapping candidates using standard non-maximum suppression. Similarly, we select background candidates with probability less than θ_{bg} and perform non-maximum suppression.

iv) Feature projection. The BoW features extracted from the labeled (samples initially annotated by the user) and unlabeled samples (selected candidates) are projected on to a 2D coordinate space by using the t-SNE algorithm [78]. This method performs a non-linear dimensionality reduction by preserving the position of the samples with respect to their neighbors when projecting from high to lower dimensions (typically 2D). t-SNE is known to preserve the distribution of the data on the projection spaces better than other methods in the literature [78] and can be used to design better classifiers by exploiting human-machine interaction [104].

v) Interactive annotation in the 2D coordinate space. Figure 5.2 illustrates the GUI for interactive annotation of image patches. The features of the unlabeled candidates are visualized as gray points, while red and blue points represent already labeled patches.

Additionally, we represent the classification probability of the candidates by showing background regions in the plot with intense red color for regions where candidates are being classified with high certainty of being a coconut tree (light red when such probability is small) and intense blue colors for the background class. In this GUI, the user is able to select a group of candidate patches (orange points in the left panel of Figure 5.2) and see their images and corresponding predicted label (center panel of Figure 5.2). By doing so, the selection is driven by the neighborhood in the feature space, rather than the geographical space used in traditional labeling on the VHR image. This type of

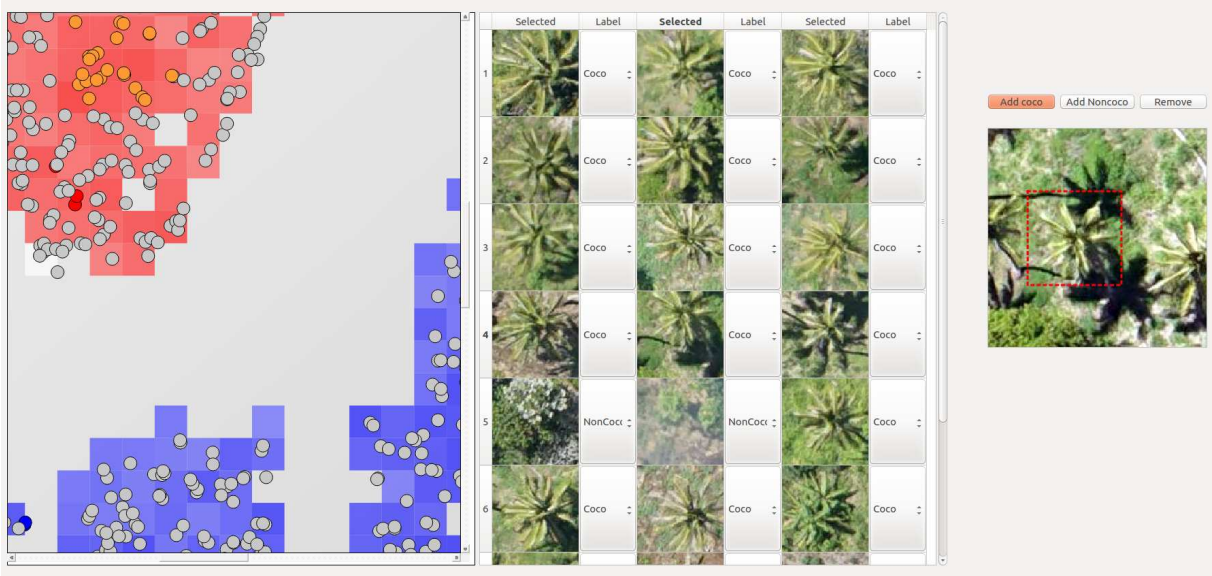


Figure 5.2: Graphical User Interface (GUI) used for interactive annotation in the 2D coordinate space.

visualization is desirable since it can exploit the manifold assumption directly: most samples in a particular region of the feature space should belong to a same class. So, the manual annotation of selected candidates becomes the task of correcting the labels of a few samples that do not belong to the predominant class in the selected region. This facilitates the annotation of several samples at once. Additionally, the user can also re-center poorly centered patches (right panel of Figure 5.2).

The user has total freedom of the selection strategy: he/she might select samples from spatially distant unannotated regions in the plot to annotate diverse samples or focus on uncertain regions, which are those highlighted respectively with light red or blue colors. After a certain time of annotation, the plot is updated using the new training samples and the new classifiers' confidence scores.

vi) Classification. Once the user is confident that the interactive annotation process has provided a large set of mostly correct annotated samples, we fine-tune a Convolutional Neural Network, more specifically AlexNet [68], to perform the final classification. As we will show in Section 5.3.2, this approach obtains better numerical results than our original feature vector (BoW) and classifier (Linear SVM). Given that usually there are overlapping patches classified as coconut trees, which correspond to the same coconut tree, we applied a non-maximum suppression to remove redundancy in the final results.

5.3 Experiments

5.3.1 Dataset and Setup

In order to evaluate our proposed method, we used Unmanned Aerial Vehicle (UAV) imagery provided by the Open AI challenge of WeRobotics [1]. This data was collected by UAVs for the Disaster Resilience Program on the Tongatapu island in the Kingdom of Tonga in October 2017. The provided optical imagery has a spatial resolution of 8cm and contains three bands (Red, Green and Blue). For validation, we used reference data of coconut trees from OpenStreetMap. In order to perform the annotation and validate the accuracy of the trained classifier, we cropped two disjoint regions from the original aerial imagery. The first image crop was used for annotation and contained 1'771 coconut trees, and the second crop was used as the test set and contained 1'373 coconut trees. The size of the candidate image patches was set to 90×90 pixels, in order to capture the coconut tree with a few of background information. To extract the candidates with a sliding window we used a stride of 15 pixels. We set $\theta_{coco} = 0.8$ and $\theta_{bg} = 0.2$ experimentally.

We used a time budget for annotation, corresponding to a maximum time period an operator is asked to annotate the trees. Annotation is performed in two steps. First, the user manually annotated for 5 minutes using the original image to obtain the initial training set. This initial training set was used as starting point for both the approaches considered: the proposed interactive annotation in the projected t-SNE space and a traditional annotation by exploring the image without human machine interaction. After obtaining the initial training set, we performed 20 minutes of annotation. In the projection space-based approach, we recomputed the t-SNE 2D plot (with the new training samples) every 5 minutes.

For the classification phase, we performed non-maximum suppression using a threshold of 0.10 of Intersection over Union to remove redundant candidates for both annotation approaches and classifiers. Regarding the CNN, we finetuned the AlexNet model trained on ImageNet for 15 epochs, with and initial learning rate value of 0.001 and reducing it by a factor of 0.1 every 5 epochs.

5.3.2 Results

We compared our proposed method with manual annotation performed directly over the aerial image. By performing traditional annotation in the image space, the user annotated 222 coconut tree samples and 198 background samples. With the feature space-based approach, 371 coconut tree samples and 420 background samples were collected. Table 5.1 shows the numerical performances. As can be observed the models trained with the proposed method for coconut tree annotation achieve better performances than the image-based annotation strategy. As expected, the CNN model largely outperforms the Linear SVM classifier with BoW features. The improvement in F-score is due to an impressive increase in precision (+10% in both cases), but at the price of a small drop in recall. Figure 5.3 shows some examples of the labels predicted by the CNN models trained with

[1] https://blog.werobotics.org/2018/01/11/open-ai-challenge-2/

Table 5.1: Performance of image- and projection space-based annotation strategies using the Linear SVM classifier with BoW features and the AlexNet model.

Classifier	sampling	Prec.	Recall	F-score
BoW+SVM	Image	0.461	0.733	0.566
	Projection space	0.554	0.683	0.612
AlexNet	Image	0.797	0.826	0.811
	Projection space	0.898	0.791	0.841

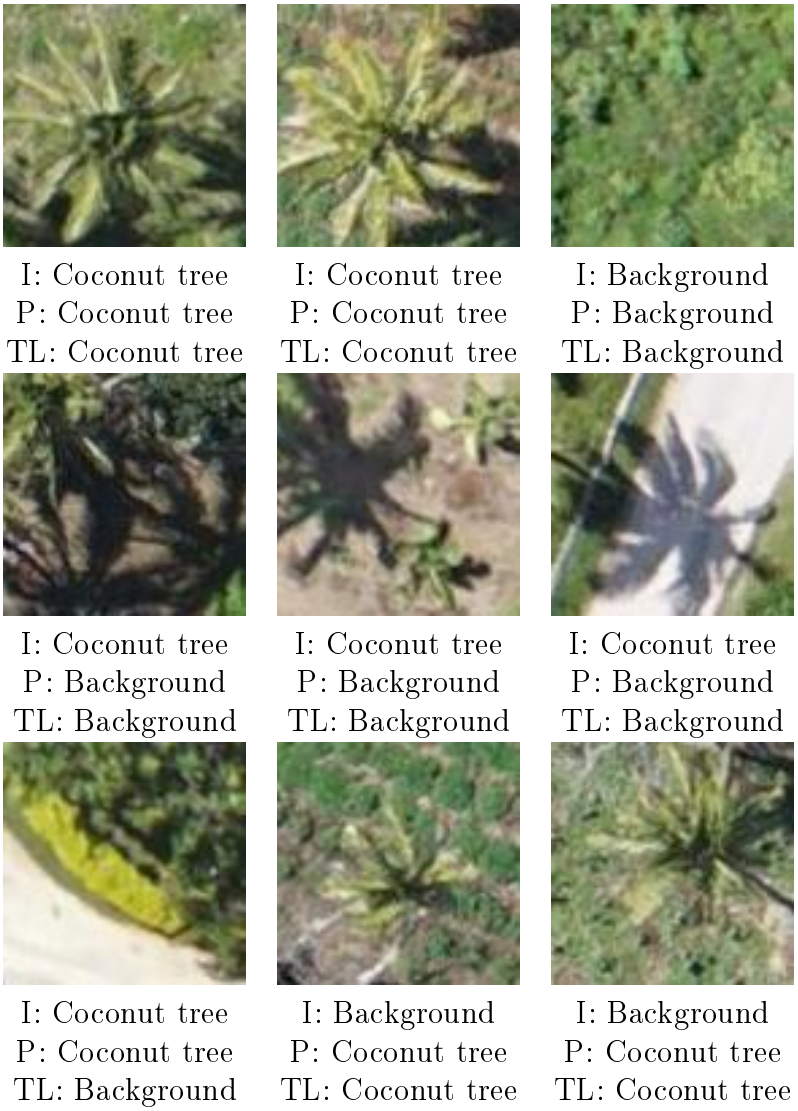


Figure 5.3: Examples of predictions of the CNN models trained using examples from the image approach (I) and the projection space (P) based strategy, together with the true label (TL).

the image- and projection space-based approaches used for sample annotation.

5.4 Conclusion

We presented a workflow for interactive annotation of coconut trees in aerial images. The proposed approach lets the user perform the annotation of samples in a projected 2D space of the original image features. Given that samples from the same class are usually close to each other in the 2D plot, the user is able to annotate several samples in a reduced time because he/she only needs to correct the labels of few samples. The visualization of the candidates in a 2D plot and the highlight by the classifier confidence allow the user to annotate more diverse samples and explore new areas of the feature space efficiently. The experimental results show that with the proposed strategy for coconut tree annotation the user can annotate more samples for a given time budget and also lead to classifier with higher precision than those trained with samples obtained by traditional annotation on the images themselves. Note that the proposed methodology can also be applied to other regularly shaped objects.

Acknowledgment

The authors acknowledge the FAPESP (grant 2016/14760-5, and 2014/12236-1) for funding this project and WeRobotics for providing the aerial imagery.

Chapter 6

Conclusion

In this thesis, we investigated how machine learning methods can be used to improve open geographical data. In particular, we focused our study on improving OpenStreetMap data since it is one of the largest VGI projects and has an active and large community of volunteers. We present methods that can greatly reduce the effort of human annotators during the correction of existing annotations and also for the addition of new ones.

In Chapter 2, we described works that have proposed methods based on machine learning to improve several aspects of OSM, such as the geometry and semantic labels of objects (points of interest, building footprints, and street networks). The earlier methods used intrinsic OSM data (i.e., semantic, geometrical and topological data, exclusively extracted from OSM) and traditional machine learning methods like Support Vector Machines and Random Forest to perform tasks, such as street type recommendation [60]. Recent works have used other sources of information, such as aerial imagery together with deep learning techniques to perform tasks, such as building and road extraction [135, 15].

The use of OSM data for commercial and non-commercial applications is increasing in recent years. Therefore, some companies like Facebook and Telenav, have created projects to improve the quality of OSM data since some of their services rely on that quality. As we show in Chapter 2, OSM data have been used in two ways for applications in other domains: i) as reference data to train a machine learning model (e.g., landuse classification) and ii) as data source to perform feature extraction (e.g., fine-scale population estimation).

In Chapter 3, we propose a methodology to correct building annotations in rural areas. The experimental results show that this methodology can attain a better accuracy than applying semantic segmentation methods. In contrast with methods that perform per-pixel classification, the output of our method is a set of polygonal building footprints that can be easily exported to geographical databases. Despite of the fact that semantic segmentation methods do not obtain accurate delineations of rural buildings they are very useful, in our methodology, to correct alignment errors of existing annotations.

Although the method proposed in Chapter 3 attains a considerable level of accuracy it can not be accurate in some geographical regions. Thus, in Chapter 4, we propose an interactive approach that instead of trying to fully automate the correction of annotations, it optimizes the work of human annotators. We perform an experiment with real human annotators in which most of the annotation errors are corrected by the annotators by

analyzing just a fraction of the dataset. The proposed methodology can be useful to optimize the annotators' labor in online mapping projects, such as Missing Maps [1].

The manual annotation of objects in aerial images is usually a time-consuming task, especially when the number of objects is large. In Chapter 5, we propose an interactive approach to annotate coconut trees in aerial images. This method extracts image features and projects them into a 2D space, in which the user selects and annotates the samples. We observed that usually, the samples of the same classes are close to each other in the 2D space, which facilitates the annotation of several samples in a few user actions. Additionally, by showing to the annotators the classifier confidence in the 2D plot allows the exploration of areas, in the feature space, where the classifier is more uncertain. The methodology allows the user to annotate more samples than traditional visual inspection and help to collect a more effective set of annotated samples to train a classifier.

The synergy between automatic and interactive approaches has a great potential for obtaining up-to-date and accurate OpenStreetMap data, with efficient use of human annotators' effort. Traditionally, information collected in-situ and remote sensing images have been widely used to update maps, but more recently social media data (e.g., pictures and tweets) have shown to be promising resources to collect up-to-date mapping information.

6.1 Future work

In this section, we describe some possible future research directions related to the improvement of the quality of open geographical data:

Improve the extraction of vectorial building footprints from aerial imagery

CNN-based methods have shown to be effective for object detection. However, object delineation obtained by CNN-based semantic segmentation techniques usually requires post-processing steps. Furthermore, in order to obtain a vectorial output from raster classification maps methods like the one proposed in [122] are needed. The method that outputs vectorial footprints proposed in [135] is restricted to buildings with predefined shapes. The authors in [87] propose a method based on Active Contour models that can generate vectorial outputs, but highly depends on a pre-computed building classification map. The results of these methods are not accurate enough for automatic building extraction.

Improve OSM search query responses

The website of OpenStreetMap provides a tool (called Nominatim) to perform searches in the OSM map. However, this tool is limited since it was designed mostly to locate a given address. Other tools, such as Pelias [2] provide additional functionalities like autocompleting the user query. However, none of these tools can appropriately answer natural

[1] https://www.missingmaps.org/
[2] https://pelias.io/

language queries, such as “Where are ATMs in Campinas downtown?". In [70], the authors propose a method that allows users to perform natural language queries over OSM data. However, they observe that their algorithms cannot answer more complex natural language queries.

Learn trust scores of volunteer annotators

The quality of OSM data depends on the annotation skills of the volunteers. OSM does not have a way to rank the volunteers according to their experience or skills. Trust scores of annotators are important to find possible mislabeled objects or inaccurate geometrical data. These indices can be computed using a hand-crafted definition or by using machine learning approaches. For the latter, we could obtain reference data by letting other experienced volunteers to independently re-annotate the same regions that were already annotated by the analyzed volunteer and compare their annotations.

Learning to recommend points of interest for tourism applications

As shown in [109] the points of interest registered in OSM are very useful for tourism applications using augmented reality in mobile devices. We believe that collecting user feedback from the users of such applications are the key to create supervised models to build more effective recommendation systems.

Study the annotation skills of different users

The annotation skills of volunteers are heterogeneous in OSM. There are studies that analyze the total contributions of volunteers in entire cities or countries. However, the analysis of the skills of individual annotators has not been performed. We hypothesize that different annotators are more skilled in particular tasks, for example, some annotators could be good at editing road networks, while others could be better in delineating building footprints.

Incentivizing users to become active volunteers through gamification

In each month between January and May of 2019 less than 1% of the volunteers registered in OSM edited data [3]. A few efforts are made to encourage volunteers to actively annotate data in OSM. Actively assigning annotation tasks to volunteers with a game-like interface and scoring systems could help to increment the number of volunteers that edit data. Some independent gamification projects have been listed in the OSM wiki website [4]. Another interesting project, inspired in reCAPTCHA [5], is Geo-reCAPTCHA that was proposed in [55] to help to delineate building footprints using aerial imagery.

[3] https://osmstats.neis-one.org/
[4] https://wiki.openstreetmap.org/wiki/Gamification
[5] https://en.wikipedia.org/wiki/ReCAPTCHA

Bibliography

[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.

[2] A. L. Ali, Z. Falomir, F. Schmid, and C. Freksa. Rule-guided human classification of Volunteered Geographic Information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:3–15, 2017.

[3] A. L. Ali, F. Schmid, R. Al-Salman, and T. Kauppinen. Ambiguity and plausibility: managing classification quality in volunteered geographic information. In *Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 143–152, 2014.

[4] F. Alsaade. Fast and accurate template matching algorithm based on image pyramid and sum of absolute difference similarity measure. *Research Journal of Information Technology*, 4(4):204–211, 2012.

[5] J. Andreas, A. Vlachos, and S. Clark. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 47–52, 2013.

[6] J. J. Arsanjani, P. Mooney, M. Helbich, and A. Zipf. An exploration of future patterns of the contributions to OpenStreetMap and development of a Contribution Index. *Transactions in GIS*, 19(6):896–914, 2015.

[7] J. J. Arsanjani and E. Vaz. An assessment of a collaborative mapping approach for exploring land use patterns for several european metropolises. *International Journal of Applied Earth Observation and Geoinformation*, 35:329–337, 2015.

[8] N. Audebert, B. Le Saux, and S. Lefèvre. Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1552–1560, 2017.

[9] M. Bakillah, S. Liang, A. Mobasheri, J. Jokar Arsanjani, and A. Zipf. Fine-resolution population mapping using OpenStreetMap points-of-interest. *International Journal of Geographical Information Science*, 28(9):1940–1963, 2014.

[10] A. Ballatore, M. Bertolotto, and D. C. Wilson. Geographic knowledge extraction and semantic similarity in OpenStreetMap. *Knowledge and Information Systems*, 37(1):61–81, 2013.

[11] C. Barrington-Leigh and A. Millard-Ball. The world's user-generated road map is more than 80% complete. *PloS one*, 12(8):e0180698, 2017.

[12] C. Barron, P. Neis, and A. Zipf. A comprehensive framework for intrinsic OpenStreetMap quality analysis. *Transactions in GIS*, 18(6):877–895, 2014.

[13] A. Basiri, P. Amirian, and P. Mooney. Using crowdsourced trajectories for automated osm data entry approach. *Sensors*, 16(9):1510, 2016.

[14] A. Basiri, M. Jackson, P. Amirian, A. Pourabdollah, M. Sester, A. Winstanley, T. Moore, and L. Zhang. Quality assessment of OpenStreetMap data using trajectory mining. *Geo-spatial Information Science*, 19(1):56–68, 2016.

[15] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. Machine-assisted map editing. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '18, pages 23–32. ACM, 2018.

[16] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2018.

[17] O. Benarchid, N. Raissouni, S. El Adib, A. Abbous, A. Azyat, N. Ben Achhab, M. Lahraoua, and A. Chahboun. Building extraction using object-based classification and shadow information in very high resolution multispectral images, a case study: Tetuan, morocco. *Canadian Journal on Image Processing and Computer Vision*, 4:3–14, 2013.

[18] B. C. Benato, A. C. Telea, and A. X. Falcão. Semi-supervised learning with interactive label propagation guided by feature space projections. In *SIBGRAPI Conf. Graphics, Patterns and Images*, 2018.

[19] R. F. Berriel, F. S. Rossi, A. F. de Souza, and T. Oliveira-Santos. Automatic large-scale data acquisition via crowdsourcing for crosswalk classification: A deep learning approach. *Computers & Graphics*, 68:32–42, 2017.

[20] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 68:259–302, 1986.

[21] M. A. Brovelli, M. Minghini, M. E. Molinari, and G. Zamboni. Positional accuracy assessment of the OpenStreetMap buildings layer through automatic homologous pairs detection: The method and a case study. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:615–620, 2016.

[22] J. Chen, Y. Zhou, A. Zipf, and H. Fan. Deep Learning From Multiple Crowds: A Case Study of Humanitarian Mapping. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–10, 2018.

[23] J. Chen and A. Zipf. DeepVGI: Deep learning with volunteered geographic information. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 771–772. International World Wide Web Conferences Steering Committee, 2017.

[24] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3322–3337, 2017.

[25] K. L. Chong, K. D. Kanniah, C. Pohl, and K. P. Tan. A review of remote sensing applications for oil palm studies. *Geo-spatial Info. Sci.*, 20(2):184–200, 2017.

[26] B. Ciepłuch, R. Jacob, P. Mooney, and A. C. Winstanley. Comparison of the accuracy of openstreetmap for ireland with google maps and bing maps. In *Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences*, page 337, 2010.

[27] M. M. Crawford, D. Tuia, and L. H. Hyang. Active learning: Any value for classification of remotely sensed data? *Proceedings of the IEEE*, 101(3):593–608, 2013.

[28] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212. Springer, 2017.

[29] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[30] M. J. Egenhofer and R. D. Franzosa. On the equivalence of topological relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.

[31] J. Estima and M. Painho. Exploratory analysis of OpenStreetMap for land use classification. In *ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, New York, NY, 2013.

[32] H. Fan, A. Zipf, Q. Fu, and P. Neis. Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science*, 28:700–719, 04 2014.

[33] P. Fleischmann, T. Pfister, M. Oswald, and K. Berns. Using OpenStreetMap for autonomous mobile robot navigation. In *Intelligent Autonomous Systems 14*, 2017.

[34] G. Floros, B. van der Zander, and B. Leibe. OpenStreetSLAM: Global vehicle localization using OpenStreetMaps. In *2013 IEEE International Conference on Robotics and Automation*, pages 1054–1059, May 2013.

[35] C. C. Fonte, L. Bastin, L. See, G. Foody, and F. Lupia. Usability of VGI for validation of land cover maps. *International Journal of Geographical Information Science*, 29(7):1269–1291, 2015.

[36] S. Funke, R. Schirrmeister, and S. Storandt. Automatic extrapolation of missing road network data in OpenStreetMap. In *Proceedings of the 2nd International Conference on Mining Urban Data-Volume 1392*, pages 27–35, 2015.

[37] S. Funke and S." Storandt. Automatic Tag Enrichment for Points-of-Interest in Open Street Map. In *Web and Wireless Geographical Information Systems*, pages 3–18, Cham, 2017. Springer International Publishing.

[38] L. Gervasoni, S. Fenet, R. Perrier, and P. Sturm. Convolutional neural networks for disaggregated population mapping using open data. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 594–603. IEEE, 2018.

[39] J. F. Girres and G. Touya. Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS*, 14(4):435–459, 2010.

[40] C. A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, 55(6):532–537, 1993.

[41] B. Glocker, A. Sotiras, N. Komodakis, and N. Paragios. Deformable medical image registration: Setting the state of the art with discrete methods. *Annual Review of Biomedical Engineering*, 12:219–244, 2011.

[42] M. Goetz and A. Zipf. Using Crowdsourced Geodata for Agent-Based Indoor Evacuation Simulations. *ISPRS International Journal of Geo-Information*, 1(2):186–208, 2012.

[43] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

[44] M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[45] A. Graser. Integrating Open Spaces into OpenStreetMap Routing Graphs for Realistic Crossing Behaviour in Pedestrian Navigation. 1:217–230, 06 2016.

[46] E. Guirado, S. Tabik, D. Alcaraz-Segura, J. Cabello, and F. Herrera. Deep-learning versus obia for scattered shrub detection with google earth imagery: Ziziphus lotus as case study. *Remote Sens.*, 9(12), 2017.

[47] C. Haas and S. Riezler. A Corpus and Semantic Parser for Multilingual Natural Language Querying of OpenStreetMap. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 740–750, 2016.

[48] S. Hahmann, J. Miksch, B. Resch, J. Lauer, and A. Zipf. Routing through open spaces – A performance comparison of algorithms. *Geo-spatial Information Science*, 21(3):247–256, 2018.

[49] M. Haklay. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning B: Planning and design*, 37(4):682–703, 2010.

[50] Ryuhei Hamaguchi and Shuhei Hikosaka. Building detection from satellite imagery using ensemble of size-specific detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 223–2234, 2018.

[51] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 447–456, 2015.

[52] P. Hashemi and R. A. Abbaspour. *Assessment of Logical Consistency in OpenStreetMap Based on the Spatial Similarity Concept*, pages 19–36. Springer, 2015.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[54] R. Hecht, C. Kunze, and S. Hahmann. Measuring completeness of building footprints in openstreetmap over space and time. *ISPRS International Journal of Geo-Information*, 2(4):1066–1091, 2013.

[55] F. Hillen and B. Höfle. Geo-reCAPTCHA: Crowdsourcing large amounts of geographic information from earth observation data. *International Journal of Applied Earth Observation and Geoinformation*, 40:29–38, 2015.

[56] S. Huber and C. Rust. Calculate travel time and distance with openstreetmap data using the open source routing machine (osrm). *The Stata Journal*, 16(2):416–423, 2016.

[57] M. Jilani, M. Bertolotto, P. Corcoran, and A. Alghanim. Traditional vs. Machine-Learning Techniques for OSM Quality Assessment. In *Geospatial Intelligence: Concepts, Methodologies, Tools, and Applications*, pages 469–487. IGI Global, 2019.

[58] M. Jilani, P. Corcoran, and M. Bertolotto. Automated quality improvement of road network in OpenStreetMap. In *Agile Workshop (Action and Interaction in Volunteered Geographic Information)*, page 19, 2013.

[59] M. Jilani, P. Corcoran, and M. Bertolotto. Multi-granular street network representation towards quality assessment of OpenStreetMap data. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science*, page 19. ACM, 2013.

[60] M. Jilani, P. Corcoran, and M. Bertolotto. Automated highway tag assessment of OpenStreetMap road networks. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 449–452, 2014.

[61] M. Jilani, P. Corcoran, and M. Bertolotto. Probabilistic graphical modelling for semantic labelling of crowdsourced map data. In *Intelligent Systems Technologies and Applications*, pages 213–224. Springer, 2016.

[62] M. Jilani, P. Corcoran, and M. Bertolotto. A Multi-layer CRF Based Methodology for Improving Crowdsourced Street Semantics. In *Proceedings of the 11th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 29–38. ACM, 2018.

[63] B. A. Johnson and K. Iizuka. Integrating OpenStreetMap crowdsourced data and Landsat time-series imagery for rapid land use/land cover (LULC) mapping: Case study of the Laguna de Bay area of the Philippines. *Applied Geography*, 67:140–149, 2016.

[64] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[65] N. Karagiannakis, G. Giannopoulos, D. Skoutas, and S. Athanasiou. OSMRec tool for automatic recommendation of categories on spatial entities in OpenStreetMap. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 337–338. ACM, 2015.

[66] A. Kashian, A. Rajabifard, K. F. Richter, and Y. Chen. Automatic analysis of positional plausibility for points of interest in OpenStreetMap using coexistence patterns. *International Journal of Geographical Information Science*, 33(7):1420–1443, 2019.

[67] T. Koukoletsos, M. Haklay, and C. Ellul. Assessing Data Completeness of VGI through an Automated Matching Procedure for Linear Data. *Transactions in GIS*, 16, 08 2012.

[68] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[69] C. Kuntzsch, M. Sester, and C. Brenner. Generative models for road network reconstruction. *International Journal of Geographical Information Science*, 30(5):1012–1039, 2016.

[70] C. Lawrence and S. Riezler. NLmaps: A Natural Language Interface to Query OpenStreetMap. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 6–10, 2016.

[71] C. Lawrence and S. Riezler. Improving a neural semantic parser by counterfactual learning from human bandit feedback. *arXiv preprint arXiv:1805.01252*, 2018.

[72] S. U. Lee, S. Y. Chung, and R. H. Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2):171–190, 1990.

[73] Q. Li, H. Fan, X. Luan, B. Yang, and L. Liu. Polygon-based approach for extracting multilane roads from OpenStreetMap urban road networks. *International Journal of Geographical Information Science*, 28(11):2200–2219, 2014.

[74] W. Li, H. Fu, L. Yu, and A. Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sens.*, 9(1):22, 2016.

[75] B. Y. Lin, F. F. Xu, E. Q. Liao, and K. Q. Zhu. Transfer Learning for Traffic Speed Prediction: A Preliminary Study. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[76] S. Lobry, J. Murray, D. Marcos, and D. Tuia. Polygonization of Binary Classification Maps Using Mesh Approximation with Right Angle Regularity. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019. to appear.

[77] D. Luxen and C. Vetter. Real-time Routing with OpenStreetMap Data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 513–516, 2011.

[78] L. Maaten and G. Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(Nov):2579–2605, 2008.

[79] E. Maggiori, G. Charpiat, Y. Tarabalka, and P. Alliez. Recurrent neural networks to correct satellite image classification maps. *IEEE Transaction on Geoscience and Remote Sensing*, 55(9):4962–4971, Sept 2017.

[80] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017.

[81] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transaction on Geoscience and Remote Sensing*, 55(2):645–657, Feb 2017.

[82] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Polygonization of remote sensing classification maps by mesh approximation. In *IEEE International Conference on Image Processing (ICIP)*, pages 560–564, 2017.

[83] I. Majic, S. Winter, and M. Tomko. Finding Equivalent Keys in Openstreetmap: Semantic Similarity Computation Based on Extensional Definitions. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, GeoAI '17, pages 24–32, New York, NY, USA, 2017. ACM.

[84] S. Malek, Y. Bazi, N. Alajlan, H. Hichri, and F. Melgani. Efficient framework for palm tree detection in uav images. *IEEE J. Sel. Topics Appl. Earth Obs. and Remote Sens.*, 7:4692–4703, 12 2014.

[85] L. A. Manfré, E. Hirata, J. B. Silva, E. J. Shinohara, M. A. Giannotti, A. P. C. Larocca, and J. A. Quintanilha. An analysis of geospatial technologies for risk and natural disaster management. *ISPRS International Journal of Geo-Information*, 1(2):166–185, 2012.

[86] D. Marcos, R. Hamid, and D. Tuia. Geospatial correspondences for multimodal registration. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[87] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun. Learning deep structure active contours end-to-end. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[88] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia. Land cover mapping at very high resolution with rotation equivariant CNNs: towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.

[89] G. Matasci, D. Tuia, and M. Kanevski. Svm-based boosting of active learning strategies for efficient domain adaptation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5):1335–1343, Oct 2012.

[90] G. Máttyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *IEEE International Conference on Computer Vision*, pages 3438–3446, 2017.

[91] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing road maps by parsing aerial images around the world. In *IEEE International Conference on Computer Vision*, pages 1689–1697, 2015.

[92] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3611–3619, 2016.

[93] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *International Conference on Machine Learning*, pages 567–574, 2012.

[94] P. Mooney and P. Corcoran. The annotation process in OpenStreetMap. *Transactions in GIS*, 16(4):561–579, 2012.

[95] P. Neis. Measuring the Reliability of Wheelchair User Route Planning based on Volunteered Geographic Information. *Transactions in GIS*, 19(2):188–201, 2015.

[96] P. Neis, M. Goetz, and A. Zipf. Towards automatic vandalism detection in OpenStreetMap. *ISPRS International Journal of Geo-Information*, 1(3):315–332, 2012.

[97] P. Neis and D. Zielstra. Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap. *Future Internet*, 6(1):76–106, 2014.

[98] P. Neis, D. Zielstra, and A. Zipf. The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, 4(1):1–21, 2011.

[99] P. Neis and A. Zipf. Analyzing the contributor activity of a volunteered geographic information project—The case of OpenStreetMap. *ISPRS International Journal of Geo-Information*, 1(2):146–165, 2012.

[100] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[101] M. Over, A. Schilling, S. Neubauer, and A. Zipf. Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *Computers, Environment and Urban Systems*, 34(6):496–507, nov 2010.

[102] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging*, 22(8):986–1004, 2003.

[103] S. Quinn and L. Yapa. Openstreetmap and food security: A case study in the city of philadelphia. *The Professional Geographer*, 68(2):271–280, 2016.

[104] P. E. Rauber, A. X. Falcão, and A. C. Telea. Projections as visual aids for classification system design. *Information Visualization*, 17(4):282–305, 2018.

[105] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[106] Y. Ren, T. Cheng, and Y. Zhang. Deep spatio-temporal residual neural networks for road-network-based data modeling. *International Journal of Geographical Information Science*, 0(0):1–19, 2019.

[107] N. Rey, M. Volpi, S. Joost, and D. Tuia. Detecting animals in african savanna with UAVs and the crowds. *Remote Sens. Environ.*, 200C:341–351, 2017.

[108] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[109] M. Ruta, F. Scioscia, D. De Filippis, S. Ieva, M. Binetti, and E. Di Sciascio. A semantic-enhanced augmented reality tool for OpenStreetMap POI discovery. *Transportation Research Procedia*, 3:479–488, 2014.

[110] S. Saito, T. Yamashita, and Y. Aoki. Multiple object extraction from aerial imagery with convolutional neural networks. *Electronic Imaging*, 2016(10):1–9, 2016.

[111] A. Schilling, M. Over, S. Neubauer, P. Neis, G. Walenciak, and A. Zipf. Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. *Urban and regional data management: UDMS annual*, pages 75–84, 2009.

[112] S. Schmitz, A. Zipf, and P. Neis. New applications based on collaborative geodata—the case of routing. In *Proceedings of XXVIII INCA international congress on collaborative mapping and space technology*, 2008.

[113] H. Senaratne, A. Mobasheri, A. L. Ali, C. Capineri, and M. Haklay. A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, 31(1):139–167, 2017.

[114] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[115] J. Severinsen, M. de Roiste, F. Reitsma, and E. Hartato. VGTrust: measuring trust for volunteered geographic information. *International Journal of Geographical Information Science*, pages 1–19, 2019.

[116] B. Sirmacek and C. Unsalan. Urban-area and building detection using sift keypoints and graph theory. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1156–1167, April 2009.

[117] S. Srivastava, J. E. Vargas-Muñoz, S. Lobry, and D. Tuia. Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data. *International Journal of Geographical Information Science*, 0(0):1–20, 2018.

[118] S. Srivastava, J. E. Vargas-Muñoz, S. Lobry, and D. Tuia. Land-use characterisation using Google Street View pictures and OpenStreetMap. In *21st AGILE Conference on Geographic Information Science (2018), Lund*, 2018.

[119] S. Srivastava, J. E. Vargas-Muñoz, and D. Tuia. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. *Remote Sensing of Environment*, 228:129–143, 2019.

[120] B. Suger and W. Burgard. Global outer-urban navigation with OpenStreetMap. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1417–1422, May 2017.

[121] F. Sunar Erbek, A. Ulubay, D. Maktav, and E. Yağiz. The use of satellite image maps for urban planning in turkey. *International Journal of Remote Sensing*, 26(4):775–784, 2005.

[122] O. Tasar, E. Maggiori, P. Alliez, and Y. Tarabalka. Polygonization of Binary Classification Maps Using Mesh Approximation with Right Angle Regularity. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6404–6407, 7 2018.

[123] G. Touya, V. Antoniou, A. M. Olteanu-Raimond, and M. D. Van Damme. Assessing crowdsourced POI quality: Combining methods based on reference data, history, and spatial relations. *ISPRS International Journal of Geo-Information*, 6(3):80, 2017.

[124] Q. Truong, G. Touya, and C. De Runz. Towards Vandalism Detection in OpenStreetMap Through a Data Driven Approach. In *GIScience 2018*, 2018.

[125] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(Sep):1453–1484, 2005.

[126] D. Tuia, G. Moser, M. Wurm, and H. Taubenböck. Land use modeling in North Rhine-Westphalia with interaction and scaling laws. In *Joint Urban Remote Sensing Event (JURSE)*, pages 1–4, March 2017.

[127] D. Tuia and J. Muñoz-Marí. Learning user's confidence for active learning. *IEEE Trans. Geosci. Remote Sens.*, 51(2):872–880, 2013.

[128] D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery. Active learning methods for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.*, 47(7):2218–2232, 2009.

[129] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Muñoz-Marí. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE J. Sel. Topics Signal Proc.*, 5(3):606–617, 2011.

[130] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios. Graph-based registration, change detection, and classification in very high resolution multitemporal remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7):2940–2951, 2016.

[131] A. Vandecasteele and R. Devillers. Improving volunteered geographic information quality using a tag recommender system: the case of OpenStreetMap. In *OpenStreetMap in GIScience*, pages 59–80. Springer, 2015.

[132] J. E. Vargas, A. X. Falcão, J. A. dos Santos, J. C. D. M. Esquerdo, A. C. Coutinho, and J. F. G. Antunes. Contextual superpixel description for remote sensing image classification. In *IGARSS*, pages 1132–1135, 2015.

[133] J. E. Vargas-Muñoz, A. S. Chowdhury, E. B. Alexandre, F. L. Galvão, P. A. Vechiatto Miranda, and A. X. Falcão. An Iterative Spanning Forest Framework for Superpixel Segmentation. *IEEE Transactions on Image Processing*, 28(7):3477–3489, July 2019.

[134] J. E. Vargas-Muñoz, D. Marcos, S. Lobry, J. A. dos Santos, A. X. Falcão, and D. Tuia. Correcting misaligned rural building annotations in OpenStreetMap using convolutional neural networks evidence. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

[135] J. E. Vargas-Muñoz, S. Lobry, A. X. Falcão, and D. Tuia. Correcting rural building annotations in OpenStreetMap using convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:283 – 293, 2019.

[136] J. E. Vargas-Muñoz, P. Zhou, A. X. Falcão, and D. Tuia. Interactive coconut tree annotation using feature space projections. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019. to appear.

[137] C. Ventura, J. Pont-Tuset, S. Caelles, K. K. Maninis, and L. Van Gool. Iterative deep learning for road topology extraction. In *British Machine Vision Conference*, 2018.

[138] M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, Feb 2017.

[139] Z. Wang and A. Zipf. Using OpenStreetMap Data to Generate Building Models with Their Inner Structures for 3d Maps. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 411–416, Sep 2017.

[140] F. F. Xu, B. Y. Lin, Q. Lu, Y. Huang, and K. Q. Zhu. Cross-region traffic prediction for China on OpenStreetMap. In *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 37–42. ACM, 2016.

[141] Y. Xu, Z. Chen, Z. Xie, and L. Wu. Quality assessment of building footprint data using a deep autoencoder network. *International Journal of Geographical Information Science*, 31(10):1929–1951, 2017.

[142] Y. Xu, Z. Xie, L. Wu, and Z. Chen. Multilane roads extracted from the OpenStreetMap urban road network using random forests. *Transactions in GIS*, 23(2):224–240, 2019.

[143] Y. Yao, X. Liu, X. Li, J. Zhang, Z. Liang, K. Mai, and Y. Zhang. Mapping fine-scale population distributions at the building level by integrating multisource geospatial big data. *International Journal of Geographical Information Science*, 31(6):1220–1244, 2017.

[144] J. Yuan, P. K. R. Chowdhury, J. McKee, H. L. Yang, J. Weaver, and B. Bhaduri. Exploiting deep learning and volunteered geographic information for mapping buildings in Kano, Nigeria. *Scientific data*, 5:180217, 2018.

[145] X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.*, 5(4):8–36, 2017.

[146] X. Zhuo, F. Fraundorfer, F. Kurz, and P. Reinartz. Optimization of OpenStreetMap Building Footprints Based on Semantic Information of Oblique UAV Images. *Remote Sensing*, 10(4):624, 2018.

Appendix A

Copyright Permission

Copyright Clearance Center RightsLink®

Home Account Info Help LIVE CHAT

Title: Correcting rural building annotations in OpenStreetMap using convolutional neural networks

Author: John E. Vargas-Muñoz,Sylvain Lobry,Alexandre X. Falcão,Devis Tuia

Publication: ISPRS Journal of Photogrammetry and Remote Sensing

Publisher: Elsevier

Date: January 2019

© 2018 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

Logged in as:
John Edgar Vargas Muñoz
Account #:
3001467131

LOGOUT

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW

Copyright © 2019 Copyright Clearance Center, Inc. All Rights Reserved. Privacy statement. Terms and Conditions.
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix B

Subsection "3.2.2 Aligning OSM rural building annotations" with better standardized mathematical symbols

The building registration problem is considered as the problem of aligning the vector shapes from OSM to the predictions of the CNN (Figure 3.2b). Such alignment is performed by estimating alignment vectors, basically shifting every OSM polygon to an area of high building probability in the CNN map.

In order to compute these alignment vectors, we need to measure how well a given shift performs. To this end, we use the correlation between the aligned annotations and the building probability map obtained previously using the image on which the annotations need to be registered. Making the hypothesis that rural buildings are gathered in small groups where each building has the same misalignment error, we align groups of buildings instead of individual buildings. This reduces greatly the computational load and is numerically more effective (see the results Section 3.4). Moreover, using groups of buildings instead of single ones makes the results less dependent on the quality of the building probability map.

Additionally, we observed that nearby groups of buildings have similar registration errors. Based on this observation, we build our building registration module on a MRF model using this prior together with the evidence provided by the building probability map. Our method aims at finding the alignment vectors $\mathbf{d} = \{d_0, d_1, \ldots, d_n\}$ that need to be applied to the annotation locations $\mathbf{x}$ based on the a probability map $\mathbf{y}$. Groups of buildings, or *sites*, are used as nodes of the MRF graph (See Figure 3.3), where sites i and j are neighbors (i.e., $j \in \mathbf{h}_i$) in the graph if they are spatially close (see Section 3.3.2 for more details on the MRF graph definition).

In our MRF formulation, the unary term is obtained by using the normalized correlation $C(A(x_i, d_i), y_i)$ between the annotation after alignment $A(x_i, d_i)$ (A is a function that moves the annotation x_i using the alignment vector d_i) and the building probability map y_i. This term is equal to the average of the predicted probability values y_i of the pixels contained in the aligned annotation $A(x_i, d_i)$. The pairwise term is defined by the dissimilarity (vector norm of the difference of two vectors) between the alignment vector d_i of the annotation i and the alignment vectors d_j of neighboring annotations $j \in \mathbf{h}_i$ [86].

Algorithm 2 – MRF-BASED ALIGNMENT ALGORITHM

INPUT: Original OSM annotations $\mathbf{m}$ and building probability map $\mathbf{y}$
OUTPUT: Alignment vectors $\mathbf{d}$

Group the original rural building annotations $\mathbf{m}$ *according to their spatial distance from each other, obtaining the set of building groups* $\mathbf{x}$.
Define the neighbors $\mathbf{h}_i$ *of each site* i *as spatially close sites.*
Initialize each d_i *to* $\arg\max_{v \in \mathbf{v}} C(A(x_i, v), y_i)$
Run Iterated Conditional Modes (ICM) for $MaxIters$ *iterations*
For $t \leftarrow 1...MaxIters$, **do**
    For each $x_i \in \mathbf{x}$, **do**
        For each $v \in \mathbf{v}$, **do**
            Compute energy $U(v|x_i, y_i)$, *equation* (B.1)
            If $U(v|x_i, y_i) < U(d_i|x_i, y_i)$, **then**
                $d_i \leftarrow v$
Return $\mathbf{d}$

The optimal set of alignment vectors $\hat{\mathbf{d}}$ for the annotations is defined by:

$$\begin{aligned} \hat{\mathbf{d}} &= \arg\min_{\mathbf{d} \in \mathbf{v}^{\mathcal{N}}} \sum_i U(d_i|x_i, y_i) \\ &= \arg\min_{\mathbf{d} \in \mathbf{v}^{\mathcal{N}}} \sum_i -\log C(A(x_i, d_i), y_i) + \beta \sum_{j \in \mathbf{h}_i} \frac{1}{\mathcal{Z}} ||d_i - d_j||_2, \end{aligned} \tag{B.1}$$

where $\mathbf{v} = \{v_1, v_2, \ldots, v_p\}$ is the set of all possible p alignment vectors, β is the spatial regularization parameter and $\mathcal{Z}$ is a normalization factor, defined as the maximum possible distance between two alignment vectors in $\mathbf{v}$. To compute the optimal $\hat{\mathbf{d}}$ by minimizing the energy function U, we use the Iterative Conditional Modes (ICM) [20] algorithm initialized with $d_i = \arg\max_{v \in \mathbf{v}} C(A(x_i, v), y_i)$. As this initialization is already a good heuristic (see Section 3.4), the ICM algorithm allows to obtain a good solution in a few iterations. The inclusion of a distance-based weight in the pairwise term does not lead to better performances, so it is omitted for clarity. We presented preliminary results of our proposed method for alignment of OSM annotations in the conference paper [134]. Algorithm 2 summarizes the proposed method for aligning OSM annotations.