Technische Universität München
TUM School of Engineering and Design

PREDICTING BUILDING FUNCTIONS BY FUSING SOCIAL MEDIA AND REMOTE SENSING DATA

EIKE JENS HOFFMANN

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. rer. nat. Martin Werner

Prüfer:innen der Dissertation:

1. Prof. Dr.-Ing. habil. Xiaoxiang Zhu
2. Prof. Nathan Jacobs, Ph.D.
3. Prof. Dr.-Ing. Frank Petzold

Die Dissertation wurde am 26.04.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 29.09.2022 angenommen.

ABSTRACT

Building functions play a crucial role in urban planning, population density estimation, and risk assessments of natural disasters. While building functions are available in developed countries, the Global South often lacks this data. Even if land use maps are created, they are outdated within a few years due to the fast pace of urban development.

The default way of creating these maps is by conducting a field study. However, manual labor is not feasible for keeping the maps up-to-date in fast-growing urban areas. Thus, automated approaches using other data sources are necessary. At the same time, social media and remote sensing data are big data sources at almost no cost. They offer complementary perspectives: One from a ground level and one from a bird's eye view. Moreover, both data sources are continuous data streams providing up-to-date information.

This thesis explores four different, new methods using Google Street View (GSV), social media, and remote sensing data for building function classification. It aims at generalizable models, and thus, each method builds upon culturally diverse datasets. Moreover, the methods are evaluated on a subset of buildings with human-validated labels. Finally, they are used in a combined way to improve individual prediction performances. The four approaches for predicting building functions are

1. Predicting building functions based on façade-oriented GSV images
2. Extracting building façade images from big social media image dataset and aligning them with the spatial surrounding
3. Creating spatio-temporal features from social media metadata and using their inherent patterns
4. Building an aerial image dataset and using this for predicting

The first approach uses state-of-the-art deep learning architectures for image classification to predict building functions with street-level imagery. Subsequently, the second approach describes a computationally efficient filtering pipeline for extracting suitable images from large-scale social media datasets. The pipeline identifies photos of building façades in big image datasets and aligns them spatially with surrounding buildings. The pipeline is evaluated on a culturally diverse dataset with 28 million social media images, and its results are compared to human-validated labels. Third, the thesis introduces a social media metadata approach that predicts building functions solely based on spatio-temporal features of social media posts. Third, it demonstrates the results of pure aerial models on the given task and finally analyzes fusion strategies for combining aerial and social media-based models. Given a high label quality, the fusion of social media images and aerial images improves individual results by up to 10.9 %.

ZUSAMMENFASSUNG

Das Wissen über Gebäudefunktionen spielt sowohl in der Städteplanung, bei der Schätzung von Einwohnerzahlen sowie bei der Risikoanalyse von Naturkatastrophen eine zentrale Rolle. Aufgrund der Urbanisierung insbesondere im globalen Süden sind Karten über die Landnutzung jedoch schnell veraltet, falls sie überhaupt exisitieren.

Gleichzeitig ist die manuelle Kartierung der Landnutzung mittels Feldstudien sehr aufwändig, so dass automatisierte Verfahren unter Verwendung von alternativen Datenquellen notwendig sind. Parallel dazu haben sich die Daten aus sozialen Medien und der Fernerkundung zu unerschöpflichen Quellen entwickelt, die mit geringen Kosten genutzt werden können. Die beiden Datenquellen haben darüber hinaus den Vorteil, dass sie gegensätzliche Perspektiven widerspiegeln: die eine betrachtet direkt vom Boden aus und die andere aus der Vogelperspektive. Zudem bieten beide Quellen einen kontinuierlichen Datenstrom, der jederzeit aktuelle Informationen liefert.

Diese Arbeit beschäftigt sich mit der Vorhersage von Gebäudefunktionen durch die Kombination von Daten aus sozialen Medien und der Fernerkundung. Dafür werden vier neue, unterschiedliche Vorhersagemethoden vorgestellt, die auf Bildern aus Google Street View (GSV), Bildern und Nachrichten aus sozialen Medien sowie Fernerkundungsdaten basieren. Bei der Entwicklung stand eine weltweite Anwendbarkeit im Zentrum, weswegen die Methoden auf Datensätzen mit globaler Verteilung entwickelt wurden. Desweiteren findet die Evaluation der Ansätze auf einem Anteil der Gebäude statt, deren Funktion manuell verifiziert wurde. Nach der Vorstellung der einzelnen Methoden werden diese zum Schluss miteinander kombiniert, um die Vorhersagegenauigkeit weiter zu erhöhen.

Die vier Ansätze zur Vorhersage von Gebäudefunktionen sind wie folgt:

1. Die Vorhersage von Gebäudefunktionen basierend auf Google Street View (GSV) Fassadenbildern

2. Die Extraktion von Gebäudefassadenbildern aus großen Bilddatensätzen und deren Verknüpfung mit der räumlichen Umgebung

3. Die Erstellung von räumlichen und zeitlichen Merkmalen aus den Metadaten von Nachrichten aus sozialen Medien und der Mustererkennung in diesen Merkmalen

4. Der Aufbau eines Datensatzes mit Luftbildern und dessen Verwendung zur Vorhersage von Gebäudefunktionen

Die erste Methode verwendet für die Vorhersage der Gebäudefunktionen aktuelle Ansätze aus dem Bereich der künstlichen Intelligenz in Kombination mit Fassadenbildern von GSV.

Darauf aufbauendend stellt der zweite Ansatz einen effizienten Algorithmus zur Extraktion von Fassadenbildern aus großen Bilddatensätzen von sozialen Medien vor. Dazu muss ein Bild sowohl eine Gebäudefassade zeigen als auch Positions- und Kompassdaten enthalten, um die dargestellten Gebäude mit Kartendaten verknüpfen zu können. Dieser Algorithmus wird auf einem global diversifizierten Datensatz von 28 Millionen Bildern evaluiert. Zusätzlich wurde ein Teil der Bilder und deren Gebäude durch menschliche Experten verifiziert.

Die dritte Methode basiert auf den Metadaten von Nachrichten aus sozialen Medien und erstellt daraus räumliche und zeitliche Merkmale. Mittels maschinellem Lernen werden daraus die Funktionen der umliegenden Gebäude vorhergesagt.

Für die vierte Methode werden Luftbilder der Gebäude verwendet und analog zu der Vorgehensweise der ersten Methode passende Deep Learning Modelle trainiert.

Nach der individuellen Analyse der einzelnen Methoden werden diese am Ende der Arbeit kombiniert, um die Vorhersagen weiter zu verbessern. Durch die Kombination der vorgestellten Methoden lassen sich auf dem validierten Datensatz die Vorhersagen um bis zu 10,9 % verbessern.

If I have seen further it is by standing on the shoulders of Giants

— Sir Isaac Newton, Letter to Robert Hooke, 1676

ACKNOWLEDGMENTS

First of all, my deepest gratitude goes to Prof. Xiaoxiang Zhu, who allowed me to be a Ph.D. candidate in her international lab and supported me constantly during the last four years. Her advice and supervision always provided solid scientific guidance. Moreover, I would like to thank Prof. Nathan Jacobs and Prof. Frank Petzold for being the reviewers of this Ph.D. thesis. Their effort is invaluable, and their expression of curiosity about this work created additional motivation while writing.

My gratefulness for Prof. Martin Werner is inexpressible. As my supervisor during my Master's thesis, he convinced me to pursue a Ph.D. in the first place. As my mentor, he always supported me with scientific and personal advice. As my friend, he helped me wherever he could and had faith in me when I was tumbling.

I am incredibly grateful to Dr. Karam Abdulahhad, a reliable expert, for discussing ideas and results. If there was any mistake or twisted number, he was the one to discover it. He constantly challenged the *status quo* and asked the right questions. Without his deep experience and accurate proofreading, this thesis would not be what it is today.

Furthermore, I would like to thank Matthias Häberle, my fellow working on the same task of building function classification for the So2Sat project. When I struggled with Flickr images, he had similar issues with Twitter text data. As we tend to have contrary opinions, our discussions have always been lively and showed me different perspectives.

During my time at TUM SiPEO and DLR's IMF DAS departments, all my colleagues enriched my life as a Ph.D. candidate. I am very grateful to all of them, and some deserve a special mention in their order of appearance. When I started, Dr. Homa Ansari and Dr. Sina Montazeri helped me get into life at DLR. It is always a pleasure to meet both and laugh about science and German culture and tradition. Dr. Lukas Krieger showed me the minimalistic window manager i3, which helped me keep focused when several tasks were on my desk. During my journey down the rabbit hole of research, Dr. Yao Sun shared her experience and helped me see things differently and brighter. Moreover, I will never forget all the good memories of laughter and joy with Dr. Chunping Qiu and Dr. Jingliang Hu at work and during conferences. Dr. Yuanyuan Wang manages the So2Sat project gently and friendly and is always in for a relaxed beer. In parallel with this Ph.D. work, my IT position at DLR would have been impossible without the support of Dr. Andreas Niedermeier. He introduced me to the task and has been an invaluable advisor. If the art of science became too abstract, Dr. Andrés Camero and Lukas Kondmann were always there to find appropriate and new interpretations. Mohsin Ali and Jakob Gawlikowski helped me understand uncertainty mathematically and supported me with their implementations for my research. During the final phase of writing this thesis, Dr. Conrad M. Albrecht backed me and took over all IT-related responsibilities. Nikolai Skuppin gave me hope that all the effort put into the infrastructure behind this work might serve as a foundation for the future work of other colleagues. Nevertheless, all scientific work needs to be structured and organized. In case of any organizational obstacle, Dr. Anja Rösel, Dr. Anna Kruspe, Irene Danhofer, and Yingjie Schreiber-Liu are there to help. Whenever I needed IT support or any kind of infrastructure, the teams from DLR's DFD-INF and Navum provided immediate answers and reliable systems.

While writing, Dr. René Cyranek provided mental and nutritional support as well as divergent feedback from an unrelated scientific field. After the first draft of this thesis was finished, Lars von den Driesch helped with careful proofreading.

During my Ph.D. journey, I made heavy use of open-source software. With the reliable operating system Ubuntu, I implemented most code in Python with the packages `numpy` [1], `pandas` [2] and `tensorflow` [3]. The plots and maps in this thesis are made with `seaborn` [4] and `cartopy` [5] on top of `matplotlib` [6] with a LaTeX backend. All maps have the stunningly minimalistic Stamen Toner tiles as background.

My sincere gratitude goes to OpenStreetMap (OSM) and its contributors for creating a free and open map of the world with incredible detail. OSM has its blind spots, but they will be fixed step by step by a dedicated team of volunteers. Although Google has strict terms of service, they enable scientists of all fields with their tremendous data treasure accessible through Google Cloud services. Without all their aerial and GSV imagery, this thesis would have been impossible. The aim for global coverage requires sophisticated data management. The database system PostgreSQL [7] with its spatial extension PostGIS [8] made it possible.

Last but not least, I want to thank my parents and my grandparents, who believed in me in times when I did not. They supported me in any aspect of life and lifted me back onto the track whenever needed. Finally, I want to express my deepest gratitude to my girlfriend Marina, who supports me unconditionally under any circumstances.

CONTENTS

I Introduction

II Fundamentals

III Related Work

IV Contributions

v Conclusion

vi Appendix

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API Application Programming Interface

CNN Convolutional Neural Network

DEM Digital Elevation Model

EO Earth Observation

EXIF Exchangeable Image File Format

GPU Graphics Processing Unit

GSD Ground Sample Distance

GSV Google Street View

LDA Latent Dirichlet Allocation

LSTM Long Short-Term Memory

LCZ Local Climate Zones

MLP Multilayer Perceptron

OSM OpenStreetMap

POI Point of Interest

SVM Support Vector Machine

SDG Sustainable Development Goal

SGD Stochastic Gradient Descent

UN United Nations

Part I

INTRODUCTION

Die Erkenntnis der Städte ist an die Entzifferung ihrer traumhaft hingesagten Bilder geknüpft.

— SIEGFRIED KRACAUER

1 INTRODUCTION

Urbanization is a global, demographic megatrend transforming our planet [9]. Back in 1950, 30 % of the world population lived in urban areas, with the majority of 70 % of people living in rural areas [10] (Figure 1.1). In the following 58 years, this fraction changed entirely, and in 2018 55 % of all people worldwide lived in urban areas. This trend will continue so that current predictions of the UN see 68 % of the global population living in urban environments by 2050. This trend will primarily affect metropolitan areas in emerging countries. Since cities are a place of education and innovation [11], the UN sees urbanization as a positive development requiring active management and support [12].

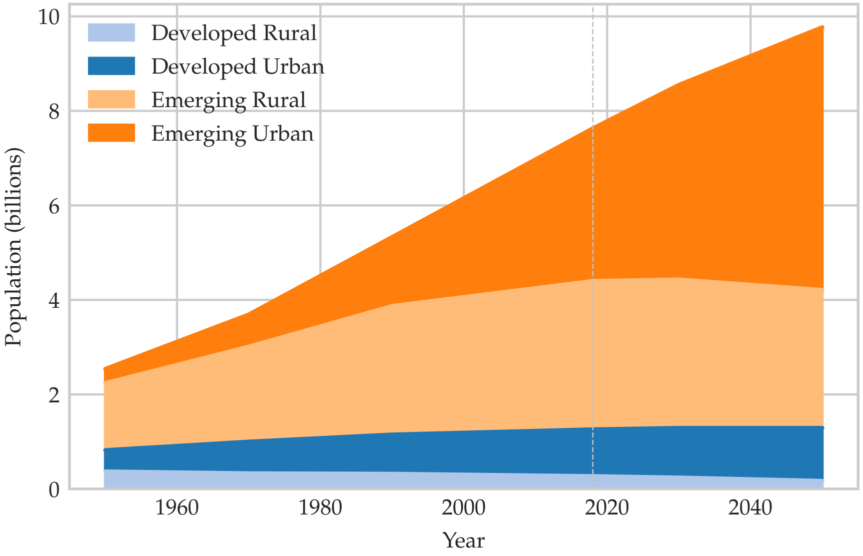


Figure 1.1: Urban and rural population growth from 1950 to 2050, predicted in 2018 by the UN for development groups. Data from [10]

1.1 MOTIVATION

With more and more people moving into urban areas, new dwellings are needed together with appropriate commercial and civic infrastructure like retail stores, schools, and hospitals—especially fast-growing metropolitan areas in the Global South struggle with the pace of urban development. For example, Mumbai's population skyrocketed from 9 million inhabitants in 1980 to 20 million in 2020 [13]. A structured urban development requires detailed knowledge of where people live, where they work, and what infrastructure they can access. The importance of this issue is highlighted by the Sustainable Development Goal (SDG) 11 of the UN: "Make cities and human settlements inclusive, safe, resilient, and sustainable" ([14]). A part of this in-depth knowledge is the function of the buildings in a city. Connecting the knowledge, where people live, and what infrastructure they have access to enables urban planners to estimate current and future demands and manage urban development. Moreover, maps of building functions allow more fine-grained population density estimation and can play a vital role in natural hazard risk assessments and disaster management.

This thesis presents new methods to predict building functions using geotagged social media and remote sensing data. Building functions are the most fine-grained level of urban land use in geosciences. In contrast to land cover, which is based on direct observation, land use "requires socio-economic interpretation of the activities that take place on that surface" ([15]). Hence, land use is not directly measurable but is a subject of pattern recognition. Performing this task on a building instance level is challenging as buildings are tiny from an urban spatial perspective, and their function can be hard to estimate from an outside view. Additionally, buildings can serve multiple functions in dense urban areas. Outside of the city centers in the suburban areas, there are functionally more homogeneous neighborhoods that exhibit distinctive patterns from a street-view [16] and an aerial perspective [17]. Thus, land use classification schemes [18, 19] are all simplifications of the real world.

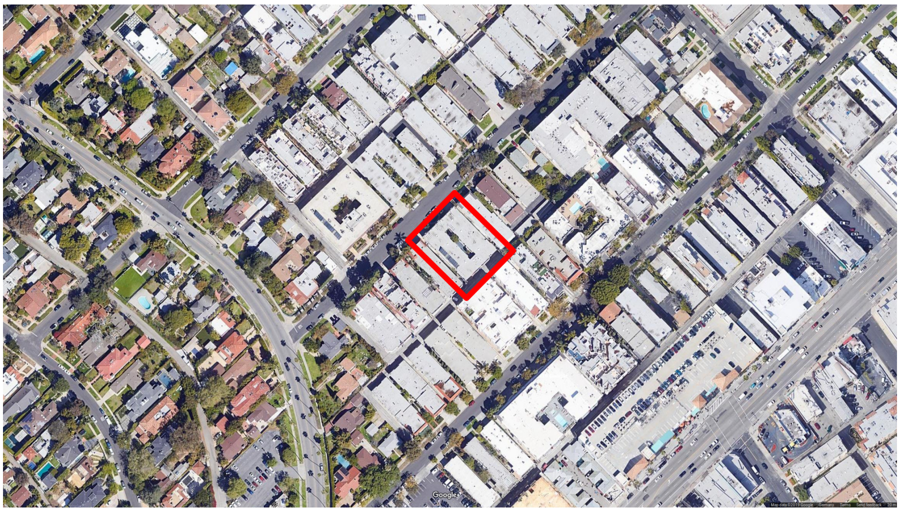
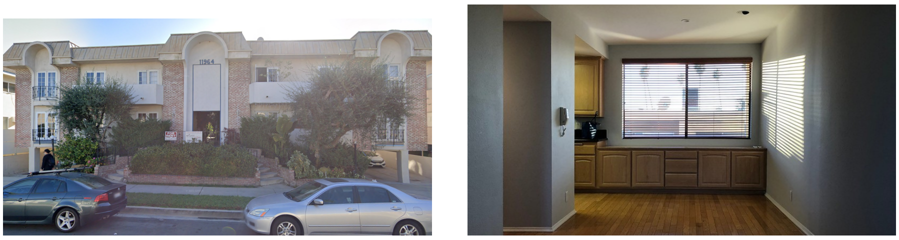


Figure 1.2: Example of an aerial image in Los Angeles, USA. One building at 11964 Kiowa Ave is highlighted with a red rectangle. Aerial image © Google Maps and their suppliers

Figure 1.2 illustrates these cases: while the single-detached houses on the left side are residential buildings, the roofs of the buildings on the right side do not provide any hint of their function. The ample parking space on the bottom right might indicate commercial activity in this area. Since the ambiguities in the aerial image cannot be resolved from this perspective, additional information is necessary. Figure 1.3 shows two examples of data sources that can provide different insights: a GSV image centered on the building façade (Figure 1.3a) and a geotagged photography from Flickr, a social media platform (Figure 1.3b). The façade image depicts windows with curtains and a sign reading *for rent*. Furthermore, there is no other sign advertising any commercial activity. For a human, these are strong indications for a residential building. Figure 1.3b shows the kitchen of an empty apartment, probably taken during a flat viewing. In combination, both sources provide strong evidence that the building highlighted in Figure 1.2 is a residential building.



(a) Example of a GSV image showing the building highlighted in Figure 1.2. © Google Maps

(b) Example of a geotagged social media photo taken inside the building highlighted in Figure 1.2. Walkthrough © by Jeremiah LaRose

Figure 1.3: Examples of images from additional data sources showing different perspectives of the building highlighted in Figure 1.2

While each modality for itself can only provide weak hints, the fusion of different data sources can resolve these ambiguities. Possible data sources include, but are not limited to, aerial [20] and satellite imagery [21], street-level imagery [16], social media text messages [22], social media images [23], social media metadata [24], Point of Interest (POI) data [25], and mobile phone cell tower data [26].

However, not every data source is freely and openly available globally. Since this thesis aims at generalizable methods, the data must be sampled from worldwide distribution. The widespread usage of social media platforms and the open access to remote sensing data make them suitable data sources and allow for combining complementary perspectives.

1.2 PROBLEM STATEMENT AND OBJECTIVES

In summary, there are five challenges involved when developing building function classification methods:

1. Buildings are tiny objects from a global perspective, even within a city
2. Building functions cannot be directly measured but require the interpretation of patterns
3. Buildings with one function have a highly diverse appearance across cultures and partly even within cultures
4. Buildings can have multiple functions at the same time on different stories or parts, and these functions are subject to change over time
5. Classification schemes for building functions are always incomplete as it is impossible to capture all possible variations and combinations of classes worldwide

This thesis proposes four methods to predict building functions based on different data sources: Street-level imagery from commercial providers, geotagged street-level imagery from social media image platforms, metadata of geotagged social media posts, and aerial images. Each method is presented individually in the first place and finally combined for improved predictions. Throughout the work, the second challenge is handled by using state-of-the-art machine learning methods that mimic human perception and detect latent patterns in structured data. The methods proposed in this thesis are developed with a globally diverse dataset and thoroughly analyzed in different cultural domains. The thesis builds upon a simple, yet globally applicable building classification scheme of three classes: *Commercial*, *residential*, and *other*, which are a subset of the built-up land use classes proposed by Theobald *et al.* [19]. For example, *commercial* buildings include shopping malls, gas stations, industrial sites, and hotels. *Residential* buildings are, for example, single-detached houses, terraced houses, or apartment buildings. All buildings that do not belong to one of these two classes are categorized as *other*, e. g. hospitals, schools, and town halls. This classification scheme allows estimating population densities at a fine-grained level by focusing on the *residential* buildings [27] and enables estimations on access and demand of public infrastructure [28]. Moreover, these classes play a crucial role in risk assessments of natural desasters [29].

1.3 STRUCTURE OF THE THESIS

While this chapter introduced the task of building function classification, its challenges, and the thesis' objectives, the next Chapter 2 provides background information that helps in understanding the principles of Earth Observation (EO) and deep learning. Equipped with this knowledge, Chapter 3 gives an overview of related work in the fields of social media analysis, remote sensing for urban land use, and the fusion of these two subjects. Chapter 4 introduces the data sources used to build datasets with global coverage. Then four methodological chapters propose different methods to generate building function predictions from the datasets mentioned above. Chapter 5 describes a method that uses GSV images. Chapter 6 extends this approach by introducing a filtering algorithm for social media images that show similar content as GSV images and compares the results with the previous chapter. The subsequent Chapter 7 analyzes the suitability of social media metadata for predicting building functions. The fourth methodology in Chapter 8 utilizes remote sensing data for the same task. Chapter 9 analyzes how the individual results from the different methodologies can be combined to improve the final results. Finally, Chapter 10 concludes the methods, results, and findings to discuss opportunities for further research.

Part II

FUNDAMENTALS

In God we trust,
all others must bring data

— W. EDWARDS DEMING

2 FUNDAMENTALS

This chapter provides an overview of all methods used in this thesis. It starts with a basic introduction to Earth Observation (EO) principles and describes how EO data can be processed with a technique called deep learning.

2.1 INTRODUCTION TO EARTH OBSERVATION

Earth Observation (EO) is often seen as a synonym for remote sensing, but this does not cover the whole picture. Remote sensing is a part of EO, while the term itself refers to a much broader field. The Group on Earth Observations (GEO) defines it as "Earth observations are data and information collected about our planet, whether atmospheric, oceanic or terrestrial. This definition includes remotely-sensed data as well as ground-based or in situ data." ([30]) Hence, EO data includes remote sensing data, commercial street-level imagery, and social media data. This section provides a brief overview of core technologies that are used across the different parts of EO. It starts with remote sensing and transfers the concept to ground-level imagery.

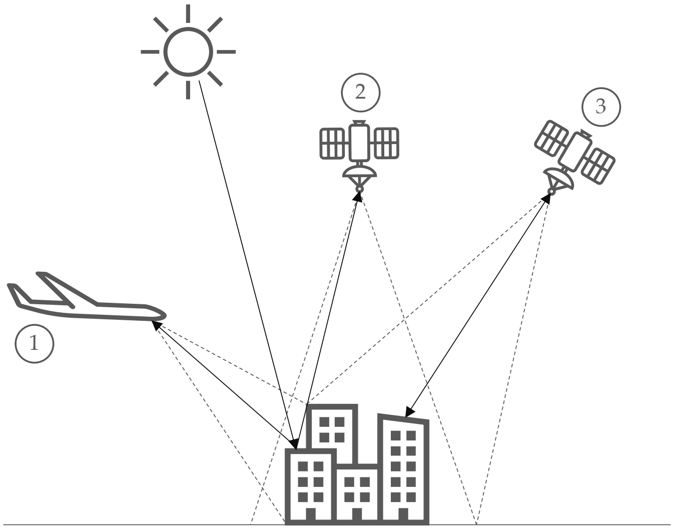
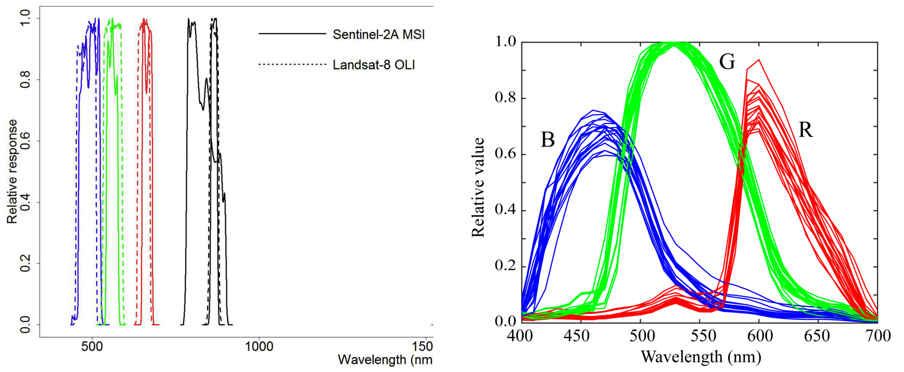


Figure 2.1: Concepts of remote sensing with airborne (1) and satellite platforms (2,3) from a vertical (2) and an oblique perspective (1, 3) as well as active (3) and passive sensors (1, 2)

"In an environmental context, remote sensing typically refers to technologies for recording *electromagnetic energy* that emanates from areas or objects on (or in) the Earth's land surface, oceans, or atmosphere" ([31]). Figure 2.1 illustrates different concepts of remote sensing. Electromagnetic energy can have its source with the sensor (satellite 3), the sun (satellite 2 and airplane 1), or other sources. The first case is an active sensor, while the latter is called a passive sensor. Examples of active sensors are radar satellites that emit a radar signal like Sentinel-1 [32] and lidar sensors mounted on airplanes. Both emit an electromagnetic signal and capture their reflectance from the Earth. Passive sensors observe the electromagnetic signals reflected or emitted by the Earth. An example is an optical satellite with a camera like Sentinel-2 [33] or the nighttime lights capturing where light signals are emitted during the night, for example, VIIRS DNB [34]. Moreover, the sensor can be mounted on different platforms: a satellite like (2) and (3) or an airplane (1). Satellite platforms have the advantage of low maintenance effort once deployed in space but yield lower-quality imagery. In contrast, capturing aerial imagery requires manual flights with airplanes but results in high-quality images. A third aspect is the perspective of the sensor. For example, the airplane (1) in Figure 2.1 has a sensor with an oblique view, looking at approximately 45° degrees ahead. Satellites (2) have a vertical view with a perpendicular angle to the ground. This view is also referred to as the nadir view.

At their core, remote sensing data are digital products similar to digital images consisting of single pixels. It is a two-dimensional array with numerical values from a computer science perspective. However, this array has a particular property: It is georeferenced. Every pixel is associated with a defined area in a geographical coordinate system allowing mappings to the Earth's surface. The area covered by a pixel is called spatial resolution or Ground Sample Distance (GSD). The higher the spatial resolution, the smaller the area covered by a pixel, and the better objects can be distinguished. While open satellite imagery achieves a GSD of up to 10 m [33], freely available aerial imagery taken by airplanes yields up to 0.1 m GSD [35]. Beyond the spatial resolution, there are three other resolutions of remote sensing data: temporal, spectral, and radiometric. The temporal resolution defines how much time is between two observations of the same area. This resolution is also referred to as the revisit period in terms of satellites. The spectral resolution denotes which electromagnetic wavelengths the sensor captures. Figure 2.2a shows the spectral response functions in visible and near-infrared bands for the Sentinel-2A and the Landsat-8 satellites. Last but not least, the sensor's sensitivity is called the radiometric resolution. It denotes the smallest differences in electromagnetic intensity that the sensor can detect and is denoted in binary bit-depth [36]. As sensors in remote sensing are carefully engineered and calibrated, the pixel values of these data are reliable physical measurements of the Earth's surface.

Compared with other EO data like social media images, remote sensing data is highly structured with well-known properties. Figure 2.2 illustrates the difference between satellite sensors (Figure 2.2a)



(a) approximately equivalent Sentinel-2A MSI (solid lines) and Landsat-8 OLI (dashed lines) [37]

(b) Relative spectral sensitivity functions of the 20 mobile phone cameras [38]

Figure 2.2: Relative spectral response functions for satellite sensors and mobile phone camera sensors

and smartphone camera sensors (Figure 2.2b). While the response functions of the satellite sensors are clearly separated with clear signal edges, the smartphone functions intersect with each other and are not separated. Furthermore, they show a high variance between different devices. However, these optical satellites and smartphones have in common that they capture imagery in three visible channels red, green, and blue (RGB). Other resolution aspects are also subject to change: the image quality can differ depending on the mobile device. The manufacturers optimize their devices to create visually appealing photos rather than physically correct representations of reality. Moreover, there is no direct comparison in spatial resolution: Although many smartphones are equipped with a GPS sensor for positioning, they have limited accuracy and precision.

Moreover, the signal from a GPS sensor specifies the position of the devices while taking a photo, but that is not sufficient for aligning image content with a geographical coordinate system. Only a combination of camera position and compass direction allows for relating the image content with spatial entities. From a temporal perspective, it depends on the users: while touristic hotspots have a high revisit period, other remote areas are never captured in a photo.

Commercial street-level imagery platforms are in between these two extreme cases. Providers like Google Street View (GSV) have developed dedicated devices for capturing street view photos. Hence, the camera sensors have similar properties to satellite sensors: They use industry-quality cameras with known properties to take 360-degree images and have high-precision GPS sensors to align them with maps. However, their limiting factor is the manual process involving a human driver during the creation process. With this limit, the temporal resolution is up to the services provider, who decides when which location is covered.

Nevertheless, the ground-level perspective covers details that are not visible from a remote-sensing perspective. Together with their global abundance, these details make them a complementary source to remote sensing and allow a richer landscape of EO products. Almost all EO data sources in this thesis generate RGB images. They can be analyzed with deep learning, a technology briefly presented in the next section.

2.2 DEEP LEARNING IN COMPUTER VISION

Computer vision aims to interpret the input from cameras similar to human perception. While this task is easy for humans and requires no effort, it is extremely challenging for computers. Daniel Kahnemann describes it as *System 1*, which acts fast, instinctive, and emotionally [39]. However, perceptional tasks have been studied for decades with little progress as researchers tried to build systems-based low-level features like edge [40] and corner [41] detection. This changed when CNNs became computationally feasible with Graphics Processing Units (GPUs) that calculates matrix operations massively parallelized. A milestone in this development is a CNN called AlexNet [42], which won the ILSVRC-2012 competition on image classification with a top-5 test error rate of 15.3 %, whereas the second-best achieved 26.2 %. This network with 60 million parameters has 650,000 neurons distributed among eight layers. Although neural networks have been a very well-established technique at this time, the number of layers and parameters was unprecedented and coined the terminology *deep learning*, a variant of machine learning. This section introduces the basic principles of machine learning, discusses how such algorithms are evaluated, and briefly presents the networks used in this thesis.

2.2.1 *Principles of Supervised Machine Learning*

Machine learning has three subdisciplines: supervised machine learning, unsupervised machine learning, and reinforcement learning. As this thesis aims at building function classification with three classes, *commercial*, *other*, and *residential*, it is in the domain of supervised machine learning. The building functions are already known, and the task is to find a function that maps the input data to one of these classes. This task is called *learning* because there is a set of input data that contains the class label for each input sample. As the function is created with algorithms based on this set, it is referred to as *machine learning*. Mitchell defines it more formally as follows: "A computer program is said to learn from experience E concerning some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E ." ([43])

As this definition is very general, this subsection illustrates its meaning for this thesis on building function classification. The task T is a classification problem, which means that the result needs to be categorical and stand for one of the three classes. Given some input data $\mathbf{x} \in \mathbb{R}^n$, the classification function $f(\mathbf{x}) = y : \mathbb{R}^n \to \{0, 1, 2\}$ maps x to a building function class that is numerically represented by its index $y \in \{0, 1, 2\}$. An alternative representation is the so-called one-hot-encoding with a vector that contains a 1 and all other values are zero. For example, an *other* building can be encoded as $\mathbf{y} = [0, 1, 0]$. This representation can be seen as a probability distribution. The experience E is packed as a dataset containing samples and their label. A set of images with labels shapes the dataset and is seen as the experience E. Finally, the performance measure P allows calculating a metric of how many samples are correctly assigned to their label. As machine learning focuses on improving the function f, the algorithm needs to learn from its errors. Therefore, the performance measure is often called error rate or loss. For classification tasks with classes C, the categorical cross-entropy loss $\mathcal{L}_{CE}$ is usually employed and also applied in this thesis for all classification algorithms. It has its foundations in information theory and compares two probability distributions: the predicted one $\hat{\mathbf{y}}$ and the true one $\mathbf{y}$:

$$\mathcal{L}_{CE}(y, \hat{y}) = -\sum_{c \in C} y_c \log \hat{y}_c \tag{2.1}$$

More formally, the objective of every machine learning algorithm is to minimize the error or loss. As an analytical solution to this optimization problem is not feasible, different algorithms iteratively minimize the loss.

2.2.2 *Supervised Deep Learning*

Deep learning is inspired by the way animals process information and make decisions. Their nervous system consists of a network of connected cells that process information from sensors like eyes or antennae and decide how muscles are used to survive in complex environments. Deep learning has its roots in the research of neural networks, which were first described in the 1940s [44]. These networks consist of single units mimicking the cell that pass on a signal if activated. Like a biological cell, the decision if a signal is forwarded should be based on adaption to the real world. Today's deep learning algorithms still build upon this principle. The following paragraphs provide a coarse intuition of the mechanisms. A more formal description can be found in Goodfellow *et al.* [45].

This adaption to the real world is an iterative process in which small parts of the training data are used to predict with a randomly initialized network. Afterward, the loss function calculates the error between the prediction and the actual ground truth. The next step is called backpropagation: The influence of each network part on the error is calculated and slightly changed so that the error in the following prediction phase is lower than before. However, a different part of the training data is used in this phase, and the process starts again. This principle of making forward and backward passes while moderately changing the network to adapt to the data is called Stochastic Gradient Descent (SGD). The rate of change is known as the learning rate. This iterative optimization process operates in a high-dimensional space and finds a minimum of the loss function given the weights. However, the SGD algorithm is prone to end in a local minimum, which can be improved by adding momentum. Adam [46] is an optimization algorithm with momentum by adaptively estimating first-order and second-order moments.

In the first neural networks, each network unit was connected to each unit of the previous layer. This kind of layer still exists and is called a dense or fully connected layer. Due to their many connections, networks with multiple dense layers can theoretically approximate any function; they are universal function approximators [47]. However, they are hard to train in practice, especially when the data has spatial correlations. LeCun *et al.* solved this problem by introducing convolutional layers [48]. They contain multiple convolution filters with trained parameters and are mostly used for image processing. For example, a convolutional filter can extract features like edges or corners in an image. Other layer types are batch normalization, which is used for re-scaling intermediate values in networks, and pooling layers, which aggregate values from convolutional layers and act as a dimensionality reduction mechanism.

The terminology in this field can be puzzling, as several terms describe different aspects of neural networks: a neural network contains different layers, and the arrangement of these layers is referred to as architecture. An architecture can be used for different tasks which depend on the training dataset. An instance of an architecture trained for a specific task is called a model.

2.2.3 *Evaluation of Classification Algorithms*

After training an architecture on a dataset, the final loss gives the first indication of how well a model performs for a task. However, for practical use of a model, one needs to know what the performance is on unseen data. Therefore, the dataset with labels is split into training and test sets. The first is used only for training, while the second is never used for training and is only applied when the model is finished. In this second step, the model predicts the test data, and the resulting predictions are compared with the actual labels.

For classification problems, this comparison starts with a confusion matrix. Intuitively, this table has every class represented in a row and a column. While the rows represent the actual classes, the columns stand for the predicted classes. The table is filled by counting how the samples in the test set are predicted.

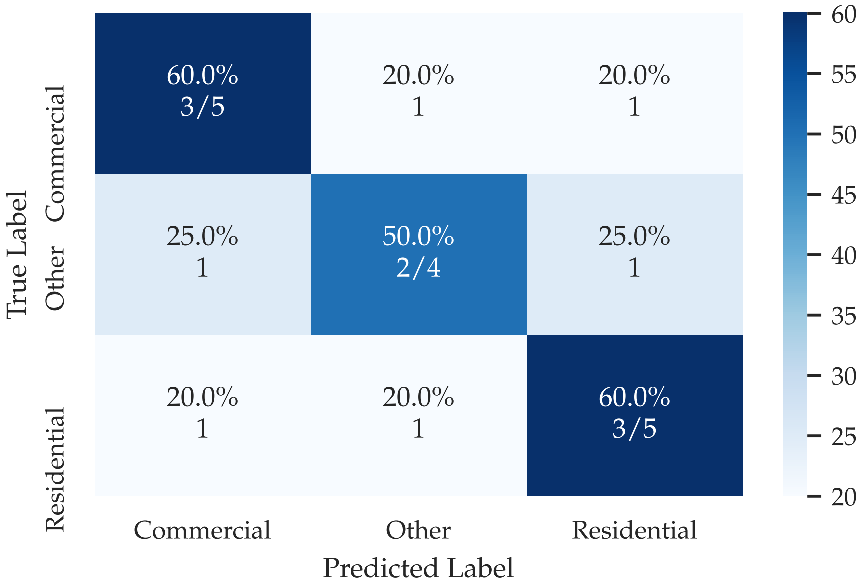


Figure 2.3: Example of a confusion matrix

Figure 2.3 shows an example of a confusion matrix. The correctly predicted samples are on the diagonal, whereas all wrong predictions are in the other cells. Three out of five commercial samples are correctly predicted in this example, and two are wrong. These two wrong samples are classified as *other* and *residential*. Based on these counts, two metrics can be calculated: Precision and recall. The precision is the number of correctly predicted samples divided by the column sum. At the same time, the recall is calculated as the number of correctly predicted samples divided by the row sum.

More formally, the two metrics are defined based on three counts:

1. true positives TP_{c_x}: samples of a class c_x that are correctly predicted
2. false positives FP_{c_x}: samples of another class c_y that are wrongly predicted as c_x
3. false negatives FN_{c_x}: samples of a class c_x that are wrongly predicted as another class c_y

The precision P_{c_x} of a class c_x is defined as

$$P_{c_x} = \frac{TP_{c_x}}{TP_{c_x} + FP_{c_x}} \tag{2.2}$$

The recall R_{c_x} of a class c_x is defined as

$$R_{c_x} = \frac{TP_{c_x}}{TP_{c_x} + FN_{c_x}} \tag{2.3}$$

Intuitively, the precision shows how often a classifier is correct if it predicts a given class. In contrast, the recall describes how many samples of a given class are predicted as such. In this example, the percentage values on the diagonal represent the recall for each class. With three out of five *commercial* samples being correctly predicted, the recall is 60.0 %. By coincidence, the precision is also 60 % in this example, but that is not always the case. There is a tradeoff between precision and recall. Suppose one optimizes for one metric, and the other decreases. Choosing the metric to optimize for is application-specific. For example, when analyzing the risk for citizens during a natural disaster, the recall of the *residential* class would be more critical. Even if a few *commercial* or *other* buildings are included as false positives, the disaster management capacities should exceed the demand rather than fail to save lives because of a lack of capacity.

Nevertheless, a classification algorithm for building function classification should not optimize for recall. Ideally, both metrics are high or well-balanced in a more realistic scenario. Hence, the F1 score is a well-suited metric as it is defined as the harmonic mean between precision and recall

$$F1_{c_x} = 2\frac{P_{c_x} \cdot R_{c_x}}{P_{c_x} + R_{c_x}} \tag{2.4}$$

When evaluating the classification models in this thesis, all metrics are shown for each class to have a comprehensive comparison.

Some publications use accuracy as a measure for a classification algorithm. It is calculated as

$$A_{c_x} = \frac{TP_{c_x} + TN_{c_x}}{TP_{c_x} + FN_{c_x} + TN_{c_x} + FP_{c_x}} \tag{2.5}$$

While it might be suited in some cases, this metric is prone to misinterpretations when the class distribution is imbalanced. Suppose a binary classification task dataset has 90 samples for class 0 and 10 for class 1. In that case, accuracies of 90 % are not convincing: A classifier could constantly return 0 and would achieve this accuracy. Therefore, this thesis uses precision, recall, and F1 score for all classes, enabling a comprehensive assessment of the methods.

2.2.4 *Deep Learning Architectures for Image Classification*

This subsection describes seven state-of-the-art architectures used in this thesis. It introduces them chronologically and highlights their novelties over each other. The revolution started with AlexNet [42] when this method won the ILSVRC-2012 competition on image classification. The competition was started in 2010 with an unprecedented dataset of 1,461,406 images labeled with 1,000 classes. The training dataset with labels was publicly released along with an unlabeled test set. The labels for the test were manually created and kept private for a fair evaluation. Participants of these competitions needed to upload predicted labels for the test dataset, and an evaluation server calculated the winner based on the best predictions.

The game-changer of AlexNet was the stacking of convolutional layers, using rectified linear units as activation functions, and their efficient implementation using two modern GPUs. Convolutional operations in neural networks were introduced in 1980 [49], but their applicability was limited due to their computational demands [42]. Intuitively, a convolutional filter slides over an image and processes step by step a small region to extract a feature on a higher level. Figure 2.4 depicts the 96 convolutional filters learned by AlexNet on ImageNet data. They are primitive edge and pattern detectors. By stacking convolutional layers on top of each other, the features become more abstract in the higher layers. The second difference is the use of rectified linear units (ReLU) [50] instead of sigmoid functions for activation. The non-linear ReLU function is defined as $f(x) = max(0, x)$ and due to its non-saturating character, it enables faster training compared to saturating sigmoid functions. Finally, AlexNet was trained on two GPUs in parallel by splitting the different convolutional filters into two partitions. In combination, these three modifications yielded the break-trough of this model. In the following years, this architecture has been refined in several ways.

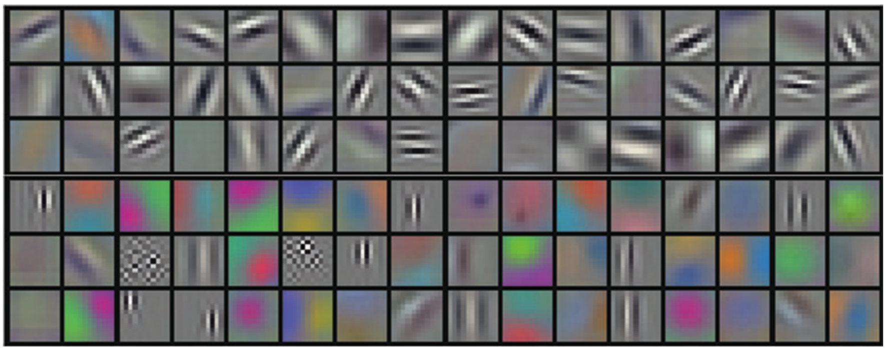


Figure 2.4: Example for 11x11x3 convolution filters of the first layer in AlexNet [42]

2.2.4.1 *VGG16*

VGG16 builds upon AlexNet by replacing the large filters in the first convolutions with smaller ones of 3×3 filters [51]. Figure 2.4 shows 11×11 filters, which are computationally expensive as they have

364 trainable parameters. For reference, a 3×3 filter has 28 trainable parameters. Simonyan *et al.* compensate for the smaller filter sizes by increasing the number of convolutional layers to 16 layers in total. Figure 2.5 illustrates the architecture. Although it is deeper with more layers, the number of parameters decreased from 60 million to 15 million parameters. This architecture achieved second place in the ILSVRC-2014 competition.

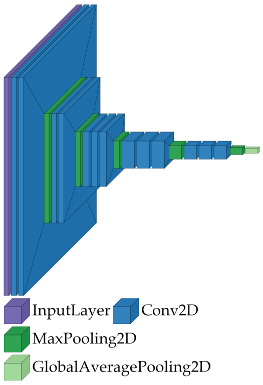


Figure 2.5: VGG16 network architecture

2.2.4.2 *Inceptionv3*

Szegedy *et al.* presented the original InceptionV1 architecture in 2015, the same year as VGG was introduced [52]. Both networks were inspired by AlexNet and participated in the ILSVRC-2014 competition. InceptionV1, called GoogLeNet, won first place ahead of VGG. Szegedy *et al.* observed that although images belong to the same class, the scale of their motifs can be highly different. So it would be beneficial to have multiple convolutional layers with different filter sizes in parallel to capture larger and smaller features simultaneously. However, a linear increase in the size of convolutional filters leads to a quadratic growth in the number of parameters. Szegedy *et al.* mitigate this issue by adding 1×1 convolutions ahead of the larger convolutional filter layers.

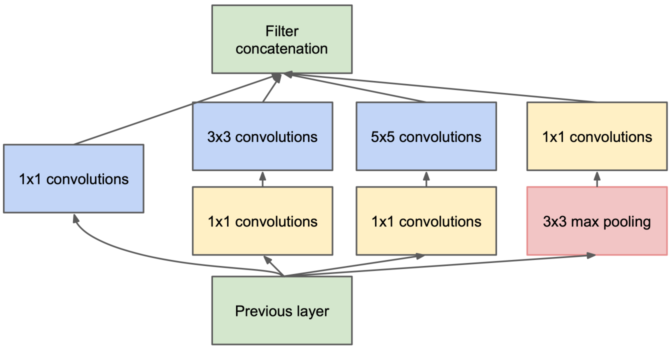


Figure 2.6: InceptionV1 module from [52]

Figure 2.6 depicts the architecture of a single inception block. The whole InceptionV1 network or GoogLeNet has nine blocks stacked on top of each other. Szegedy *et al.* used auxiliary losses at the fourth and the seventh block to handle the vanishing gradient problem. This architecture has 5 million parameters, a 12x reduction compared to AlexNet [53]. The subsequent versions two and three were presented in a follow-up work investigating further computational improvements [53]. The InceptionV2 architecture introduced two significant optimizations. One of them was factorizing the convolutional operations into smaller blocks, e.g., two stacked 3×3 convolutions are comparable with one 5×5 convolution but require 28 % fewer computations. The other optimization was reducing the grid size, i.e., the number of features, without a loss in the representational capacity. The final InceptionV2 architecture is 42 layers deep, while the computational costs are 2.5 times higher than the initial architecture. At the same time, the top-1 error decreased from 29 % to 23.4 % on the ILSVRC-2012 dataset. The third version of this architecture added a new path to the inception module introducing a 7×7 convolutional layer factorized into 3×3 convolutions. Moreover, it uses RMSprop instead of SGD for optimizing the loss function, added batch normalization layers to the auxiliary classification branches, and label smoothing.

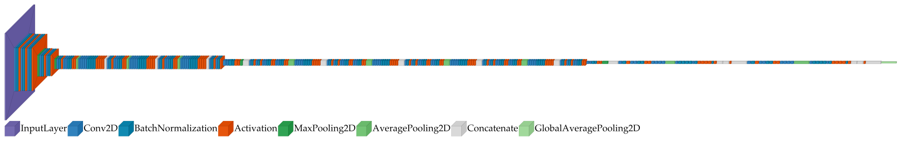


Figure 2.7: Inceptionv3 network architecture

Figure 2.7 shows the whole Inceptionv3 architecture as a flattened visualization without a parallel path of inception blocks. It achieves a top-1 error of 21.2 % on the ILSVRC-2012 dataset.

2.2.4.3 *ResNetV2*

The approach of VGG16, increasing depth with smaller convolutional filters, does not scale beyond tens of layers [54]. If more and more layers are stacked on top of each other, then the gradients from the loss function become smaller with every step of backpropagation. This issue is known as the vanishing gradient problem [55, 56]. As a result, the early convolution layers are not properly adjusted to the errors, and the deeper the network is, the worse its performance is. To address this issue, He *et al.* proposed to introduce so-called skip connections that add the results from the previous layer to the current one.

Figure 2.8 shows the skip connections introduced in the first version of ResNet [54] on the left and the improved version of ResNetV2 on the right. Adding the previous layers mitigates the vanishing gradient

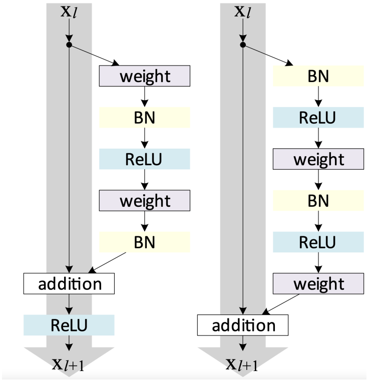


Figure 2.8: Identity mappings in ResNet architectures. Left: skip connection in ResNet50V1 architecture [54], right skip connection in ResNet50v2 architecture [57]. Illustration from [57].

problem as the skip connection introduces direct gradients for the previous layer. The difference between the first and the second version is the position where the addition takes place. The addition was computed ahead of the activation in the first version, while the second version made the architecture more modular. In this case, a ResNet block is shaped from batch normalization layers and convolution layers with ReLU activation functions. The skip connection adds the result from the previous block and the current block. This change allows faster training due to more distinctive gradients [57].

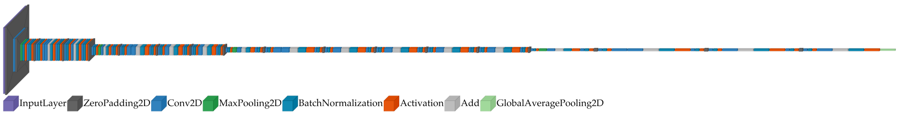


Figure 2.9: ResNet50v2 network architecture

These skip connections allow the training of architectures with 50 layers and more. Figure 2.9 illustrates the depth of the ResNet50v2 network but omits the skip connections. The extended version with 152 layers, ResNet152V1, won first place in the ILSVRC 2015 classification task.

2.2.4.4 *DenseNet*

Inspired by the ResNet idea of skip connections, Huang *et al.* introduced the DenseNet architecture [58]. They extend the idea of skip connections to all convolutional blocks so that every block obtains the feature maps from all preceding blocks. Figure 2.10 illustrates this with an example of five blocks: the last block receives the features from all other blocks. The direct information flow eliminates the need to pass on data from one layer to another and allows blocks with fewer filters. The term DenseNet is a reference to dense layers, where all input units are connected with all units from the previous layer. In this case, the units are convolutional blocks instead of single neurons.

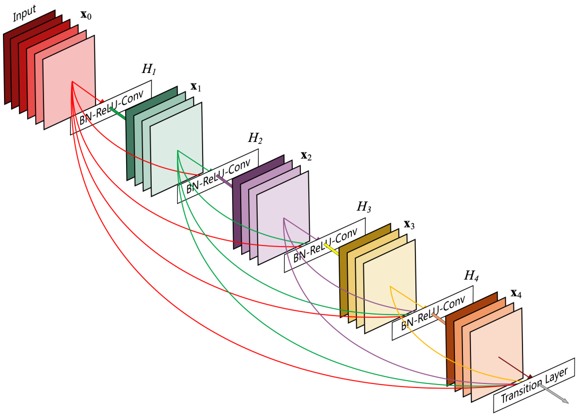


Figure 2.10: DenseNet architecture with five blocks. Each block is connected to all other blocks. Illustration from [58]

.

This architecture yields a similar performance as ResNet with less than 12x fewer parameters. Figure 2.11 shows the DenseNet121 backbone used in this thesis without skip connections. This architecture yields a top1-error rate of 25.02 % on the ILSVRC-2021 dataset.

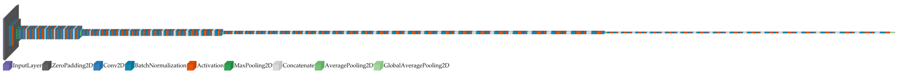


Figure 2.11: DenseNet121 network architecture

2.2.4.5 *InceptionResNetv2*

Szegedy *et al.* picked up their Inceptionv3 architecture and combined it with the idea of skip connection with InceptionResNet in two versions [59]. They replaced the filter concatenations of Inceptionv3 with the additive residual connections and improved the top1-error to 18.7 % with ResNet151 achieving 21.4 % and Inceptionv3 achieving 19.8 % on the same baseline. Figure 2.12 illustrates the depth of the InceptionResNetv2 architecture used in this thesis as a CNN backbone.

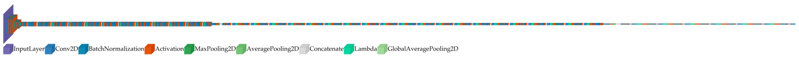


Figure 2.12: InceptionResNetv2 network architecture

2.2.4.6 *MobileNetV2*

The error rate of a model determines the winner of an ILSVRC competition. Hence, all architectures are optimized for a high accuracy achieved with enormous computational power. Moreover, as only the predicted labels for the test are reported, the runtime was neither a critical point. At the same time, smart, portable devices are ubiquitous but have limited computational power. Therefore, Howard *et al.* presented a lightweight network structure called MobileNet to run on mobile devices [60]. They replaced the standard convolutional filters with depthwise separable convolutions, which require 8 to 9 times less computation when used with a 3×3 filter. This change is similar to the improvements of the InceptionV2 architecture: By splitting the convolutional filter into two parts, the computation becomes easier. The default convolution acts on RGB images with three color channels. In contrast, the depthwise separable convolution works first on each channel individually and performs the channel convolution with a second 1×1 filter. Trained on ImageNet, this architecture with 4.2 million parameters achieved an accuracy of 70.6 %, which is similar to VGG (71.5 % accuracy, 138 million parameters) or InceptionV1 (69.8 % accuracy, 6.8 million parameters). Sandler *et al.* created a second version of MobileNet by adding skip connections as in the ResNet architecture [61]. However, their skip connections are called inverted residuals, which connect bottleneck layers instead of high-dimensional feature maps. The feature maps introduce a high complexity and require more computation, whereas bottleneck layers contain the information in a memory-efficient, compressed form. Finally, all operations on these layers need less computational effort.

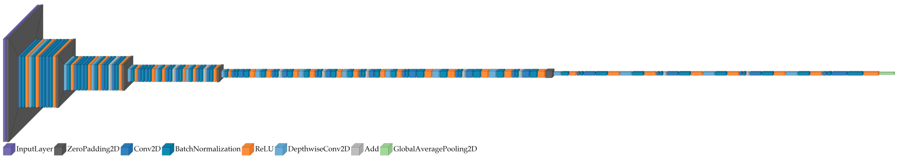


Figure 2.13: MobileNetV2 network architecture

Figure 2.13 sketches the depth of the MobileNetV2 CNN backbone used in this thesis. A full MobileNetV2 model trained on ImageNet yields higher accuracy than the initial architecture: 72.0 % compared to 70.6 % with the same number of parameters. However, the number of operations decreased from 575 million to 300 million, resulting in 33.6 % faster calculation on a Google Pixel 1 phone.

2.2.4.7 *Xception*

This architecture is a synthesis of Inceptionv3 and the depthwise separable convolutions from MobileNetV1 [62]. It replaces the standard convolutions in the Inceptionv3 architecture with the new convolutions from MobileNetV1. This yields slightly higher performance than Inceptionv3: The accuracy of Inceptionv3 with 78.2 % is increased to 79.0 %, but the number of parameters decreased by 770,776 to 22,855,952 parameters. Hence, it uses the parameters more efficiently

Architecture	Year	#Para (train)	#Features
DenseNet121 [58]	2017	6,953,856	1,024
InceptionResNetv2 [59]	2017	54,276,192	1,536
Inceptionv3 [53]	2016	21,768,352	2,048
MobileNetV2 [61]	2018	2,223,872	1,280
ResNet50v2 [57]	2016	23,519,360	2,048
VGG16 [51]	2015	14,714,688	512
Xception [62]	2017	20,806,952	2,048

Table 2.1: Year of publication, number of trainable parameters in the backbone, and output dimensions (#Features) per deep vision architecture in alphabetic order

than the Inceptionv3 architecture. Figure 2.14 illustrates the depth of the CNN backbone of this network.

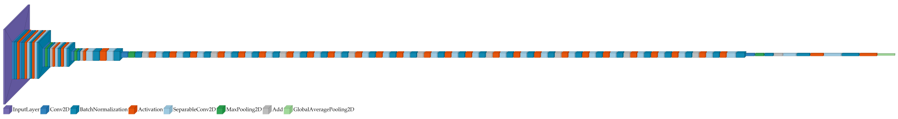


Figure 2.14: Xception network architecture

2.2.4.8 *Summary*

Starting with AlexNet in 2012, several CNN architectures have been proposed with more depth and new concepts for tackling vanishing gradients and efficient computation. This thesis uses seven of these state-of-the-art architectures for building function classification. Table 2.1 gives an overview of all architectures with their year of publication, the number of trainable parameters, and the number of features resulting from their CNN backbone after a pooling layer. The number of parameters in this table is lower than the official number in the publications as only the CNN backbone is considered, and the final classification layers for ImageNet are removed. The number of trainable parameters varies highly, starting with MobileNetV2 with 2.2 million parameters and going up to 54.3 million parameters of InceptionResNetv2. However, the number of features from these architectures is similar, with 512 features from VGG16 up to 2,048 features from Inceptionv3, ResNet50v2, and Xception.

2.2.5 *Deep Learning Architectures for Object Detection*

The success of deep learning methods for image classification enabled new methods for a downstream task: object detection in images. An image is assumed to be centered on a single object with an irrelevant background in image classification. However, in reality, many photos contain many objects of different sizes. Object detection aims to find each of them and describe the object class and its position in the image. Figure 2.15 illustrates how the result can look: objects of different sizes are detected with bounding boxes and class names. The bus covers a significant part of this photo, but the cars in the background are also detected. The fire hydrant in the center is wrongly detected as a person.

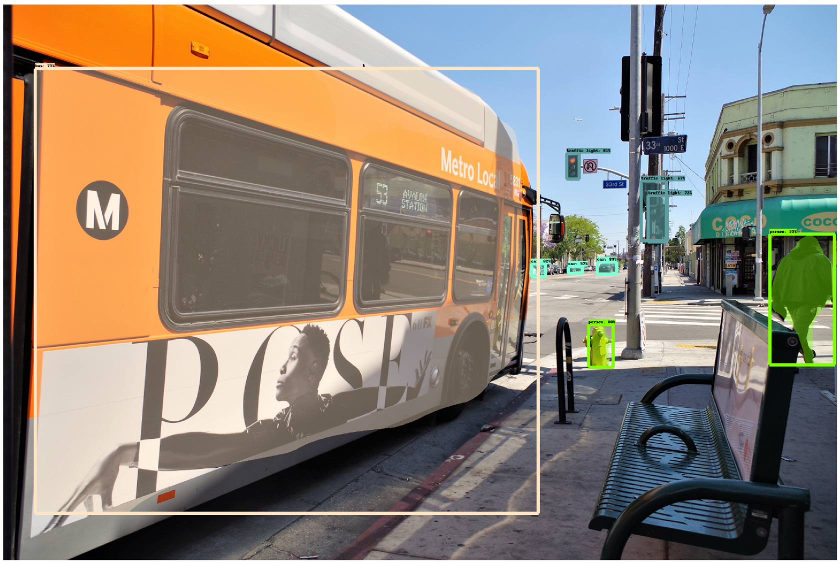


Figure 2.15: Object detections with Faster R-CNN [63] in a social media image. Photo Historic Corridor of Central Ave. L.A. by joey zanotti is licensed under CC BY 2.0

In 2014 Girshick *et al.* presented an object detection algorithm using the power of AlexNet [64]. Their algorithm splits the image into smaller parts that might contain an object and pass all of them through an AlexNet trained on ImageNet. However, they cut off the classification part of AlexNet and used it as a feature extractor for the object candidates. They use a Support Vector Machine (SVM) to predict the classes of an object based on the extracted features. They find their object candidates with a selective search algorithm [65] and call them region proposals. "Since we combine region proposals with CNNs, we call our method R-CNN: Regions with CNN features." ([64]) This algorithm had a better performance than the best model in the ILSVRC2013 object detection competition with an mAP of 31.4 % (compared to 24.3 % of the ILSVRC2013 detection winner OverFeat [66]). However, it had three drawbacks: a multi-stage training pipeline, a time-consuming training process, and a slow object detection [67]. Girshick proposed the Fast R-CNN architecture based on a pre-trained VGG network trained in an end-to-end fashion to overcome these limitations. He replaced the SVM classifier with two new outputs of the VGG network: One for predicting the object class and a second one for predicting the object bounding box. Trained with a multi-task loss, he optimized both outputs in parallel. This setup improved the detection performance by up to 9.6 % and the detection runtime by up to 213x compared to R-CNN. However, as the region proposal generation remained unchanged with a selection search algorithm, this became a computational bottleneck. Ren *et al.* solved this issue by creating a *region proposal network (RPN)* [63]. This subnetwork slides over the feature maps of a CNN like VGG and predicts regions of interest. With all functions being merged in one network architecture, it can be trained end-to-end, and predictions become faster. They report a framerate of 5 fps with pure GPU implementation with a similar detection performance as Fast R-CNN.

In this thesis, a Faster R-CNN architecture is used with an Inception-ResNetv2 CNN backbone [68]. The network was trained on the Open Images V4 dataset [69, 70]. This dataset has 15.4 million bounding boxes for 600 object classes, 15x more than the next largest datasets.

2.3 SUMMARY

This chapter introduced the fundamentals of EO data and their analysis with state-of-the-art deep learning architectures. It presented the basic principles of how EO data is captured with different sensors and the different types of resolution. Subsequently, this chapter provided a high-level overview of the principles of supervised machine learning and the evaluation of machine learning algorithms. Moreover, the key deep learning architectures of the last decade were presented as they build the foundation of this thesis for every image classification task: For street-level, social media, and remote sensing images.

This knowledge opens the way to the next chapter, which introduces the related work for this thesis. The breakthrough of CNNs in computer vision enabled several downstream tasks, which are presented in this following chapter.

Part III

RELATED WORK

Pour y voir plus clair,
il suffit souvent de changer
la direction de son regard.

— ANTOINE DE SAINT-EXUPÉRY, Citadelle

3 RELATED WORK ON URBAN ANALYSIS

This chapter introduces related work for all modalities used in this thesis. Although the main focus of this work is building function classification, the following subsections present land use classification methods and related applications based on the data sources. The order of data sources and applications follows the overall structure of this thesis, especially the part of the contributions. Hence, it starts with methods and applications of commercial street view level imagery and continues with approaches based on social media images. Afterward, there is related work on social media text data and combinations of images and text. The subsequent sections focus on land use classification with remote sensing data and fusion methods for combining ground view and remote sensing data.

3.1 APPLICATIONS OF STREET-LEVEL IMAGES

Among the different commercial platforms, three are commonly used in research. Firstly, Google Street View (GSV), the most popular one; secondly, Mapillary, with crowdsourced image data; and thirdly, Tencent Street View, mainly covering Asia. GSV has gained high popularity because of its worldwide coverage with high-quality data. Its Application Programming Interface (API) allows fine-grained settings on the imagery. However, as of today, Google prohibits downloading, scraping, or storing GSV images in their terms of service [71, §3.2.3]. These restrictions prevent the use of GSV images in research with creating and sharing datasets. So downloading image data is technically possible but legally prohibited. Since there are many publicly available datasets with GSV images for research [16, 72–75], Google seems to tolerate academic use of this data. Nevertheless, GSV continues to be the most crucial source of street-level imagery as their image quality is assured by a fleet of dedicated cars keeping their database up-to-date with specialized cameras. In contrast to this approach, Mapillary provides a platform for crowdsourced street view data captured with a smartphone app by volunteers. Their significant advantage is the lower effort, but it comes at the cost of varying image quality, a limited angle of view focused on the street, and uneven spatial distribution. Last but not least, Tencent offers street view data in China, but it has a limited spatial extent and an API documentation in Chinese. Therefore, its user basis concentrates on Chinese researchers.

Among the first studies using GSV for building function classification is a work from Movshovitz-Attias *et al.*. They predict 208 services offered in buildings based on the façade images [76]. Their labels are based on an ontology from Google Map Maker, and they sample 1.3 million images globally. With a GoogLeNet architecture [52], they report a 69 % accuracy on a spatial split test set of 100,000 images.

Kang *et al.* extended the task to more building types, e.g., apartments or churches, while simplifying the classification scheme to eight classes [16]. They obtain building labels from OSM and download one or more GSV images for these buildings. After decision-level fusion of all predictions for one building, they achieve an F1 score of 0.58 with a VGG16 architecture. With a multiscale AlexNet architecture, Qiao *et al.* gain 54.7 % overall accuracy on the same dataset. However, as both publications report different metrics and models, they are not comparable.

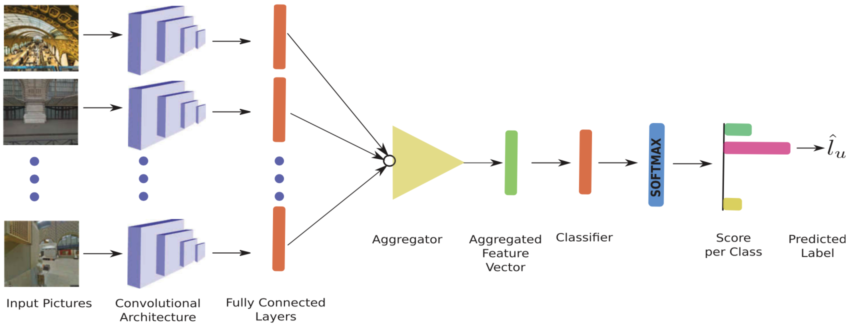


Figure 3.1: Multi stream CNN architecture of Srivastava *et al.* with averaging aggregation function. Illustration from [77].

The approach of Srivastava *et al.* is similar to Kang *et al.*, but instead of decision-level fusion, they average the VGG16 features of multiple images [77]. Figure 3.1 illustrates their multi-stream CNN architecture with an average aggregation function. In their study area of Île-de-France, the Parisian metropolitan area, they report an average accuracy of 60 %. Fang *et al.* present a study using Tencent Street View data from Wuhan, China, to predict five land use classes at a parcel level [78]. They use 360-degree panoramas and cut them into different parts according to the position of the parcels. Their evaluation compares seven state-of-the-art architectures and finds the ResNet50 architecture best suited for the given task with an accuracy of 71 %.

Beyond land use classification in urban areas, tree mapping is an essential application as they have a severe impact on health [79, 80]. This mapping can be either done manually with handcrafted features from super-resolution pixels [81] or in an automated fashion [82]. The latter study shows that existing tree databases have an approximate address of the trees but lack precise geo coordinates. Their solution to this issue relies on an object detection algorithm to find the trees in GSV images and a municipality-wide optimization that minimizes the total distance between all pairs of trees. This method assigns a geographic location to 56 % of all 1,100,952 trees in 48 Californian municipalities. Krylov *et al.* present a similar, more generic pipeline for geocoding objects in GSV images [83]. In addition to a CNN for object detection, they predict a depth map with a second CNN and combine this information in a triangulation algorithm based on a custom Markov random field model. They report a position precision of approximately 2 m for traffic lights and telegraph poles.

Moreover, the abundance of GSV images inspired further applications. For example, Gebru *et al.* predict socio-economic attributes based on car models in the US [84]. Using a deep learning algorithm, they classify the car models and use this information as a proxy for inferring income, race, education, and voting patterns. Goel *et al.* analyze if GSV images are suitable for estimating travel patterns at a city level in Great Britain [85]. By correlating their manual object counts with survey data, they found strong correlations between the use of bicycles and buses and their occurrence in GSV images. While social media images are mostly used for disaster response [86–88], GSV images can help for risk assessment in case of flooding or earthquakes. Ianelli and Dell'Acqua estimate the number of floors per building using a VGG16 architecture [89]. They report a classification accuracy of almost 85 % in their San Francisco, US, study area. For the latter use case of seismic risk assessment, Aravena Pelizari *et al.* develop a fine-grained hierarchical scheme to classify building characteristics [29]. They report an accuracy above 80 % on their building structure scheme of 14 classes in their study area of Santiago de Chile.

Despite all research success with GSV imagery, the legal issue is still prevalent. Thus, other publications investigate the suitability of open social media data for similar applications. The following section provides an overview of these efforts.

3.2 Applications of Social Media Data

This section is split into two subsections: The first one discusses different applications of social media images focusing on land use classification at different scales. However, it also describes other applications of social media images, including but not limited to landscape aesthetics. The second part is structured similarly but considers publications on social media text and metadata.

Geotagged Social Media Images

With social media image platforms becoming more and more popular, the number of geotagged images increased heavily. This enabled studies on landmark detection and touristic routes as in [90]. In their work, Crandall *et al.* use a dataset of 35 million geotagged Flickr images to identify landmarks in metropolitan areas around the world using clustering of location, visual, and textual features. Snavely *et al.* use sets of geotagged images around landmarks from Flickr to create 3D models [91]. They compute handcrafted SIFT features [92] and use them with a structure-from-motion approach for computing the 3D point clouds for eight landmarks. Additionally, they show how these point clouds can be aligned geographically with world coordinates and Digital Elevation Model (DEM). Paldino *et al.* investigate the attractiveness of different cities for residents, domestic, and foreign tourists [93]. Based on the spatial distribution of Flickr images, they analyze differences in habits between US and European tourists when visiting each other.

Many publications have studied land cover mapping, and land use as this application is highly intuitive. Among the first is a work from Leung and Newsam [94], who used geotagged Flickr and Geograph images for land cover classification in the London metropolitan area. They generate image features with edge histogram descriptors and train an SVM. This method yields an accuracy of 75 % on a binary land cover scheme. Last but not least, they coin the term *proximate sensing* in this work when using publicly available photos for mapping purposes. Oba *et al.* extend this approach with a larger spatial scale, a different feature extractor, and a more fine-grained classification scheme [95]. Instead of edge histogram descriptors, they use handcrafted SURF features [96] with an SVM. Their training set is built using a keyword search on Flickr to obtain photos labeled with a keyword from the classification scheme. They achieve an accuracy of at least 60 % on a classification scheme with six classes for the US.

Xie and Newsam build upon the results from Leung and Newsam to create a scenicness map of Great Britain using Geograph images [97]. They use handcrafted gist features [98] and learn a regression function that predicts the scenicness of a location with nearby photos. The novelty of this work is the combination of image location and image content information in one regression function. Langemeyer *et al.* describe an alternative, manual approach based on handcrafted categories and expert labeling to assess landscape aesthetics [99]. Havinga *et al.* take the work from Xie and Newsam into the deep learning era. They substitute the handcrafted features with two CNNs and combine their results with a random forest to predict a scenicness score in Great Britain [100]. While the first CNN uses a ResNet50 architecture with Places365 weights, the second one utilizes image attributes predicted with a CNN trained on the SUN dataset [101]. They use a grid with 5 km by 5 km cells and aggregate the individual predictions of all images within a cell using average fusion. Together with an environmental indicator model, the aggregated predictions are converted into a scenicness score by a random forest model.

Land use is a slightly different use case than land cover but is also latently encoded in social media images. Leung and Newsam exploit

Flickr images to predict three different land use classes, *academic*, *residential*, and *sports* on two campuses in California, US [102]. They train an SVM with handcrafted features, a bag of visual words (BoW) [103], to predict land use on a rasterized map of the campuses. Their approach yields an accuracy of 60 % when trained on one campus and tested on the other. Object Bank features [104] pose an alternative to BoW features. Fang *et al.* apply them to Geograph pictures from London metropolitan area to predict urban land use on a hierarchical level [105]. They utilize the completeness of the road network in OSM to define different block levels: the lower block levels are defined by small roads like local and neighborhood streets. Subsequently, each level-up is defined by secondary roads, primary roads, and primary highways. Based on the Object Bank features, they calculate the land use with a majority voting algorithm. They report an overall accuracy of 76.5 % for a classification scheme with five classes. With the rise of deep learning methods, the methods switched from handcrafted features and SVMs to CNN-based approaches. Zhu and Newsam use an existing CNN trained on the Places365 dataset [106] to split a set of geotagged Flickr images first into indoor and outdoor scenes [107]. Furthermore, they use the CNN features for training an SVM for a final land use classification. This hierarchical approach achieves up to 76 % accuracy on a classification scheme with eight classes.

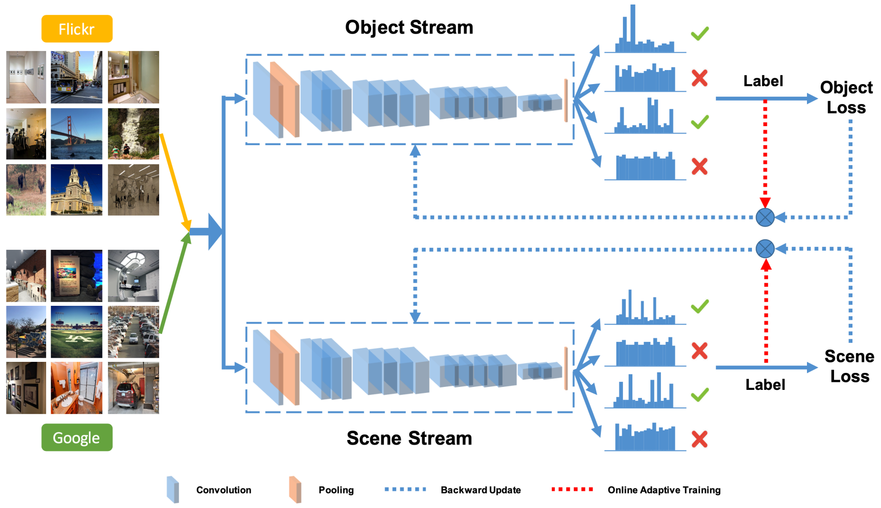


Figure 3.2: Two stream CNN architecture of Zhu *et al.* with scene and object branch trained with Flickr and Google images. Illustration from [108].

A more sophisticated method from Zhu *et al.* extends this work with a more fine-grained classification scheme, an improved training set collection, and a two-stream CNN [108]. They pruned a hierarchical taxonomy from the American Planning Association to three levels with 5, 16, and 45 classes each. Their training dataset is built using a Google image search with each of the 45 classes to create a labeled image set unrelated to the Flickr images for testing. The two-stream network has one branch for object classification and one for scene classification, both with fixed convolutional layers from different training sets (Figure 3.2). While the object stream is initialized with ImageNet weights [109], the scene-oriented stream is set up with weighted from Places365 [106]. The authors assume a domain shift from the Google image set and the Flickr image set. Hence, they train the network with mixed batches of Google and Flickr images. The Flickr images are assigned to the next parcel if they are inside or less than five meters away. This approach yields an accuracy of 49.54 % on the most fine-grained level of 45 classes. A more comprehensive review of urban land use classification based on imagery can be found in [110].

Closely related to land use classification is activity mapping. Zhu and Newsam use geotagged Youtube videos to predict human activities using a two-stream CNN [111]. One stream uses the MotionNet [112] for spatial features, and one stream is a CNN for temporal features. They are combined with an average late fusion to predict the final ten activity classes. This network yields an accuracy of 90.94 % on the task.

Beyond urban environments, crowdsourced images are also used for crop monitoring. D'Andrimont *et al.* evaluate the availability of Mapillary images across the European Union to monitor eight crop types [113]. They conclude that the heterogeneous data quality of the crowdsourced street view imagery is challenging, and the spatial and temporal sparsity create additional data uncertainty. Although Flickr images tend to focus on art photography, Chaudhary *et al.* show that they contain helpful information for disaster response. In their work on water height estimation, they use a multitask learning approach to predict water levels in flooded areas [86]. As accurate, numerical water height levels are hard to obtain for single Flickr images, they apply a comparative loss and train a VGG architecture with image pairs to differentiate between higher and lower water levels. They report an RMSE of less than 12 cm for their final regression model.

Geotagged Social Media Text

Apart from GSV images and social media images, urban land use can also be predicted using metadata and textual features. Huang *et al.* group tweets with Latent Dirichlet Allocation (LDA) into topics and use this as features for a Long Short-Term Memory (LSTM) [24]. The LSTM aggregates tweets that belong to the same spatial building block together with their temporal metadata. Based on a classification scheme with three classes, they report an average accuracy of 63 % in their study area of Munich, Germany. Häberle *et al.* investigate the suitability of Twitter tweets for building function classification [114]. Their study analyzes three different word embedding methods and how building functions are represented in the embeddings of surrounding tweets. They conclude that although there are clusters in the embeddings, no general patterns can be derived. In their follow-up study, they predict buildings functions in Berlin, Germany, with a feed-forward network trained on fastText [115] sentence vectors [22]. They achieve F1 scores up 0.87 for the best performing *accommodation* class but report F1 scores of 0.2 for *commercial* buildings. Their conclusion summarizes that the linguistic patterns between the five classes are not distinctive due to multilingual input. While Häberle *et al.* have a building-centric approach, Terroso-Saenz and Muñoz create land-use areas using a clustering method [116]. They cluster locations of Flickr posts with a density-based approach and obtain a land use label for these clusters using the most abundant venue type from Foursquare. Their Flickr datasets in the study areas of New York City, US, and San Francisco, US, are extracted from the Yahoo Flickr Creative Commons 100M dataset [117]. Their approach does not take the images into account but focuses on the text of each post. They report an average F1 score of 0.71 in New York City and 0.67 in San Francisco with a random forest classifier on a six-class scheme. Terroso-Saenz *et al.* extend this work by combining multiple data sources. They use taxi trajectories, social media posts from Twitter and Flickr, and the public transport network from subways and busses to create time-based features [118] and thus, predict urban land use in New York City and Chicago. They achieve an accuracy above 80 % on 14 land use classes from Km4City ontology [18] with a random forest classifier. As land use classification schemes always simplify reality and cannot capture a variety of possible combinations of usages, Dax and Werner introduce the concept of abstaining classifiers [119]. They use the abstaining principle to decide if a classifier's prediction is helpful for a task or not. Their approach increases the precision at the cost of a lower recall in their study area of Los Angeles, US.

Furthermore, Twitter data can be used to identify the activities of citizens in urban green spaces [120]. Hamstead *et al.* investigate how people interact with parks and conclude that from "a social equity perspective, the findings may imply that parks in high-minority neighborhoods are not as accessible, do not accommodate as many visitors, and/or are of lower quality than those in low-minority neighborhoods" ([121]). From a socio-economic perspective, Twitter data shows high correlations with income. For example, Mitchell *et al.* perform a sentiment analysis on tweets from the United States and reveal a strong correlation between household income and positive sentiment [122]. Li *et al.* investigate where tweets and Flickr photos are posted and find that "tweet density is highly dependent on the percentage of well-educated people with an advanced degree and a good salary who work in the areas of management, business, science, and arts. The second model suggests that high photo density is correlated with a high percentage of white and Asian people with an advanced degree in the areas of management, business, science, and arts." ([123]) Bokányi *et al.* confirm these findings and conclude there is a strong correlation between the words and language used in tweets and the socio-economical and cultural similarities, like degree of urbanization, religion, or ethnicity [124]. Beyond these influences, Twitter data helps to identify mobility patterns, e.g., during the first lockdown in Germany during the Covid-19 pandemic [125], or how people perceive public transport like the metro in Madrid [126].

3.3 URBAN LAND USE PREDICTION WITH REMOTE SENSING DATA

Urban remote sensing creates geoinformation products from EO data to analyze socio-economic, ecological, and cultural aspects of human life [127]. Parts of these products are land cover and land use maps. Although they are related concepts, they reflect different perspectives of spatial data. "*Land cover* is determined by direct observation while *land use* requires socioeconomic interpretation of the activities that take place on that surface." ([15])

Land use classification is not based on direct measurements but on interpretation, and hence, it requires high-resolution optical images [128]. Therefore, Yang and Newsam use aerial images sampled from different states of the United States Geological Survey (USGS) National Map and create a new dataset called UC Merced Land Use Dataset [129]. This dataset became a benchmark for several studies that improved the original baseline of 81.19 % overall accuracy. The dataset consists of 100 images for 21 classes, 2,100 in total. Yang and Newsam propose a bag-of-visual-words based on SIFT features [92] to create the baseline. With the rise of deep learning methods, CNNs models yielded above 90 % accuracy on the UC Merced Land Use Dataset. For example, Castelluccio *et al.* report an accuracy of 97.10 % on this dataset when using a fine-tuned GoogLeNet architecture [130]. However, the relatively small size of the dataset makes CNN-based models with millions of parameters prone to overfitting. Albert *et al.* propose an alternative land use classification dataset for benchmarks with ten classes, including agricultural and water bodies [131]. Their labels are from the Urban Atlas [132], and hence, its classification scheme is a mixture between land use and land cover. This dataset contains more than 140,000 image patches from Google Maps optical images at zoom level 17, i.e., approximately 1.2 m GSD, covering six European cities. They report 68 % - 83 % accuracy within cities and 23 % - 54 % accuracy on cross-city evaluations with a fine-tuned ResNet50 architecture.

The launch of high-resolution optical satellites like QuickBird, IKONOS, and RapidEye enabled unprecedented studies on land use classification with remote sensing data from space. Their increased spatial resolution enabled clear distinctions of objects in urban environments. An example of such a study is a work by Hu and Wang. They use handcrafted features and decision trees in a study area of Austin, Texas, US, to predict four land use classes with an overall accuracy of 61.68 % [133]. Ruiz Hernandez and Shi propose a similar approach with more sophisticated features and random forests to classify land use in Ciudad Juarez, Mexico [134]. They combine textural and spatial features and yield an overall accuracy of 92.3 % for five land use classes. As deep learning models showed their superiority over handcrafted features, different studies on using CNNs for urban land use classification were published. An example is a work from Huang *et al.* who use a two-stream CNN for predicting 13 land use classes in Shenzhen and Hong Kong [21]. Their network takes high spatial resolution, multispectral remote sensing imagery as input: WorldView-3 with eight channels and WorldView-2 with four channels. Their two-stream CNN model has one branch for three-band RGB channels with a large receptive field and one branch for all multispectral channels with a small receptive field. They achieve an overall accuracy of 80 % in Shenzhen and 91 % overall accuracy in Hong Kong. However, with more and more studies being published in different datasets, the need for available benchmark datasets became inevitable. The image scene classification dataset NWPU-RESISC45 fills this gap [135]. It consists of 45 classes with 700 images per class; 31,500 images in total. Cheng *et al.* use Google Earth imagery on multiple scales from 0.2 m GSD to 30 m GSD. They propose a fine-tuned VGG16 model with an accuracy of 90.36 % as a baseline.

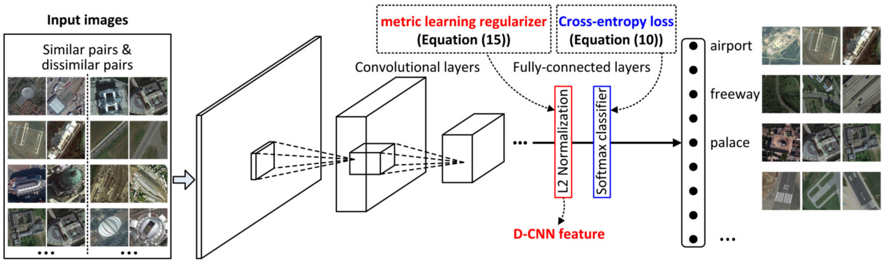


Figure 3.3: CNN architecture with metric learning term for regularization of Cheng *et al.*. Illustration from [136].

In a follow-up publication, Cheng *et al.* extend their work by using a discriminative CNN model [136]. They add a metric learning regularization term to the dense layer before the softmax, which forces the network to learn more discriminative features (Figure 3.3). This regularization term is based on a contrastive embedding [137] pushing features of the same class closer together and those of other classes apart. Their discriminative CNN with a VGG16 backbone gains 91.89 % accuracy on the NWPU-RESISC45 dataset. The Functional Map of the World (FMoW) poses an alternative benchmark dataset [138]. It is based on QuickBird-2, GeoEye-1, WorldView-2, and WorldView-3 data and contains crowdsourced object boxes from experts. Moreover, it features a global coverage with 63 categories and a multi-temporal image series for each sample. Christie *et al.* set an LSTM model as a baseline with an average F1 score of 0.731. The LSTM combines the deep features from different time steps and uses a DenseNet121 backbone together with a metadata branch. Minetto *et al.* improve this result with an ensemble of CNNs called *Hydra* [139]. Their combination of DenseNet and ResNet architectures yields an average F1 score of 0.781, ranking them third in the official competition. Moreover, they report an accuracy of 94.51 % on the NWPU-RESISC45 dataset with *Hydra*. All recent successes on the NWPU-RESISC45 and remote sensing scene classification have been reviewed by Cheng *et al.* in [140].

Apart from urban areas, Campos-Taberner *et al.* propose a bidirectional LSTM for agricultural land use classification with multitemporal Sentinel-2 data [141]. They report an overall accuracy of 98.7 % in their study area of València, Spain.

Zhang *et al.* exploit the close relationship between land cover and land use by learning them jointly with two networks [142]. A Multilayer Perceptron (MLP) predicts the land cover for an image patch, which is used as a conditional probability for an object-based CNN that predicts the land use. This output is used as a priori knowledge for the next training step of the MLP. Hence, this process can be formulated as a Markov process. They test their method in two study areas in Manchester, GB, and Southampton, GB, on aerial images with 0.5 m GSD and report an overall accuracy of 90.18 % for land cover classification and 87.92 % for land use classification.

A related concept to land use is Local Climate Zones (LCZ) [143]. They capture the morphological features of urban environments and are suited, e.g., for analyzing urban heat islands. Qiu *et al.* propose an LSTM that combines features from a ResNet architecture to predict LCZ using multi-seasonal Sentinel-2 imagery [144]. For each time step, the ResNet backbone extracts image features from a scene, while the LSTM combines the multitemporal features for the final classification. Their approach achieves an overall accuracy of 84 % on six classes. They extend this work by simplifying the method and replacing the LSTM with decision level fusion and yield an overall accuracy of 86.7 % [145] LCZ definition.

3.4 FUSION OF REMOTE SENSING AND GROUND-LEVEL DATA

As ground-level data and remote sensing data have a complementary perspective on the world, several studies proposed methods to combine the best of both modalities. However, the temporal and spatial alignment of different modalities poses a major challenge, as well as handling the many-to-many relationships between multiple input data points and the matching spatial targets. This section introduces seven selected methods focusing on land use classification exemplary for the state-of-the-art in this research field.

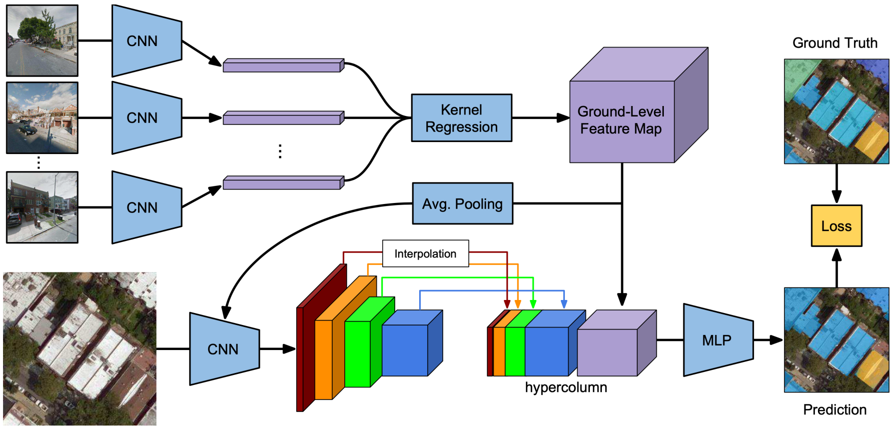


Figure 3.4: CNN architecture of Workman *et al.* that combines street-level and aerial images. Illustration from [146].

Workman *et al.* propose a CNN architecture that integrates deep features from ground-level imagery as an additional feature map to the CNN processing the aerial imagery [146]. They combine multiple GSV images with a kernel regression to a ground-level feature map that is stacked together with feature maps from convolutions of the aerial image (Figure 3.4). Workman *et al.* apply their method to two study areas in New York City, US: Brooklyn and Queens. Based on aerial imagery from Bing Maps and panoramic GSV images, they report a top-1 accuracy of 45 % in Brooklyn for a building function classification scheme of 206 classes. An extension of their approach includes a geospatial attention mechanism, which increases the accuracy to 60 % [147].

As a pilot study for this thesis, Hoffmann *et al.* investigate the combination of aerial and ground-level imagery with different multimodal fusion strategies [20]. Their study is based on Google Maps images at three different zoom levels together with GSV images for building function classification. They sample their dataset from labeled OSM buildings of all 52 states of the US. They train two CNN architectures, Inceptionv3 and VGG16, for each modality and analyze which fusion strategy yields the best performance. The fusion strategies are early feature fusion (stacking feature maps from convolutional layers), late feature fusion (concatenating feature vectors of dense layers), decision level fusion (averaging the probabilities of the softmax layer), and model stacking (training a naïve Bayes classifier to predict the best probability vector). They achieved the best results with decision-level fusion on a four-class scheme and increased the precision of 67 % from the best unimodal to 76 % when fusing all modalities.

Similar work is presented by Srivastava *et al.*, who extend their CNN based on GSV images [77] with a second CNN branch that includes Google Maps aerial imagery [148]. They concatenate the dense features from the ground-level CNN with the dense features from the aerial CNN. All CNN branches are based on a VGG16 backbone. Additionally, they show how to handle missing modalities, aerial or GSV images, by substituting them with the next neighbor image in a feature embedding space. They improve their result of pure GSV from 60 % to 70 % on 16 classes when using the two-stream CNN in the metropolitan area of Paris, France.

Leichter *et al.* show how Twitter data can be fused with optical remote sensing data. They propose a method for LCZ classification based on Sentinel-2 data and features derived from geotagged Twitter tweets [149]. Their custom CNN takes Sentinel-2 channels at multiple resolutions and augments the resulting feature maps from the convolutional layers with six feature maps calculated on Twitter data. Their Twitter feature maps contain, among others, the number of tweets and the mean text length and are rasterized based on the geotags. Leichter *et al.* achieve a performance increase by 1.3 % to 78.6 % when adding the Twitter feature maps in their study area of Washington, DC, US.

Zhang *et al.* combine remote sensing data from GaoFen-2, a Chinese panchromatic and multispectral satellite, with social media posts from Weibo and POI data from Baido to predict urban land use on a parcel level in Beijing [150]. They derive the parcels from segmentations defined by the OSM road network. Their method is based on handcrafted features used by a random forest and achieves an accuracy of 77.83 % in their study area of Haidian District, Beijing, China.

A similar study uses Google Earth data and POI data from Amap for land use classification in Wuhan. Lin *et al.* predict seven land use classes based on eight geometrical features from building footprints and eight textural features from Google Earth imagery [151]. They obtain their POI data from Amap, a Chinese geoinformation provider Alibaba owns. Their method distinguishes if there is sufficient data in the vicinity and predicts the land use class by nearest neighbor classification in the feature space or based on kernel density estimation (KDE) that builds on the neighborhood similarity of buildings. They report an accuracy of 68 % for the spatial similarity approach and 66 % when applying KDE.

Salem *et al.* present a generic method for generating time-aware embeddings that combine ground and aerial views [152]. They create their Cross-View Time (CVT) dataset with global coverage based on two other datasets: First, Archive of Many Outdoor Scenes (AMOS), a collection of webcam images around the world [153, 154] and second, Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M) [117]. Together with aerial data from Bing maps, they train a CNN-based embedding that takes ground views along with their metadata as additional input. The final embedding can retrieve ground-level images for a given month and hour, reflecting the visual appearance at that date and time. They state that "our objective is to construct a map that represents the expected appearance at any geographic location and time." ([152]) This embedding and the resulting map have three applications: Cross-view image retrieval, image localization, and metadata verification. They report an accuracy of 31.2 % for localization and an accuracy of 40.3 % for time verification.

A more general review focussing on data fusion of multi- and hyperspectral data, SAR data, and LiDAR data is provided by Ghamisi *et al.* in [155].

3.5 DIFFERENTIATION OF THIS WORK

The contribution of this thesis concerning all presented works is in the combination of six aspects:

1. The underlying data distribution is global, with 42 study areas from different cultural zones
2. All inferencing is executed on real-world social media data
3. All experiments are based on large-scale raw datasets
4. Urban land use classification is performed at the most fine-grained level of individual building instances
5. The final prediction results are evaluated on a human-verified subset of buildings
6. Three different data modalities, two of them based on social media data, are used for building instance prediction and finally fused

The following part of this thesis describes the datasets in more detail and discusses the methods for predicting building functions on a global scale.

Part IV

CONTRIBUTIONS

Zum Beispiel mit gelehrten Sachen
kann man sich vielfach nützlich machen.

— WILHELM BUSCH, Die Haarbeutel

4 DATASETS

For this thesis, five different sources of data were considered:

1. OpenStreetMap for global building information
2. Google Maps aerial images as remote sensing data
3. Google Street View images as the gold standard for façade-oriented ground-level images
4. Geotagged Flickr images as a source of social media images
5. Geotagged Twitter tweets as a source of social media posts

The first source, OpenStreetMap (OSM), provides ground truth information for extracting labels on a building instance level, also called target data. All other sources are used as input for creating, developing, and training machine learning models.

OSM provides building function information on different levels of granularity. OSM has three different, optional tags for indicating building functions: *building*, *amenity*, and *shop*. This thesis uses a unified scheme of three classes for building function classification: *commercial*, *residential*, and *other*. Although this simple scheme has its limitations, it allows fine-grained population density estimations by including only residential buildings and improved estimations of the economic strength by providing details about commercial activities. *Commercial* buildings include offices and retail buildings as well as malls and halls on industrial sites. *Residential* places can be single, detached houses, large apartment blocks, and high-rise skyscrapers with a primary focus on providing permanent housing for people. The class *other* consists mainly of civic buildings, e.g., schools, universities, city halls, and built landmarks.

As input data, four different sources are considered: Google Maps aerial images as very high-resolution remote sensing data, Google Street View (GSV) images as ground-level imagery focusing on building façade images, geotagged Flickr images as a source of social media images and geotagged Twitter tweets for spatially and temporally dense social media posts.

This thesis focuses on 42 metropolitan areas as defined in the So2Sat LCZ classification benchmark [156] covering cities in all cultural zones and continents. Figure 4.1 shows the distribution of these areas on the globe. This work is part of the So2Sat project [157], which combines social media and satellite data to create 3D maps of urban settlements with detailed semantics. Further, these cities will be referred to as the LCZ42 cities.

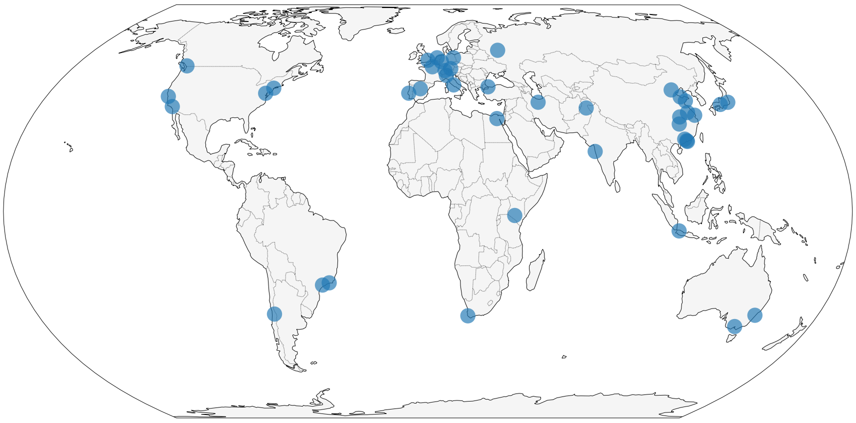


Figure 4.1: Distribution of LCZ42 cities [156] on the globe

4.1 OPENSTREETMAP BUILDINGS AS GROUND TRUTH DATA

OpenStreetMap (OSM) was founded as a wiki-style, collaborative mapping project in London, 2004 [158], and has more than 7.9 million registered contributors [159]. They insert and correct geographical data based on aerial imagery and field studies. OSM data covers, for example, streets, buildings, administrative boundaries, and topological information, but it is not limited to these entities. However, the contributors of OSM add and correct the data voluntarily without strict guidelines. Thus, the data quality is mixed: E. g. urban areas are more frequently updated than rural areas [160]. Moreover, building footprints and their semantic annotations show high variations concerning accuracy [161] and completeness [162].

This thesis builds upon a global OSM data dump from July 2020. Based on this dump, all building information has been extracted. This information includes building footprint information as well as semantic details about building types, amenities, and shops inside these buildings. In summary, there are 400,800,001 buildings worldwide included in this dataset, 73,889,355 of them have a known function according to the mapping defined in Table A.1 and Table A.2.

Figure 4.2 shows the number of buildings in each LCZ42 city as the total number of buildings mapped in a city and the number of buildings having a semantic label. These cities have significant differences, even when considering their spatial extents. Paris and Jakarta have more than three million mapped buildings, but less than ten percent have labels. Los Angeles has the best coverage of labeled buildings, both absolute and relative numbers. More than 2.2 million labeled buildings, which is 94.5 % of all buildings. Amsterdam has the second-highest number of labeled buildings with 1.2 million but compared to the total number of mapped buildings, just 57.9 % of all buildings have semantics.

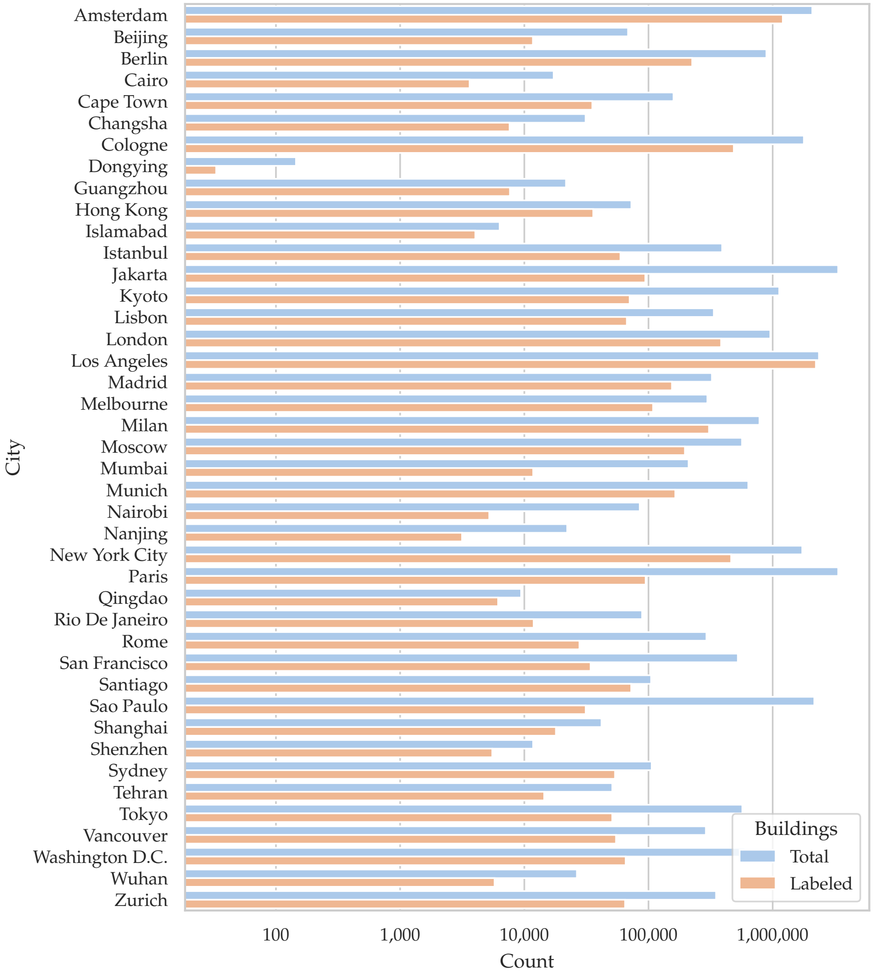


Figure 4.2: Total number of buildings and number of labeled buildings mapped in OSM for each LCZ42 city.

Further notable examples are Santiago, Qingdao, and Islamabad, with a labeling rate of more than 63 % (69.2 %, 65.3 %, and 63.7 %, respectively). They all have very few mapped buildings but a high labeling coverage. This finding indicates that mostly the POIs are annotated. Generally, China has low coverage in OSM as private mapping is illegal and strictly prosecuted, with researchers going to prison for collecting geodata [163].

4.2 GOOGLE AERIAL IMAGES AS REMOTE SENSING DATA

Finding remote sensing data suitable for building function classification is challenging. Given that 50 % of all labeled building footprints in the LCZ42 cities are smaller than 275.2 m^2 (Figure A.1), free satellite imagery can not be used here. For example, Sentinel-2 has 10 m GSD in the human visible bands [164]. Assuming that a building with the median footprint size of 275.2 m^2 has a quadratic shape, its side length would be 16.6 m. Hence, with 10 m GSD, the whole building is covered by four pixels. This resolution is not sufficient for the given task [165].

Therefore, this thesis uses aerial imagery from Google Earth at zoom level 18, which has been successfully applied for building instance detection [166, 167]. This zoom level yields a spatial resolution of approximately 0.48 m in our study areas.

Google Earth uses the WGS84 web Mercator standard and provides tiles with a resolution of 256 × 256 pixels on up to 22 zoom levels[1]. Therefore, the GSD gsd on a given zoom level z and a latitude lat is

$$gsd(z, lat) = \frac{2\pi r_E \cos(lat)}{2^{(z+8)}} \tag{4.1}$$

with r_E as the equatorial radius of 6,378,137 m [2]. For example, New York is at latitude 40, and Los Angeles is at latitude 33. Hence, the ground sample distance of our image patches is 0.46 m to 0.50 m in these cities.

Based on multiple map tiles stitched together, patches of 256 × 256 pixels centered on the building centroid were cropped. Hence, each aerial image covers an area of approximately 15,100 m^2 around the building center. For every labeled building in the LCZ42 cities, there is the respective aerial image from Google Earth.

4.3 GOOGLE STREET VIEW IMAGES AS A SOURCE OF GROUND-LEVEL IMAGES

Google Street View (GSV) is a commercial service from Google providing geolocated panoramic images around the world. The company collected this data first by itself but later opened it to volunteers and professional photographers. Most imagery is collected while systematically driving cars through the city. These cars are equipped with a custom high-resolution 3D camera system called R7. It has 15 5-megapixel CMOS image sensors and custom, low-flare, controlled-distortion lenses [168]. In October 2021, GSV had imagery from every cultural zone in the world (Figure 4.3), but at different temporal and spatial resolutions. However, GSV does not cover every single street: Especially extremely poor and rich neighborhoods tend to be inaccessible and are therefore not captured [169].

The Google Cloud API allows programmatic access to the Street View data as a pay-per-use service. There are two primary endpoints: a free one, which can be used to check if Street View data are available at a given location, and a paid one, yielding the actual image at a given position. Details of an image can be adjusted on a fine-grained level, starting with a position; the endpoint supports heading, pitch, and field-of-view [170].

[1] https://developers.google.com/maps/documentation/javascript/coordinates

[2] https://wiki.openstreetmap.org/wiki/Zoom_levels

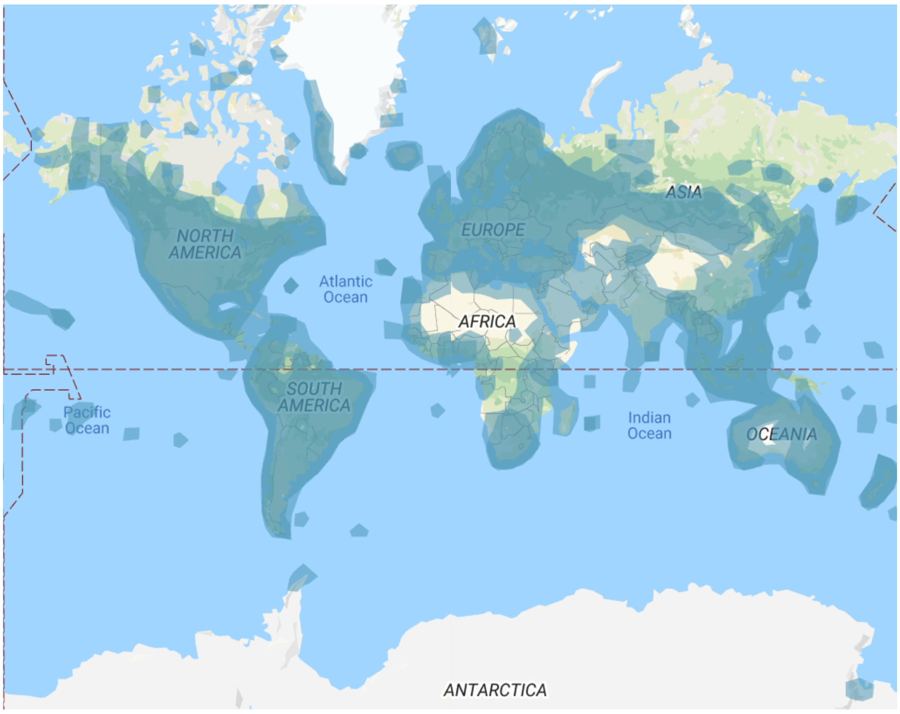


Figure 4.3: Coverage of Google Street View data; image © Google 2021 (from `https://www.google.com/streetview/explore/`)

This thesis uses a dataset of 43,392 globally sampled GSV images, with 14,512 *commercial*, 14,184 *other*, and 14,696 *residential* images. The sampling algorithm tries to obtain a GSV image of labeled OSM buildings for every administrative region in the world and every building class. However, if a given number of trials is reached, it will continue with the next class or region. For this thesis, the maximum number of trials was set to 25.

Figure 4.4 shows the spatial distribution of the dataset. It shows a high density in Europe as there are many small countries. Especially, the Benelux area and central European countries are highly represented, whereas Germany and France are comparably sparse. The American continent is evenly covered from North to South without a strong focus on the USA. However, some middle American countries are missing, e.g., Cuba, Nicaragua, El Salvador, or small island states in the Caribean sea.

In Africa, only eleven countries show up on the map: The highest density has Tunesia, Senegal, Kenya, Botswana, and South Africa. Hotspots in Asia are Israel, Oman, Kirgisitan, Sri Lanka, Bangladesh, Taiwan, Japan, Indonesia, and the Philippines. Australia and New Zealand are also part of the dataset but with fewer buildings.

Generally speaking, the distribution is based on an intersection of labeled building footprints in OSM and image data from GSV.

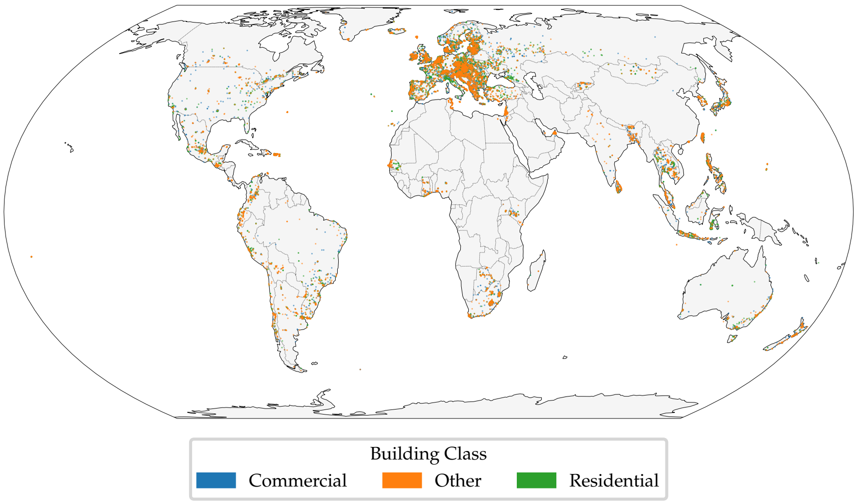


Figure 4.4: Global distribution of sampled dataset with labeled OSM buildings and a corresponding GSV image

4.4 FLICKR AS A SOURCE OF SOCIAL MEDIA IMAGES

Flickr is a platform for uploading and sharing photos founded in 2004 with a focus on high-quality photography [171]. Its users can comment and like shared photos of other users as well as follow updates from other users. The platform hosts billions of photos for more than 100 million registered users [172]. Following the stream of newly uploaded photos in 2018 showed that approximately 5 % of all uploaded images are geotagged; i. e., have latitude and longitude data from a GPS sensor. This work uses only geotagged Flickr images.

Besides an app for smartphones and a website, Flickr allows sharing and exploring content via an official web API. The API endpoints give access to all entities, e.g., users, groups, photos, cameras, or places. Using the *photos.search* endpoint, any machine can look up photos for a given bounding box defined by two points containing the latitude and longitude of the start and end. This method returns details about all geotagged photos taken in the specified bounding box. Based on the URL in these details, it is possible to download each photo with a separate request.

Figure 4.5 illustrates the variety of motifs in geotagged Flickr images from Venice, Italy. The feature vectors from a VGG16 network trained on ImageNet are embedded in a two-dimensional space using T-SNE [173]. The lower right part of the Figure shows images of water and sky, while the upper center contains photos of people. On the left side, there are pictures from night activities as well as macro photographs of flowers and food.

The dataset for this thesis was generated using random sampling of bounding boxes in the LCZ42 cities. By continuously querying

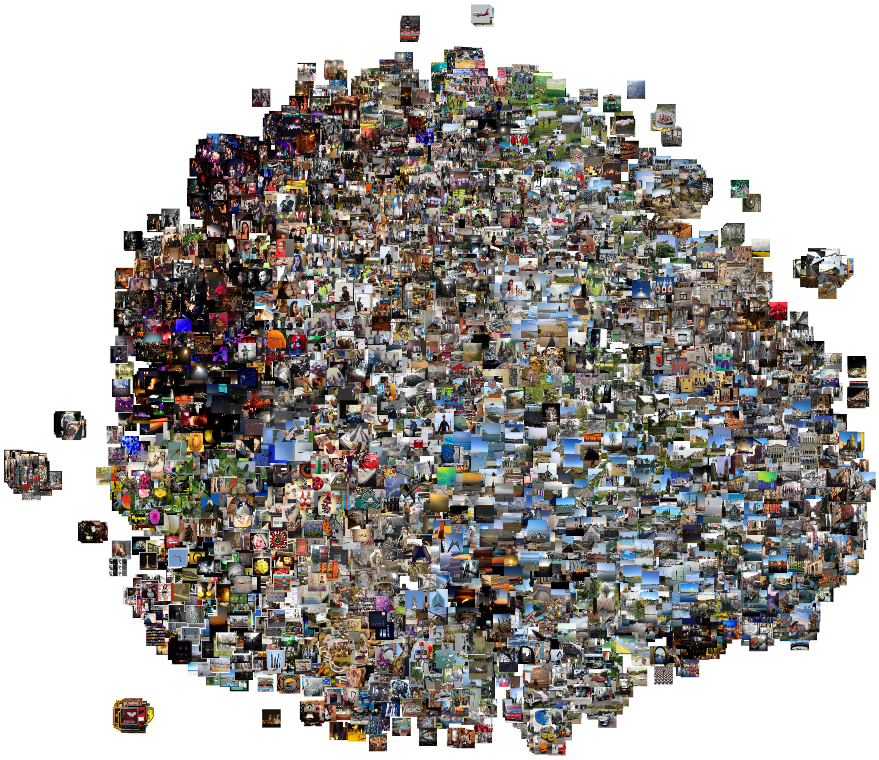


Figure 4.5: Two-dimensional embedding of feature vectors from geotagged images in Venice, Italy

the endpoint mentioned above from May 2018 to October 2021, the download script yielded 28,818,438 images. Figure 4.6 shows the number of Flickr images found in each LCZ42 city. The most images were taken in London, with almost 4 million images, followed by New York City and Los Angeles (2.4 million and 2.0 million). Eight cities have more than one million images with a geotag; seven are in the western hemisphere, and only Tokyo is from Asia.

A notable example of sparse coverage in social media is Dongying. There are 153 images, while all other cities have a few thousand Flickr images. A recent study found that Dongying is likely to be a ghost city, meaning that the number of housing possibilities outnumbers the number of inhabitants [174].

4.5 TWITTER AS SOURCE OF SOCIAL MEDIA TEXT

Twitter is a social media service focussing on short, public text messages. It allows direct and private messages, but every profile is public by default. Twitter was founded in 2006 [175] and has more than 330 million monthly active users worldwide [176]. A user's post is called a tweet and is up to 280 characters long.

Twitter provides access to its data via an official API to obtain a one percent sample of its continuous data stream. An endpoint called

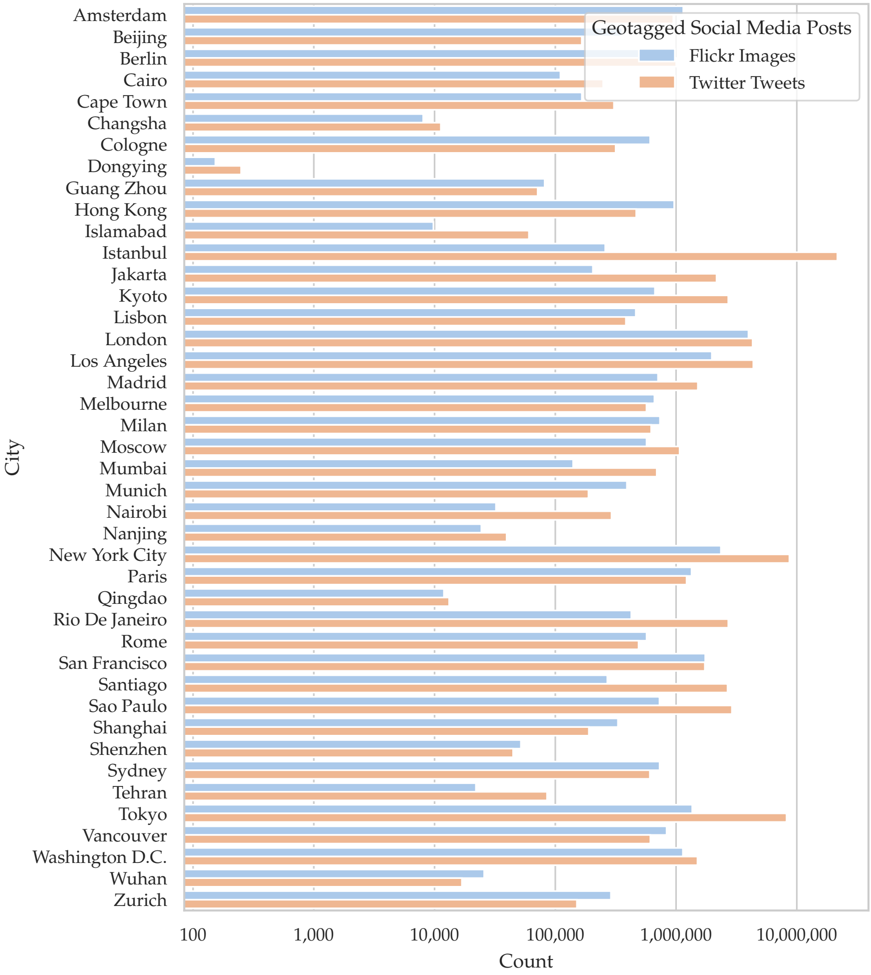


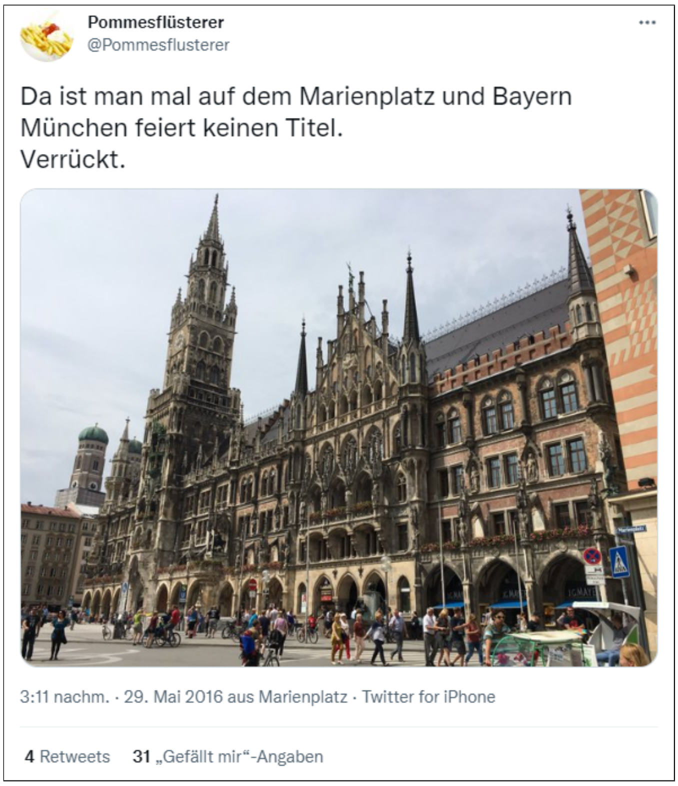
Figure 4.6: Number of geotagged Flickr images and Twitter posts for each LCZ42 city.

statuses/filter [177] enables authenticated users to capture parts of a live stream of tweets. Additional filters allow focusing on hashtags, user ids, or bounding boxes of latitude and longitude.

Tweets can be geotagged based on two properties: Geolocation, technically *coordinates* of latitude and longitude, or place, e.g., a POI, neighborhood, or city. Figure 4.7 shows an example of a geotagged tweet from Munich, Germany, posted in May 2016 that includes an optional image[3]. It includes a username, a text, a timestamp, and a location. Moreover, a tweet contains information about the client app, the number of retweets, and likes. In 2019, Twitter announced discontinuing precise locations in tweets because most users did not use them [178]. However, there are still geotagged tweets with a geolocation [179].

This thesis is based on a continuous one percent sample from the Twitter API covering November 2017 to October 2021. A bounding-

[3] https://twitter.com/Pommesflusterer/status/736907779034775552



Pommesflüsterer
@Pommesflusterer

Da ist man mal auf dem Marienplatz und Bayern München feiert keinen Titel.
Verrückt.

3:11 nachm. · 29. Mai 2016 aus Marienplatz · Twitter for iPhone

4 Retweets 31 „Gefällt mir“-Angaben

Figure 4.7: Example of a geotagged tweet from Munich, Germany, including a picture of the town hall

box filter covering all possible GPS coordinates ensured that only geotagged tweets were captured. This database consists of 589,764,252 tweets, with 76,061,274 coming from one of the LCZ42 cities. Figure 4.6 shows the distribution of tweets for these cities.

Most of the tweets, 21.7 million (28.5 %), are from Istanbul. Second, New York City has 8.6 million tweets, and the third is Tokyo, with 8.2 million tweets. The lowest tweet coverage has Qingdao (13,206 tweets), Changsha (11,276), and Dongying (250). Most cities have between 100,000 and 3 million tweets.

4.6 summary

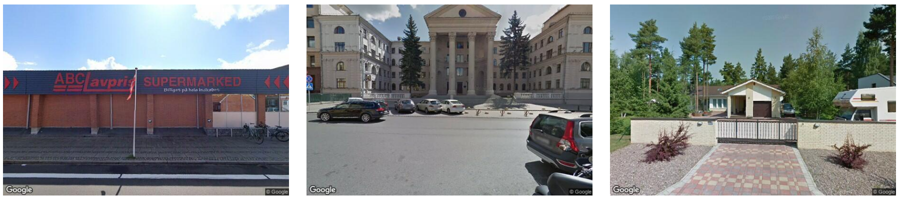
Another part of the So2Sat project [157] is a multimodal social media dataset for building function prediction. This dataset consists of 655,425 labeled OSM buildings from the LCZ42 cities. Each building comes with at least one tweet in its 50 m surrounding, and one very high-resolution aerial image focused on the building centroid.

This thesis uses this dataset in two ways for training models: First, for training classification algorithms on social media metadata in Chapter 7. Second, for training CNNs on the aerial images in Chapter 8. Parallelly, this thesis describes an independent dataset of labeled OSM buildings from social media images in Chapter 6, which are predicted using GSV models from Chapter 5. Thus, there are independent datasets for training models on different modalities and one common set of labeled OSM buildings that are used across all chapters for evaluation.

The next chapter introduces CNNs that predict building functions from GSV images (Chapter 5). These CNNs are re-used in Chapter 6 on social media images. The set of labeled OSM buildings from this chapter serves as a common test set for all subsequent methods in Chapter 7, Chapter 8, and Chapter 9. This approach has two advantages: First, all images from Flickr can be used for testing and the sparsity of Flickr images is mitigated. Second, the independent, large-scale training datasets allow for building robust models that are finally all evaluated on the same test set. Hence, the strength and weaknesses of different methods can be directly compared.

5 PREDICTING BUILDING FUNCTIONS USING STREET VIEW IMAGERY

As commercial services for street view imagery evolved, they soon became a subject of research on how they could be used for socioeconomic studies. In urban analysis, street view images are considered the gold standard for building-level imagery. For example, Google Street View (GSV) provides fine-grained settings for the location, the compass direction, the heading, the pitch, and the field-of-view of an image. This chapter introduces seven state-of-the-art architectures for building function classification using GSV data. They serve as baseline models and are re-used in different parts of this thesis.



(a) GSV image for *commercial* OSM building 12174077 in Ulfborg, Denmark

(b) GSV image for *other* OSM building 50655248 in Minsk, Belarus

(c) GSV image for *residential* OSM building 53770628 in Koidu, Estonia

Figure 5.1: Examples for GSV images showing buildings with clear function. Images © Google

Figure 5.1 illustrates how GSV images look like. For this thesis, all images are centered on the building they represent with a field-of-view of 90° and zero pitch. Figure 5.1a shows a supermarket in Ulfborg, a town in the west of Denmark, as an example for a *commercial* building. The theater in Figure 5.1b is the Belarusian State Academy of Music, located in Minsk, Belarus. Figure 5.1c depicts a single-detached house in Koidu, a village near Estonia's capital Tallinn. All images in Figure 5.1 are captured during clear, sunny weather conditions, but this is not generally the case.

5.1 FINE-TUNING METHODOLOGY

Google provides an API for obtaining street view images with fine-grained control over the image content. It allows specifying either the viewpoint or location of the image. Moreover, the API has parameters for the field-of-view, the compass direction, and the pitch. This level of detail ensures high-quality image content.

As a drawback, Google charges users for every API request with a small amount of money. While a single image comes at moderate costs, building a global dataset to train a deep neural network from scratch is comparably expensive. Furthermore, the Terms of Service for Google Maps are very restrictive on usage.

Therefore, this thesis uses the GSV dataset introduced in Section 4.3 for fine-tuning pre-trained ImageNet models. The fine-tuning approach is organized in two steps: first, a newly added classification layer is adopted to the backbone model. Second, the whole network is fine-tuned in an end-to-end fashion. An intuitive explanation is: The first step aligns a new prediction layer with the existing backbone, while the second step adopts the whole model to the task.

Fine-tuning describes a method that takes the weights of an existing deep model and adopts them to a new but related task. It relies on the observation that all deep vision models have generic feature extractors in the early layers and aggregate them in the subsequent layers. Only the last layers of a network contain the actual, task-specific weights.

The more general a vision task is, the more general feature extractors are expected in a trained vision model [180]. Hence, this thesis uses seven deep neural networks trained on ImageNet [181] to predict building functions based on street view imagery. The architectures are DenseNet121 [58], InceptionResNetv2 [59], Inceptionv3 [53], MobileNetV2 [61], ResNet50 [57], VGG16 [51], and Xception [62].

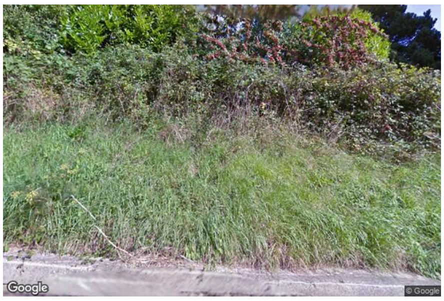
ImageNet models are trained to predict 1,000 image classes in their final output layer, unsuitable for the given task. Therefore, all task-specific dense layers are cut off so that only feature extraction layers remain. These layers are frozen in the first training step, i.e., their weights are not adjusted during training. A newly added, randomly initialized output layer predicts the final building function class using a softmax activation function. This layer is trained in the first step using a categorical cross-entropy loss and Adam [46] as optimizer. The learning rate in this step is defined as $lr_1 = 10^{-4}$.

The training runs for $n_1 = 16$ epochs with model checkpointing based on the validation loss. After training, the model with the lowest validation loss is used for the next step to prevent overfitting. In this case, all layers are trained with a learning rate $lr_2 = 10^{-5}$, while all other parameters stay the same. Again, the resulting final model is chosen based on the lowest validation loss during training.

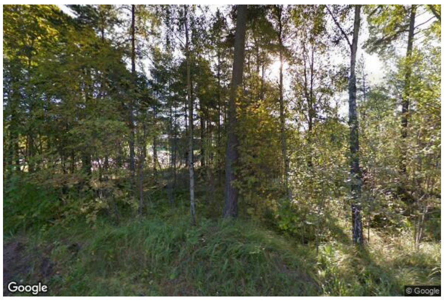
5.2 EVALUATION

5.2.1 *Creating Subsets for Training and Testing*

For evaluation, the dataset from Section 4.3 is divided into two independent sets for training and testing. Visual inspection of the dataset revealed that not all images depict a building. Figure 5.2 shows two images that are heading toward buildings, but they are not visible due to occlusions from vegetation. Therefore, an object detection algorithm was applied, and only images containing a *house* or *building* were kept. All objects were detected using a Faster R-CNN trained on OID v4 (see Section 2.2.5).



(a) GSV image for *residential* OSM building 5662533



(b) GSV image for *other* OSM building 4631033

Figure 5.2: Examples for GSV images showing no building, but task unrelated content. Images © Google

This step reduced the dataset by 14 % from 43,383 to 37,295 images. Table 5.1 shows the class distribution in the training and test set after filtering and applying an 80:20 split. These steps retained the class balance from the original dataset and left approximately 10,000 images per class for fine-tuning.

Dataset	Original	Filtered	
Class		Train	Test
Commercial	14,509	9,979	2,497
Other	14,181	9,438	2,288
Residential	14,693	10,419	2,674
Sum	43,383	29,836	7,459

Table 5.1: Training and test set of Google Street View (GSV) images

5.2.2 *Architecture Analysis*

After fine-tuning with the two-step approach, all architectures were evaluated on the test set. They all show a similar precision and recall performance between 0.513 and 0.568 (Table 5.2). The best architecture is InceptionResNetv2, with a precision of 0.563 and a recall of 0.568. Close to this is the related Inceptionv3 model gaining the second highest values of precision and recall (0.560 and 0.564, respectively). In contrast, the ResNet50 architecture shows the lowest performance of 0.513 precision and 0.519 recall. All other architectures are in between these results but always with a high balance between precision and recall. The difference between both metrics is 0.002 on average, with recall as the higher value.

From a class-level perspective, there are more differences. The *residential* class yields generally the highest F1 scores, 0.602 on average, with one exception: the VGG16 architecture has a higher F1 score for *commercial* than for *residential*. *Commercial* buildings are predicted with an average F1 score of 0.595. Finally, all architectures have the lowest F1 score for *other* buildings (0.428 on average). The precision and recall values show no distinctive pattern beyond these findings on the F1 score due to their balanced values.

Since InceptionResNetv2 has the most parameters of all architectures, it also has the largest capacity to adopt fuzzy patterns, which poses a possible explanation for its performance. The following subsection analyzes the InceptionResNetv2 model in more detail.

5.2.3 *Analysis of Best Performing Model*

The InceptionResNetv2 architecture yielded a weighted F1 score of 0.562 on the GSV image test dataset, which is the best value of all architectures. Figure 5.3 shows the confusion matrix of this model for a deeper analysis. The majority of *commercial* images, 60.8 %, is correctly predicted. The remaining wrong classified images are almost evenly distributed among the two other classes with 18.5 % on *other* and 20.7 % on *residential* images. *Other* images show the most confusion: 38.3 % of all *other* images are correctly classified, but almost the same number, 36.1 % is predicted to be residential. Less confusion is between *other* and *commercial* with 25.5 % wrongly classified images. Last but not least, the *residential* class yields the best results with 69.0 % of all *residentail* images being predicted correctly. It is mostly confused with *other* buildings in 18.0 % and less confused with *commercial* (13.1 %). While the *commercial* and *residential* classes are relatively well-defined, the *other* class combines very different classes like stations, hospitals, schools, churches, mosques, and temples. Hence, the inter-class variance of *other* buildings is substantially higher than for the other two classes. Moreover, the first three examples, stations, hospitals, and schools, exhibit often similar façades as *commercial* or *residentail* buildings.

These findings are also reflected by the geographical evaluation in Figure 5.4. The F1 score results for *commercial* and *residential* buildings is mostly above 0.50 across all regions. The InceptionResNetv2 architecture shows comparable results in industrialized regions and developing areas for both classes. It yields the best F1 scores for *commercial* in Polynesia, Australia and New Zealand, and Western Asia. However, the score for Polynesia needs to be taken with care as it is based on one building. For the two other regions, the F1 scores are cal-

	Metric	F1-score	Precision	Recall
Model	Class			
DenseNet121	Commercial	0.605	0.602	0.608
	Other	0.427	0.467	0.394
	Residential	0.612	0.578	0.650
	Weighted	0.553	0.552	0.557
InceptionResNetv2	Commercial	**0.614**	**0.619**	0.608
	Other	0.427	**0.482**	0.383
	Residential	**0.629**	0.578	0.690
	Weighted	**0.562**	**0.563**	**0.568**
Inceptionv3	Commercial	0.608	0.609	0.606
	Other	0.445	0.475	0.418
	Residential	0.616	0.587	0.649
	Weighted	0.561	0.560	0.564
MobileNetv2	Commercial	0.573	0.611	0.539
	Other	0.419	0.473	0.376
	Residential	0.610	0.542	**0.697**
	Weighted	0.539	0.544	0.546
ResNet50v2	Commercial	0.565	0.544	0.588
	Other	0.396	0.436	0.363
	Residential	0.569	0.551	0.588
	Weighted	0.515	0.513	0.519
VGG16	Commercial	0.607	0.591	**0.623**
	Other	**0.463**	0.431	**0.499**
	Residential	0.575	**0.641**	0.522
	Weighted	0.551	0.560	0.549
Xception	Commercial	0.591	0.589	0.592
	Other	0.419	0.464	0.382
	Residential	0.599	0.561	0.642
	Weighted	0.541	0.541	0.546

Table 5.2: Prediction results of different deep architectures trained with a global GSV dataset. *Weighted* denotes the weighted average of all classes.

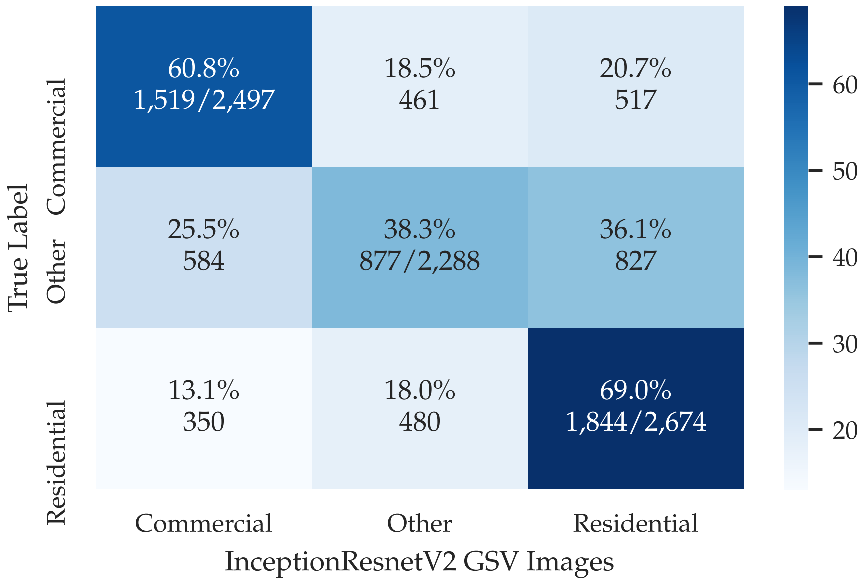


Figure 5.3: Confusion matrix of InceptionResNetv2 architecture fine-tuned on original GSV dataset

culated with 50 samples in Australia and New Zealand, reaching 0.750 and 119 images in Western Asia with a score of 0.681. Figure 5.5a depicts a correctly predicted *commercial* building in Western Asia. Apart from the Seven Seas region as an outlier with one sample, *commercial* buildings have the lowest scores in Northern Africa and Western Africa: 0.421 and 0.460. Figure 5.6a presents a cafe, which is predicted as *other*, which might result from the neatly ordered chairs. *Other* buildings are best predicted in Western Asia (0.589 F1 score on 129 images), Northern Africa (0.536 F1 score on 29 images), and Central Asia (0.492 F1 score on 28 images). An example of an *other* building is shown in Figure 5.5b. As there are no *other* buildings in Polynesia and the Seven Seas, the lowest F1 scores for *other* buildings come from Eastern Africa with 0.341, Southern Africa with 0.286, and Australia and New Zealand with 0.286. All three have a substantial number of buildings, with 25, 58, and 51 samples, respectively. Figure 5.6b illustrates a possible reason for misclassification: the building, a city hall, contains an estate agency. Regarding *residential* buildings, three unrelated regions yield the highest F1 scores: Micronesia with 0.784 on 47 buildings, the Caribbean with 0.716 on 104 buildings, and Southern Europe with 0.661 on 353 buildings. Figure 5.5c gives a correctly predicted example of a *residential* building in Micronesia. Polynesia is considered an outlier because its F1 score of 1.0 is based on two samples. The lowest F1 values for *residential* images are from Southern Asia (0.545 on 83 images), Western Africa (0.540 on 79 images), Northern Africa (0.438 on 40 images). The cafe displayed in Figure 5.6c is predicted as *commercial* but the label is *residential*.

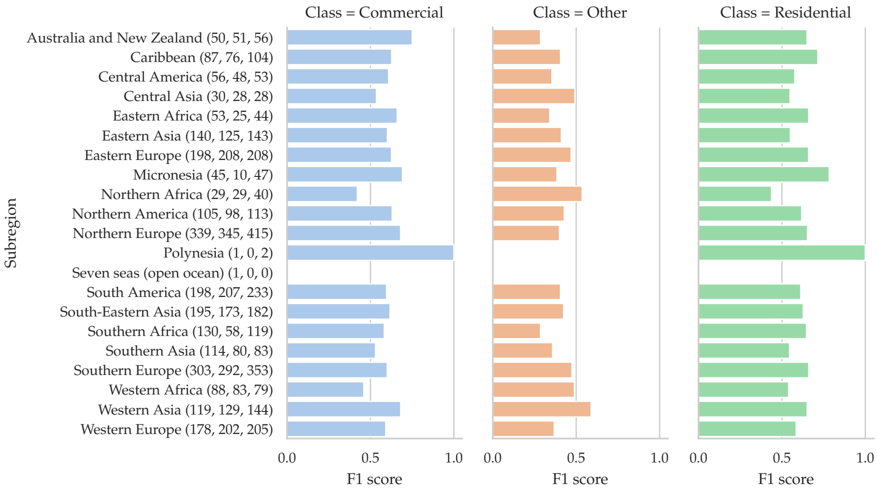
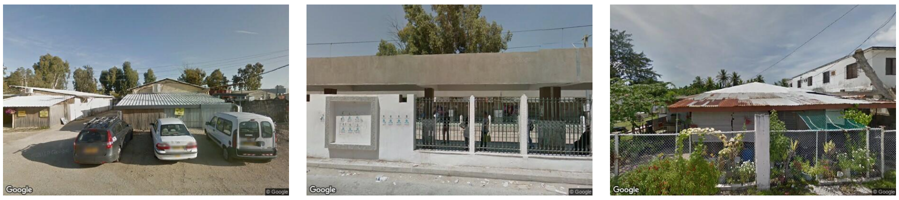


Figure 5.4: F1 scores of building predictions in LCZ42 cities on a class-wise level for InceptionResNetv2 model trained on the original global GSV dataset. The numbers in brackets behind the region names indicate the number of buildings in the same order as the plot columns.



(a) GSV image for *commercial* OSM building 829179350 in Western Asia

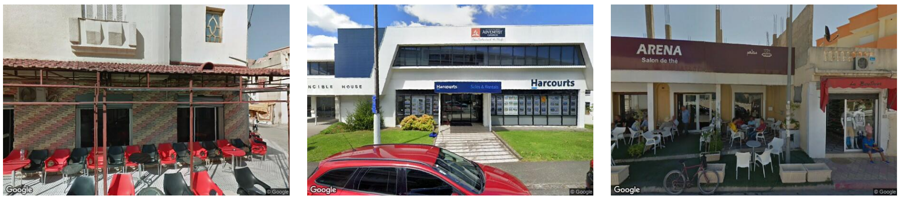
(b) GSV image for *other* OSM building 1361736724 in Northern Africa

(c) GSV image for *residential* OSM building 690199384 in Micronesia

Figure 5.5: Examples for GSV images correctly predicted by the InceptionResNetv2 model. Images © Google

5.3 SUMMARY

Predicting building functions from GSV images of building façades is still a challenging task if performed on a global scale. Different aspects are limiting the prediction performance: First, the quality of OSM labels is not consistent across the globe. Furthermore, the labeling scheme of three classes, *commercial*, *other*, and *residential* cannot model types of mixed-use buildings. The consistent results of all architectures are a strong hint that the label quality is a limiting factor in this case. A more fine-grained classification scheme could give more insights into what confuses the *other* class while going towards a multilabel classification model would take mixed uses into account. However, as the ground truth is still mostly for only one class, such an approach is challenging to evaluate. On the positive side, the InceptionResNetv2 model shows no bias towards the Global North but gains similar results across all



(a) GSV image for *commercial* OSM building 770071764 in Northern Africa, predicted as *other*

(b) GSV image for *other* OSM building 660972198 in Australia, predicted as *commercial*

(c) GSV image for *residential* OSM building 804768742 in Northern Africa, predicted as *commercial*

Figure 5.6: Examples for GSV images wrongly predicted by the InceptionResNetv2 model. Images © Google

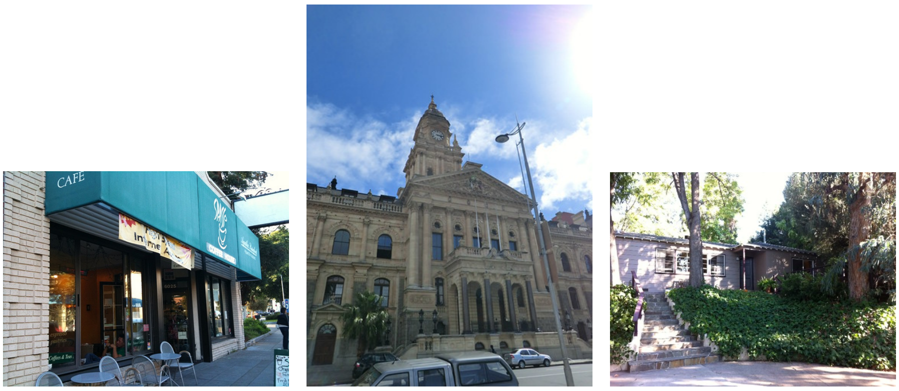
regions. Its results are balanced between America, Europe, and Asia. Nevertheless, the weaker results are primarily from Africa. With these strengths and weaknesses in mind, the next chapter takes the models to the next level with social media images.

6

PREDICTING BUILDING FUNCTIONS USING SOCIAL MEDIA IMAGES

This chapter focuses on social media images that are geotagged and publicly available. However, the methodological approach is not limited to this data or task. The application, in this case, is building function prediction, i.e., estimating the usage of a building using a façade image.

Social media images are taken for the given task but serve other purposes. However, they contain a small fraction of images beneficial for building function prediction. Figure 6.1 shows three examples of Flickr images that depict building façades giving a clear hint towards the building function. This chapter introduces a filtering method to extract such images from large-scale social media datasets and uses the models from the previous chapter to predict building functions.



(a) Flickr image showing *commercial* OSM building 107655590. Photo spasso @ 6021 college by Jed Schmidt is licensed under CC BY-NC-SA 2.0

(b) Flickr image for *other* OSM building 31826960. Photo ©CIMG2016 by bwilliamsdc

(c) Flickr image for *residential* OSM building 857226308. Photo ©Sneak preview of my new listing by Tracy King

Figure 6.1: Examples for Flickr images showing building façades with clear function

6.1 FILTERING RELEVANT SOCIAL MEDIA IMAGES

Social media images cover different content and motifs, including but not limited to photography, digital art, and cartoons. However, given a task like building function classification, most images do not help solve the task. For the task of building function classification, an image must have three features:

1. Shows a building façade
2. Has a valid geotag
3. Has a known compass direction

The image content allows for predicting the building function based on the façade. The geotag determines the position where the image was taken, and the compass direction is necessary to map the image content to a nearby building.

A filtering pipeline is needed to identify all images that fulfill these three criteria in a social media image dataset. Additionally, it must account for big data to work on datasets with more than 20 million images.

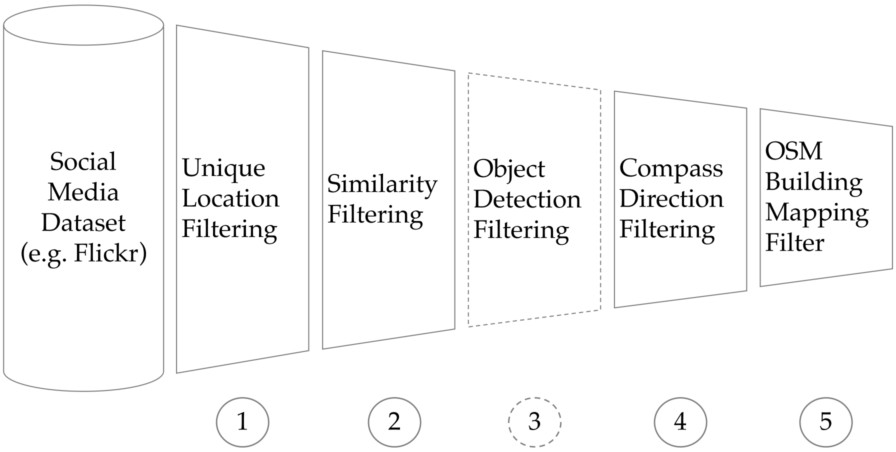

Social Media Dataset (e.g. Flickr)
Unique Location Filtering
Similarity Filtering
Object Detection Filtering
Compass Direction Filtering
OSM Building Mapping Filter
1
2
3
4
5

Figure 6.2: Filter pipeline for extracting Street View-like images from Flickr image database

Figure 6.2 shows the pipeline used in this thesis. It consists of five steps, starting with unique location filtering. This filter is a heuristic to discard images with an invalid geotag by checking if more than one image is from one location. The following two steps focus on the image content and ensure that the first two criteria are matched. Similarity filtering checks if a social media image has potentially helpful content by comparing it to a set of images that are known to be helpful. Object detection filtering is an optional step that applies a deep object detection algorithm to the images yielded from the previous step. If a social media image contains an object defined as valuable, it will be passed on to the subsequent step. At this point, there is evidence for each image showing a building, and its geotag is probably from a GPS sensor. The next filter step checks the third criterion: the compass direction. An image passes this filter if the tag *GPSImgDirection* is present in the Exchangeable Image File Format

(EXIF) metadata. Please note that this pipeline builds on the assumption that the EXIF data is not present and needs to be downloaded separately for each image. The final step, OSM building mapping filter, is most relevant for evaluation. It ensures that the first building within the line of sight is labeled, and predictions can be compared with this label. However, it would be sufficient to relate the image content to an unlabeled building in a pure inference case.

Having the pipeline in this order allows fast filtering and reduces the required communication with external servers.

The following subsections describe the filter steps in more detail.

6.1.1 *Unique Location Filtering*

This filter is a heuristic to identify images that were manually tagged. Geotags can be created in two different ways: either automatically by a GPS sensor of the camera or manually by the user. GPS sensors in mobile devices have a warm-up phase, in which the location is constantly updated based on visible satellites and nearby WiFi identifiers [182]. If users have to pick locations of images by hand, they tend to do it batch-wise, tagging multiple images at the same place.

Therefore, two images taken at the same position with the same device will have a slightly different geotag. If two images have precisely the same geotag, up to the 16th digit, likely, they were manually tagged while post-processing. In this case, a geotag is not considered to be valid.

More formally, an image i_x with location $l(i_x)$ passes this filter if

$$\forall i \in I, i \neq i_x \nexists l(i) = l(i_x) \tag{6.1}$$

If naïvely done, the geotag for each image needs to be compared with all geotags in the database, a so-called sequential scan. A geospatial index decreases the necessary checks by excluding geotags far away. An R-tree allows finding the images in a very close neighborhood, and a subsequent check on true equality is performed only on the geotags of these images.

6.1.2 *Similarity Filtering*

This first content-based step is a coarse filtering step aiming at finding images that are potentially helpful for building function classification. Previous studies showed the relevance of façade images to predict building functions [16, 20, 77, 183]. Therefore, this step is formulated as an image retrieval problem with a sample of GSV images as a seed dataset and a social media dataset.

This sample is compiled from the global GSV dataset as described in Section 4.3. To ensure that all seed images show a meaningful object, they are filtered to contain either a building or a house using object detection with Faster RCNN (see Subsection 2.2.5).

Features from deep neural networks are well-suited for finding structurally similar images. As they aggregate information with every layer, the final layers of a network are an abstract representation of the whole image. For example, the deep features of VGG16 have been successfully applied in different domains for image retrieval [184–187].

For this thesis, features are taken from the last hidden layer of a VGG16 network trained on ImageNet. This process yields feature vectors $v \in \mathbb{R}^{4096}$. To assess similarity between pairs of images i_1, i_2, the cosine similarity s_{cos} is calculated based on the feature vectors v_1, v_2:

$$s_{cos}(v_1, v_2) = \frac{v_1 v_2^T}{\|v_1\| \, \|v_2\|} \tag{6.2}$$

For efficient calculation, the features for all images of the seed dataset are calculated beforehand. Then, the features for all social media images are computed batch-wise. Next, the pair-wise cosine similarity is calculated between the batch and the seed dataset. The similarity score sim_{score} of an image with feature vector v_s is defined as the maximum similarity with all seed images:

$$sim_{score}(v_s) = \max\left(\{s_{cos}(v_1, v_s), ..., s_{cos}(v_n, v_s)\}\right) \tag{6.3}$$

A threshold t_{sim} is set as a minimum similarity value and all social media images with $sim_{score} < t_{sim}$ are discarded.

6.1.3 *Object Detection Filtering*

The previous step is a fast check for structural similarity to a given seed dataset but does not ensure that the social media images contain a building façade. Therefore, this step uses an object detection algorithm to find all objects in the images that passed the previous filter.

There are two main criteria for selecting an object detection algorithm. First, it needs to detect the object types of interest, and second, its trade-off between speed and accuracy must be appropriate for the task. There are no images with bounding boxes available for training, and only a pre-trained detection algorithm could be applied. The Open Image Dataset [70] is the only dataset with buildings and houses as part of the training data to the best knowledge of the author. A suitable architecture trained on this dataset is Faster R-CNN [63] with an mAP of 37 and a runtime of 727 ms [68].

Applying the object detection algorithm yields a list of objects for each image. If this list contains either a *house* or a *building* it is a candidate for passing this filter. Each detected object comes with a size relative to the image and a confidence score. Based on these variables, there are two thresholds for adjusting if a candidate image passes the filter: t_{size} and t_{score}. Only if there is a building or a house that is larger than t_{size} and has confidence higher than t_{score} the image is passed to the next step. The filter is a logical OR, i.e., if there is at least one object matching the criteria, the image passes the filter.

6.1.4 *Compass Direction Filtering*

The geotag of an image defines where its photographer was standing while taking it. This information is essential but does not say anything about the camera's viewing direction. Without a compass orientation of the camera, it is almost impossible to align the image content with the spatial surrounding. Therefore, this step checks if this value is available for an image.

This step is based on metadata of images, so-called Exchangeable Image File Format (EXIF) data. EXIF is a standard established by the Camera and Imaging Products Association (CIPA) and the Japan Electronics and Information Technology Industries Association (JEITA) [188]. It defines fields for saving details about images, including the date and time of capturing, camera model, and camera settings. Moreover, it specifies how data from GPS sensors can be incorporated. Possible options for this data are GPS positions with longitude and latitude, compass orientation, or compass direction.

An intermediate step is downloading the EXIF data for all images that pass the previous filter. As this is not part of the filtering pipeline, it is omitted in Figure 6.2. However, every request that can be avoided with the unique location filtering reduces network traffic and hence, the overall runtime of the filtering pipeline.

Assuming that EXIF data is present for all images that passed the filters so far, this step checks if the tag *GPSImgDirection* is present and rejects all images that do not have this tag.

6.1.5 *OSM Building Mapping Filter*

This final step establishes a connection between buildings shown in an image and their representations in OSM. Having a position and a compass orientation allows one to draw a line of sight. The first building intersecting with this line is defined as the building shown in the image with p_{dist} as the distance between the position and the building in meters. Figure 6.3 illustrates the approach.

Based on this parameter, a fourth threshold is introduced: t_{dist} sets the maximum distance for a building. For evaluation, all images looking at unlabeled buildings are skipped.

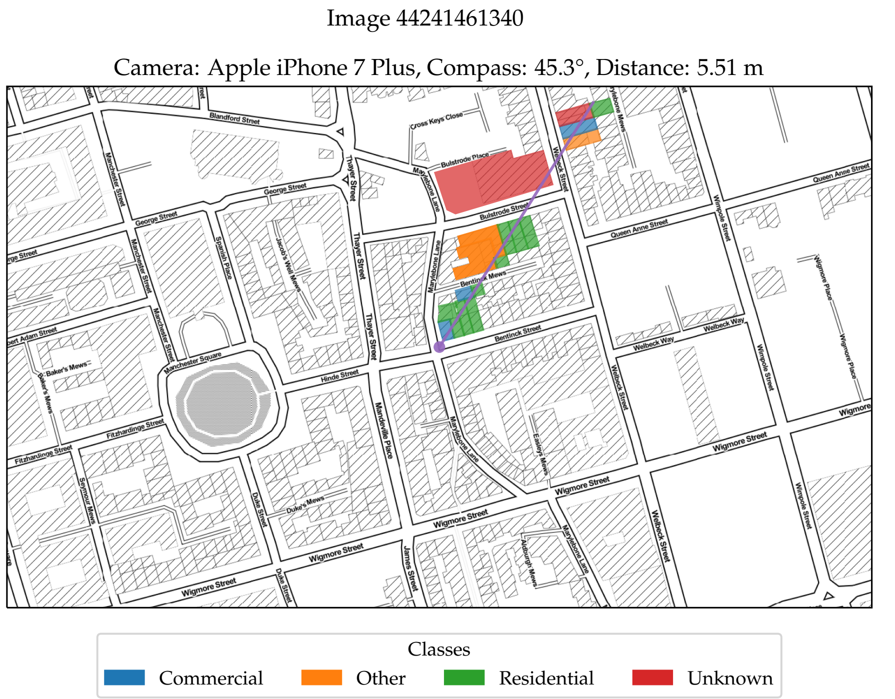


Figure 6.3: Example for mapping a Flickr image to an OSM building. The image location and line-of-sight are in purple. Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

6.1.6 *Filtering Pipeline Summary*

Having the pipeline in this order enables a content-first strategy while keeping the computational effort low. Additionally, the number of hyperparameters is small with four thresholds:

1. minimum seed similarity t_{sim}
2. minimum object size t_{size}
3. minimum object score t_{score}
4. maximum building distance t_{dist}

6.1.7 *Predicting Building Functions*

Instead of training new models, this thesis builds upon the models from the previous Chapter 5. The similarity filtering implicates that the distributions of the GSV dataset and the social media images are close to each other. Additionally, the social media image labels will probably have a lower quality as there are more error sources in the process. Last but not least, reusing models saves energy and is environmentally more friendly.

6.1.8 *Human Label Verification*

Using labels from OSM has the advantage of potentially global coverage, but the activity of contributors limits them. Moreover, the simplicity of the classification scheme cannot cover mixed usages.

The human visual processing system is still among the top classification mechanisms despite the recent success of CNNs [189]. Therefore, a group of humans was asked to verify the labels of the social media images to assess the label quality. They were shown an image and a corresponding label obtained from the filtering pipeline. Based on this information, they had to vote if a label was correct or not. For unclear cases, a third option *unsure* was offered. Figure 6.4 shows the user interface of the tool with a sample image. Once n users voted on an image, it was not shown to any other user to have the same number of votes for every image.

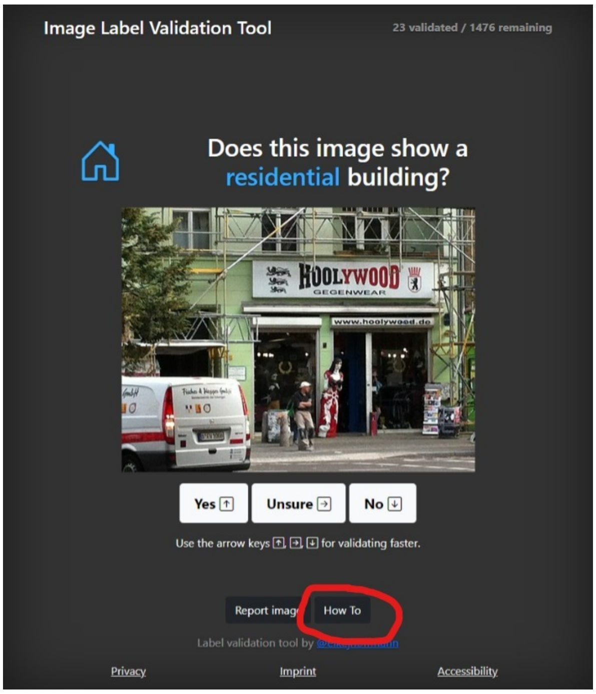


Image Label Validation Tool

23 validated / 1476 remaining

Does this image show a residential building?

Use the arrow keys ↑, →, ↓ for validating faster.

Label validation tool by [illegible]

Figure 6.4: Web-based tool for human validation of image labels

6.2 EVALUATION

This section is structured based on the pipeline order. Therefore, it investigates the effects of each filtering step in the first place. Afterward, an end-to-end evaluation analyzes the performance of different state-of-the-art architectures for this task.

Subsequently, an analysis of human-validated labels shows the accuracy of OSM labels from the pipeline and how the results can be assessed when applied in inference mode. Finally, a discussion of the results summarizes this chapter.

The filtering starts with the dataset from Section 4.4 containing 28,818,438 images. The unique location filter first processes this dataset.

6.2.1 *Unique Location Filtering*

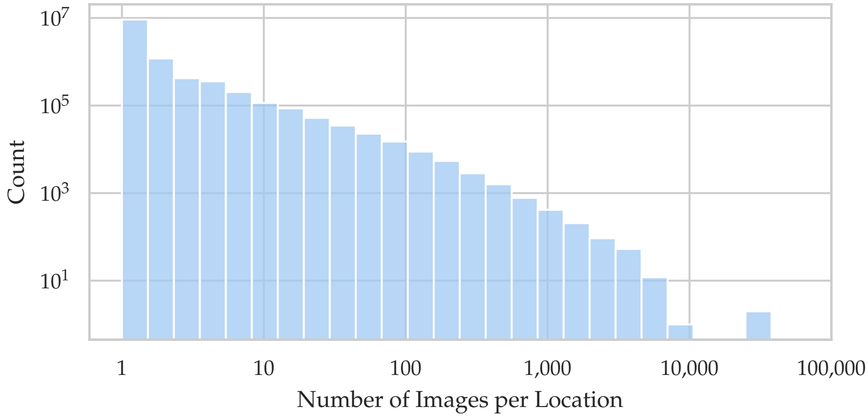


Figure 6.5: Number of Flickr images per unique location on **log-log** scale

9,188,589 out of 28,818,438 images have no other image at precisely the same location (31.9 %). Hence, more than two-thirds of all images in this dataset do not pass this filter. Figure 6.5 shows the distribution of the number of images per location on a log-log-scale. Twenty-five percent of all images share their position with two to ten other images. Further, 23 % have eleven to 100 other images at the same position. Generally, the histogram can be described with a power-law distribution having a long tail. In total, there are 487 locations with more than 1,000 images. However, only 9,188,589 are unique and used for the next step.

6.2.2 *Similarity Filtering*

Figure 6.6 shows the distribution of similarity values from the unique location-filtered images. These images have a mean similarity of 0.539 to the GSV seed dataset (variance 0.016). The minimum similarity is 0.159, and the maximum similarity is 0.895, with a skewness of -0.307. Hence, the distribution is skewed towards higher values than the mean.

This indicates that the previous step is a good heuristic as the distribution of similarity scores from the overall dataset follows a normal distribution almost perfectly. Figure A.2 depicts little skewness, while the mean similarity is 0.506. Without taking the content into

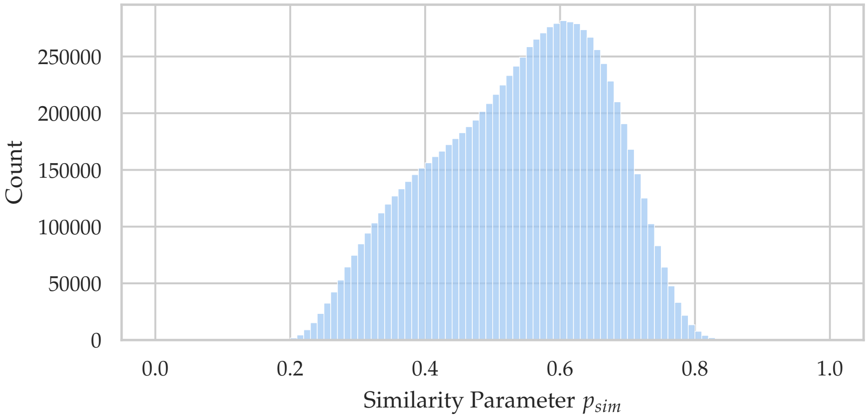


Figure 6.6: Distribution of similarity parameter p_{sim} after unique location filtering

account, unique location filtering helps to focus on the more street-view-like images.

For further analysis, a minimum threshold $t_{sim} = 0.70$ is set to keep the number of images for object detection and the number of requests to download EXIF metadata reasonable. This results in 821,110 images that are used for the next steps. If object detection filtering is applied, this is the next step. Otherwise, these images are passed to the compass direction filtering directly.

6.2.3 *Object Detection Filtering*

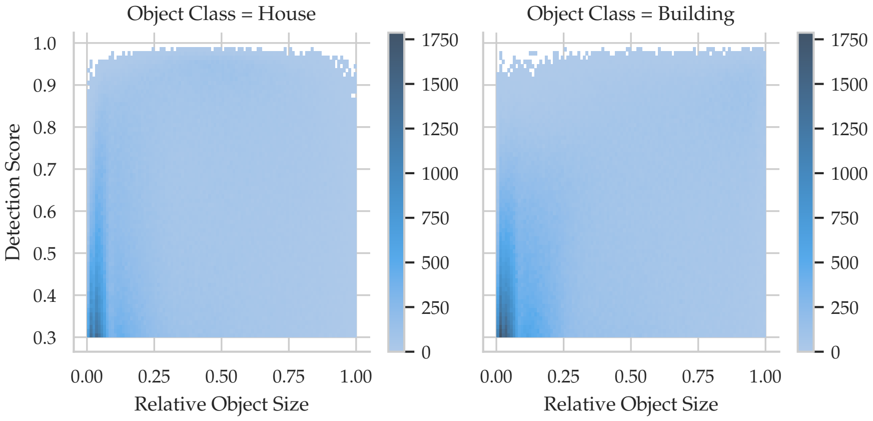


Figure 6.7: Number of detected houses and buildings as a function of detection scores and relative object sizes in social media image dataset

This optional step removes all images that do not show a house or a building. Figure 6.7 depicts the distribution of the confidence scores and the relative object sizes in the dataset. Almost all combinations of size and scores are present, but most houses and buildings cover less than 20 % of an image and are detected with scores < 0.8. Since the algorithm does not report objects with a score lower than 0.3, this value represents the start of the detection score axis. For further analysis, thresholds are set to $t_{score} = 0.0$ and $t_{size} = 0.0$. Hence, an image passes this filter if a house or a building was detected independent of any score or size. The effects of higher thresholds are discussed below. Out of 821,110 images, 76 % (624,099) fulfill this requirement.

6.2.4 *Compass Direction Filtering*

This binary filter checks if an image has the tag *GPSImgDirection* in its EXIF data. It is assumed that this data is not present and needs to be downloaded individually for each image.

If object detection is omitted, this step yields 168,456 images out of 821,110 (20.5 %). Otherwise, 88,809 images pass this filter after object detection.

6.2.5 *OSM Building Mapping Filtering*

Out of 168,456 images from the previous step, 120,547 have an OSM building within their line-of-sight, and 43,526 of these buildings are labeled. If object detection filtering is applied, the numbers are 98,604 (all buildings) and 35,568 (labeled buildings). There is a notable difference in the proportions: without object detection, 71.5 % of images can be mapped to a building. By using object detection, this proportion increases to 78.6 %. However, in absolute numbers omitting object detection results in 22.3 % more images mapped to labeled buildings.

The images are almost equally distributed among the three classes: With object detection, there are 10,950 *commercial*, 11,958 *other*, and 12,660 *residential* images. If object detection is not applied, there are 13,525 *commercial*, 14,474 *other*, and 15,527 *residential* images.

The following analysis focuses on the numbers gained without object detection. Figure 6.8 shows how many images are mapped to one building. In most cases, there is one image per building for each class. However, there are two outliers of buildings having more than 50 images. Two *other* buildings have 81 (ID 806812206) and 60 (ID 6516601) images. The first one, 806812206, is the Kinkaku-ji temple in Kyoto, a World Heritage Site. The second building, 6516601, is also a World Heritage Site located in Lisbon: Mosterio dos Jerónimos, a former monastery. Both are tourist hot spots, and hence many pictures are taken there.

The *residential* building with the most images assigned has 38 images. This building (ID 476482044) is the White House in Washington, DC. For the class *commercial* the building with ID 579548282 has 33 images. It is a supermarket in the south of London, a region with very few

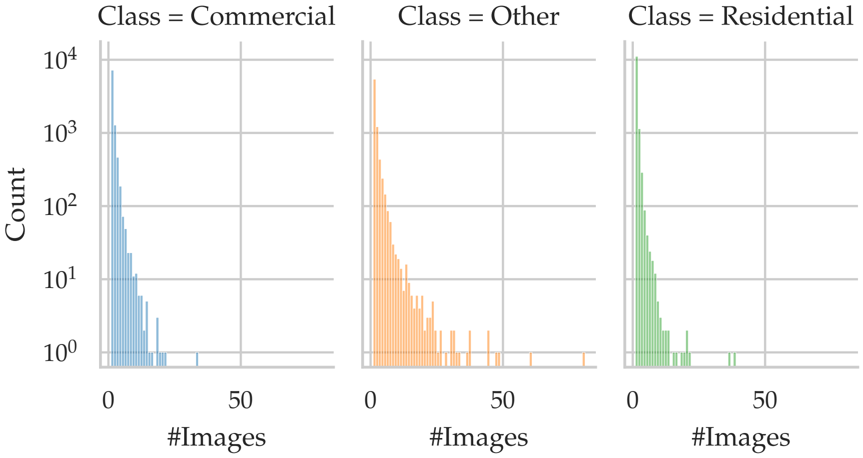


Figure 6.8: Distribution of Flickr images per OSM building on a class-wise level

building footprints in OSM. This building is the only one in the broader area, and it gets all images assigned.

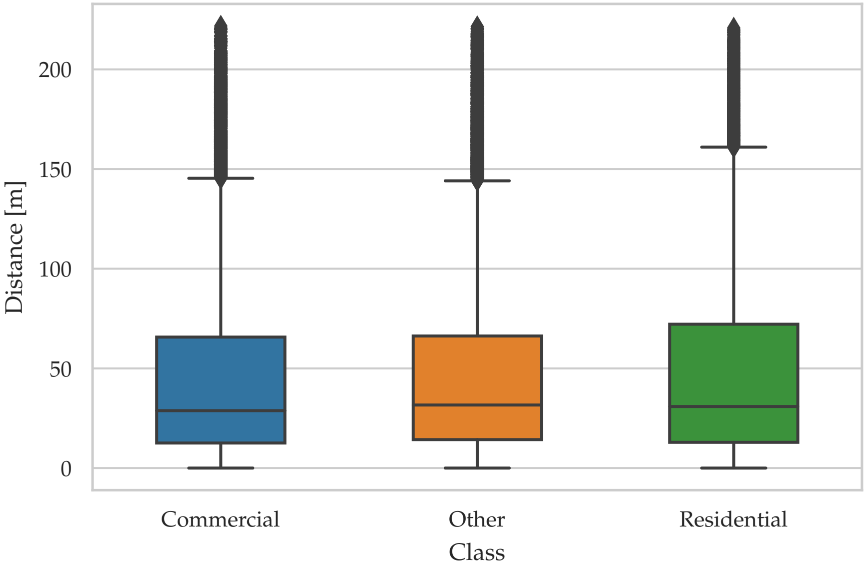


Figure 6.9: Distance in meters of Flickr images to OSM building on a class-wise level

As an example, Figure 6.9 depicts the distance of images to buildings. Between the classes, there are a few differences. More than 75 % of all images are assigned to buildings less than 75 m away. The median distance is similar for all classes with 28.8 m for *commercial*, 31.6 m for *other*, and 30.9 m for *residential* buildings. Nevertheless, there are outliers of more than 160 m distance to a building. Here, no threshold is defined as the full parameter range is a subject of further analysis in Subsection 6.2.8.

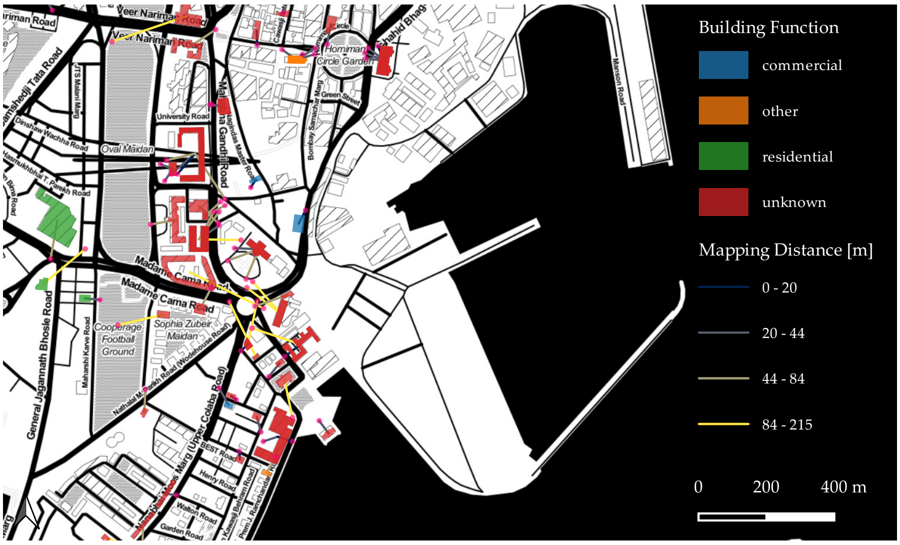


Figure 6.10: Map of Flickr images and assigned OSM buildings in Mumbai, India. Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Figure 6.10 shows social media images identified in Mumbai, India, and their corresponding buildings according to the algorithm. In this city, there are 192 images in total, which are mapped to 154 buildings, with 73.3 % of them unlabeled (filled with red in Figure 6.10). Although social media images and building data are sparse in this area, there is still a high potential to fill blind spots in the Global South.

6.2.6 *Pipeline Summary and Runtime Analysis*

Table 6.1 summarizes the number of images remaining after each pipeline step. Furthermore, it shows the time needed to process one image sample. Although the absolute time values change with different machines, they allow a relative comparison between the steps. In this case, all calculations were performed on an Nvidia DGX-1 server with 80 cores à 2.2 GHz, 512 GB memory, and one Nvidia Tesla V100 GPU with 32 GB

The fastest step is unique location filtering, which takes 0.0002 s. It reduces the initial dataset to approximately one-third with an overall runtime of 96 minutes. Similarity filtering requires computing dense feature vectors, which takes 0.0236 seconds. This is 100 times slower than the previous step, taking 60.2 h to process the remaining 9,188,590 images. Object detection (OD) takes 144.1 h, with each image being processed in 0.6319 s. Image direction filtering is the slowest of all steps taking 1.3 s per image sample. The slowest part of this step is downloading the EXIF data and waiting to make the subsequent request. If one continues with the results from the OD step, it takes 231.1 h to check all 624,099 images. If OD is omitted, the process

Filtering Step	#Images	% of Dataset	Time [s]
Flickr LCZ42 Dataset	28,818,438	100.00 %	
Unique location filtering	9,188,590	31.88 %	0.0002
Similarity filtering	821,110	2.85 %	0.0236
Object detection filtering	624,099	2.17 %	0.6319
Image direction filtering	125,375	0.44 %	1.3333
w/o OD	168,456	0.58 %	
OSM building in line-of-sight	98,604	0.34 %	0.0008
w/o OD	120,547	0.42 %	
Labeled OSM building in line-of-sight	35,568	0.12 %	
w/o OD	43,526	0.15 %	

Table 6.1: Number of Images remaining after each filtering step when using $t_{sim} = 0.70$ and $t_{dist} = 250$. Time [s] describes the execution time for one image. *w/o OD* stands for *without object detection* and describes the pipeline results if object detection filtering is omitted. Execution time per image sample in seconds averaged over 10,000 samples

requires 304.1 h. Hence, there is no benefit in using OD from a time perspective. It is 23.3 % slower than checking the results of similarity filtering directly. The last step, checking if any OSM building is within the line-of-sight, is the second-fastest (0.0008 s) because this is also a spatial index-based database query. The smaller dataset (including OD) takes 100 s, and without it, it takes 135 s. However, the number of resulting images is different: OD gives 35,568 images, whereas omitting it yields 43,526 images, an increase of 22.3 %. Furthermore, the overall runtime is lower if OD is omitted.

Moreover, this analysis shows the potential of the approach: Without OD, there are 120,547 images, which are mapped to 86,658 OSM buildings. 56,857 out of these buildings, a majority of 65.6 %, are unlabeled and could be enriched with a semantic tag.

6.2.7 *End-to-end Performance Analysis*

The resulting images are fed into seven state-of-the-art architectures trained on ImageNet and fine-tuned on GSV data for this analysis (Subsection 5.2.2). They are evaluated against the label of the building they are mapped to, i.e., the performance of a GSV model to predict a building function based on a façade image from Flickr. Table 6.2 shows the F1-score, precision, and recall for all architectures at a class-wise and model level weighted by the number of samples.

As the class distribution is relatively even, the weighted metrics are always close to the mean of all class-wise metrics. Additionally, the table enables comparisons between images filtered with and without OD because both results are in two primary columns.

		Without OD			With OD		
	Metric	F1	Pre	Rec	F1	Pre	Rec
Model	Class						
DenseNet121	Com	0.467	0.433	0.507	0.462	0.440	0.486
	Oth	0.490	0.417	0.593	0.506	0.429	0.617
	Res	0.334	0.531	0.244	0.359	0.542	0.268
	Wgt	0.427	0.463	0.442	0.440	0.473	0.453
InceptionResNetv2	Com	**0.516**	0.430	**0.643**	**0.516**	0.435	**0.633**
	Oth	0.490	**0.510**	0.472	**0.511**	**0.527**	0.496
	Res	0.428	**0.550**	0.350	0.453	**0.568**	0.377
	Wgt	**0.476**	**0.499**	**0.482**	**0.492**	**0.513**	**0.496**
Inception v3	Com	0.485	**0.438**	0.542	0.480	**0.447**	0.517
	Oth	0.472	0.454	0.492	0.493	0.466	0.523
	Res	0.421	0.506	0.361	0.443	0.517	0.388
	Wgt	0.458	0.467	0.461	0.471	0.479	0.473
MobileNetv2	Com	0.469	0.432	0.513	0.463	0.438	0.491
	Oth	**0.491**	0.408	**0.615**	0.505	0.418	**0.635**
	Res	0.282	0.529	0.192	0.312	0.540	0.220
	Wgt	0.410	0.459	0.433	0.423	0.468	0.443
ResNet50v2	Com	0.436	0.423	0.449	0.426	0.430	0.423
	Oth	0.413	0.435	0.393	0.433	0.450	0.417
	Res	**0.450**	0.442	**0.459**	**0.467**	0.449	**0.486**
	Wgt	0.433	0.434	0.434	0.443	0.443	0.443
VGG16	Com	0.472	0.431	0.523	0.465	0.436	0.499
	Oth	0.480	0.400	0.601	0.496	0.410	0.627
	Res	0.276	0.537	0.186	0.300	0.553	0.206
	Wgt	0.405	0.458	0.429	0.417	0.469	0.438
Xception	Com	0.496	0.419	0.609	0.494	0.426	0.588
	Oth	0.465	0.456	0.474	0.487	0.468	0.508
	Res	0.365	0.504	0.287	0.387	0.520	0.308
	Wgt	0.439	0.462	0.449	0.454	0.474	0.461

Table 6.2: Prediction results of different deep architectures trained with GSV data on filtered Flickr images. Filtering is done with and without object detection (OD). Metrics are abbrivated as **F1**-score, **Pre**cision, and **Rec**all. Classes are abbrivated as **Com**mercial, **Oth**er, and **Res**idential. ***Wgt*** denotes the weighted average of all classes.

Generally, the results of all architectures are on a similar level, with weighted F1 scores between 0.405 and 0.492. With OD applied, the F1 scores are on average 3 % better than without OD. Moreover, most models have slightly better precision than recall (0.016 on average). All further analysis will focus on the pipeline without OD considering the higher runtime, fewer images, and a slight performance gain.

From a model perspective, InceptionResNetv2 outperforms all other architectures concerning all metrics at a weighted level. This finding is following the results from Subsection 5.2.2 thus, the InceptionResNetv2 shows the best performance on GSV images and filtered social media images. These results are another indicator for the hypothesis that the large capacity of the network is best suited for fuzzy patterns. However, at a class-wise level, some other architectures gain better results than InceptionResNetv2. For example, MobileNetV2 has 0.143 better recall for *other* buildings, but this comes with a very low precision of 0.508 (InceptionResNetv2 0.510). Overall, InceptionResNetv2 has an appropriate balance between precision and recall while obtaining the best results. Figure 6.11 shows an example for which this model is predicting the correct class while most other models fail.

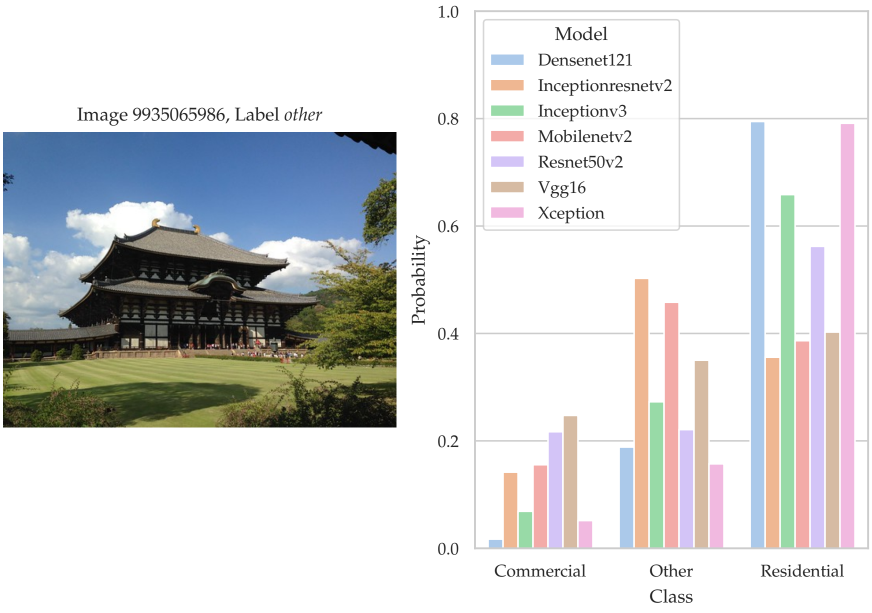
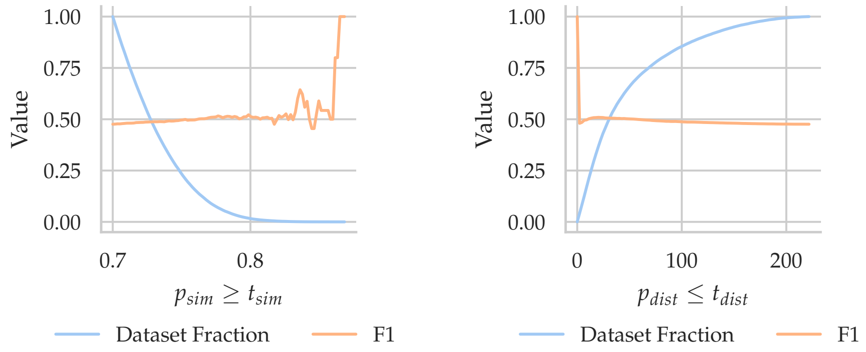


Figure 6.11: Example of a Flickr image that is correctly predicted by InceptionResNetv2 and MobileNetV2. Photo Daibutsuden - Todaiji Temple by John Dunsmore is licensed under CC BY-ND 2.0

The InceptionResNetv2 model has a high precision of 0.550 on *residential* images but a lower recall of 0.361. This finding is in contrast to the *commercial* class, for which it has a high recall of 0.643 and a comparably low precision of 0.43.

6.2.8 *Influence of Pipeline Parameters on Performance*

This evaluation looks at how the four parameters of the pipeline affect the classification performance. All metrics are computed using the InceptionResNetv2 architecture. Figure 6.12 shows the weighted F1 score and the size of the resulting dataset when the thresholds for minimum similarity t_{sim} and maximum distance t_{dist} are changed.



(a) F1-score as function of similarity threshold t_{sim}

(b) F1-score as function of distance threshold t_{dist}

Figure 6.12: Influence of pipeline parameters on classification performance of InceptionResNetv2 model without applying object detection

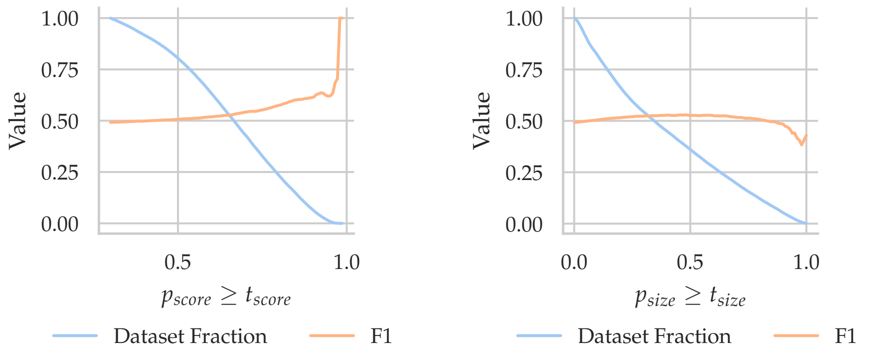
Increasing t_{sim} from 0.7 to 0.8 improves the F1 score from 0.476 to 0.512, an increase of 7.6 % (Figure 6.12a). At the same time, the dataset size decreases to 681 images, which is 1.6 % of the original dataset with 43,525 images. Using $t_{sim} > 0.85$ yields seven images resulting in an unstable F1 score of 0.543.

A decrease of t_{dist} from 250 m to 100 m increases the F1 score from 0.476 to 0.487 while maintaining a dataset size of 85.6 % (37,292 images, Figure 6.12b). The highest F1 score of 0.509 is reached for $t_{dist} = 20$ with 36.6 % of the dataset. Going below these values decreases the F1 score again before it becomes unstable when approaching 0.0.

Figure 6.13 depicts the influence of parameters from object detection on the classification performance. This analysis is also based on the InceptionResNetv2 architecture predicting the 35,567 images from the filtering pipeline with OD.

An increase of $t_{score} = 0.50$, the minimum object detection score, leads to a F1 score of 0.507, which is 0.015 more than the initial 0.492 F1 score (Figure 6.13a). In this case 80.2 % of the dataset is still available. Above $t_{score} > 0.5$ the F1 score increases up to 0.636 at $t_{score} = 0.92$. However, at this threshold, the dataset is reduced to 3.5 % (1,233 images). Further increases result in an unstable F1 score due to one remaining image at $t_{score} > 0.973$.

Changing the minimum building t_{size} yields a F1 score of 0.522 at $t_{size} = 22.4$ (Figure 6.13b). Using this threshold results in 15,215



(a) F1-score as function of building score threshold t_{score}

(b) F1-score as function of building size threshold t_{size}

Figure 6.13: Influence of additional pipeline parameters on classification performance of InceptionResNetv2 model with applying object detection

images, which is 42.8 % of the whole dataset. Lower and higher values of this threshold result in lower F1 scores.

All parameters significantly influence the dataset size and have a low effect on the prediction results. Setting more strict values reduces the dataset size to small fractions of the original one. However, there is not much change in the prediction performance. This insight indicates that the classification results are influenced more by the label quality than the image quality. The following section investigates this aspect in more detail.

6.2.9 *Results of Human Label Verification*

A subset of 1,500 images was created to assess the task's difficulty, with 500 images for each class. Thirty-four humans checked the image labels and voted if the label was correct, wrong, or they could not tell. If an image received three votes from different reviewers, it was defined as done and taken out of the set.

756 image labels out of 1,500 got full agreement from all three reviewers. Figure 6.14 shows the confusion matrix of these human labels and the labels from OSM. The overall accuracy of OSM is 69 %, but there are differences between the classes. *Commercial* has a recall of 0.635, whereas *residential* gains 0.725. *Commercial* is most often confused with *other* (54 images). The reverse holds true as well: *other* is confused with *commercial* in 45 images. *Residential* images are equally confused with the other classes: 33 images as *commercial* and 30 images as *other*.

There are two conclusions from this verification. First, only half of the images show a building façade that humans can use to identify the

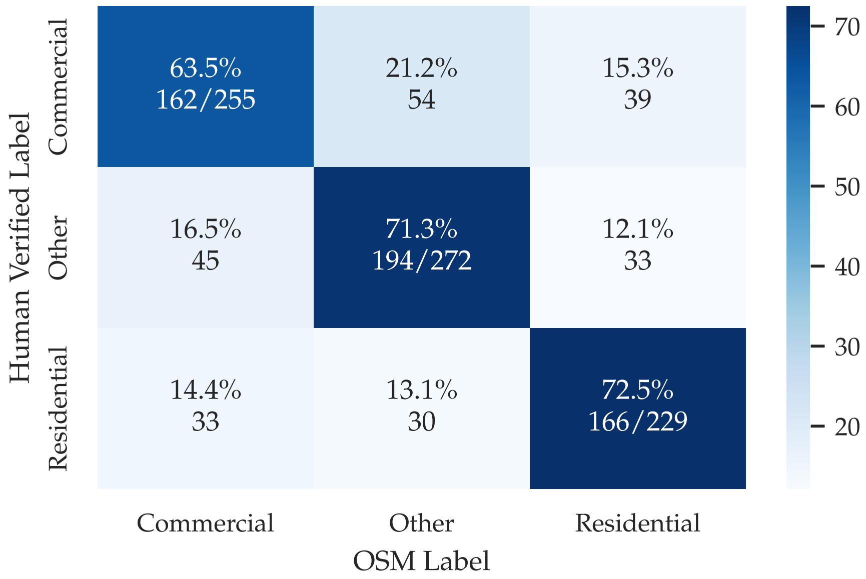


Figure 6.14: Agreement between images labels from OSM and human-verified labels as confusion matrix

building's function. Second, only two-thirds of OSM building labels are correct.

These insights enable a new assessment of the classification performance. Figure 6.15 depicts the confusion matrix for the InceptionResNetv2 model on images with human-verified labels. In this case, the weighted F1 score is 0.78, which increases 62.5 % compared to the F1 score of 0.48 on the full filtered dataset. Especially the *commercial* class gains a high F1 score of 0.81, which is mostly due to a recall of 0.92. In contrast, the *residential* class shows a precision of 0.87 and a recall of 0.64.

Generally, the relatively low classification performance from Subsection 6.2.7 is not a result of an underperforming model but rather due to label noise and unclear images. This issue needs to be considered when analyzing the performance at a building level in the next part.

6.2.10 *Performance Analysis at the Building Level*

So far, all evaluations have been performed at the level of individual images. However, the overall task is building function classification, so aggregation is needed at a building instance level. This aggregation is accomplished by fusing the predicted probability vectors for each building image using an unweighted average.

The mean fusion at the building level yields the following class distribution: 13,525 *commercial* images are aggregated to 9,345 buildings, 14,474 *other* images to 7,774 buildings, and 15,527 *residential* images to

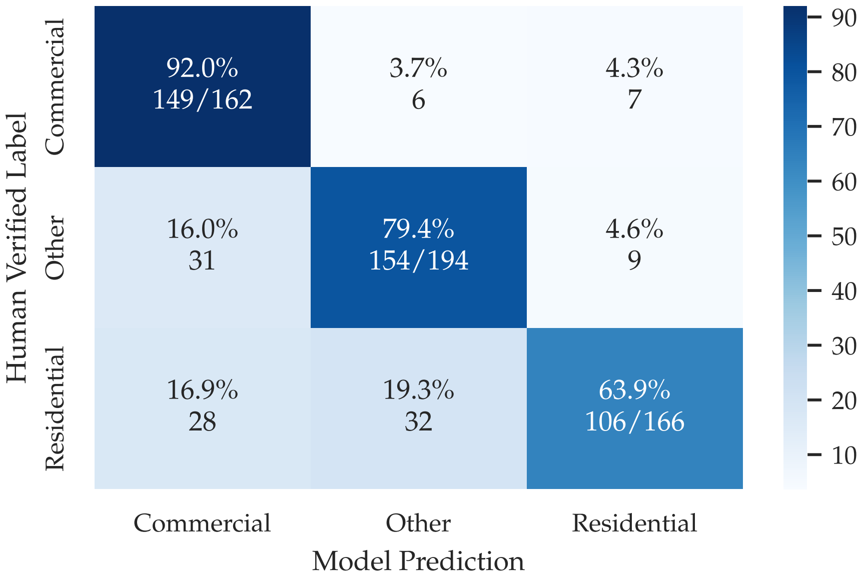


Figure 6.15: Confusion matrix of InceptionResNetv2 model on images with human-verified labels

12,748 buildings. In summary, the total number of 43,526 images is aggregated to 29,867 buildings.

The following analysis takes a closer look at how building functions are predicted in different LCZ42 cities (Figure 6.16). The number of buildings for each class is in brackets behind the city names. Most buildings are located in London, with 5,508 samples. Other cities in the top 5 are Los Angeles (4,255), Amsterdam (2,794), Berlin (2,304), and New York City (1,528).

The top 5 cities with the lowest number of buildings are Nanjing (12), Tehran (11), Changsha (10), Wuhan (7), and Islamabad (5). Dongying does not appear in this list as there is no intersection between filtered images and labeled buildings. These cities are considered outliers and are not part of any further analysis.

Most F1 scores are in line with the overall performance of 0.48 from Subsection 6.2.7. However, there are some notable examples. *Commercial* buildings show high F1 scores in Los Angeles (0.687) and San Francisco (0.631), which is related to the geographical distribution of training samples. These building types are also identified with high scores in Jakarta, 0.750 on 33 buildings, Shenzhen, 0.699 on 94 buildings, and Beijing, 0.654 on 60 buildings. On the other hand, *commercial* buildings gain the lowest F1 scores in Berlin (0.300), Rome (0.241), and Qingdao (0.200). A low number of samples can explain the latter: there are six *commercial* buildings in Qingdao.

Other buildings yield high F1 scores in cities that are famous for historical sights. Twenty *other* buildings in Cairo show 0.800 F1 score, 230 buildings in Rome reach 0.733 F1 score, and 85 buildings in

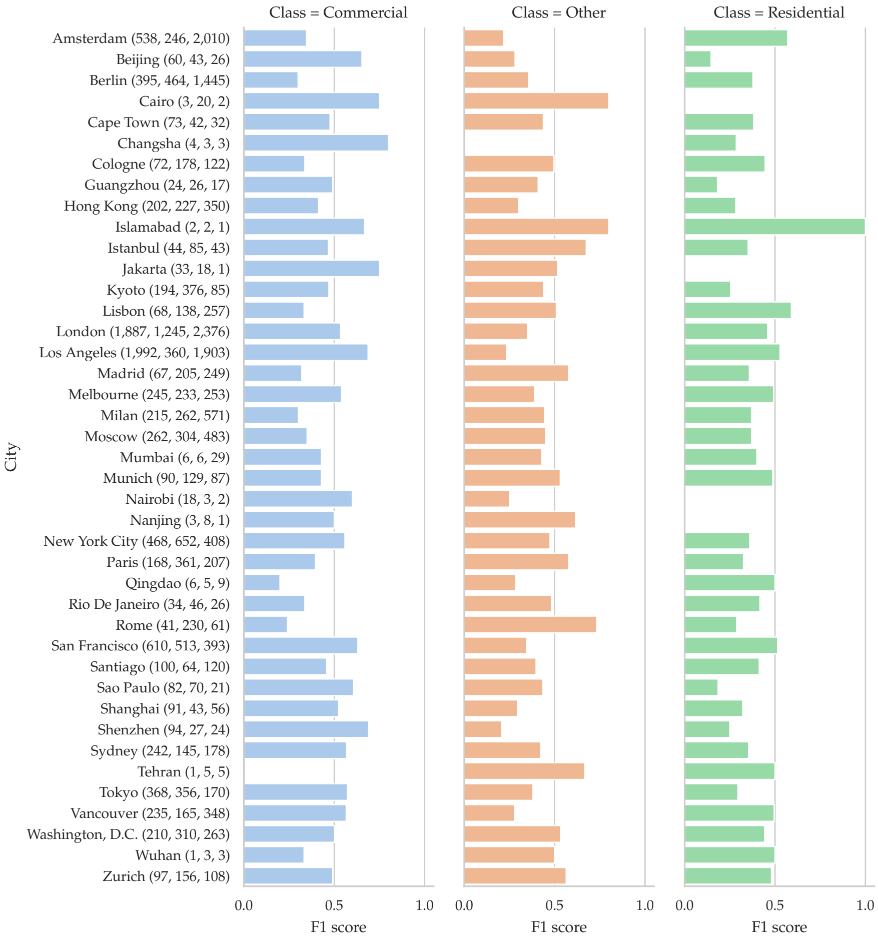


Figure 6.16: F1 scores of building predictions in LCZ42 cities at a class-wise level for InceptionResNetv2 model. The numbers in brackets behind the city names indicate the number of buildings in the same order as the plot columns.

Istanbul get 0.675 F1 score. Surprisingly, two western cities are among the three cities with the lowest scores for *other* buildings: Los Angeles with 0.233 on 360 buildings and Amsterdam with 0.218 on 246. Only Shenzhen is even lower, with 0.207 on 27 buildings.

On the other hand, these two western cities appear in the list of top 3 for *residential* buildings. Lisbon shows the highest F1 score for this class with 0.590 on 257 buildings. Next, Amsterdam has 0.570 on 2,010 buildings, and Los Angeles with 0.528 on 1,903 buildings. The cities with the lowest F1 scores of 0.0 for *residential* buildings are Cairo, Jakarta, and Nairobi, but they have just two or one building of this class.

There are no geographical patterns, but all cities show similar results. While this is probably an effect of the label noise, it also shows that the prediction model is robust and generalizes well to different cultural regions.

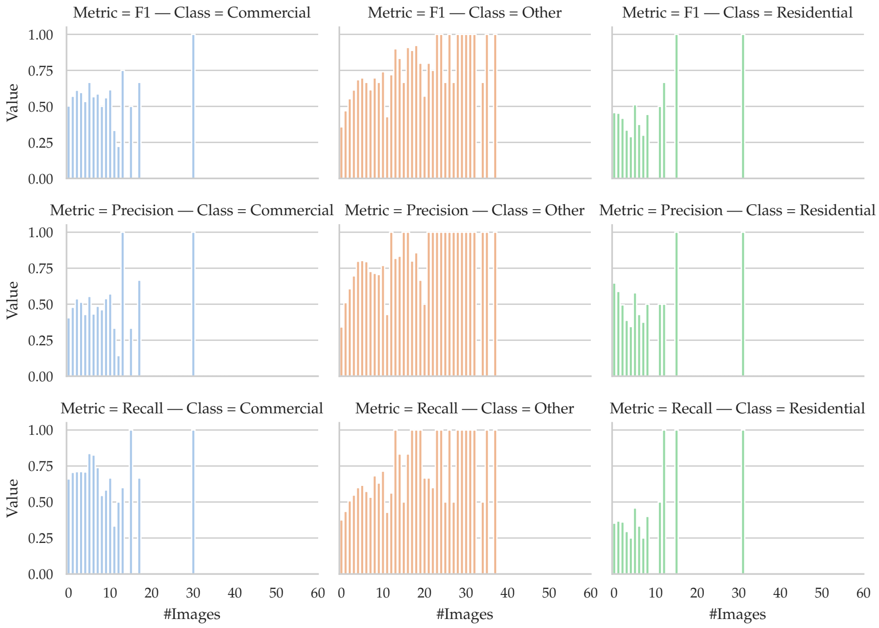


Figure 6.17: F1 score, precision, and recall as a function of the number of images per building at a class-wise level for InceptionResNetv2 model.

A summarizing confusion matrix of the results at a building instance level can be found in the Appendix, Figure A.3. Last but not least, the mean fusion approach introduces a new variable: The number of images per building. Figure 6.17 shows the F1 score, precision, and recall as functions of the number of images per building. The classification performance increases with more images per building, and the predictions become more robust. This effect is most significant for up to ten images afterward; the number of buildings with more than ten images becomes too small for statistical evaluation. As only one building has $n > 10$ images per class, the metrics become 1 or 0. However, up to ten images increase the classification performance.

6.3 SUMMARY

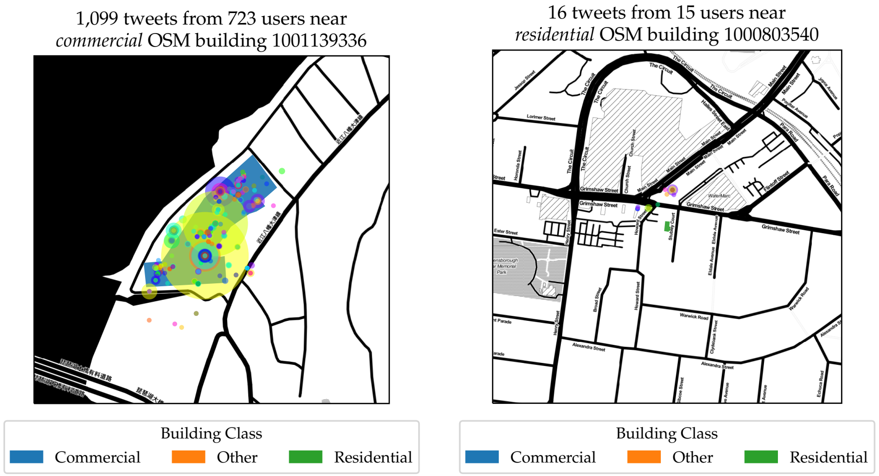
This chapter introduced a computationally efficient filtering pipeline to identify images in large social media datasets for building function classification. It shows how the content of the resulting images can be related to geospatial information to map images to buildings.

An extensive analysis investigates the effects of filtering steps on a dataset of 28 million social media images and the classification performance. Since 65.6 % of all identified OSM buildings are unlabeled, there is considerable potential for closing gaps in semantic tags of OSM. Seven state-of-the-art deep learning architectures are evaluated for predicting the building functions based on the filtered images. The results of the best-performing model received an in-depth analysis concerning the correctness of labels, geographical distribution, and effects of the image on building aggregation.

Building function classification is challenging for humans and computers on real-world datasets. Additionally, the labels of OSM data cannot be considered as ground truth as the label quality is far from perfect. A possible option to improve the label quality could be adding additional data from other platforms, e.g., Foursquare or Wikipedia, which also contain POIs data. Although these platforms have little data about *residential* buildings, they could be used for validating *commercial* and *other* labels. Going from image content to spatial knowledge requires a chain of multiple tools that introduce further uncertainty. The next chapter will propose a method independent of any social media content and focus only on its metadata to avoid these shortcomings.

7 PREDICTING BUILDING FUNCTIONS USING METADATA OF SOCIAL MEDIA POSTS

All approaches mentioned earlier use visual features from aerial or ground-level imagery. Visual data is high-dimensional and requires sophisticated algorithms combined with tremendous computing power. This chapter presents an alternative way to predict building functions using social media data by focusing on the metadata. When a user creates a social media post, its content is saved along with several other data fields describing the context during creation. For example, a Flickr image has 20 attributes apart from the image identifiers, and a tweet has 83 other attributes beyond the text. This approach works with three essential attributes of every geotagged social media post: the user id, the location, and the timestamp. While this information is trivial for a single post, it becomes a rich data source as different users create them at different times and locations. If the posts are aggregated for multiple years, spatio-temporal patterns become visible.


1,099 tweets from 723 users near
commercial OSM building 1001139336
16 tweets from 15 users near
residential OSM building 1000803540
Building Class
Commercial
Other
Residential
Building Class
Commercial
Other
Residential

(a) Tweets around *commercial* OSM building 1001139336 near Kyoto, Japan

(b) Tweets around *residential* OSM building 1000803540 in Melbourne, Australia

Figure 7.1: Examples for spatial distributions of tweets around different building classes, colors indicate different users, marker sizes illustrate the number of tweets from one user and location. Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Figure 7.1 illustrates the hypothesis behind this method: in the surrounding of *commercial* buildings, different users post a lot of content, while the number of posts around *residential* buildings is relatively low and mainly from the same users. Figure 7.1a shows the distribution of 1,099 tweets from 723 users around Pieri Moriyama, a shopping mall near Kyoto, Japan. In contrast to this pattern, the distribution of tweets in Figure 7.1b shows only 16 tweets from 15 users in the surrounding of a residential building in Melbourne, Australia.

This approach is based on spatio-temporal features derived from the metadata to keep the computational effort low. The following sections describe how the features are generated and used for training machine learning classifiers. Furthermore, Section 7.3 shows the results and discusses the performance of different classification algorithms.

7.1 CREATING FEATURES FROM METADATA

This approach puts buildings in the first place. It has two primary hyperparameters: First, a maximum distance threshold $t_d \in \mathbb{R}$ defining how far away a social media post can be, and second, a minimum post threshold $t_p \in \mathbb{N}$ that sets the minimum number of posts, which must be within the distance threshold t_d. If a building has at least t_p posts within distance t_d, the building is considered valid, and 181 features are calculated. These thresholds ensure that there is sufficient information in the defined spatial context. Thus, if a building does not fulfill these criteria, the features become meaningless and introduce noise. Therefore, these buildings are omitted. Otherwise, there are equal or more than t_p social media posts around a building. Let this resulting set of social media posts be S_b for a building b, with each post s being a tripel of user id, timestamp, and location $s = (u, t, l)$.

This section assumes that all locations are in a cartesian coordinate system to speed up distance calculations. All data sources considered in this thesis, Flickr and Twitter, provide locations in WGS 84, and hence, all location information needs to be reprojected to a suitable, local UTM zone. In this case, a suitable UTM zone is defined using the building footprint.

7.1.1 *Spatial Features*

Spatial features combine the location of social media posts with other attributes, e.g., users, distances, or point patterns.

7.1.1.1 *Density Features*

Density features represent the number of an attribute relative to the size of the region of interest.

First, there is the *Spatial Post Density* den_{post} defined as the number of posts per search area:

$$\text{den}_{\text{post}} = \frac{|S_b|}{(2t_d)^2} \tag{7.1}$$

Second, there is the number of unique users in a search area den_{user}

$$\mathrm{den}_{user\_area} = \frac{|\{(u, \dots) \in S_b\}|}{(2t_d)^2} \tag{7.2}$$

Third, there is the number of users per post

$$\mathrm{den}_{user\_post} = \frac{|\{(u, \dots) \in S_b\}|}{|S_b|} \tag{7.3}$$

This feature might seem counterintuitive in the first place. However, it has the same information as the number of posts per user, but the inverse scales between 0 and 1. Otherwise, the numbers might become large compared to all other features.

7.1.1.2 *Distance Features*

All four features are statistical measures based on distances between a building centroid c_b and the locations of surrounding social media posts. They are the minimum distance, the mean distance, the maximum distance, and the standard deviation of the mean. As these statistics are intuitive, no formulas are given.

7.1.1.3 *Spatial Entropy*

In information theory, entropy describes the degree of information in a random variable. Claude Shannon was the first who adopted the entropy known in physics and formulated a mathematical theory of communication [190]. Given a discrete random variable X with possible outcomes $x_1, \dots, x_n, n \in \mathbb{N}$ with probabilities $p(x_1), \dots, p(x_n)$, the Shannon entropy $H(X)$ is defined as

$$H_S(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{7.4}$$

There are two possible outcomes of the random variable in a binary scenario: 0 and 1. A data source that only contains zeros has 0 entropy as there is no change or surprise in the data. The maximum entropy has a data source with an equal number of zeros and ones, making the output unpredictable. The maximum entropy of a discrete random variable is $\log(n)$ with n as the number of possible outcomes.

Shannon formulated his entropy with a sender and a receiver in mind, but in geographical science, there are at least two dimensions to take into account. Michael Batty built upon Shannon's entropy and evolved it into a spatial entropy measure by discretizing a region of interest into small, random partitions [191]. His entropy compares how often an outcome of a random variable occurs in a partition. More formally, the Batty entropy H_B is defined as

$$H_B(X) = \lim_{\Delta s_i \to 0} - \sum p(x_i) \log \frac{p(x_i)}{\Delta s_i} \tag{7.5}$$

with s_i as the size of the partition i that is close to zero.

Karlström and Ceccato extended the idea of a spatial entropy measure based on partitioning by including spatial correlation [192]. Their S-statistics takes the surrounding partitions into account and calculates entropy based on how much a partition differs from its neighbors. Intuitively, the resulting entropy is low if neighborhoods are very similar. They formulate it as

$$H_{KC}(X) = \sum_{i=1}^{n} p(x_i) \log \sum_{j=1}^{n} w_{ij} \frac{p(x_i)}{q(x_i)} \tag{7.6}$$

with w_{ij} being the weight from an adjacency matrix W and $q(x_i)$ the probabilities from other partitions. The adjacency matrix w_{ij} is 1 if two partitions p and q are neighbors and 0 if not.

This thesis is based on an implementation of entropy measures in R [193] providing Battys entropy H_B and Karlström and Ceccatos entropy H_{KC} as an absolute abs and a relative value rel: $H_{B_{abs}}$, $H_{B_{rel}}$, $H_{KC_{abs}}$, $H_{KC_{rel}}$. All four values are considered metadata features derived from the locations of social media posts.

7.1.2 *Temporal Features*

These features take only the timestamps of social media posts into account; any user or location information is omitted. Moreover, all timestamps are converted to local time based on the geotag of the social media post.

7.1.2.1 *Time Difference Features*

For calculating time differences, all posts are put in chronological order. This order is used to compute two features: First, the mean difference between two subsequent posts in days, and second, the respective standard deviation.

7.1.2.2 *Hour of Week Features*

These features capture when users create social media posts. Divided into 24 hours for all seven days, this yields 168 features counting the number of posts during a given hour of the week. As all timestamps are represented in local time, they reflect the actual time patterns of their location and are not shifted toward Greenwich Mean Time (GMT).

7.1.3 *Normalization of Features*

All spatial and temporal features have different scales, e.g., spatial distance-based features range between 0 and t_d, whereas time distance can be more than 365 days if there are posts between years.

However, machine learning methods require evenly scaled data across all features. Hence, all features are normalized. The spatial density features are scaled using the area of interest or the number of posts. Spatial distance features are scaled with the maximum distance threshold t_d. The four spatial entropy features are within the range of 0 to $\log(n_{classes})$ and are normalized by definition. Time difference features are normalized by the number of days per year, 365, whereas the hour-of-week features are scaled using the number of posts $|S|$.

7.2 TRAINING CLASSIFICATION ALGORITHMS USING METADATA FEATURES

As the features are simple and numerical, there is no need for a sophisticated classification algorithm. Simplicity is key in this approach, with easy-to-compute features and lightweight models for prediction.

Therefore, two different algorithms are compared: First, a gradient-boosted tree algorithm, and second, a multilayer perceptron.

7.2.1 *Gradient Boosted Trees*

Gradient-boosted trees are combinations of multiple decision trees trained subsequently. Each decision tree itself is a weak classifier, but together they compensate for their weaknesses and form a powerful committee [194].

The general principle behind it is called *boosting*. Basic committee or ensemble methods use a set of independent classifiers and combine their results to predict. In contrast, boosting starts with an initial, weak classifier that is slightly better than random. Afterward, the method analysis for which samples of the dataset this classifier fails and trains an additional classifier with these samples. The more classifiers are trained, the more weight is put on samples that are hard to predict. Once a given number of classifiers is trained, the final prediction for a new sample is calculated by weighting the results of individual classifiers.

Boosting became popular with the introduction of *AdaBoost* [195] and was extended towards learning with gradients in 2001 [196]. This theoretical work on gradient boosting has been implemented in different frameworks, e.g., MART [197], CatBoost [198], or LightGBM [199]. However, there is one outstanding implementation that has proven successful in different prediction challenges: XGBoost [200]. It showed superiority against various other classification methods, especially when handling different feature types [201].

XGBoost uses decision trees as weak classifiers and trains them subsequently with Boosting. Three parameters heavily influence its performance:

1. Learning rate lr
2. Number of trees n_{tree}
3. Maximum tree depth n_{level}

For our approach, XGBoost is trained using a cross-entropy loss. The optimal configuration of the learning rate, number of trees, and maximum tree depth is subject to analysis in the subsequent evaluation 7.3.

7.2.2 *Neural Networks*

An alternative algorithm for predicting building functions is a neural network with hidden dense layers. A network needs one input neuron for each feature. These input neurons are succeeded by an arbitrary number of hidden units that build the first hidden layer. An optional second hidden layer takes more non-linearities into account. The final output layer consists of three neurons indicating a pseudo-probability for each class to predict.

The network is trained using Adam [46] as an optimizer with a sparse categorical cross-entropy loss for at most eight epochs. Checkpointing prevents overfitting to the training set by monitoring the validation loss. The final model is restored from the weights that yielded the lowest validation loss during the training process.

7.3 EVALUATION

This evaluation is structured as follows: First, the creating phase of the dataset is described briefly. Afterward, the results of the two different model types are analyzed, and the best model is selected. Subsequently, this model is evaluated concerning feature importance, confusion matrix, and the results in different geographical areas.

7.3.1 *Creating a Dataset*

The dataset used in this thesis is created based on the list of buildings from the So2Sat social media dataset having at least one tweet in a surrounding of 50 m. This threshold is also used as a distance threshold $t_d = 50$ for creating the dataset. The minimum number of posts is $t_p = 5$.

Out of 655,425 buildings in the So2Sat social media dataset, 385,975 fulfill the criteria above. Hence, the dataset used in this chapter consists of 385,975 buildings, each with 181 features. For training and testing, the dataset is split randomly into 67 % training and 33 % test data.

7.3.2 *Comparison of Classification Algorithms*

Both model types come with multiple hyperparameters that have a tremendous influence on the final performance. Therefore, an extensive grid search was applied to find the best models. All models were trained on an Nvidia DGX-1 server with 80 cores à 2.2 GHz, 512 GB memory, and one Nvidia Tesla V100 GPU with 32 GB. For XGBoost, nine learning rates, thirteen values for the number of estimators, and nine values for max depth were trained and evaluated, 1,053 models in total. The following parameters were used:

$lr \in \{0.6, 0.3, 0.1, 0.06, 0.03, 0.01, 0.006, 0.003, 0.001\}$

$n_{tree} \in \{4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4098, 8192, 16384\}$

$n_{level} \in \{3, 5, 7, 9, 11, 13, 15, 17, 19\}$

Table 7.1 shows the top ten models concerning F1 score in descending order. Overall, the F1 score is very similar among all models with 0.72. Small differences start with the third digit. All models are in the middle of the parameter space, which indicates that they are close to the optimal model, and further extension of the space would not gain more performance. The top-performing model has 4,096 trees, at most 13 levels, and was trained at a learning rate of 0.06. This model gained the highest F1 score of 0.723, while the training took 11.8 minutes. All other models have a higher complexity with more parameters and a longer training time (except for #6). Hence, this fast and lightweight model is used for further comparisons and analysis.

A similar evaluation was done for neural network models: The grid search investigated eight learning rates, four batch sizes, nine sizes for the first layer, and ten sizes for the second layer. The number of units for the second layer has 0 as an additional value to analyze single-layer models. In total, 2,880 were trained and evaluated using the following parameters:

$lr \in \{0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001, 0.00003, 0.00001\}$

$bs \in \{8, 16, 32, 64\}$

$n_{first} \in \{4, 8, 16, 32, 128, 256, 512, 1,024, 2,048\}$

$n_{second} \in \{0, 4, 8, 16, 32, 128, 256, 512, 1,024, 2,048\}$

Table 7.2 contains the performance of the top ten models sorted descending by their F1 score. The NN models show a similar pattern as the XGBoost models: All gain a similar F1 score of approximately 0.50, although their architectures are very different. For example, the third model is shallow, with 256 units in the first and 16 in the second layer. In contrast to this model, the fourth one has 1,024 units in the first and 2,048 units in the second layer. Hence, the #3 model has 50,755

	Duration	lr	n_{level}	n_{tree}	F1 score
1	11.8min	0.06	13	4096	0.723016
2	34.1min	0.03	13	16384	0.722678
3	15.1min	0.06	15	4096	0.722639
4	17.2min	0.06	13	8192	0.722545
5	23.0min	0.06	17	8192	0.722537
6	11.5min	0.10	15	4096	0.722435
7	23.1min	0.03	13	8192	0.722404
8	15.4min	0.10	15	8192	0.722372
9	25.1min	0.03	11	16384	0.722364
10	1.2h	0.01	15	16384	0.722349

Table 7.1: Parameters of top-10 XGBoost models w.r.t. F1 score, sorted descending. lr is learning rate, n_{level} the maximum tree depth, and n_{tree} the maximum number of trees

trainable parameters while the #4 model has 2,291,715 parameters. The top-performing model has 179,715 parameters based on 256 units in the first layer and 512 units in the second layer. It was trained for 3.6 minutes using a learning rate of 0.003 and batch size of 64, yielding an F1 score of 0.505. Two patterns show up: Firstly, larger batch sizes are better as six out of ten models were trained using a batch size of 64, three with 32, one with 16, and none with 8. Secondly, bottleneck architectures are predominant. Seven models have a first layer that is larger than the second one.

Despite all structural differences, the final performance is quite similar among all models, indicating that an F1 score of 0.51 is an upper bound. NN models are at least 0.21 below the XGBoost models compared with the gradient-boosted tree models. This finding is in line with [202, p. 19], stating that *gradient boosting is used for problems where structured data is available, whereas deep learning is used for perceptual problems*.

7.3.3 *Analysis of Best Performing Model*

This subsection takes a closer look at the top-performing XGBoost model with 4,096 estimators, a maximum depth of 13, and a learning rate of 0.06. This model was trained in 11.8 minutes and gained an F1 score of 0.723. Figure 7.2 shows the confusion matrix of this classifier on the test data. *Commercial* buildings are correctly predicted in 59.3 % and in 37.2 % confused with *residential* buildings. In 3.4 % of the cases, a *commercial* building is wrongly predicted to be *other*. Truly *other* buildings are accurately classified in 16.4 %, hence the majority is

	Duration	lr	bs	n_{first}	n_{second}	F1 score
1	3.6min	0.0030	64	256	512	0.504886
2	7.1min	0.0030	32	512	32	0.503232
3	13.9min	0.0003	16	256	16	0.501860
4	4.5min	0.0010	64	1024	2048	0.500750
5	7.3min	0.0030	32	256	1024	0.499757
6	3.4min	0.0030	64	128	64	0.495546
7	3.5min	0.0100	64	2048	4	0.494862
8	4.2min	0.0300	64	2048	32	0.494674
9	3.4min	0.0001	64	2048	512	0.494259
10	6.5min	0.0001	32	2048	64	0.493770

Table 7.2: Parameters of top-10 neural network models w.r.t. F1 score, sorted descending. lr is learning rate and bs the batch size. n_{first} and n_{second} denote the number of hidden units in the first and second layer of the neural network.

mixed-up with *residential* (45.6 %) and *commercial* (38.0 %). *Residential* buildings are correctly identified in 89.2 %. The remaining *residential* samples are predicted as *commercial* in 9.3 % and as *other* in 1.5 %.

As the test dataset is unbalanced towards *residential* buildings, one might suspect that the model is predicting according to the class frequencies. 60.6 % of all 127,372 samples are *residential*, but 89.2 % are correctly classified, which indicates that the model captures more than a random, frequentistic classifier. For *commercial* one could expect 27.5 % correct predictions, but there are more than twice the relative numbers of accurate classifications. For *other* buildings, the difference is not that significant: While expecting 12.0 % accurate predictions, the number is 16.4 %. Hence, the features capture information about building functions, which is straightforward mainly for *residential* buildings and fuzzier for *other* buildings.

Gradient-boosted models have a significant advantage over deep learning models: They are based on decision trees, and thus, the input features can be analyzed concerning their importance. There are different ways in which feature importance can be measured: gain and weight. The gain value tells how discriminative a feature is when used for a split. The weight value counts how often a feature was used to create a split.

A simple example illustrates the relation: One wants to predict if a forest can adapt to climate change. The features are, among others, a binary feature whether a forest is a monoculture and the age of the forest. The binary attribute can only be used once in a tree, but it is rather distinctive [203]. As the age feature is numerical, it can serve multiple times as a split criterium with different values in the same

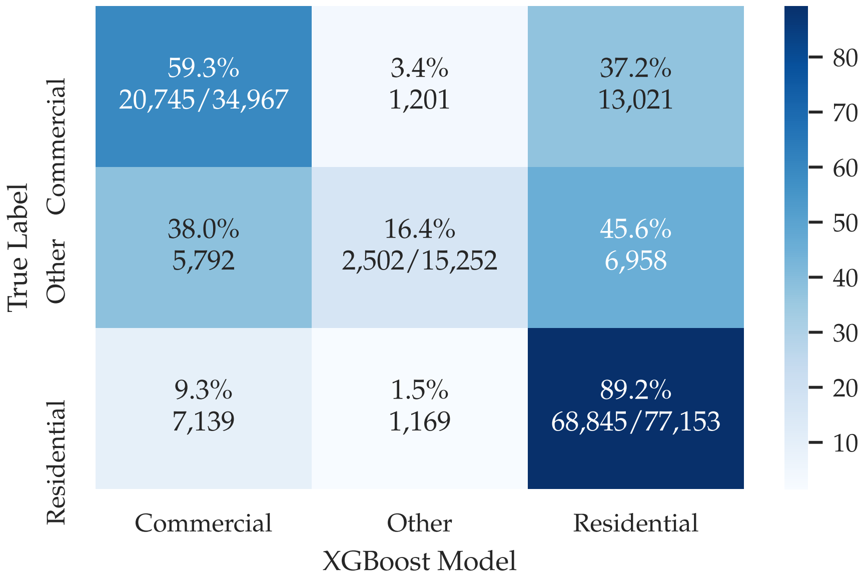


Figure 7.2: Confusion matrix of Top1 XGBoost model when predicting the test dataset

tree. Hence, the monoculture feature has a significant gain but low weight. Vice versa, the age will appear with a low gain but a high weight.

Therefore, this analysis looks at both metrics: gain and weight. Figure 7.3 shows the top 25 features by gain. This value has no units and denotes the average gain if this feature is used. In this case, the most important feature is the user area density, i.e., the number of unique users relative to the area, with an average gain of 5.8. The second most gain comes from the relative Karlstrom entropy with 3.8. All other top-23 features have a similar gain of approximately 0.94 to 1.10 and cover the early morning hours (except for #13, the Karlstrom entropy). As these features are very sparse, they do not often appear in the decision trees, but they are very distinctive if they do.

Figure 7.4 shows the 25 most important features concerning their weight, i.e. how often they appear in a tree. In this case, two distance-based features are at the top: the mean distance between posts and buildings is used most often (249,009), followed by the minimum distance between the two entities (247,423). Third, there is the Batty entropy with a weight of 240,190. Afterward, the standard deviation of the distance between posts and buildings is the fourth-ranked feature (229,738). Subsequently follows the first time-based feature, the standard deviation of days between two posts with a weight of 228,823. Next is the last distance-based feature, the maximum distance between posts and buildings (226,040). Seven-ranked is the mean time delta between two posts with a weight of 209,563, followed by the user post density with a weight of 195,175. The next two

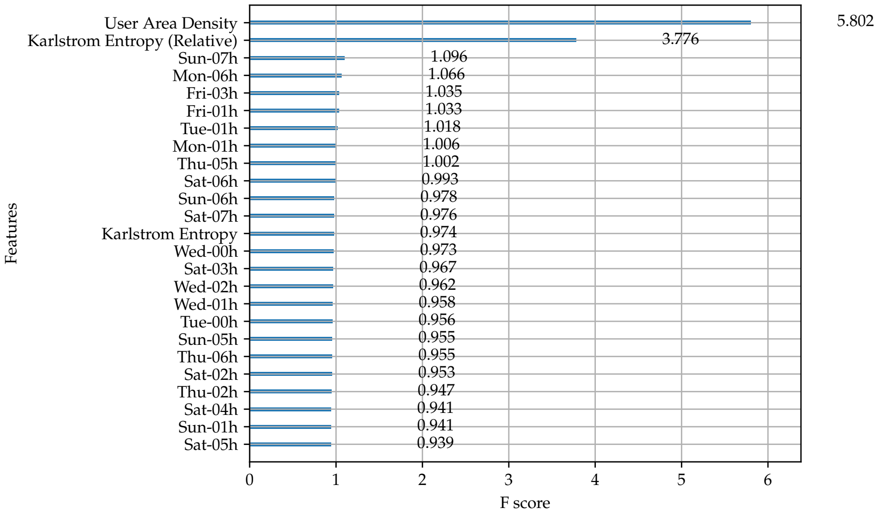


Figure 7.3: 25 features with the highest average gain in gradient boosted tree model with 4,096 estimators, a maximum depth of 13, and a learning rate of 0.06

features appeared also in Figure 7.3: the Karlstrom entropy (192,181 weight) and the user area density (106,939 weight). The latter has approximately the same weight as the remaining density-based feature: the spatial post density with 106,610. Starting with rank twelve, hours of the week features are dominating with lower weights between 26,407 and 30,434. In contrast to the feature importance by gain, the midday hours have more importance, especially during the weekend. The relative Karlstrom entropy has the lowest weight of 8 (Figure A.5), which was the second most important feature regarding gain.

Although some features have more importance than others, the two sides of this evaluation show that all features contribute to the final decision of the XGBoost models. Some are more versatilely helpful, i. e.have a high weight, and some are more helpful in some exceptional cases, which means a high gain. Only the user area density showed up in both evaluations, highlighting that the number of unique users is the most important feature for predicting a building function.

7.3.4 *Predictions on Buildings from Social Media Image Approach*

The results from the previous chapter cannot be directly compared to the results of this chapter as the datasets are different. Therefore, an additional metadata dataset was created that contains all buildings identified in Subsection 6.2.10. These buildings were identified on Flickr images, but the metadata feature vectors are always calculated based on Twitter tweets. Out of 29,867 buildings, 18,489 have at least five tweets in their surrounding. However, 38 of these buildings are also in the training set of this approach and therefore are excluded.

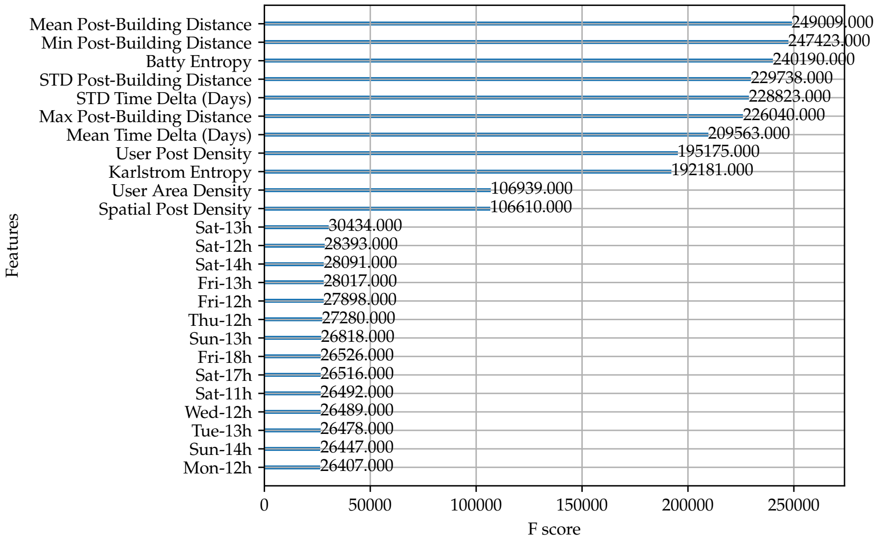


Figure 7.4: 25 features with the highest weight in gradient boosted tree model with 4,096 estimators, a maximum depth of 13, and a learning rate of 0.06

Hence, this evaluation of the best-performing XGBoost model takes place on 18,489 buildings identified in the social media image approach. As both datasets are independent, buildings from the social media image set can be part of the training dataset for this method. The intersection between both sets contains 9,565 buildings, which must be excluded for an unbiased evaluation. Therefore, the following analysis is performed on 8,924 buildings, 3,018 of them *commercial*, 2,073 *other*, and 3,833 *residential*.

The top-performing XGBoost model from above was used to predict the functions of these buildings yielding a weighted F1 score of 0.557. Figure 7.5 shows the confusion matrix, which is similar to the one from the test set. *Commercial* buildings show a higher recall than before: it raised from 59.4 % by 9.6 % to 69 %. This class is still mostly confused with *residential* buildings but with a lower frequency of 26.4 % compared to 37.2 % in the test set. The *other* class has a slightly increased value as well: from 16.4 % to 19.7 %. In contrast to the test set, it is mostly confused with *commercial* buildings (47.4 %) rather than *residential* buildings (32.9 %). *Residential* buildings gain the highest recall of all classes again but at a lower number. While the model was correct at 89.2 % for the test set, the value decreased to 70.2 % in this case.

Going toward a city level provides more insights into how this model performs in different cultures (Figure 7.6). However, four LCZ42 cities are not covered here due to a lack of data: Changsha, Dongying, Tehran, and Wuhan. The F1 score was used for this analysis, and cities with less than five buildings for a given class are considered outliers. *Commercial* buildings are predicted best in Jakarta, Istanbul, and Cape

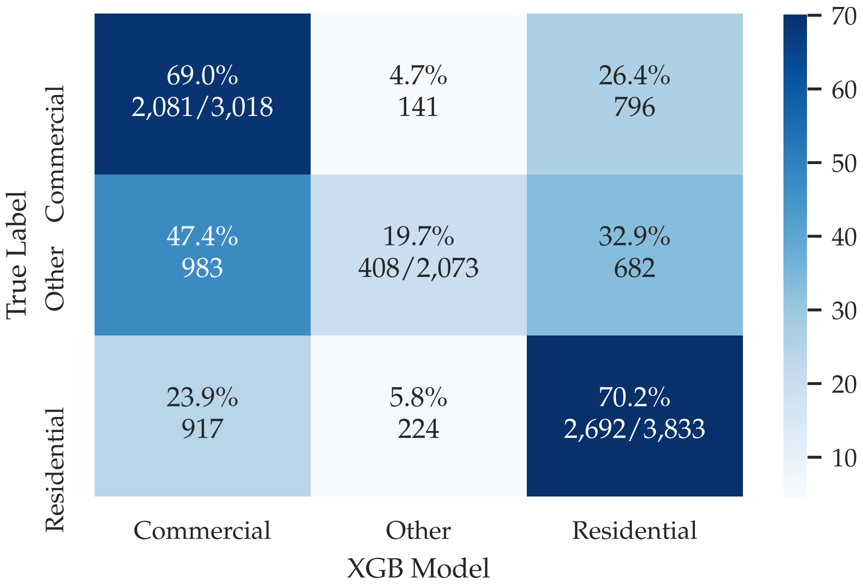


Figure 7.5: Confusion matrix of Top1 XGBoost model when predicting the metadata of buildings from the social media image approach

Town with scores of 0.833 on 15 samples, 0.800 on seven samples, and 0.769 on 23 samples, respectively. The lowest F1 scores are found in Bejing with 0.333 (9 samples), Madrid with 0.317 (16 samples), and Rome with 0.212 (11 samples). While *commercial* buildings are predicted with an F1 score of 0.594, *other* buildings have generally a lower score of 0.287. Seven cities have less than five *other* buildings and are therefore not discussed. The highest F1 scores for *other* buildings are from Kyoto with 0.569 on 152 samples, Hong Kong with 0.524 on 29 samples, and Sydney with 0.451 on 48 samples. On the three last ranks, there are Amsterdam (0.136 on 90 samples), Madrid (0.127 on 56 samples), and Cairo (0.000 on six samples). Although Amsterdam has the third-lowest result for *other* buildings, it yields the highest F1 score for residential buildings with 0.788 on 903 samples. Next, Hong Kong has a score of 0.769 on 68 buildings, and Madrid gains 0.758 on 90 samples. The lowest F1 scores are from Instanbul (0.286 on five samples), New York City (0.275 on 41 samples), and Cape Town (0.182 on five samples). Moreover, nine cities have less than five *residential* buildings and are excluded.

Generally, there is no apparent bias towards any culture: the method shows its strengths and weaknesses across all LCZ42 cities. A lack of data in Chinese cities prevents insights into this region, but the limited good results from Beijing and Guangzhou indicate that the method generalizes. However, there is likely a bias towards neighborhoods with high income and good education as this is the primary user basis of Twitter [123]. This issue opens the opportunity for further research

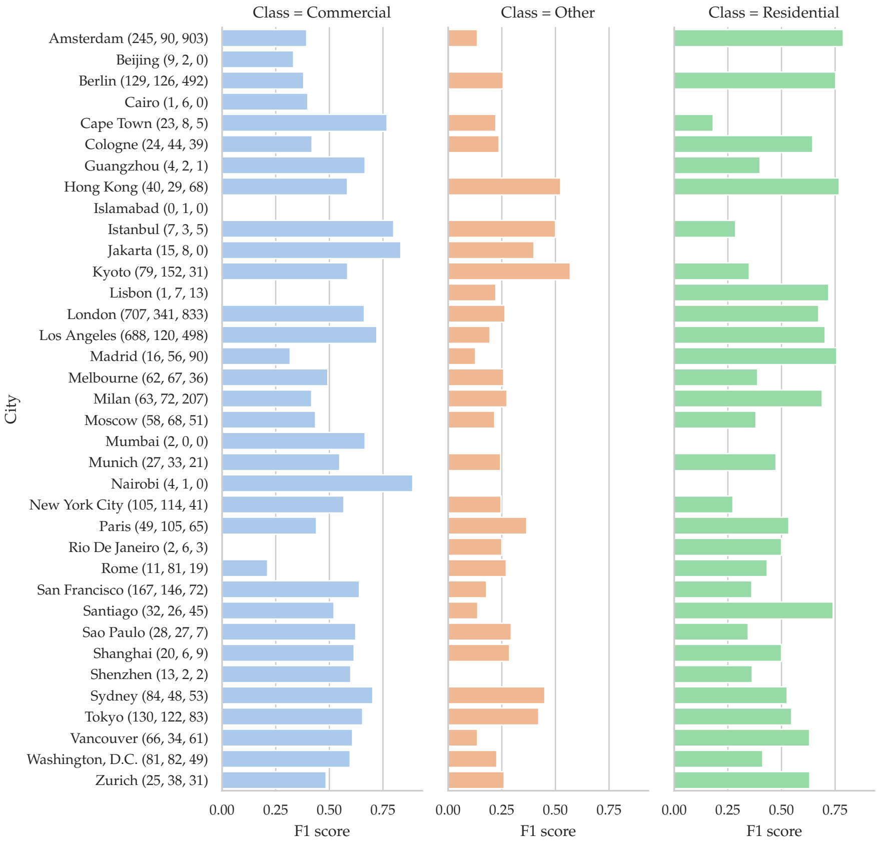


Figure 7.6: Geographical analysis of Top1 XGBoost model when predicting the metadata of buildings from the social media image approach. Numbers in brackets behind city names denote the number of buildings per class in column order.

investigating if there is a correlation between mapped buildings and socioeconomic parameters.

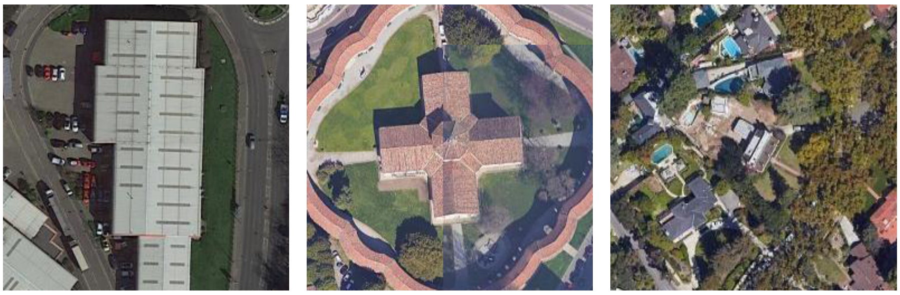
7.4 SUMMARY

This chapter described a method to predict building functions based on three attributes of social media metadata: user id, location, and time. After defining a region of interest around a building, a set of social media posts is collected in the vicinity of a building. The method aggregates this set in 181 spatial and temporal features. Two different classification algorithms were evaluated for their suitability to predict building functions based on these features: gradient-boosted trees and neural networks. An extensive grid search showed the optimal hyperparameter settings for both algorithms. A gradient-boosted tree model with 4,096 estimators, a maximum depth of 13, and a learning rate of 0.06 yields the best prediction performance with a weighted F1 score of 0.723. An in-depth analysis of this model showed that *commercial* and *residential* buildings are predictable with high accuracy, but *other* buildings are confused with the two other classes. Evaluating the feature importance highlights that all features contribute to the final predictions, some because they are very distinctive (Karlstrom entropy) and others as they often help in indifferent situations (spatial distance features). A second dataset was created using the set of OSM buildings from the previous chapter to check the model's performance on different data distributions. This analysis confirmed the findings from the test dataset and explored the prediction performance in the LCZ42 cities.

Furthermore, no cultural bias is evident. As social media is widely adopted worldwide, temporal and spatial patterns around *commercial* and *residential* buildings are culture-independent and globally similar. Hence, this approach is versatile and independent of regional differences as long as there is sufficient data from a social media platform. Nevertheless, it shares the effects of label noise with the previously introduced methods. Especially the *other* building class combines buildings with highly different usage patterns, e. g. schools, hospitals, and public transport stations. They are all used at different times but have the same label, making it difficult for a machine-learning algorithm to detect patterns. The next chapter will introduce a social media-independent approach: A remote sensing method to classify building functions. It uses aerial images to predict building functions based on roof types and spatial surroundings from a nadir view.

8 PREDICTING BUILDING FUNCTIONS USING REMOTE SENSING DATA

Remote sensing data is not always suitable for predicting building functions, especially in dense urban areas [17]. However, its worldwide coverage provides a good baseline for solving this task. Nevertheless, some buildings show characteristic patterns that allow predictions with high confidence in their function.



(a) Google aerial image for *commercial* OSM building 12357640 in London, GB

(b) Google aerial image for *other* OSM building 28393 in Milan, Italy

(c) Google aerial image for *residential* OSM building 15946583 in Los Angeles, USA

Figure 8.1: Examples for Google aerial images showing buildings with a clear function. Imagery © Google

Figure 8.1 shows clear examples for all building function classes. *Commercial* buildings like malls tend to have roofs with metal sheeting and are often surrounded by parking spaces for their customers (Figure 8.1a, a warehouse in northern London, Great Britain). A cemetery with a church in the middle is an example of an *other* building as in Figure 8.1b. It shows the *Rotonda della Besana* in Milan, Italy. Figure 8.1c depicts residential buildings in the north of Los Angeles surrounded by green gardens, pools, and trees. More generally, this task is also referred to as aerial scene classification. This chapter describes creating a culturally diverse but class-wise balanced aerial imagery dataset. This dataset is used to train models with the aforementioned fine-tuning approach from Section 5.1. All models are evaluated on a dedicated test set and aerial images of buildings from the social media image dataset.

8.1 METHODOLOGY

Naturally, there is a huge class imbalance between *other* and *residential* buildings. For example, schools, hospitals, and train stations as *other* buildings serve multiple *residential* buildings at the same time. The number of *Commercial* buildings is usually between both other classes as there are more shops, restaurants, and industry than *other* institutions, but less than *residential* buildings with one or more households. Additionally, a generalizable model needs to be trained on samples from different regions worldwide. However, the number of labeled buildings varies among different countries and cities. If naturally sampled, this leads to imbalanced datasets from two aspects: cultural and class-wise. Moreover, most machine learning algorithms are prone to predict the majority class as this yields the lowest loss while training [204].

8.1.1 *Creating a Culturally and Class-wise Balanced Dataset*

A balancing algorithm for this case has two objectives in parallel: It needs to balance classes and cities simultaneously. For this thesis, an undersampling method was implemented that has three main steps:

1. Calculate the aspired number of samples per class and city n_a
2. Create an initial list of samples by random sampling of n_a samples from each city and class
3. Backfill with unused samples for classes and cities that have less than n_a samples

Algorithm 1 contains the pseudocode for the procedure. This algorithm takes a set of triples (b, l, c) with building id b, label l, and city c as input, and returns a subset of the input. The first step identifies the class with the lowest support and divides the number of samples by the number of cities (lines 2-4). This number is denoted as n_a. Next, for each class and city, n_a samples are drawn from the original set and added to an initial sample list (lines 8-19). If the number of samples is larger than n_a, a random sample of size n_a is added to the initial list (lines 11-14). All remaining samples are saved for the subsequent step. If there are not enough samples, the number of missing samples is saved for the next step (lines 15-17). In this third backfilling step, the sum of missing samples for each class is calculated and randomly drawn from the remaining class sets (lines 20-22). Hence, the number of samples for each class is balanced, and the distribution at a city level follows roughly the original distribution. This balanced dataset is ready to use for fine-tuning.

8.1.2 *Fine-tuning on Aerial Images*

The fine-tuning methodology from Section 5.1 is applied again to create models that predict building functions using aerial imagery.

Algorithm 1: Class- and city-wise dataset balancing algorithm

```
Input  : $D = \{(b_1, l_1, c_1), ..., (b_n, l_n, c_n)\}, n \in \mathbb{N}$
begin
    assert $\forall(b_x, l_x, c_x) \in D,\ \nexists(b_y, l_y, c_y) \in D : b_x = b_y$
    cities $\leftarrow \{c_1, \ldots, c_n\}$
    labels $\leftarrow \{l_1, \ldots, l_n\}$
    $n_a \leftarrow \min_{lb \in labels}\{|\{b, l = lb, c\}\}|/|labels|$
    result $= \emptyset$
    surplus $= \emptyset$
    missing $= \emptyset$
    /* Sampling the initial set                              */
    for city in cities do
        for label in labels do
            subset $\leftarrow \{b, l = label, c = city\}$
            if $|subset| > n_a$ then
                data, plus $\leftarrow$ random_split(subset, $n_a$)
                result[label] = result[label] $\cup$ data
                surplus[label] = surplus[label] $\cup$ plus
            else if $|subset| < n_a$ then
                result[label] = result[label] $\cup$ subset
                missing[label]+ = $n_a - |subset|$
            else
                result[label] = result[label] $\cup$ subset
    /* Backfill missing data                                 */
    for label in labels do
        data, _ $\leftarrow$ random_split(surplus[label], missing[label])

        result[label] = result[label] $\cup$ data)
Output: result as balanced subset $D'$
```

Although ImageNet models are trained on object photos, a study showed that basic features from such models are also helpful for optical remote sensing data [205]. Moreover, ImageNet-based CNN models can also help when applied to night light remote sensing data [206]. The third-ranked solution in the FMoW challenge initialized its models with ImageNet weights [139].

8.2 EVALUATION

8.2.1 *Balancing Algorithm*

The So2Sat social media dataset comprises 655,425 buildings in the LCZ42 cities, with at least one tweet surrounding 50 m. Each building comes with an aerial image centered on the building centroid. The class distribution is as follows: 158,435 are *commercial*, 87,863 are *other*, and 409,138 are *residential*. Figure A.6 shows the original distribution of the dataset. For evaluation purposes, this dataset is split into a training and test set by 80:20. After applying the balancing algorithm and train-test-split, 70,363 samples are available for training from each class. Figure 8.2 shows the details on a log scale. The most *commercial* and *residential* buildings are from Los Angeles, while most *other* buildings are from London. Cities from the US and Great Britain dominate the first three ranks, but Moscow and Jakarta are in ranks four and five, introducing cultural variation. The list continues with Amsterdam, Istanbul, Kyoto, Sao Paulo, and Madrid. Hence, the ten cities with the most samples partly cover America, Europe, and Asia. Chinese cities are in the last six positions with less than 350 buildings samples each.

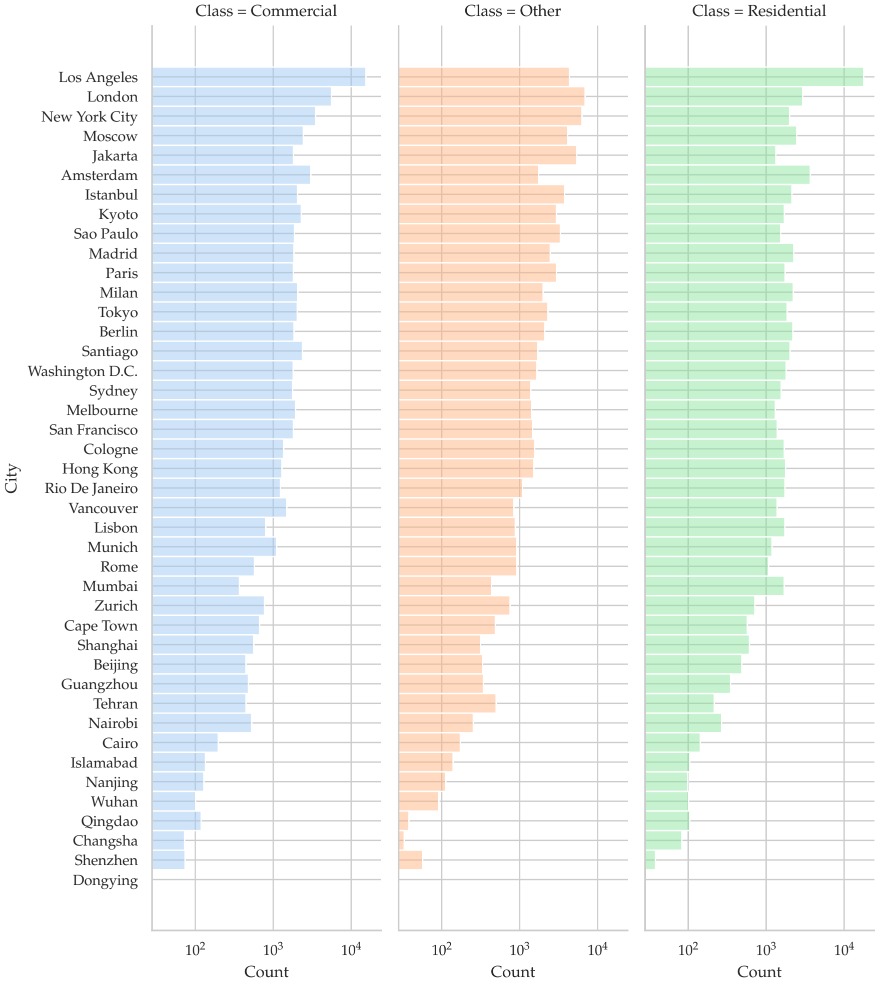


Figure 8.2: Number of aerial building images after balancing in the training dataset, sorted in descending order by the number of buildings on a log scale.

8.2.2 *Predictions on Test Dataset*

Training and test sets follow the same distribution as the balancing is done before the split is applied. Table 8.1 shows the results of all seven architectures fine-tuned on the training set and evaluated on the test set. The VGG16 model outperforms all other models concerning precision (0.738), recall (0.738), and, hence, the F1 score (0.736). The slightly lower F1 score is due to numerical instabilities. This model gains the highest F1 scores for all classes with 0.717 for *commercial*, 0.683 for *other*, and 0.809 for *residential*. There are only three metrics for which the VGG16 model is outperformed. In every case, the InceptionResNetv2 model yields better results: it has better precision for *commercial* (0.707 vs 0.694) and *residential* (0.781 vs 0.780). Moreover, the InceptionResNetv2 has a better recall for *other* buildings with 0.689 compared to 0.633 of VGG16. All other models gain lower values, with ResNet50v2 as the worst-performing model (weighted F1 score of 0.677). A possible explanation for this result might be that the VGG16 features for ImagetNet are more generic than those from other architectures. As VGG16 has fewer convolutional layers and, thus, less capacity than all other architectures, these layers must be highly generic to capture different patterns in ImageNet.

The following analysis looks in-depth at the results of the best-performing model, VGG16. Figure 8.3 shows the confusion matrix of this model on the balanced test set from the So2Sat dataset. 74.2 % of all *commercial* buildings are correctly classified. The most errors, 15.5 %, occur with *other* buildings, but 10.2 % are confused with *residential* buildings. *Other* buildings are correctly predicted in 63.3 % and show the highest confusion with *commercial* buildings (23.2 %). 13.4 % of *other* buildings are wrongly predicted as *residential* ones. *Residential* buildings show the highest recall of all classes with 84.0 %. The highest confusion occurs with *commercial* buildings with 9.1 % and 6.9. % wrongly predicted as *other* buildings.

Overall, the aerial model with a VGG16 architecture can predict aerial building images with an accuracy of 73.8 %. The following analysis applies this model to aerial images of the buildings identified in Chapter 6.

8.2.3 *Predictions on Buildings from Social Media Image Approach*

Additionally, this VGG16 model is evaluated on the aerial images of the buildings identified in Chapter 6. Hence, for each building, a corresponding aerial image was downloaded. There is no overlap between the buildings in the training set and those from the social media image part, and all 29,867 building images are used for this analysis. Figure 8.4 shows the confusion matrix of the VGG16 model predicting these images. The results are similar to the ones from the

	Metric	F1-score	Precision	Recall
Model	Class			
DenseNet121	Commercial	0.678	0.694	0.663
	Other	0.669	0.662	0.676
	Residential	0.772	0.763	0.781
	Weighted	0.706	0.706	0.707
InceptionResNetv2	Commercial	0.687	**0.707**	0.669
	Other	0.681	0.674	**0.689**
	Residential	0.793	**0.781**	0.805
	Weighted	0.721	0.721	0.721
Inceptionv3	Commercial	0.695	0.700	0.690
	Other	0.678	0.691	0.665
	Residential	0.783	0.764	0.804
	Weighted	0.719	0.718	0.720
MobileNetv2	Commercial	0.680	0.678	0.681
	Other	0.652	0.678	0.627
	Residential	0.768	0.743	0.796
	Weighted	0.700	0.700	0.701
ResNet50v2	Commercial	0.653	0.648	0.658
	Other	0.631	0.639	0.622
	Residential	0.747	0.742	0.751
	Weighted	0.677	0.676	0.677
VGG16	Commercial	**0.717**	0.694	**0.742**
	Other	**0.683**	**0.741**	0.633
	Residential	**0.809**	0.780	**0.840**
	Weighted	**0.736**	**0.738**	**0.738**
Xception	Commercial	0.674	0.662	0.686
	Other	0.654	0.656	0.653
	Residential	0.761	0.774	0.748
	Weighted	0.696	0.697	0.696

Table 8.1: Prediction results for aerial images on So2Sat balanced data

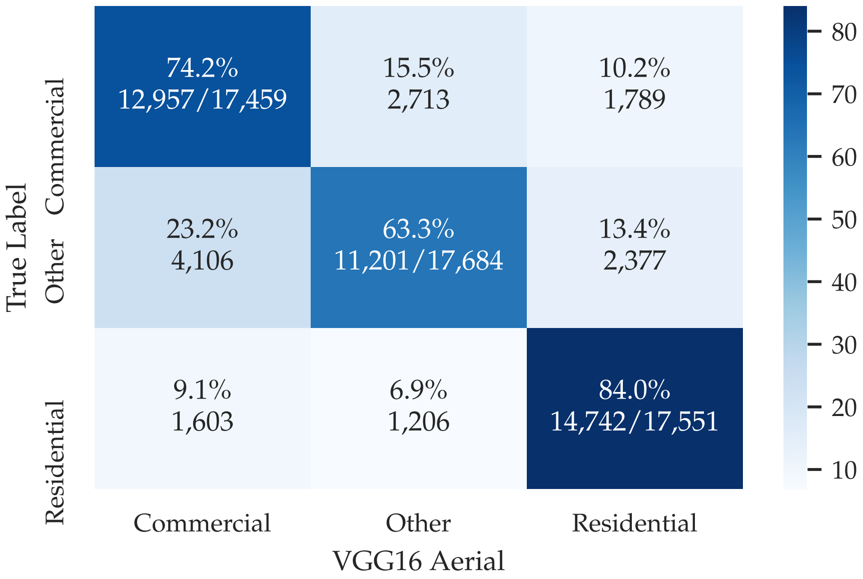


Figure 8.3: Confusion matrix of VGG16 architecture fine-tuned on So2Sat aerial image dataset

test set: *commercial* buildings are predicted with 74.4 % recall, which is an increase of 0.2 %. For *other* buildings the value raises from 63.3 % to 66.4 %. However, *residential* buildings show a decrease by 5.5 % from 84.0 % to 78.5 %. The relative numbers of wrong predictions are also comparable to the previous results. If the dataset is limited to buildings with human-verified social media images, the findings are confirmed as well (Figure A.7).

A look at the prediction performance at a city level shows that the results are stable across cultural zones. Figure 8.5 contains the F1 scores for each class and city. This analysis focuses on cities with more than five samples per class and considers all other outliers. For *commercial* buildings, Los Angeles shows the best results with an F1 score of 0.882 on 1,922 buildings. As most training samples are also from this city, this finding is intuitive. On the second and third ranks, there are Beijing and Jakarta, with F1 scores of 0.79 on 60 samples and 0.794 on 33 samples. The three cities with the lowest scores are Amsterdam (0.506 on 538 samples), Lisbon (0.471 on 68 samples), and Rome (0.364 on 41 samples). However, for *other* buildings, Rome yields the second-highest F1 score of 0.851 on 231 samples, behind Nanjing with 0.933 on eight samples. On the third rank for *other* buildings is Cairo, with an F1 score of 0.833 on 20 samples. Both Cairo and Rome are well known for their historical sites leading to a high number of *other* buildings with distinct shapes and patterns from an aerial view. On the lowest ranks are Shenzhen (0.536 on 27 samples), Amsterdam (0.455 on 246 samples), and Mumbai (0.222 on six samples). *Residential* buildings are best predicted in Vancouver with 0.883 on 348 samples,

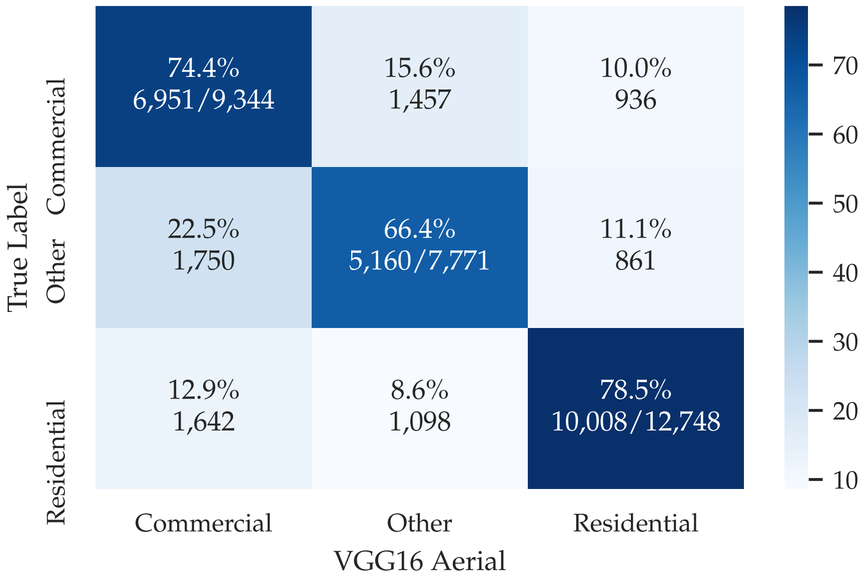


Figure 8.4: Confusion matrix of VGG16 architecture fine-tuned on Flickr image dataset

Los Angeles with 0.879 on 1,903 samples, and Berlin with 0.877 on 1,445 samples. The worst results for *residential* buildings are from Tokyo (0.641 on 170 samples), Sao Paulo (0.615 on 21 samples), and Kyoto (0.500 on 85 samples).

Amsterdam appears two times with comparably low numbers for *commercial* and *other* buildings. In both cases, the low F1 scores result from low precision. Since Amsterdam has a high number of labeled buildings (c.f. Figure 4.2), this finding stands out. Compared to all other cities, Amsterdam has a high number of labeled *residential* buildings, 2,010, ranking second after London with 2,373 *residential* buildings. However, the number of identified *commercial* and *other* buildings is relatively low (784) compared to the number of *residential* buildings (2,010). Since 407 *residential* buildings are wrongly presented as *commercial* or *other*, this has a low impact on the recall of the *residential* class but a severe impact on the precision of the other two classes. In these cases, approximately half of the samples are actually *residential* and thus, decrease the precision value. Hence, the low values are numerical artifacts from a highly imbalanced distribution. A scattered occurrence of urban green spaces in Amsterdam likely reinforces this effect [207].

Nevertheless, the overall results from aerial models are better than any other modality. While social media images yielded a top-weighted F1 score of 0.463 (Figure 6.16) and the metadata approach showed a top-weighted F1 score of 0.557 (Figure 7.6), the aerial model outperforms them with a top F1 score of 0.741. Additionally, Chapter 6

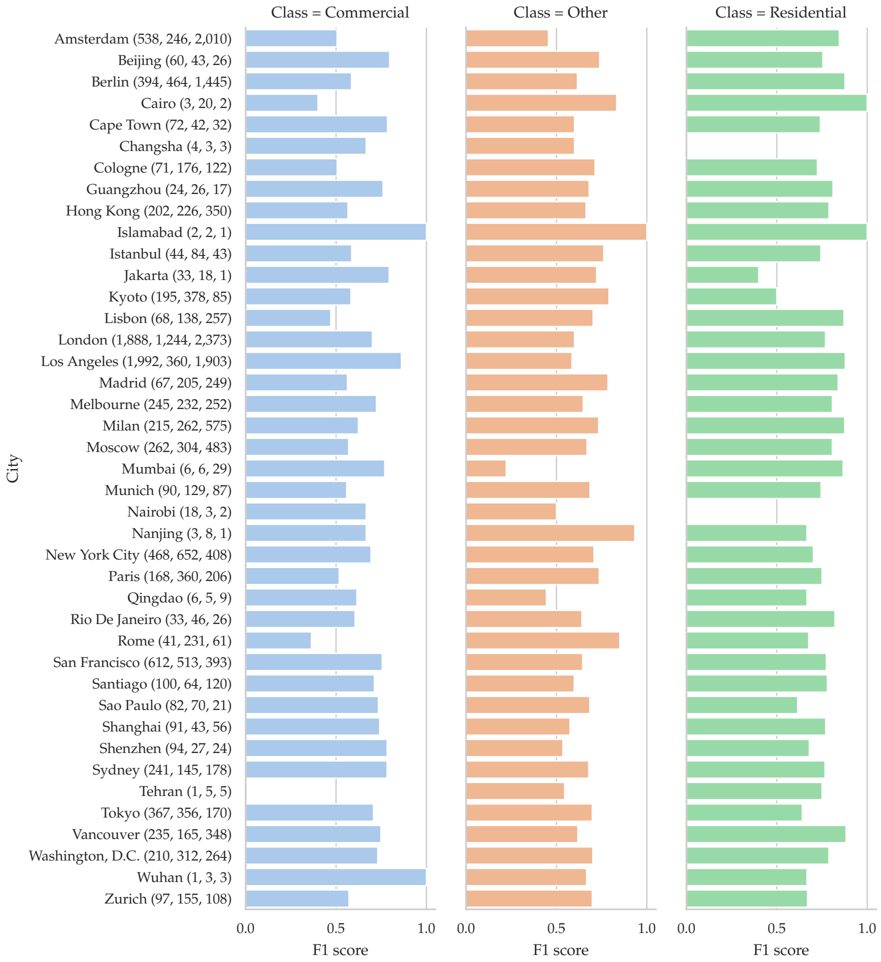


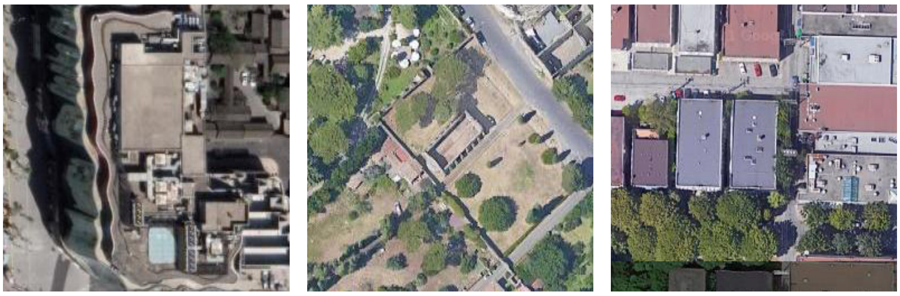
Figure 8.5: F1 scores of building predictions in LCZ42 cities on a class-wise level for VGG16 model trained on the So2Sat aerial image dataset. The numbers in brackets behind the region names indicate the number of buildings in the same order as the plot columns.

revealed that OSM labels have an accuracy of 69 %. Hence, the results of the aerial model are better than the label quality.

A probable explanation for this behavior is the core principle of machine learning. The algorithms are generic function approximators without any prior knowledge of a task. They map patterns in the features to labels they have seen before. The clearer the patterns are, the better the prediction results. In the case of social media images, the patterns are fuzzy. Thus they predict the correct class only in obvious cases. As the aerial images have more evident visual patterns than the social media images or the GSV images, the models show a better performance.

Figure 8.6 illustrates this with examples of correctly predicted aerial images. Figure 8.6a shows a shopping mall in Beijing, which is not taken from a perfect nadir position but contains parts of the façade. Moreover, it has a darker appearance than the other two examples.

Figure 8.6b depicts the ruins of a church in Rome, Chiesa di San Nicola a Capo di Bove, as an example of an *other* building outside of a dense urban area surrounded by trees and dry grass. Overall, the contrast is lower than in the other two examples. Figure 8.6c shows an example of a *residential* building in Vancouver with high contrast and a perfect nadir perspective.



(a) Google aerial image for *commercial* OSM building 952257478 in Beijing, China

(b) Google aerial image for *other* OSM building 607234690 in Rome, Italy

(c) Google aerial image for *residential* OSM building 656482824 in Vancouver, Canada

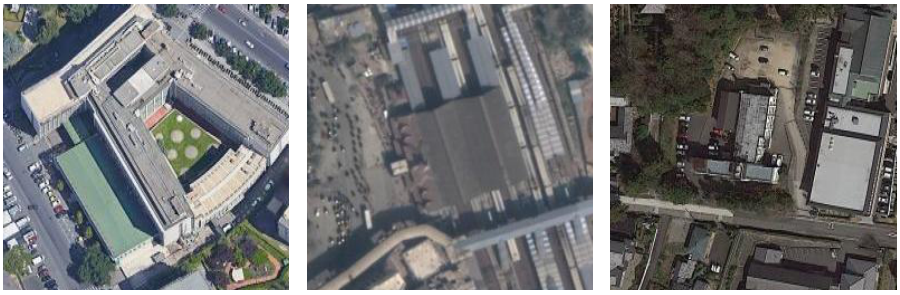
Figure 8.6: Examples for Google aerial images correctly predicted by the VGG16 model. Imagery © Google

In contrast to these correctly predicted examples, Figure 8.7 provides samples of images that are misclassified. The *commercial* building in Figure 8.7a is predicted as *other* most likely because of its unique shape. It serves as the headquarters of Acea SpA, Italy's main national operator of water infrastructure. Figure 8.7b depicts the main building of Bandra Railway Station in Mumbai, an *other* building predicted as *commercial*. The image quality is lower than in all other samples, with less sharpness and lower contrast. However, the water supplier in Rome and the train station in Mumbai are borderline cases, which could be arguable in the ground truth class and the predicted class. Figure 8.7c is an example of a *residential* building in Kyoto. As it is surrounded by several cars in parking lots, the prediction is reasonable for a human expert.

Generally, the predictions of the VGG16 model are sensible, and it performs well across different cultures and on varying image qualities. It is robust to nadir and close-nadir perspectives, contrast, and sharpness.

8.3 SUMMARY

This chapter introduced a baseline for predicting building functions using high-resolution aerial imagery. It presented an undersampling algorithm to create perfectly balanced datasets concerning prediction classes and maintain the spatial distribution. By re-using the fine-tuning approach from Chapter 5 deep learning models for predicting



(a) Google aerial image for *commercial* OSM building 7209745 in Rome, Italy, predicted as *other*

(b) Google aerial image for *other* OSM building 1538393938 in Mumbai, India, predicted as *commercial*

(c) Google aerial image for *residential* OSM building 720173914 in Kyoto, Japan, predicted as *commercial*

Figure 8.7: Examples for Google aerial images wrongly predicted by the VGG16 model. Imagery © Google

building functions were created and evaluated. The evaluation showed that the VGG16 architecture best suits the given task and gains comparable results for all three classes. Analyzing the performance of the independent set of buildings from the social media image pipeline revealed that the predictions are robust and work across cultural zones and varying image qualities. Generally, the results from the aerial approach outperform all previous methods using GSV images, social media images, or social media metadata because the signal-to-noise ratio is higher in the social media-based approaches. Nevertheless, all models have their strengths and weaknesses, which might complement each other. The next chapter examines if combining different modalities can improve overall performance.

9 FUSION STRATEGIES

The previous four chapters introduced different data sources and approaches to predict building functions. As this thesis focuses on social media data rather than GSV data, this chapter uses the results from Chapter 6 and Chapter 7 and analyses the opportunities to combine the strengths of them with the remote sensing method from Chapter 8.

Social media images, social media metadata, and aerial imagery are different data types, also referred to as modalities. The literature summarizes different modalities for a common goal as multimodal fusion. Most state-of-the-art approaches for multimodal fusion implicitly assume a one-to-one relationship between the modalities [208] or create one [209]. Additionally, deep learning methods with end-to-end learning require a common architecture that allows gradient-based learning. However, the social media image approach yields a $1 : n$ relationship between buildings and images as landmarks and touristic hotspots are popular motifs. Chapter 7 showed that gradient-boosted trees are better suited for structured metadata features. The XGBoost algorithm is not compatible with a CNN architecture to be trained at once end-to-end. Hence, deep multimodal architectures are not directly applicable to the given task—however, all prediction algorithms yield probability vectors, which opens the space for decision-level fusion strategies. Last but not least, different studies showed the effectiveness of decision-level fusion over feature-level fusion [20, 145]. Since the evaluation with human-verified labels revealed severe label noise, this chapter focuses on the subset of buildings that have a clear function, which experts have confirmed in Subsection 6.2.9.

9.1 DECISION-LEVEL FUSION STRATEGIES

All fine-tuned architectures have a dense layer with softmax activation to indicate the final prediction result. The output of a softmax layer can be interpreted as a probability vector; however, it needs to be treated with care as the probabilities are not well calibrated [210]. The probability vectors do not represent actual probabilities but are overconfident as the models are trained using one-hot encodings and not with actual probabilities. Gradient boosted trees are also affected by this issue [211]. Nevertheless, the prediction vector reflects this uncertainty with more equally distributed values, especially in fuzzy cases. If a second modality has a more confident prediction, this can be decisive for the final prediction. The following paragraph presents four functions to combine prediction vectors:

1. Average fusion
2. Prediction-entropy weighted average fusion
3. Model-entropy weighted average fusion
4. Minimum prediction-entropy fusion

Let P_b be a set of prediction vectors $P_b = \{p_1, \dots, p_n\}, p \in \mathbb{R}^x, n \in \mathbb{N}$ for a building b with n denoting the number of probability vectors and x as the number of classes to predict. The set of all buildings is denoted as B.

The average fusion is defined as

$$c_p = \operatorname{argmax} \frac{1}{|P_b|} \sum_{p \in P_b} p \tag{9.1}$$

with an *argmax* function that transforms the average probability vector to a class index. As the *argmax* returns the index of the vector element with the highest value, the normalization with the cardinality of P_b has an illustrational purpose.

The prediction-entropy weighted average fusion is defined as

$$c_p = \operatorname{argmax} \sum_{p \in P} H_S(p)^{-1} p \tag{9.2}$$

and similar to Equation 9.1: each probability vector is weighted using with $H_S(p)$ as the Shannon entropy in Equation 7.4. By weighting inversely with the entropy it emphasizes predictions with higher confidence.

The model-entropy weighted average fusion is defined as

$$c_p = \operatorname{argmax} \sum_{p \in P} H'_S(p)^{-1} p \tag{9.3}$$

with

$$H'_S(p_x) = \frac{1}{|B|} \sum_{p_x \in P_b, b \in B} H_S(p_x)$$

In contrast to Equation 9.2, it weights prediction vectors using the model entropy rather than with individual prediction entropies. The model entropy H'_S is calculated as the average entropy of all predictions of one model.

The minimum entropy fusion is defined as

$$c_p = \operatorname{argmax} \operatorname*{argmin}_{p \in P} \{H_S(p)\} \tag{9.4}$$

and considers only the prediction with the lowest entropy.

As all functions are based on set operations, they can handle an arbitrary number of predictions individually for each building. The functions are evaluated on two fusion sets: one with social media images and aerial data and one with metadata and aerial data.

9.2 COMBINING SOCIAL MEDIA AND AERIAL IMAGES

This evaluation focuses on the dataset of 522 images with human-verified labels from Subsection 6.2.9. There is a one-to-one relation between images and buildings in this subset, so there is exactly one image per building. Table 9.1 shows the results for the single modalities in the first two blocks, followed by the results of the four fusion methods, and finalized by a block with the theoretically optimal fusion result. The optimal fusion results are based on the assumption that one would know which model is correct for a sample and pick the corresponding prediction. If both are wrong or correct, the choice would be random, making no difference.

For *other* and *residential* buildings, the basic average method outperforms the single modalities as well as the other fusion methods. For example, the F1 score for *residential* buildings is 0.891, which is 6 % higher than the aerial F1 score and 20 % higher than the F1 score using only social media images. However, *commercial* buildings show a higher F1 score when predicted using the prediction-entropy weighted average with 0.839 compared to 0.835 from the plain average fusion. Two examples of a higher score from other methods are the precision of *commercial* buildings and the recall of *residential* buildings when using the model-entropy weighted average. Generally, the differences between the average fusion methods are negligible, with changes after the second digit. The minimum entropy fusion is always higher than the single modality results but does not reach the average-based performances. Using the basic average fusion improves the weighted F1 score of the aerial predictions by 6.9 % and the social media images results by 10.9 %. Nevertheless, the optimal fusion shows that there is still room for improvement: The weighted F1 score of the average fusion is 0.863, which is 6.6 % below the theoretically best possible weighted F1 score of 0.924.

The confusion matrix in Figure 9.1 shows the results of the average fusion. Compared to the results of the single modalities (Figure A.4 and Figure A.7) the *other* and *residential* classes benefit from the fusion, while the score for *commercial* stays at 90.7 % as it is with pure social media images. The prediction of *other* buildings raises from 78.4 % in social media images and 75.8 % in aerial images to 82.5 % using average fusion. The same applies to *residential* buildings that are correctly predicted with social media in 65.1 % and aerial in 83.7 % to 86.1 % with fusion.

	Metric	F1	Precision	Recall
Method	Class			
Aerial	Commercial	0.780	0.734	0.833
	Other	0.803	0.855	0.758
	Residential	0.837	0.837	0.837
	Weighted	0.807	0.812	0.807
Social Media Images	Commercial	0.788	0.697	0.907
	Other	0.798	0.813	0.784
	Residential	0.745	0.871	0.651
	Weighted	0.778	0.795	0.780
Average	Commercial	0.835	0.774	0.907
	Other	**0.863**	**0.904**	**0.825**
	Residential	**0.891**	**0.923**	0.861
	Weighted	**0.863**	**0.869**	**0.862**
Prediction-entropy weighted average	Commercial	**0.839**	0.775	**0.914**
	Other	0.859	0.903	0.820
	Residential	0.885	0.916	0.855
	Weighted	0.861	0.868	0.860
Model-entropy weighted average	Commercial	0.832	**0.783**	0.889
	Other	0.851	0.897	0.809
	Residential	0.875	0.883	**0.867**
	Weighted	0.853	0.857	0.852
Minimum entropy	Commercial	0.815	0.747	0.895
	Other	0.836	0.890	0.789
	Residential	0.888	0.917	0.861
	Weighted	0.846	0.854	0.845
Optimal	Commercial	0.905	0.869	0.944
	Other	0.926	0.951	0.902
	Residential	0.939	0.951	0.928
	Weighted	0.924	0.926	0.923

Table 9.1: Individual prediction results of aerial and social media image approach in the first two blocks, results of fusion methods in the subsequent four blocks, and theoretically optimal fusion results in the last block

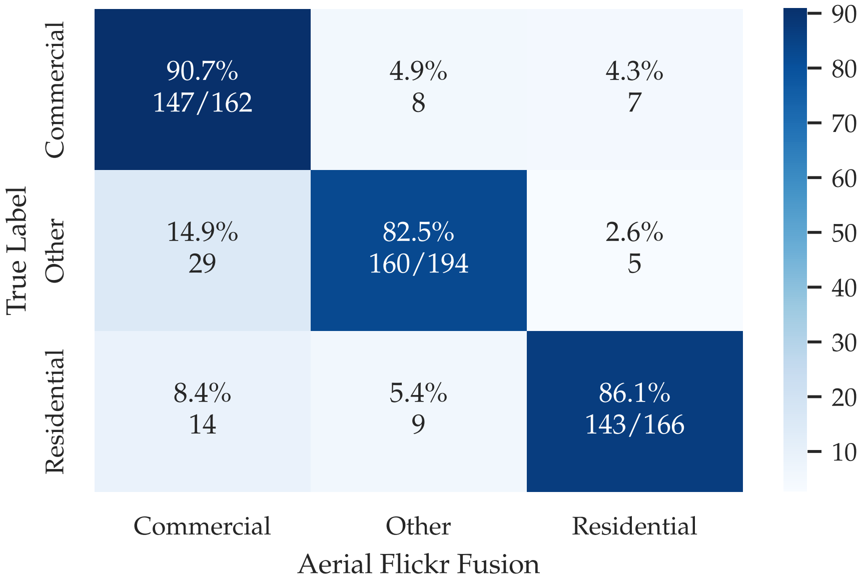


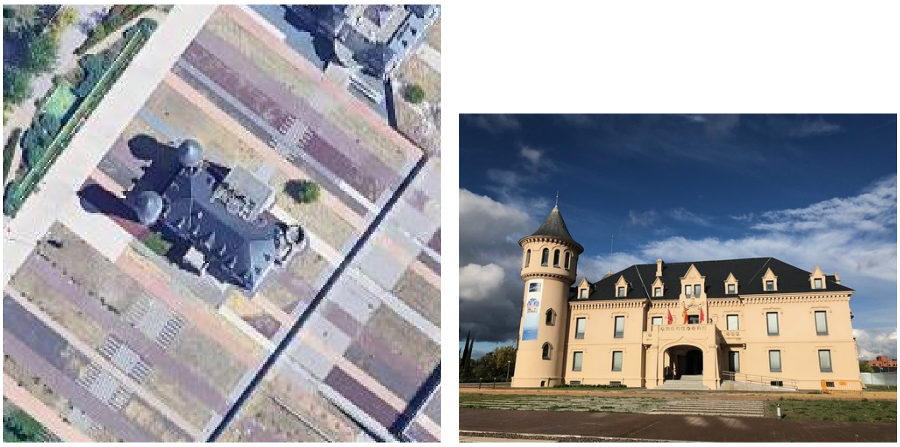
Figure 9.1: Confusion matrix of average fusion of predictions from aerial images and social media images based on buildings with human-verified labels.

Modality		Social media image prediction					
		Wrong			Correct		
		% FW	% FC	#Spl	% FW	% FC	#Spl
Aerial pred.	Wrong	97.5 %	2.5 %	40	36.1 %	63.9 %	61
	Correct	14.7 %	85.3 %	75	0.0 %	100.0 %	346

Table 9.2: Contribution analysis of aerial and social media image modalities to average fusion. FW is for **F**usion **W**rong, FC abbreviates **F**usion **C**orrect. These values are relative to the number of samples (#Spl) for each modality.

Table 9.2 allows an in-depth analysis of the contribution of each modality to the final average fusion result. It shows the relative number of samples for which each modality is wrong and correct and how this influences the average fusion result. For example, both modalities are wrong in 40 cases, but for one case (2.5 %), the average of both prediction vectors yields the correct prediction. Figure 9.2 depicts the two images for this special example. The social media image prediction is correct for 61 buildings when the aerial prediction is wrong. Averaging these predictions leads to 39 correct predictions (63.9 %). On the other hand, the aerial prediction is correct on 75 buildings, whereas the social media image prediction is wrong. The final average prediction is 85.3 % correct, which indicates that the prediction vectors of the aerial model have lower entropy than the social media model. The correct social media prediction leads only to 63.9 % correct fusion results. Finally, the fusion is always correct if both predictions are correct (346 buildings).

There is one example in which both individual predictions are wrong, but the fusion prediction is correct. How can this happen? Figure 9.2 shows the two images. The aerial VGG16 model predicts 0.491 *commercial*, 0.440 *other*, and 0.068 *residential*, so the probabilities for *commercial* and *other* are almost the same. However, as *commercial* has a slightly higher value it "wins" the final prediction. The GSV model with an InceptionResNetv2 architecture predicts 0.114 *commercial*, 0.393 *other*, and 0.493 *residential*. Hence, it is unsure if the image shows an *other* or a *residential* building. Fusing both probability vectors using the average yields 0,303 *commercial*, 0.417 *other*, and 0.281 *residential*. Hence, both individual predictions are unsure between the two classes and decide on the wrong one, while neglecting the second-ranked correct prediction. As the uncertainty is different for each modality, the fusion averages them out and emphasizes the second-ranked *other* class.



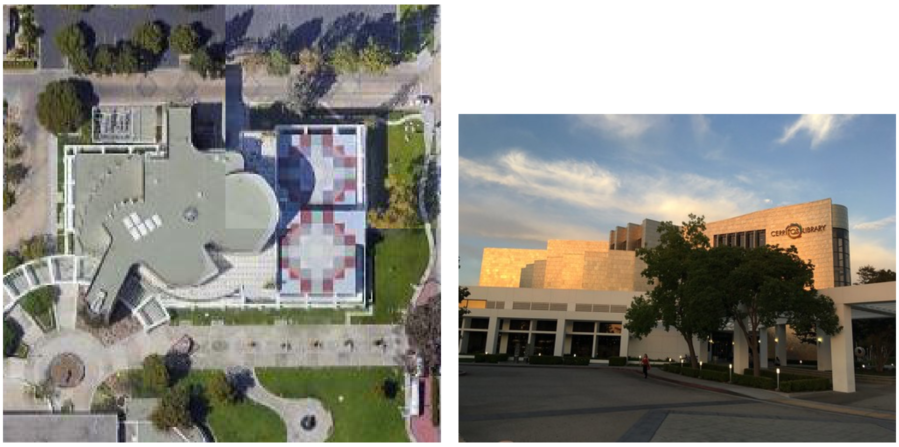
(a) Google aerial image. Imagery © Google

(b) Flickr image. Photo ©Los Castillos by Luz D. Montero Espuela

Figure 9.2: Example of both modalities, aerial and social media image, being wrong but fusion is correct with *other* OSM building 386627182, Museo Municipal de Arte en Vidrio (MAVA), Madrid, Spain

Figure 9.3 illustrates a fusion example, for which the aerial prediction is correct, but the social media model fails. The aerial VGG16 model outputs a distinctive prediction of 0.028 *commercial*, 0.970 *other*, and 0.002 *residential* for Figure 9.3a. However, based on Figure 9.3b the InceptionResNetv2 model tends to *commercial* with 0.485, *other* with 0.355, and *residential* with 0.160. The clear prediction from the aerial model contributes most to the fusion results and vanishes the uncertain prediction from the GSV model.

As Table 9.2 shows, social media images can help if the aerial model is wrong. Figure 9.4 depicts such an example. The aerial image

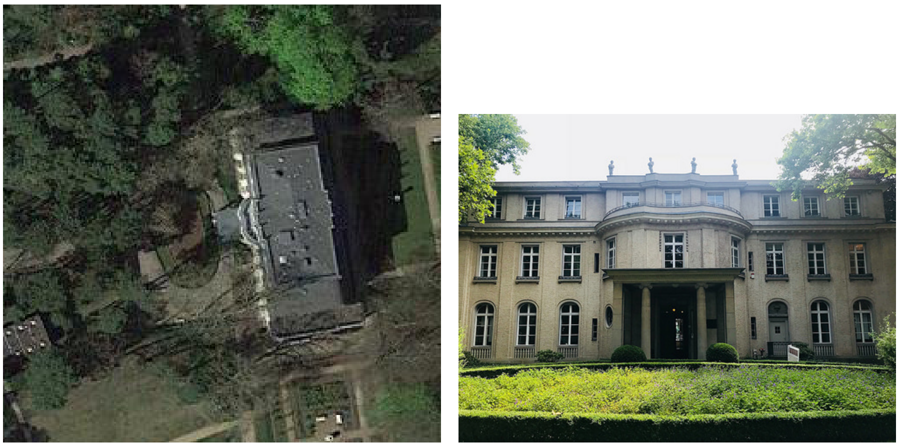


(a) Google aerial image. Imagery © Google

(b) Flickr image. Photo Cerritos by Sergei Gussev is licensed under CC BY 2.0

Figure 9.3: Example of correct aerial prediction and wrong social media image prediction *other* OSM building 699832958, Cerritos Public Library, Los Angeles, USA

from Figure 9.4a is predicted as 0.307 *commercial*, 0.269 *other*, and 0.426 *residential*. Thus, it is classified as *residential*. The prediction based on the social media image Figure 9.4b yields 0.156 *commercial*, 0.528 *other*, and 0.316 *residential*. Although the fusion results are close between *other* with 0,399 and *residential* with 0.371, it is finally correctly predicted as *other*.



(a) Google aerial image. Imagery © Google

(b) Flickr image. Photo ©iPhone 7 by Håkan Dahlström

Figure 9.4: Example of wrong aerial prediction and correct social media image prediction *other* OSM building 690278732, House of the Wannsee Conference, Berlin, Germany

This analysis shows that the two modalities, aerial images, and social media images, are complementary when predicting building functions. A simple average of the prediction probability vectors from both models improves the classification performance to a weighted F1 score of 0.863. Analyzing the contributions of both modalities to the fusion revealed that it is in every case beneficial: the performance is always better than with a single modality. Even two wrong individual predictions can be corrected by fusing the vectors in exceptional cases. However, there is still room for improvement, as the theoretically optimal fusion result shows. Moreover, the aggregation of predictions from multiple social media images provides opportunities for further investigations. For example, the set of social media images for a building can consist of several hundred samples. In this case, the influence of the aerial image would fade out compared to the social media image. A straightforward solution for this issue could be a two-step aggregation: first, fusing all social media image predictions, and second, a fusion of the aggregated social media prediction with the aerial image. A more sophisticated approach could involve clustering social images based on their content to aggregate based on similarity or distance to the building. Last but not least, an end-to-end deep learning architecture could be a two-stream network that processes the aerial image in one stream and the social media images in a second stream. At the same time, a recurrent layer or an attention layer combines multiple social media images. However, due to the large amount of training data needed, a feasibility study should investigate such an approach.

The following section has the advantage that the n:1 relation between social media data and buildings has already been resolved during the feature calculation step. It focuses on the combination of gradient-boosted tree models and CNN-based models.

9.3 COMBINING METADATA AND AERIAL IMAGES

The dataset from Section 7.3 contains features from all buildings that appear in the filtered social media images but not in the So2Sat metadata training set. For a fair comparison with the previous section, the buildings of this dataset are filtered again based on the set of buildings with human-verified labels. This results in a test set of 264 buildings with 70 *commercial*, 77 *other*, and 117 *residential* samples.

Table 9.3 shows the results analogous to the previous section. The first two blocks contain individual modalities, aerial images, and social media metadata. They are followed by the results of the four fusion methods, and in the last block, there are the theoretically optimal fusion results. Overall, the results from the aerial model are not improved by a fusion method in most cases. Two notable exceptions are the model-entropy weighted average fusion and the minimum entropy fusion, which achieve higher results on the precision of the *other* class and the recall of the *commercial* and *residential* class. In the first case, the precision of *other* buildings increases from 0.868 (aerial) and 0.824 (metadata) to 0.909. The recall of *commercial* buildings raises to 0.871

after fusing aerial predictions (0.829) and metadata predictions (0.800) with the model-entropy weighted average. A similar pattern occurs for the recall of the *residential* class: Aerial predictions with a recall of 0.897 and metadata predictions with a recall of 0.761 are combined with a model-weighted entropy average to a recall of 0.915. Generally, the fusion methods yield better results if the individual modalities perform at a similar level with comparable scores. Nevertheless, the optimal scores in the last table block reveal room for improvement for each class and evaluation metric. For example, the optimal weighted F1 score is 0.928, 0.087 better than the best-performing standalone aerial model.

Table 9.4 provides deeper insights into how both modalities contribute to the results of average fusion. If both predictions are wrong, then the fusion is also wrong. If the metadata prediction is correct and the aerial prediction is wrong, the fusion is correct in 14 out of 23 cases. Vice versa, if the aerial model is right and the metadata is wrong, then 51 of 86 samples are correctly predicted using the average fusion. The missing 35 samples, for which the aerial model is correct, but the metadata is wrong, make the difference between the best performing, single aerial modal and the fusion models with the lower performance. This difference becomes evident as the fusion model is always correct if both modalities are correct.

Two examples illustrate the cases of one modality being correct and one failing: Figure 9.5 and Figure 9.6. In the first case, the aerial image, Figure 9.5a is correctly predicted as *other* with 0.970, whereas the metadata model classifies it as *commercial* with 0.648, as *other* with 0.147, and as *residential* with 0.342. Figure 9.5b depicts the spatial distribution of 286 tweets around the building. They are clustered in 17 locations around the building without a clear pattern. The same applies to Figure 9.6b, which is an example of the metadata model being correct. The hotel is predicted as *commercial* with 0.725, as *other* with 0.112, and as *residential* with 0.162. Based on Figure 9.6a the aerial model classifies the building as *other* with a probability of 0.743, whereas the *commercial* class gains 0.235 and the *residential* class gets a probability of 0.027.

Generally, no fusion method in this thesis is suited for combining metadata and aerial prediction vectors. The analysis shows that the prediction performance must of both modalities be on a similar level to gain a benefit. The fusion will yield no profit from a combination if one is weaker.

9.4 SUMMARY

The modalities in this thesis are different in terms of their methods and relation to buildings. Therefore, an abstract method for fusion is needed. This chapter introduced four methods to combine prediction

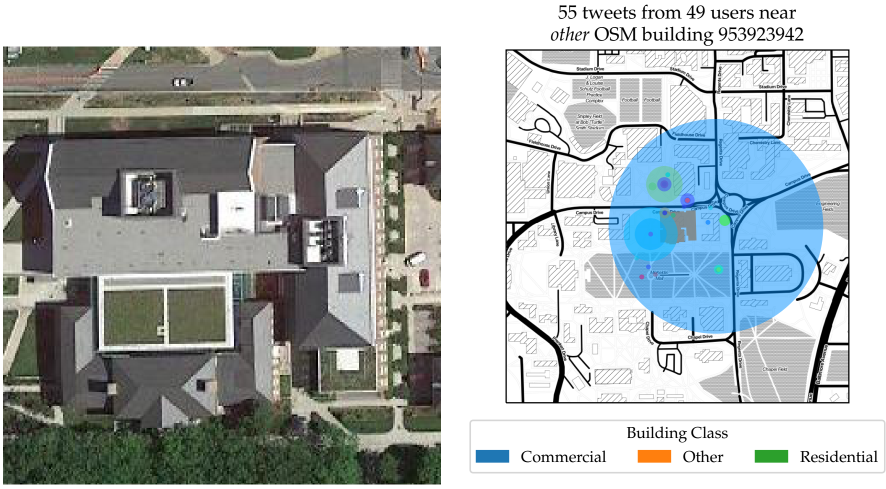
	Metric	F1	Precision	Recall
Method	Class			
Aerial	Commercial	**0.784**	**0.744**	0.829
	Other	**0.814**	0.868	**0.766**
	Residential	**0.894**	**0.890**	0.897
	Weighted	**0.841**	**0.845**	**0.841**
Social Media Metadata	Commercial	0.583	0.459	0.800
	Other	0.298	0.824	0.182
	Residential	0.736	0.712	0.761
	Weighted	0.568	0.677	0.602
Average	Commercial	0.732	0.638	0.857
	Other	0.598	0.875	0.455
	Residential	0.858	0.815	0.906
	Weighted	0.749	0.786	0.761
Prediction-entropy weighted average	Commercial	0.736	0.645	0.857
	Other	0.615	0.900	0.468
	Residential	0.863	0.817	0.915
	Weighted	0.757	0.796	0.769
Model-entropy weighted average	Commercial	0.758	0.670	**0.871**
	Other	0.661	**0.909**	0.519
	Residential	0.870	0.829	**0.915**
	Weighted	0.779	0.810	0.788
Minimum entropy	Commercial	0.753	0.663	**0.871**
	Other	0.661	**0.909**	0.519
	Residential	0.873	0.836	**0.915**
	Weighted	0.780	0.811	0.788
Optimal	Commercial	0.907	0.850	0.971
	Other	0.901	0.985	0.831
	Residential	0.958	0.950	0.966
	Weighted	0.928	0.933	0.928

Table 9.3: Individual prediction results of aerial and social media metadata approach in the first two blocks, results of fusion methods in the subsequent four blocks, and theoretically optimal fusion results in the last block

Modality		Metadata predictions					
		Wrong			Correct		
		% FW	% FC	#Spl	% FW	% FC	#Spl
Aerial Pred.	Wrong	100.0 %	0.0 %	19	39.1 %	60.9 %	23
	Correct	40.7 %	59.3 %	86	0.0 %	100.0 %	136

Table 9.4: Contribution analysis of aerial and metadata modalities to average fusion. FW is for **F**usion **W**rong, FC abbreviates **F**usion **C**orrect. These values are relative to the number of samples (#Spl) for each modality.

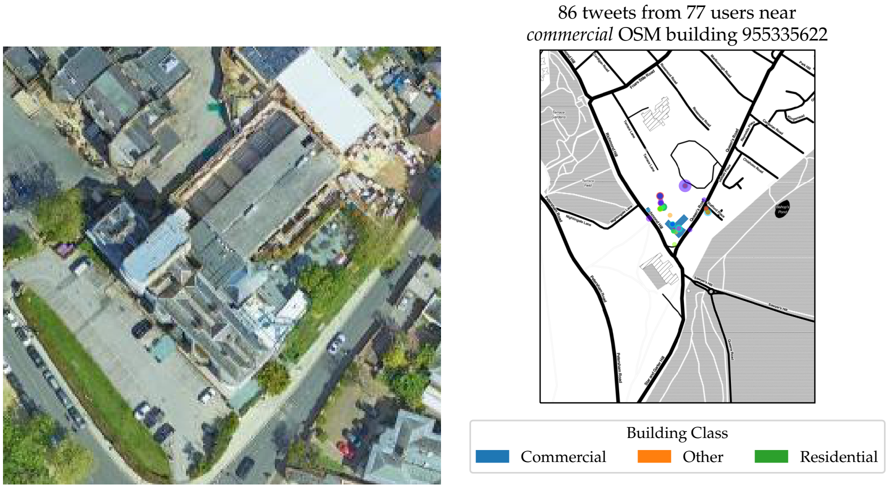
probability vectors from different methods. This decision-level fusion has the advantage of requiring no training or setup, handling n:1 relations between predictions and buildings, and combining different methods. If applied to social media images and aerial images, the basic average fusion is effective on the subset of buildings with human-verified labels. It improves the weighted F1 score of the aerial predictions by 6.9 % and the social media image results by 10.9 %. The additional weighting of the predictions does not yield any improvement. Combining metadata and aerial images does not improve the performance because the metadata predictions were much weaker than the aerial predictions. Nevertheless, the theoretically optimal fusion result reveals room for improvement with both combinations. A pilot study of this thesis investigated different fusion methods, including deep feature level fusion and model stacking if there is a 1:1 relation between ground-level image and aerial image [20]. It concludes that average fusion is the best approach if the different modalities share no common features and are different in any aspect. Hence, further research could develop multimodal networks that cope with different modalities. Approaches like CLIP [212], DALL·E [213], or MAGMA [214] that use text for better image classification or generate images from the text are promising points for going forward to multimodal fusion.



(a) Google aerial image. Imagery © Google

(b) Spatial distribution of tweets. Colors indicate users, marker sizes indicate the number of tweets per user and location. Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Figure 9.5: Example of correct aerial prediction and wrong social media metadata prediction *other* OSM building 699832958, Edward St. John Learning and Teaching Center, Washington, D.C., USA


86 tweets from 77 users near
commercial OSM building 955335622
Building Class
Commercial
Other
Residential

(a) Google aerial image. Imagery © Google

(b) Spatial distribution of tweets. Colors indicate users, marker sizes indicate the number of tweets per user and location. Background map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Figure 9.6: Example of wrong aerial prediction and correct social media metadata prediction *commercial* OSM building 955335622, Richmond Harbour Hotel, Washington, D.C., USA

Part V

CONCLUSION

One of the lessons I have learned in the different stages of my career is that science is not done alone. It is through talking with others and sharing that progress is made.

— CAROL W. GREIDER

10 CONCLUSION

Unprecedented volumes of EO data help to see the world in high spatial and temporal resolution. They show us the world from a different perspective allowing large-scale analysis and improving our understanding of urbanization. Part of this process is the rapid and sometimes uncoordinated construction of new buildings. This chapter briefly reviews the concepts and results of the four automatic methods for building function classification introduced in this work and presents future work that could improve and extend the existing approaches.

10.1 SUMMARY

Building function classification with EO data is challenging: As the task cannot be directly measured, its prediction requires the interpretation of patterns. This thesis aims at developing generalizable methods for the given task and thus, uses a simple yet universal classification scheme of three classes: *commercial*, *other*, and *residential*. A global dump from OpenStreetMap (OSM) forms the basis to extract building footprints and functions for 400 million buildings worldwide. Four different methods for building function classification are presented and evaluated on a culturally diverse dataset of 42 metropolitan areas around the Globe. The four methods rely on different data sources to predict building functions:

1. Fine-tuning of deep learning architectures for image classification with façade imagery from GSV images
2. Filtering street-level imagery from big social media datasets with re-use of the GSV models
3. Using spatio-temporal features from social media metadata to predict building functions
4. Fine-tuning of deep learning architectures for image classification with aerial imagery from Google Maps

The first method establishes a basis for the social media image-based approach. It builds a seed dataset of reference images and creates state-of-the-art prediction models for building function classification based on façade images (Chapter 5). These models are seven fine-tuned deep learning architectures based on ImageNet weights and adapted to the given task with a globally sampled GSV image dataset. The best-performing model with an InceptionResNetv2 architecture yields a weighted F1 score of 0.562. An in-depth analysis on a subregion basis reveals that the model's prediction performance is stable across different cultural zones. The individual examples show that some missed predictions are due to label noise.

However, data from GSV is not free, and hence, other open data sources become interesting. Social media image platforms like Flickr contain massive image data with open licenses. Thus, applying the existing street-level-based models to social media images is evident. A suitable filter algorithm is necessary because social media platforms are not made explicitly for street-level imagery but contain various motifs and different photography styles. This thesis proposes a filtering pipeline that extracts GSV-like images from extensive collections and maps them to buildings in Chapter 6. The pipeline uses the GSV image set from the previous method as a seed dataset and finds all social media images that show similar motifs using feature vectors from CNNs. It combines spatial heuristics and geoinformatics methods to create direct relationships between images and buildings.

The evaluation of the filtering pipeline shows that out of 28 million images, 0.34 % can be mapped to an OSM building. However, as 65.6 % of these buildings are unlabeled, 56,857 OSM buildings could profit from an additional semantic tag. A more detailed analysis of the prediction results confirms the findings of the previous chapter: The InceptionResNetv2 model yields the best-weighted F1 score of 0.476. Since this number is 0.086 lower than the F1 score of the GSV images, a group of human experts validated a subset of the images and their labels.

This study shows that only 50.4 % of all images show a building with a precise function, and the building labels from OSM are correct in 69 % of the cases. If evaluated on the human-validated subset, the InceptionResNetv2 model yields a weighted F1 score of 0.78. This result demonstrates the vast impact of label noise on prediction performance.

The third method from Chapter 7 presents a different approach that ignores the noisy, user-generated content of social media and focuses on the metadata of social media posts. A building-first approach starts with a building and analyzes the spatial and temporal patterns of social media posts surrounding a building. With an extensive hyperparameter search, two different classification algorithms are trained on 181 features from 385,975 buildings. It turns out that the gradient-boosted tree algorithm outperforms the neural network-based approach by a 0.218 difference. The tree-based method yields a weighted F1 score of 0.723 on the test set compared to a weighted F1 score of 0.505. Other publications confirm that gradient-boosted trees are better suited for structured data than neural networks [45, 201]. An in-depth analysis of the feature importance for the best-performing model reveals two different groups of features: those with a high gain, like the user area density, and those with a high weight, e.g., the mean distance between social media posts and buildings. When applied to the buildings resulting from the image filtering pipeline (Chapter 6), the best-performing tree-based model yields a weighted F1 score of 0.557.

The fourth method for predicting building functions utilizes the same fine-tuning approach as the GSV method but uses aerial images from the So2Sat social media dataset (Chapter 8). As the fine-tuning approach assumes a balanced training dataset, an algorithm for undersampling is introduced that ensures a class-wise balance and equalizes the city-wise sample distribution. The best-performing model for this data is based on a VGG16 architecture with a weighted F1 score of 0.736. A city-wise analysis shows that it generalizes across all LCZ42 cities. The results from the test set are confirmed when the method is applied to the buildings identified with the image filtering pipeline.

Chapter 9 proposes four fusion strategies for decision-level fusion. A pilot study for this thesis found the decision level best suited for fusing multimodal data. It has the additional benefit of seamlessly resolving 1:n relations between images and buildings. The evaluation is performed on the subset of buildings with human-validated functions to eliminate the effect of label noise. The analysis shows that an average mean fusion of the filtered social media images and the aerial images improves the weighted F1 score to 0.863. This fusion method can improve the results if one modality provides a wrong prediction, but it never turns a correct prediction into a wrong one. The fusion of metadata and aerial images does not yield an overall improvement in the results. With the generally superior predictions of the aerial model, the relatively weak metadata predictions do not provide any benefit, except for some exceptional cases.

A general pattern across all methods is the comparably low prediction results of the *other* class. *Commercial* and *residential* buildings have clear patterns either in their façade images, their social media metadata patterns or in their aerial images. As these two classes summerize precise functions, they have more apparent patterns and are easier to predict for machine learning algorithms. The *other* class combines a wide variety of building functions, like schools, hospitals, airports, and stations, resulting in a high inter-class variance. This variance makes it hard for prediction models to detect any distinctive patterns.

10.2 FUTURE WORK

All methods in this thesis have a huge potential for further development and research. As all of them are severely affected by the label noise, this could be one direction of further investigations. Possible options include accepting the noisiness of labels and applying a less rigid scheme with multiple labels, e.g., a building can be *commercial* and *residential* at the same time [215]. Such an approach needs to cope with obtaining ground truth labels: Very few OSM buildings have this fine-grained information. Alternatively, the hierarchical relation between building functions might be exploited. Since the *other* class consists of multiple different subclasses, a hierarchical learning approach could help to distinguish better between subclasses [216]. Moreover, the *commercial* and *residential* classes have the potential to benefit from this method as well. In this case, the balancing between subclasses is a major challenge as there are few samples for rare subclasses, e.g., monasteries.

While the different classes are not equally well recognizable, all models yield a comparable performance across different cultural zones and nationalities. This finding indicates that the CNN models have sufficient capacity to include versatile patterns of single classes. Nonetheless, some studies suggest having local expert models that are trained in specific regions [217]. An investigation of regional models versus a global model poses another opportunity for further research.

However, mapping based on ground-level data risks systematic biases toward society's upper and lower parts. Extremely poor neighborhoods are neither appropriately covered by GSV [169] nor by Twitter [218]. There are different reasons for this finding: First, the commercial power of these communities is low, which reduces the attractivity for commercial platforms. Second, these areas are not accessible with the mapping equipment, and third, a mapping campaign in these areas might put the staff at risk of becoming a victim of crime. Last but not least, the inhabitants of these areas are not users of social media platforms. The other extreme is gated communities of rich people that prevent trespassing. With closed roads for the public, they are inaccessible for every volunteer cartographer. Moreover, as Li *et al.* stated that the user basis of Flickr is "white and Asian people with an advanced degree" ([123]), their neighborhoods are likely overrepresented in the social media image approach of this thesis. The analysis of this aspect poses another research opportunity, especially when the results of these methods are possibly used for policymaking.

Beyond these generally applicable ideas, each method offers several opportunities for further development. Although GSV imagery sampling is aimed at a globally diverse dataset, some types of building façades might be missed because they are unique to some regions. A taxonomy-guided sampling strategy could help identify blind spots and enrich the existing dataset. However, the legal barriers are still existing and need to be respected. Moreover, the analysis of architectures for street-level images in Chapter 5 revealed that other architectures have better performance for some classes. Hence, a combination of multiple architectures with decision level fusion [20] or advanced feature level fusion [219] might improve the overall prediction results.

The filtering pipeline for social media images could be evolved with a more task-specific image retrieval algorithm. Content-based image retrieval (CBIR) is a research field on its own and provides task-specific methods that allow the filtering of more images with higher relevance [220]. An alternative opportunity for further development is considering the camera's focal length. This information allows computing a field of view, and thus, all buildings in an image can be identified. Hence, a building-wise prediction might help to predict multiple buildings with one image. Moreover, façade images are just one type of image that can be related to a building function. With other seed datasets, this pipeline could identify images for activity recognition [221, 222], which is a different way of describing land use.

Concerning the metadata approach, its features are handcrafted. Similar to the evolution in computer vision moving from handcrafted features to learned features, this transformation is also possible for metadata. For example, the occurrences of tweets in a specific time interval could be processed with a CNN for predicting building functions. Such an approach would implicitly encode spatio-temporal features rather than taking the spatial and temporal features separately as in the current status. However, while this method offers an end-to-end training phase, the discretization of the tweets in terms of spatial and temporal resolution requires thorough analysis. In any way, the discretization will result in sparse input data, which requires a tailored CNN architecture.

The aerial image classification is based on single-scene classification. However, most buildings are not solitary but part of a neighborhood that often shares the same function. A segmentation approach has the potential to scale the method to larger areas. For example, the aerial images could be predicted using a semantic segmentation method that returns its predictions together with an uncertainty measure [17]. This approach would show where additional, complementary data from other sources are necessary. In such an environment, social media data might be used in a more targeted manner. Alternatively, this opens the opportunity to switch from passive data capturing to active data collection based on citizen science [223]. For example, this data can be collected by citizens or tourists with a mobile app. If they register for a dedicated data collection platform, they will receive push notifications asking them to take a picture of a given building. In combination with a monetary incentive, this approach allows for closing blind spots and data lack, similar to Amazon Mechanical Turk. Especially in developing countries, an active data collection platform could be a win-win for scientists and citizens.

10.3 OUTLOOK

Urbanization has a tremendous impact on metropolitan areas, especially in the Global South, where little data for planning is available [224]. However, shaping this process according to the SDGs is challenging but necessary. EO data can help to monitor the *status quo*, detect changes, and ultimately help policymakers to understand the dynamics and make data-driven solutions that increase the well-being of citizens [225, 226]. In the best case, this data is free and open as in OSM to enable democratic and informed processes about future development. Although OSM has its deficits, it is still the best platform for sharing geospatial data and has vast potential, especially when combining expert knowledge with algorithmic approaches [227]. With data and algorithms becoming a core foundation of today's world and making decisions for the future [228], their outcome shall be accessible by the public to allow informed discussions.

This thesis tries to contribute a small part to this goal by providing generalizable methods to gather semantic building information. The author of this thesis sincerely hopes that the goals can be fulfilled and would be humbled if some of the results of this thesis help other researchers to bring the SDGs closer to reality.

Part VI

APPENDIX

A

APPENDIX

value	description	class
apartments	A building arranged into individual dwellings, often on separate floors. May also have retail outlets on the ground floor.	residential
bungalow	A single-storey detached small house, Dacha.	residential
cabin	A cabin is a small, roughly built house usually with a wood exterior and typically found in rural areas.	residential
detached	A detached house, a free-standing residential building usually housing a single family.	residential
dormitory	For a shared building, as used by college/university students (not a share room for multiple occupants as implied by the term in British English). Alternatively, use building=residential plus residential=university.	residential
farm	A residential building on a farm (farmhouse). For other buildings see below building=farm_auxiliary, building=barn, . . . If in your country farmhouse looks same as general residential house then you can tag as building=house as well. See also landuse=farmyard for the surrounding area	other
hotel	A building designed with separate rooms available for overnight accommodation. Normally used in conjunction with tourism=hotel for the hotel grounds including recreation areas and parking.	commercial
house	A dwelling unit inhabited by a single household (a family or small group sharing facilities such as a kitchen). Houses forming half of a semi-detached pair, or one of a row of terraced houses, should share at least two nodes with joined neighbours, thereby defining the party wall between the properties.	residential
houseboat	A boat used primarily as a home	residential
residential	A general tag for a building used primarily for residential purposes. Where additional detail is available consider using 'apartments', 'terrace', 'house' or 'detached'.	residential
static_caravan	A mobile home (semi)permanently left on a single site	residential
terrace	A single way used to define the outline of a linear row of residential dwellings, each of which normally has its own entrance, which form a terrace (row-house in North American English). Consider defining each dwelling separately using 'house'.	
commercial	A building where non-specific commercial activities take place, not necessarily an office building. Consider tagging the surrounding area using landuse=commercial. Use 'retail' if the building consists primarily of shops.	commercial
industrial	A building where some industrial process takes place. Use warehouse if the purpose is known to be primarily for storage/distribution. Consider using landuse=industrial for the surrounding area and the proposed industrial=* tag to describe the industrial activity.	commercial
kiosk	A small one-room retail building.	commercial
office	An office building. Use building=office with office=* to describe the type of office. Consider tagging the surrounding area using landuse=commercial. Use 'retail' if the building consists primarily of shops.	commercial
retail	A building primarily used for selling goods that are sold to the public; use shop=* to identify the sort of goods sold or an appropriate amenity=* (pub, cafe, restaurant, etc.). Consider use landuse=retail for the surrounding area.	commercial
supermarket	A building constructed to house a self-service large-area store.	commercial
warehouse	A building primarily used for the storage or goods or as part of a distribution system.	commercial
cathedral	A building that was built as a cathedral. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the cathedral grounds where it is in current use.	other
chapel	A building that was built as a chapel. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the chapel grounds where it is in current use.	other
church	A building that was built as a church. Used in conjunction with amenity=place_of_worship, religion=* , denomination=* and landuse=religious for the church grounds where it is in current use.	other
mosque	A mosque. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the grounds where it is in current use.	other
religious	Unspecific religious building. Prefer more specific values if possible.	other
shrine	A building that was built as a shrine. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the grounds where it is in current use. Small buildings should consider historic=wayside_shrine.	other
synagogue	A building that was built as a synagogue. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the grounds where it is in current use.	other

temple	A building that was built as a temple. Used in conjunction with amenity=place_of_worship, religion=*, denomination=* and landuse=religious for the grounds where it is in current use.	other
bakehouse	A building that was built as a bakehouse (i.e. for baking bread). Often used in conjunction with another node amenity=baking_oven and oven=wood_fired.	other
civic	For any civic amenity, for example amenity=community_centre, amenity=library, amenity=toilets, leisure=sports_centre, leisure=swimming_pool, amenity=townhall etc. Use amenity=* or leisure=* etc. to provide further details. See building=public as well.	other
government	For government buildings in general, including municipal, provincial and divisional secretaries, government agencies and departments, town halls, (regional) parliaments and court houses.	other
hospital	A building which forms part of a hospital. Use amenity=hospital for the hospital grounds.	other
kindergarten	For any generic kindergarten buildings. Buildings for specific uses (sports halls etc.) should be tagged for their purpose. Use amenity=kindergarten for the perimeter of the kindergarten grounds.	other
public	A building constructed as accessible to the general public (a town hall, police station, court house, etc.).	other
school	For any generic school buildings. Buildings for specific uses (sports halls etc.) should be tagged for their purpose. Use amenity=school for the perimeter of the school grounds.	other
toilets	A toilet block.	other
train_station	A building constructed to be a train station building, including buildings that are abandoned and used nowadays for a different purpose.	other
transportation	A building related to public transport. You will probably want to tag it with proper transport related tag as well, such as public_transport=station. Note that there is a special tag for train station buildings - building=train_station.	other
university	A university building. Use amenity=university for the whole university area.	other
barn	An agricultural building used for storage and as a covered workplace.	other
conservatory	A building or room having glass or tarpaulin roofing and walls used as an indoor garden or a sunroom (winter garden).	other
cowshed	A cowshed (cow barn, cow house) is a building for housing cows, usually found on farms.	other
farm_auxiliary	A building on a farm that is not a dwelling (use 'farm' or 'house' for the farm house).	other
greenhouse	A greenhouse is a glass or plastic covered building used to grow plants. Use landuse=greenhouse_horticulture for an area containing multiple greenhouses	other
stable	A stable is a building where horses are kept.	other
sty	A sty (pigsty, pig ark, pig-shed) is a building for raising domestic pigs, usually found on farms.	other
grandstand	The main stand, usually roofed, commanding the best view for spectators at racecourses or sports grounds.	other
pavilion	A sports pavilion usually with changing rooms, storage areas and possibly an space for functions & events. Avoid using this term for other structures called pavilions by architects (see Pavilion)	other
riding_hall	A building that was built as a riding hall.	other
sports_hall	A building that was built as a sports hall.	other
stadium	A building constructed to be a stadium building, including buildings that are abandoned and used nowadays for a different purpose.	other
hangar	A hangar is a building used for the storage of airplanes, helicopters or space-craft. Consider adding aeroway=hangar, when appropriate.	other
hut	A hut is a small and crude shelter. Note that this word has two meanings - it may be synonym of building=shed, it may be a residential building of low quality.	other
shed	A shed is a simple, single-storey structure in a back garden or on an allotment that is used for storage, hobbies, or as a workshop.	other
carport	A carport is a covered structure used to offer limited protection to vehicles, primarily cars, from the elements. Unlike most structures a carport does not have four walls, and usually has one or two.	other
garage	A garage is a building suitable for the storage of one or possibly more motor vehicle or similar. See building=garages for larger shared buildings. For an aircraft garage, see building=hangar.	other
garages	A building that consists of a number of discrete storage spaces for different owners/-tenants. See also building=garage.	other
parking	Structure purpose-built for parking cars.	other
digester	A digester is a bioreactor for the production of inflatable biogas from biomass.	other
service	Service building usually is a small unmanned building with certain machinery (like pumps or transformers).	other
transformer_tower	A transformer tower is a characteristic tall building comprising a distribution transformer and constructed to connect directly to a medium voltage overhead power line. Quite often the power line has since been undergrounded but the building may still serve as a substation. If the building is still in use as a substation it should additionally be tagged as power=substation + substation=minor_distribution.	other
water_tower	A water tower	other
bunker	A hardened military building. Also use military=bunker.	other
bridge	A building used as a bridge. Can also represent a gatehouse for drawbridges. See also bridge=yes for highway=*. Don't use this tag just for marking bridges (their outlines). For such purposes use man_made=bridge.	other

construction	Used for buildings under construction. Use construction=* to hold the value for the completed building.	other
roof	A structure that consists of a roof with open sides, such as a rain shelter, and also gas stations	other
ruins	Frequently used for a house or other building that is abandoned and in poor repair. However, some believe this usage is incorrect, and the tag should only be used for buildings constructed as fake ruins (for example sham ruins in an English landscape garden).	other
yes	Use this value where it is not possible to determine a more specific value.	
user defined	All commonly used values according to Taginfo, generally building types	

Table A.1: Mapping of possible values in OSM tag *building* to the unified classification scheme of this thesis. Value and description are taken from the OSM wiki [229], the class stands for the mapped value in the unified classification scheme.

value	description	class
bbq	BBQ or Barbecue is a permanently built grill for cooking food, which is most typically used outdoors by the public. For example these may be found in city parks or at beaches. Use the tag fuel=* to specify the source of heating, such as fuel=wood;electric;charcoal. For mapping nearby table and chairs, see also the tag tourism=picnic_site. For mapping campfires and firepits, instead use the tag leisure=firepit.	other
biergarten	Biergarten or beer garden is an open-air area where alcoholic beverages along with food is prepared and served. See also the description of the tags amenity=pub;bar;restaurant. A biergarten can commonly be found attached to a beer hall, pub, bar, or restaurant. In this case, you can use biergarten=yes additional to amenity=pub;bar;restaurant.	commercial
cafe	Cafe is generally an informal place that offers casual meals and beverages; typically, the focus is on coffee or tea. Also known as a coffeehouse/shop, bistro or sidewalk cafe. The kind of food served may be mapped with the tags cuisine=* and diet=*. See also the tags amenity=restaurant;bar;fast_food.	commercial
drinking_water	Drinking water is a place where humans can obtain potable water for consumption. Typically, the water is used for only drinking. Also known as a drinking fountain or bubbler.	other
fast_food	Fast food restaurant (see also amenity=restaurant). The kind of food served can be tagged with cuisine=* and diet=*.	commercial
food_court	An area with several different restaurant food counters and a shared eating area. Commonly found in malls, airports, etc.	commercial
ice_cream	Ice cream shop or ice cream parlour. A place that sells ice cream and frozen yoghurt over the counter	commercial
pub	A place selling beer and other alcoholic drinks; may also provide food or accommodation (UK). See description of amenity=bar and amenity=pub for distinction between bar and pub	commercial
restaurant	Restaurant (not fast food, see amenity=fast_food). The kind of food served can be tagged with cuisine=* and diet=*.	commercial
college	Campus or buildings of an institute of Further Education (aka continuing education)	other
driving_school	Driving School which offers motor vehicle driving lessons	commercial
kindergarten	For children too young for a regular school (also known as preschool, playschool or nursery school), in some countries including afternoon supervision of primary school children.	other
language_school	Language School: an educational institution where one studies a foreign language.	other
library	A public library (municipal, university, . . .) to borrow books from.	other
music_school	A music school, an educational institution specialized in the study, training, and research of music.	other
school	School and grounds - primary, middle and seconday schools	other
university	An university campus: an institute of higher education	other
bicycle_parking	Parking for bicycles	other
bicycle_repair_station	General tools for self-service bicycle repairs, usually on the roadside; no service	other
bicycle_rental	Rent a bicycle	commercial
boat_rental	Rent a Boat	commercial
boat_sharing	Share a Boat	other
bus_station	May also be tagged as public_transport=station.	other
car_rental	Rent a car	commercial
car_sharing	Share a car	other
car_wash	Wash a car	commercial
vehicle_inspection	Government vehicle inspection	other
charging_station	Charging facility for electric vehicles	other
ferry_terminal	Ferry terminal/stop. A place where people/cars/etc. can board and leave a ferry.	commercial

fuel	Petrol station; gas station; marine fuel; ... Streets to petrol stations are often tagged highway=service.	commercial
grit_bin	A container that holds grit or a mixture of salt and grit.	other
motorcycle_parking	Parking for motorcycles	other
parking	Car park. Nodes and areas (without access tag) will get a parking symbol. Areas will be coloured. Streets on car parking are often tagged highway=service and service=parking_aisle.	other
parking_entrance	An entrance or exit to an underground or multi-storey parking facility. Group multiple parking entrances together with a relation using the tags type=site and site=parking. Do not mix with amenity=parking.	other
parking_space	A single parking space. Group multiple parking spaces together with a relation using the tags type=site and site=parking. Do not mix with amenity=parking.	other
taxi	A place where taxis wait for passengers.	other
atm	ATM or cash point: a device that provides the clients of a financial institution with access to financial transactions.	other
bank	Bank or credit union: a financial establishment where customers can deposit and withdraw money, take loans, make investments and transfer funds.	commercial
bureau_de_change	Bureau de change, money changer, currency exchange, Wechsel, cambio – a place to change foreign bank notes and travellers cheques.	commercial
baby_hatch	A place where a baby can be, out of necessity, anonymously left to be safely cared for and perhaps adopted.	other
clinic	A medium-sized medical facility or health centre.	other
dentist	A dentist practice / surgery.	other
doctors	A doctor's practice / surgery.	other
hospital	A hospital providing in-patient medical treatment. Often used in conjunction with emergency=* to note whether the medical centre has emergency facilities (A&E (brit.) or ER (am.))	other
nursing_home	Discouraged tag for a home for disabled or elderly persons who need permanent care. Use amenity=social_facility + social_facility=nursing_home now.	other
pharmacy	Pharmacy: a shop where a pharmacist sells medications\ndispensing=yes/no - availability of prescription-only medications	other
social_facility	A facility that provides social services: group & nursing homes, workshops for the disabled, homeless shelters, etc.	other
veterinary	A place where a veterinary surgeon, also known as a veterinarian or vet, practices.	other
arts_centre	A venue where a variety of arts are performed or conducted	other
brothel	An establishment specifically dedicated to prostitution	commercial
casino	A gambling venue with at least one table game(e.g. roulette, blackjack) that takes bets on sporting and other events at agreed upon odds.	commercial
cinema	A place where films are shown (US: movie theater)	commercial
community_centre	A place mostly used for local events, festivities and group activities; including special interest and special age groups. .	other
fountain	A fountain for cultural / decorational / recreational purposes.	other
gambling	A place for gambling, not being a shop=bookmaker, shop=lottery, amenity=casino, or leisure=adult_gaming_centre. \nGames that are covered by this definition include bingo and pachinko.	commercial
nightclub	A place to drink and dance (nightclub). The German word is "Disco" or "Discothek". Please don't confuse this with the German "Nachtclub" which is most likely amenity=stripclub.	commercial
planetarium	A planetarium.	other
public_bookcase	A street furniture containing books. Take one or leave one.	other
social_centre	A place for free and not-for-profit activities.	other
stripclub	A place that offers striptease or lapdancing (for sexual services use amenity=brothel).	commercial
studio	TV radio or recording studio	commercial
swingerclub	A club where people meet to have a party and group sex.	commercial
theatre	A theatre or opera house where live performances occur, such as plays, musicals and formal concerts. Use amenity=cinema for movie theaters.	other
animal_boarding	A facility where you, paying a fee, can bring your animal for a limited period of time (e.g. for holidays)	commercial
animal_shelter	A shelter that recovers animals in trouble	other
baking_oven	An oven used for baking bread and similar, for example inside a building=bakehouse.	other
bench	A bench to sit down and relax a bit	other
childcare	A place where children of different ages are looked after which is not an amenity=kindergarten or preschool.	other
clock	A public visible clock	other
courthouse	A building home to a court of law, where justice is dispensed	other
crematorium	A place where dead human bodies are burnt	other
dive_centre	A dive center is the base location where sports divers usually start scuba diving or make dive guided trips at new locations.	commercial
embassy	An embassy, consulate or diplomatic office. Also see office=diplomatic	other
fire_station	A station of a fire brigade	other

firepit	Deprecated. For campfires and firepits, see Tag:leisure=firepit	other
grave_yard	A (smaller) place of burial, often you'll find a church nearby. Large places should be landuse=cemetery instead.	other
gym	Do no use, leisure=fitness_centre or leisure=sports_centre is preferred! A place which houses exercise equipment for the purpose of physical exercise.	commercial
hunting_stand	A hunting stand: an open or enclosed platform used by hunters to place themselves at an elevated height above the terrain	other
internet_cafe	A place whose principal role is providing internet services to the public.	commercial
kitchen	A public kitchen in a facility to use by everyone or customers	other
kneipp_water_cure	Outdoor foot bath facility. Usually this is a pool with cold water and handrail. Popular in German speaking countries.	other
marketplace	A marketplace where goods and services are traded daily or weekly.	commercial
monastery	Monastery is the location of a monastery or a building in which monks and nuns live.	other
photo_booth	Photo Booth – A stand to create instant photo.	commercial
place_of_worship	A church, mosque, or temple, etc. Note that you also need religion=*, usually denomination=* and preferably name=* as well as amenity=place_of_worship. See the article for details.	other
police	A police station where police officers patrol from and that is a first point of contact for civilians	other
post_box	A box for the reception of mail. Alternative mail-carriers can be tagged via operator=*	other
post_depot	Post depot or delivery office, where letters and parcels are collected and sorted prior to delivery.	commercial
post_office	Post office building with postal services	commercial
prison	A prison or jail where people are incarcerated before trial or after conviction	other
public_bath	A location where the public may bathe in common, etc. japanese onsen, turkish bath, hot spring	other
public_building	A generic public building. Don't use! See office=government.	other
ranger_station	National Park visitor headquarters: official park visitor facility with police, visitor information, permit services, etc	other
recycling	Recycling facilities (bottle banks, etc.). Combine with recycling_type=container for containers or recycling_type=centre for recycling centres.	other
sanitary_dump_station	A place for depositing human waste from a toilet holding tank.	other
sauna	Deprecated. For sauna use: leisure=sauna	other
shelter	A small shelter against bad weather conditions. To additionally describe the kind of shelter use shelter_type=*.	other
shower	Public shower or bath.	other
telephone	Public telephone	other
toilets	Public toilets (might require a fee)	other
townhall	Building where the administration of a village, town or city may be located, or just a community meeting place	other
vending_machine	A machine selling goods – food, tickets, newspapers, etc. Add type of goods using vending=*	other
waste_basket	A single small container for depositing garbage that is easily accessible for pedestrians.	other
waste_disposal	A place where canal boaters, caravaners, etc. can dispose of rubbish (trash/waste).	other
waste_transfer_station	A waste transfer station is a location that accepts, consolidates and transfers waste in bulk.	other
watering_place	Place where water is contained and animals can drink	other
water_point	Place where you can get large amounts of drinking water	other
user defined	All commonly used values according to Taginfo	other
shop	All commonly used values according to Taginfo	commercial

Table A.2: Mapping of possible values in OSM tag *amenity* to the unified classification scheme of this thesis. Value and description are taken from the OSM wiki [230]

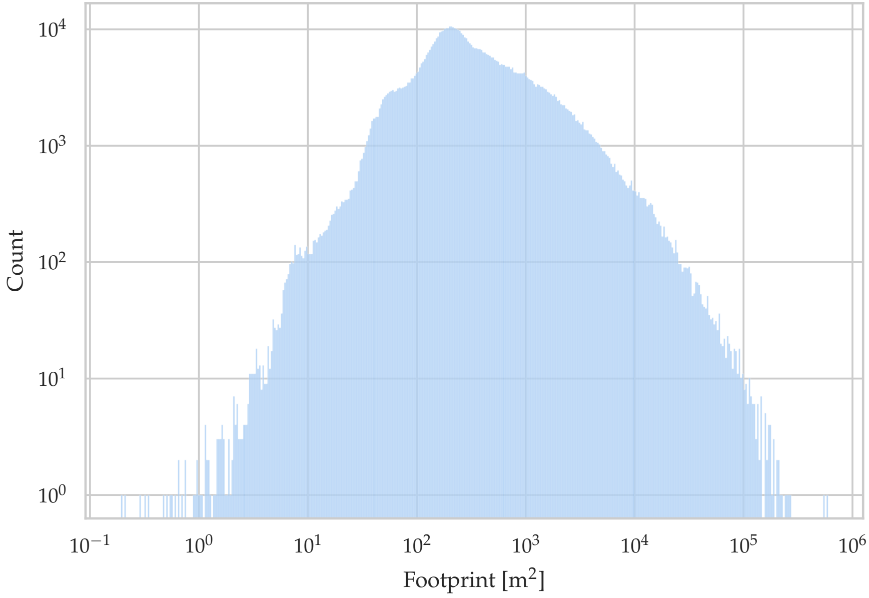


Figure A.1: Distribution of OSM building footprint sizes in LCZ42 cities

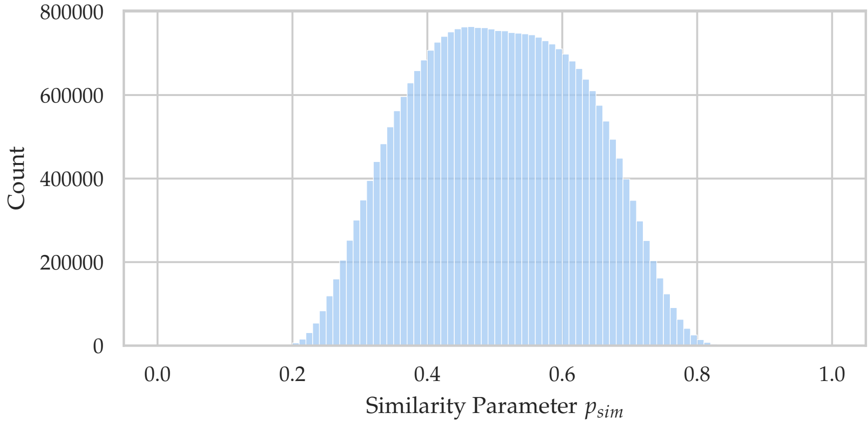


Figure A.2: Distribution of similarity parameter p_{sim} in Flickr LCZ42 dataset. The histogram is based on 28,818,438 images with a minimum similarity of 0.143, a maximum similarity of 0.904, a mean similarity of 0.506, a variance of 0.015, and a skewness of 0.014.

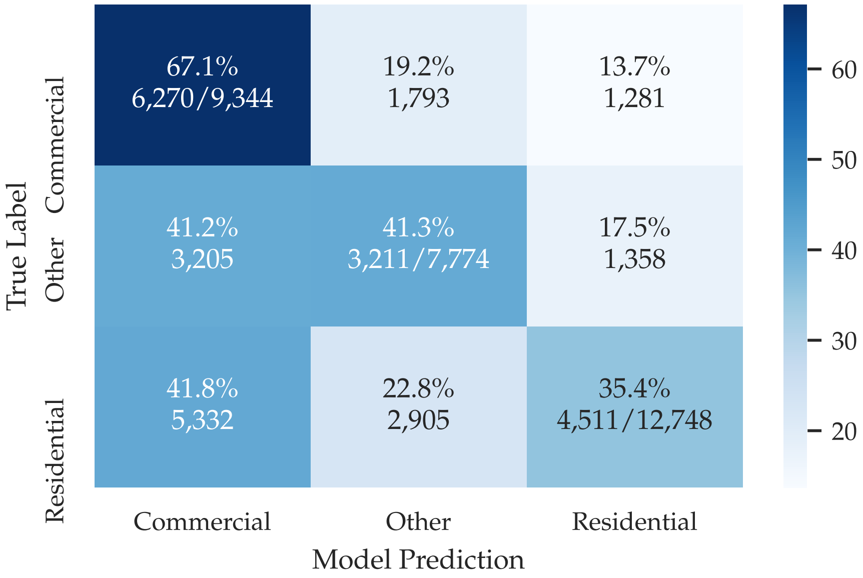


Figure A.3: Confusion matrix of building instance predictions based on a mean fusion of social media image predictions from InceptionResNetv2 model

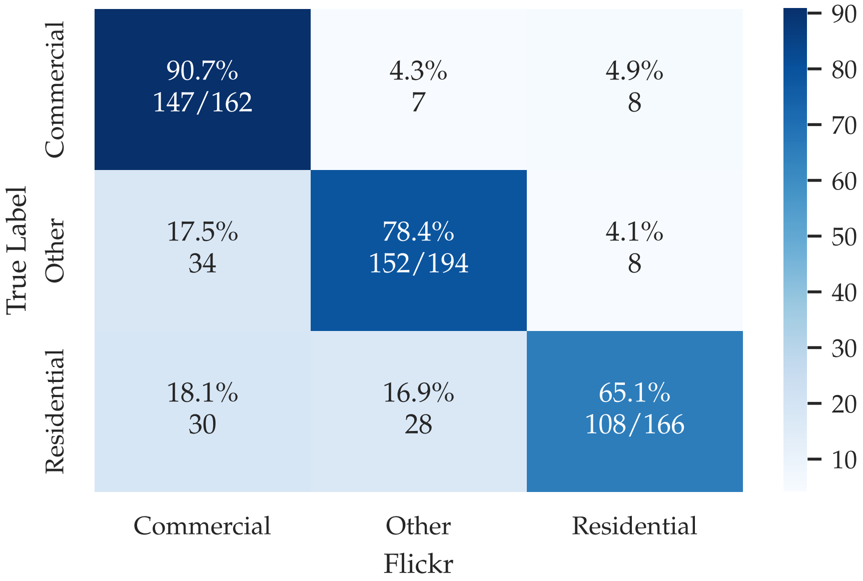


Figure A.4: Confusion matrix of building instance predictions from human-verified image labels using InceptionResNetv2 model

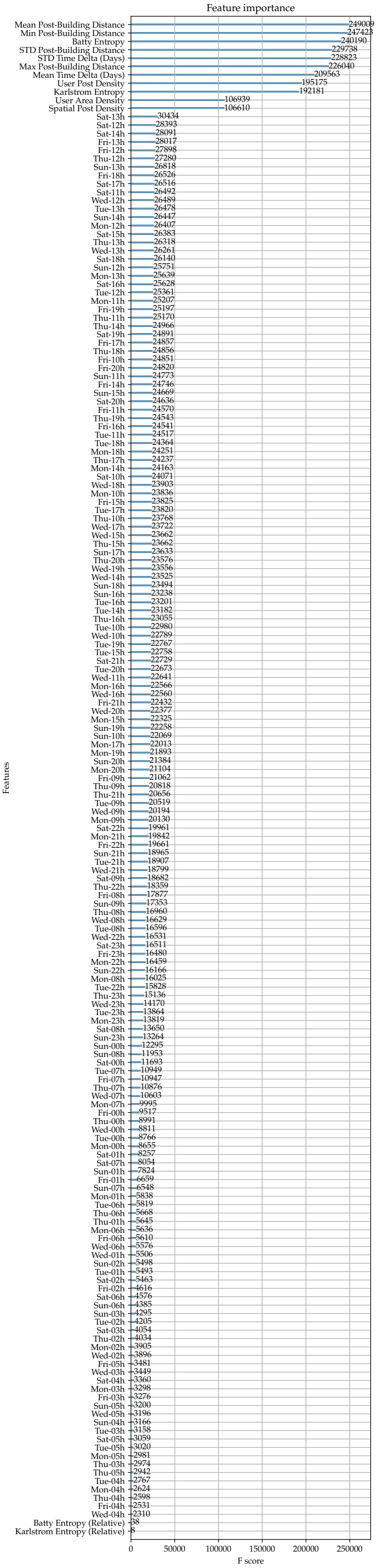


Figure A.5: Feature importance ordered by weight in gradient boosted tree model with 4,096 estimators, a maximum depth of 13, and a learning rate of 0.06

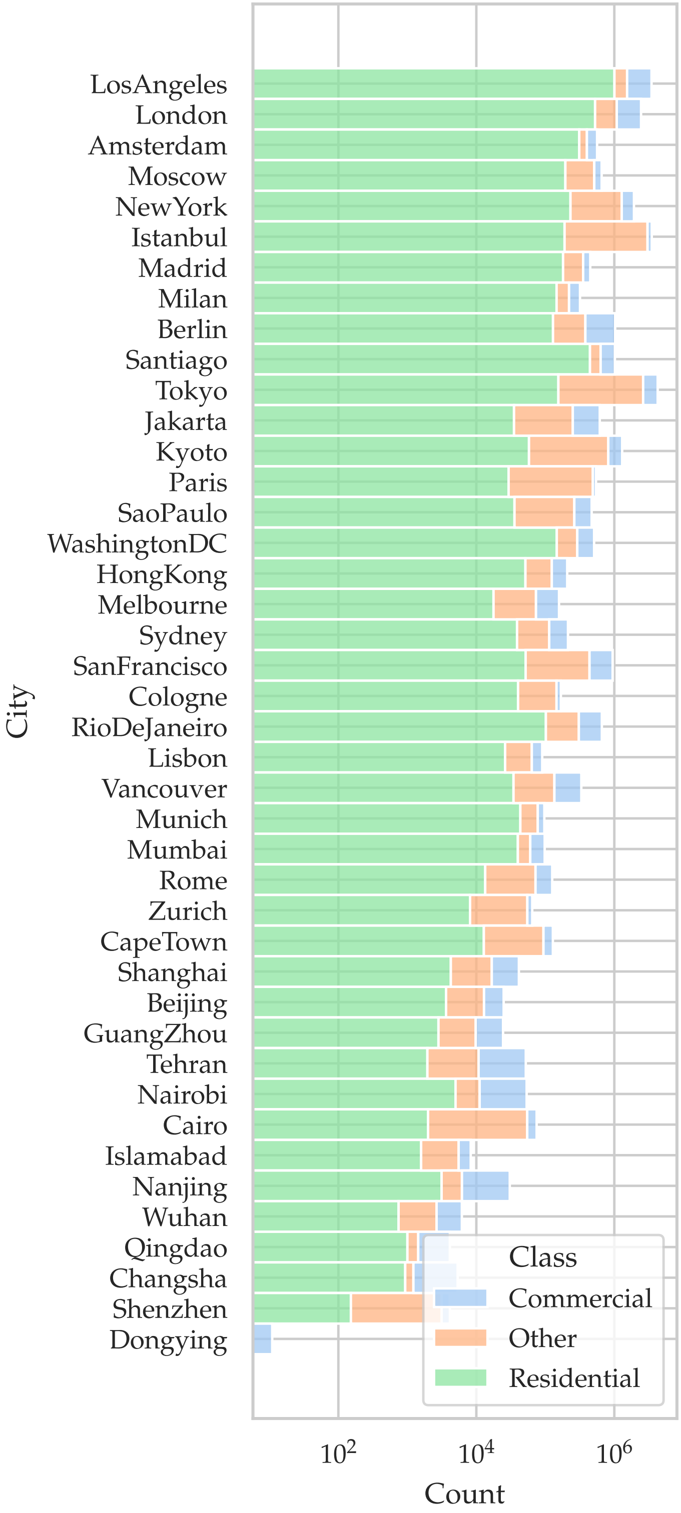


Figure A.6: Number of aerial building images in the original So2Sat dataset before balancing, sorted in descending order by the number of buildings on a log scale.

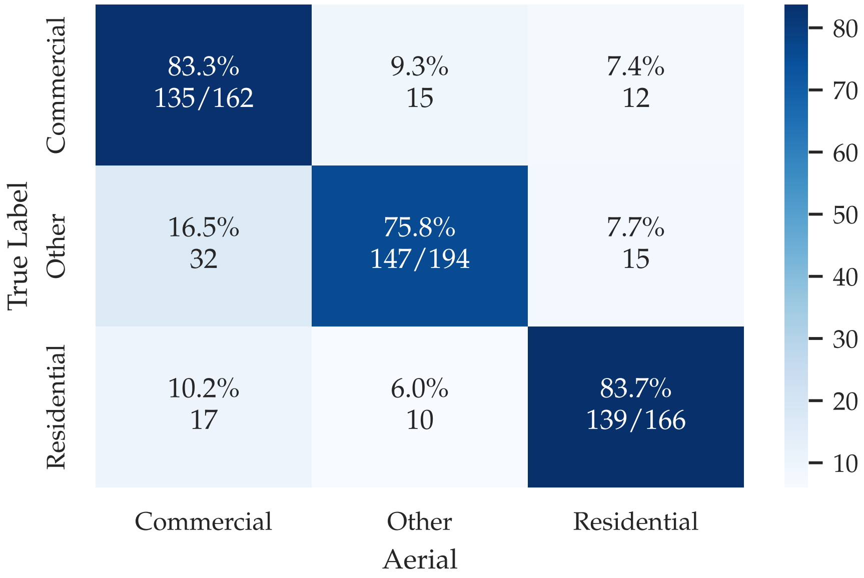


Figure A.7: Confusion matrix of VGG16 architecture fine-tuned on human-verified social media image dataset

BIBLIOGRAPHY

[1] Charles R. Harris et al. "Array Programming with NumPy." In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2.

[2] Wes McKinney. "Data Structures for Statistical Computing in Python." In: *Proceedings of the 9th Python in Science Conference* (2010), pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[3] TensorFlow Developers. *TensorFlow*. Zenodo. Feb. 2022. DOI: 10.5281/zenodo.5949169.

[4] Michael L. Waskom. "Seaborn: Statistical Data Visualization." In: *Journal of Open Source Software* 6.60 (Apr. 2021), p. 3021. ISSN: 2475-9066. DOI: 10.21105/joss.03021.

[5] Phil Elson et al. *SciTools/Cartopy: V0.20.2*. Zenodo. Jan. 2022. DOI: 10.5281/zenodo.5842769.

[6] John D. Hunter. "Matplotlib: A 2D Graphics Environment." In: *Computing in Science Engineering* 9.3 (May 2007), pp. 90–95. ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55.

[7] Michael Stonebraker and Lawrence A. Rowe. "The Design of POSTGRES." In: *ACM SIGMOD Record* 15.2 (June 1986), pp. 340–355. ISSN: 0163-5808. DOI: 10.1145/16856.16888.

[8] Christian Strobl. "PostGIS." In: *Encyclopedia of GIS*. Ed. by Shashi Shekhar and Hui Xiong. Boston, MA: Springer US, 2008, pp. 891–898. ISBN: 978-0-387-35973-1. DOI: 10.1007/978-0-387-35973-1_1012.

[9] United Nations, Department of Economic and Social Affairs, Population Division. *World Population Prospects 2019: Highlights (ST/ESA/SER.A/423)*. 2019. ISBN: 978-92-1-148316-1.

[10] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2018 Revision (ST/ESA/SER.A/420)*. United Nations, New York, 2019. ISBN: 978-92-1-148319-2.

[11] Edward Glaeser. *Triumph of the City: How Our Greatest Invention Makes Us Richer, Smarter, Greener, Healthier, and Happier*. Reprint Edition. New York, NY: Penguin Books, Jan. 2012. ISBN: 978-0-14-312054-4.

[12] UN. *Report of the UN Economist Network for the UN 75th Anniversary :* UN, 2020. ISBN: 978-92-1-104696-0.

[13] United Nations, Department of Economic and Social Affairs, and Population Division. *World Urbanization Prospects: 2018 : Highlights*. 2019. ISBN: 978-92-1-148318-5.

[14] UN Desa et al. “Transforming Our World: The 2030 Agenda for Sustainable Development.” In: (2016).

[15] Peter Fisher and David Unwin. *Re-Presenting GIS*. John Wiley & Sons, Nov. 2005. ISBN: 978-0-470-01735-7.

[16] Jian Kang, Marco Körner, Yuanyuan Wang, Hannes Taubenböck, and Xiao Xiang Zhu. “Building Instance Classification Using Street View Images.” In: *ISPRS Journal of Photogrammetry and Remote Sensing*. Deep Learning RS Data 145 (Nov. 2018), pp. 44–59. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2018.02.006.

[17] Eike Jens Hoffmann, Mohsin Ali, and Xiao Xiang Zhu. “Zooming into Uncertainties: Towards Fusing Multi Zoom Level Imagery for Urban Land Use Segmentation.” In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. July 2021, pp. 2090–2093. DOI: 10.1109/IGARSS47720.2021.9554756.

[18] Pierfrancesco Bellini, Monica Benigni, Riccardo Billero, Paolo Nesi, and Nadia Rauch. “Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-City Services.” In: *Journal of Visual Languages & Computing*. Distributed Multimedia Systems DMS2014 Part I 25.6 (Dec. 2014), pp. 827–839. ISSN: 1045-926X. DOI: 10.1016/j.jvlc.2014.10.023.

[19] David M. Theobald. “Development and Applications of a Comprehensive Land Use Classification and Map for the US.” In: *PLOS ONE* 9.4 (Nov. 2014), e94628. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0094628.

[20] Eike Jens Hoffmann, Yuanyuan Wang, Martin Werner, Jian Kang, and Xiao Xiang Zhu. “Model Fusion for Building Type Classification from Aerial and Street View Images.” In: *Remote Sensing* 11.11 (Jan. 2019), p. 1259. DOI: 10.3390/rs11111259.

[21] Bo Huang, Bei Zhao, and Yimeng Song. “Urban Land-Use Mapping Using a Deep Convolutional Neural Network with High Spatial Resolution Multispectral Remote Sensing Imagery.” In: *Remote Sensing of Environment* 214 (Sept. 2018), pp. 73–86. ISSN: 00344257. DOI: 10.1016/j.rse.2018.04.050.

[22] Matthias Häberle, Martin Werner, and Xiao Xiang Zhu. “Building Type Classification from Social Media Texts via Geo-Spatial Textmining.” In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. July 2019, pp. 10047–10050. DOI: 10.1109/IGARSS.2019.8898836.

[23] Eike Jens Hoffmann, Martin Werner, and Xiao Xiang Zhu. “Building Instance Classification Using Social Media Images.” In: *2019 Joint Urban Remote Sensing Event (JURSE)*. May 2019, pp. 1–4. DOI: 10.1109/JURSE.2019.8809056.

[24] Rong Huang, Hannes Taubenböck, Lichao Mou, and Xiao Xiang Zhu. "Classification of Settlement Types from Tweets Using LDA and LSTM." In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. July 2018, pp. 6408–6411. DOI: 10.1109/IGARSS.2018.8519240.

[25] Xiuyuan Zhang, Shihong Du, and Qiao Wang. "Hierarchical Semantic Cognition for Urban Functional Zones with VHR Satellite Images and POI Data." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 132 (Oct. 2017), pp. 170–184. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.09.007.

[26] Marco De Nadai, Jacopo Staiano, Roberto Larcher, Nicu Sebe, Daniele Quercia, and Bruno Lepri. "The Death and Life of Great Italian Cities: A Mobile Phone Data Perspective." In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 413–423. ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883084.

[27] Hannes Taubenböck and Michael Wurm. "Ich weiß, dass ich nichts weiß – Bevölkerungsschätzung in der Megacity Mumbai." In: *Globale Urbanisierung: Perspektive aus dem All*. Ed. by Hannes Taubenböck, Michael Wurm, Thomas Esch, and Stefan Dech. Berlin, Heidelberg: Springer, 2015, pp. 171–178. ISBN: 978-3-662-44841-0. DOI: 10.1007/978-3-662-44841-0_18.

[28] Anand Sahasranaman and Luís M. A. Bettencourt. "Life between the City and the Village: Scaling Analysis of Service Access in Indian Urban Slums." In: *World Development* 142 (June 2021), p. 105435. ISSN: 0305-750X. DOI: 10.1016/j.worlddev.2021.105435.

[29] Patrick Aravena Pelizari, Christian Geiß, Paula Aguirre, Hernán Santa María, Yvonne Merino Peña, and Hannes Taubenböck. "Automated Building Characterization for Seismic Risk Assessment Using Street-Level Imagery and Deep Learning." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 180 (Oct. 2021), pp. 370–386. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2021.07.004.

[30] *GEO at a Glance*. https://earthobservations.org/geo_wwd.php.

[31] Siamak Khorram, Cynthia F. van der Wiele, Frank H. Koch, Stacy A. C. Nelson, and Matthew D. Potts. "Remote Sensing: Past and Present." In: *Principles of Applied Remote Sensing*. Ed. by Siamak Khorram, Cynthia F. van der Wiele, Frank H. Koch, Stacy A. C. Nelson, and Matthew D. Potts. Cham: Springer International Publishing, 2016, pp. 1–20. ISBN: 978-3-319-22560-9. DOI: 10.1007/978-3-319-22560-9_1.

[32] *Sentinel-1 - Missions - Sentinel Online - Sentinel Online*. https://sentinels.copernicus.eu
1.

[33] *Sentinel-2 - Missions - Sentinel Online - Sentinel Online*. https://sentinel.esa.int/web/s
2.

[34] NASA VIIRS Land Science Investigator-Led Processing System. *VIIRS/NPP Daily Gridded Day Night Band 500m Linear Lat Lon Grid Night*. 2019. DOI: 10.5067/VIIRS/VNP46A1.001.

[35] *Zoom Levels – OpenStreetMap Wiki*. https://wiki.openstreetmap.org/wiki/Zoom_leve

[36] John R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Fourth. USA: Prentice Hall Press, 2015. ISBN: 978-0-13-405816-0.

[37] Hankui K. Zhang, David P. Roy, Lin Yan, Zhongbin Li, Haiyan Huang, Eric Vermote, Sergii Skakun, and Jean-Claude Roger. "Characterization of Sentinel-2A and Landsat-8 Top of Atmosphere, Surface, and Nadir BRDF Adjusted Reflectance and NDVI Differences." In: *Remote Sensing of Environment* 215 (Sept. 2018), pp. 482–494. ISSN: 0034-4257. DOI: 10.1016/j.rse.2018.04.031.

[38] Shoji Tominaga, Shogo Nishi, and Ryo Ohtera. "Measurement and Estimation of Spectral Sensitivity Functions for Mobile Phone Cameras." In: *Sensors (Basel, Switzerland)* 21.15 (July 2021), p. 4985. ISSN: 1424-8220. DOI: 10.3390/s21154985.

[39] Daniel Kahneman. *Thinking, Fast and Slow*. New York: Macmillan US, Oct. 2011. ISBN: 978-0-374-27563-1.

[40] John Canny. "A Computational Approach to Edge Detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1986.4767851.

[41] Chris Harris and Mike Stephens. "A Combined Corner and Edge Detector." In: *In Proc. of Fourth Alvey Vision Conference*. 1988, pp. 147–151.

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386.

[43] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. ISBN: 978-0-07-115467-3.

[44] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity." In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Ed. by Francis Bach. Adaptive Computation and Machine Learning Series. Cambridge, MA, USA: MIT Press, Nov. 2016. ISBN: 978-0-262-03561-3.

[46] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980 [cs].

[47] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators." In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.

[48] Yann LeCun and Yoshua Bengio. "Convolutional Networks for Images, Speech, and Time Series." In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, Oct. 1998, pp. 255–258. ISBN: 978-0-262-51102-5.

[49] Kunihiko Fukushima. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position." In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251.

[50] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines." In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Madison, WI, USA: Omnipress, June 2010, pp. 807–814. ISBN: 978-1-60558-907-7.

[51] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *arXiv:1409.1556 [cs]* (Apr. 2015). arXiv: 1409.1556 [cs].

[52] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[53] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision." In: *arXiv:1512.00567 [cs]* (Dec. 2015). arXiv: 1512.00567 [cs].

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90.

[55] Y. Bengio, P. Simard, and P. Frasconi. “Learning Long-Term Dependencies with Gradient Descent Is Difficult.” In: *IEEE Transactions on Neural Networks* 5.2 (Mar. 1994), pp. 157–166. ISSN: 1941-0093. DOI: 10.1109/72.279181.

[56] Xavier Glorot and Yoshua Bengio. “Understanding the Difficulty of Training Deep Feedforward Neural Networks.” In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Identity Mappings in Deep Residual Networks.” In: *arXiv:1603.05027 [cs]* (July 2016). arXiv: 1603.05027 [cs].

[58] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. “Densely Connected Convolutional Networks.” In: *arXiv:1608.06993 [cs]* (Jan. 2018). arXiv: 1608.06993 [cs].

[59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.” In: *arXiv:1602.07261 [cs]* (Aug. 2016). arXiv: 1602.07261 [cs].

[60] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” In: *arXiv:1704.04861 [cs]* (Apr. 2017). arXiv: 1704.04861 [cs].

[61] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” In: *arXiv:1801.04381 [cs]* (Mar. 2019). arXiv: 1801.04381 [cs].

[62] François Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions.” In: *arXiv:1610.02357 [cs]* (Apr. 2017). arXiv: 1610.02357 [cs].

[63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” In: *arXiv:1506.01497 [cs]* (Jan. 2016). arXiv: 1506.01497 [cs].

[64] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.” In: *arXiv:1311.2524 [cs]* (Oct. 2014). arXiv: 1311.2524 [cs].

[65] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. “Selective Search for Object Recognition.” In: *International Journal of Computer Vision* 104.2 (Sept. 2013), pp. 154–171. ISSN: 1573-1405. DOI: 10.1007/s11263-013-0620-5.

[66] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. "OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks." In: *arXiv:1312.6229 [cs]* (Feb. 2014). arXiv: 1312.6229 [cs].

[67] Ross Girshick. "Fast R-CNN." In: *arXiv:1504.08083 [cs]* (Sept. 2015). arXiv: 1504.08083 [cs].

[68] Jonathan Huang et al. "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors." In: *arXiv:1611.10012 [cs]* (Apr. 2017). arXiv: 1611.10012 [cs].

[69] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. "We Don't Need No Bounding-Boxes: Training Object Class Detectors Using Only Human Verification." In: *arXiv:1602.08405 [cs]* (Apr. 2017). arXiv: 1602.08405 [cs].

[70] Alina Kuznetsova et al. "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale." In: *International Journal of Computer Vision* 128.7 (July 2020), pp. 1956–1981. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-020-01316-z. arXiv: 1811.00982.

[71] *Google Maps Platform Terms Of Service*. https://cloud.google.com/maps-platform/terms.

[72] Kai Wang, Boris Babenko, and Serge Belongie. "End-to-End Scene Text Recognition." In: *2011 International Conference on Computer Vision*. Nov. 2011, pp. 1457–1464. DOI: 10.1109/ICCV.2011.6126402.

[73] Amir Roshan Zamir and Mubarak Shah. "Image Geo-Localization Based on MultipleNearest Neighbor Feature Matching UsingGeneralized Graphs." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (Aug. 2014), pp. 1546–1558. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2014.2299799.

[74] Scott Workman, Richard Souvenir, and Nathan Jacobs. "Wide-Area Image Geolocalization with Aerial Reference Imagery." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 3961–3969. DOI: 10.1109/ICCV.2015.451.

[75] Piotr Mirowski et al. "The StreetLearn Environment and Dataset." In: *arXiv:1903.01292 [cs]* (Mar. 2019). arXiv: 1903.01292 [cs].

[76] Yair Movshovitz-Attias, Qian Yu, Martin C. Stumpe, Vinay Shet, Sacha Arnoud, and Liron Yatziv. "Ontological Supervision for Fine Grained Classification of Street View Storefronts." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1693–1702. DOI: 10.1109/CVPR.2015.7298778.

[77] Shivangi Srivastava, John E. Vargas Muñoz, Sylvain Lobry, and Devis Tuia. "Fine-Grained Landuse Characterization Using Ground-Based Pictures: A Deep Learning Solution Based on Globally Available Data." In: *International Journal of Geographical Information Science* 34.6 (June 2020), pp. 1117–1136. ISSN: 1365-8816. DOI: 10.1080/13658816.2018.1542698.

[78] Fang Fang, Yafang Yu, Shengwen Li, Zejun Zuo, Yuanyuan Liu, Bo Wan, and Zhongwen Luo. "Synthesizing Location Semantics from Street View Images to Improve Urban Land-Use Classification." In: *International Journal of Geographical Information Science* 35.9 (Sept. 2021), pp. 1802–1825. ISSN: 1365-8816. DOI: 10.1080/13658816.2020.1831515.

[79] G. S. Lovasi, J. W. Quinn, K. M. Neckerman, M. S. Perzanowski, and A. Rundle. "Children Living in Areas with More Street Trees Have Lower Prevalence of Asthma." In: *Journal of Epidemiology & Community Health* 62.7 (July 2008), pp. 647–649. ISSN: 0143-005X, 1470-2738. DOI: 10.1136/jech.2007.071894.

[80] David J. Nowak, Satoshi Hirabayashi, Allison Bodine, and Eric Greenfield. "Tree and Forest Effects on Air Quality and Human Health in the United States." In: *Environmental Pollution* 193 (Oct. 2014), pp. 119–129. ISSN: 0269-7491. DOI: 10.1016/j.envpol.2014.05.028.

[81] Ian Seiferling, Nikhil Naik, Carlo Ratti, and Raphäel Proulx. "Green Streets - Quantifying and Mapping Urban Trees with Street-Level Imagery and Computer Vision." In: *Landscape and Urban Planning* 165 (Sept. 2017), pp. 93–101. ISSN: 01692046. DOI: 10.1016/j.landurbplan.2017.05.010.

[82] Daniel Laumer, Nico Lang, Natalie van Doorn, Oisin Mac Aodha, Pietro Perona, and Jan Dirk Wegner. "Geocoding of Trees from Street Addresses and Street-Level Images." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (Apr. 2020), pp. 125–136. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2020.02.001.

[83] Vladimir A. Krylov, Eamonn Kenny, and Rozenn Dahyot. "Automatic Discovery and Geotagging of Objects from Street View Imagery." In: *Remote Sensing* 10.5 (May 2018), p. 661. ISSN: 2072-4292. DOI: 10.3390/rs10050661.

[84] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. "Using Deep Learning and Google Street View to Estimate the Demographic Makeup of Neighborhoods across the United States." In: *Proceedings of the National Academy of Sciences* 114.50 (Dec. 2017), pp. 13108–13113. DOI: 10.1073/pnas.1700035114.

[85] Rahul Goel, Leandro M. T. Garcia, Anna Goodman, Rob Johnson, Rachel Aldred, Manoradhan Murugesan, Soren Brage, Kavi Bhalla, and James Woodcock. "Estimating City-Level Travel Patterns Using Street Imagery: A Case Study of Using Google Street View in Britain." In: *PLOS ONE* 13.5 (Feb. 2018), e0196521. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0196521.

[86] P. Chaudhary, S. D'Aronco, J. P. Leitão, K. Schindler, and J. D. Wegner. "Water Level Prediction from Social Media Images with a Multi-Task Ranking Approach." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (Sept. 2020), pp. 252–262. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2020.07.003.

[87] Anna Kruspe and Jens Kersten. "Detecting Event-Related Tweets by Example Using Few-Shot Models." In: (2019), p. 11.

[88] Anna Kruspe, Jens Kersten, and Friederike Klan. *Review Article: Detection of Informative Tweets in Crisis Events*. Preprint. Databases, GIS, Remote Sensing, Early Warning Systems and Monitoring Technologies, July 2020. DOI: 10.5194/nhess-2020-214.

[89] Gianni Cristian Iannelli and Fabio Dell'Acqua. "Extensive Exposure Mapping in Urban Areas through Deep Analysis of Street-Level Pictures for Floor Count Determination." In: *Urban Science* 1.2 (June 2017), p. 16. ISSN: 2413-8851. DOI: 10.3390/urbansci1020016.

[90] David J. Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. "Mapping the World's Photos." In: *Proceedings of the 18th International Conference on World Wide Web*. WWW '09. New York, NY, USA: Association for Computing Machinery, Apr. 2009, pp. 761–770. ISBN: 978-1-60558-487-4. DOI: 10.1145/1526709.1526812.

[91] Noah Snavely, Steven M. Seitz, and Richard Szeliski. "Modeling the World from Internet Photo Collections." In: *International Journal of Computer Vision* 80.2 (Nov. 2008), pp. 189–210. ISSN: 1573-1405. DOI: 10.1007/s11263-007-0107-3.

[92] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints." In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000029664.99615.94.

[93] Silvia Paldino, Iva Bojic, Stanislav Sobolevsky, Carlo Ratti, and Marta C. González. "Urban Magnetism through the Lens of Geo-Tagged Photography." In: *EPJ Data Science* 4.1 (Dec. 2015), pp. 1–17. ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-015-0043-3.

[94] Daniel Leung and Shawn Newsam. "Proximate Sensing: Inferring What-Is-Where from Georeferenced Photo Collections." In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June 2010, pp. 2955–2962. DOI: 10.1109/CVPR.2010.5540040.

[95] Hirotaka Oba, Masaharu Hirota, Richard Chbeir, Hiroshi Ishikawa, and Shohei Yokoyama. "Towards Better Land Cover Classification Using Geo-tagged Photographs." In: *2014 IEEE International Symposium on Multimedia*. Dec. 2014, pp. 320–327. DOI: 10.1109/ISM.2014.78.

[96] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features." In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 404–417. ISBN: 978-3-540-33833-8. DOI: 10.1007/11744023_32.

[97] Ling Xie and Shawn Newsam. "IM2MAP: Deriving Maps from Georeferenced Community Contributed Photo Collections." In: *Proceedings of the 3rd ACM SIGMM International Workshop on Social Media*. WSM '11. New York, NY, USA: Association for Computing Machinery, Nov. 2011, pp. 29–34. ISBN: 978-1-4503-0989-9. DOI: 10.1145/2072609.2072620.

[98] Aude Oliva and Antonio Torralba. "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope." In: *International Journal of Computer Vision* 42.3 (May 2001), pp. 145–175. ISSN: 1573-1405. DOI: 10.1023/A:1011139631724.

[99] Johannes Langemeyer, Fulvia Calcagni, and Francesc Baró. "Mapping the Intangible: Using Geolocated Social Media Data to Examine Landscape Aesthetics." In: *Land Use Policy* 77 (Sept. 2018), pp. 542–552. ISSN: 0264-8377. DOI: 10.1016/j.landusepol.2018.05.049.

[100] Ilan Havinga, Diego Marcos, Patrick W. Bogaart, Lars Hein, and Devis Tuia. "Social Media and Deep Learning Capture the Aesthetic Quality of the Landscape." In: *Scientific Reports* 11.1 (Oct. 2021), p. 20000. ISSN: 2045-2322. DOI: 10.1038/s41598-021-99282-0.

[101] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. "The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding." In: *International Journal of Computer Vision* 108.1 (May 2014), pp. 59–81. ISSN: 1573-1405. DOI: 10.1007/s11263-013-0695-z.

[102] Daniel Leung and Shawn Newsam. "Exploring Geotagged Images for Land-Use Classification." In: *Proceedings of the ACM Multimedia 2012 Workshop on Geotagging and Its Applications in*

Multimedia. GeoMM '12. New York, NY, USA: Association for Computing Machinery, Oct. 2012, pp. 3–8. ISBN: 978-1-4503-1590-6. DOI: 10.1145/2390790.2390794.

[103] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. "Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval." In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*. CIVR '07. New York, NY, USA: Association for Computing Machinery, July 2007, pp. 494–501. ISBN: 978-1-59593-733-9. DOI: 10.1145/1282280.1282352.

[104] Li-jia Li, Hao Su, Li Fei-fei, and Eric Xing. "Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification." In: *Advances in Neural Information Processing Systems*. Vol. 23. Curran Associates, Inc., 2010.

[105] Fang Fang, Xiaohui Yuan, Lu Wang, Yuanyuan Liu, and Zhongwen Luo. "Urban Land-Use Classification From Photographs." In: *IEEE Geoscience and Remote Sensing Letters* 15.12 (Dec. 2018), pp. 1927–1931. ISSN: 1558-0571. DOI: 10.1109/LGRS.2018.2864282.

[106] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 Million Image Database for Scene Recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (June 2018), pp. 1452–1464. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2017.2723009.

[107] Yi Zhu and Shawn Newsam. "Land Use Classification Using Convolutional Neural Networks Applied to Ground-Level Images." In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '15. New York, NY, USA: Association for Computing Machinery, Nov. 2015, pp. 1–4. ISBN: 978-1-4503-3967-4. DOI: 10.1145/2820783.2820851.

[108] Yi Zhu, Xueqing Deng, and Shawn Newsam. "Fine-Grained Land Use Classification at the City Scale Using Ground-Level Images." In: *IEEE Transactions on Multimedia* 21.7 (July 2019), pp. 1825–1838. ISSN: 1941-0077. DOI: 10.1109/TMM.2019.2891999.

[109] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[110] Zhinan Qiao and Xiaohui Yuan. "Urban Land-Use Analysis Using Proximate Sensing Imagery: A Survey." In: *International Journal of Geographical Information Science* 35.11 (Nov. 2021), pp. 2129–2148. ISSN: 1365-8816. DOI: 10.1080/13658816.2021.1919682.

[111] Yi Zhu, Sen Liu, and Shawn Newsam. "Large-Scale Mapping of Human Activity Using Geo-Tagged Videos." In: *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '17. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–4. ISBN: 978-1-4503-5490-5. DOI: 10.1145/3139958.3140055.

[112] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. "Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness." In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 3–10. ISBN: 978-3-319-49409-8. DOI: 10.1007/978-3-319-49409-8_1.

[113] Raphaël D'Andrimont, Momchil Yordanov, Guido Lemoine, Janine Yoong, Kamil Nikel, and Marijn Van der Velde. "Crowdsourced Street-Level Imagery as a Potential Source of In-Situ Data for Crop Monitoring." In: *Land* 7.4 (Dec. 2018), p. 127. ISSN: 2073-445X. DOI: 10.3390/land7040127.

[114] Matthias Häberle, Martin Werner, and Xiao Xiang Zhu. "Geo-Spatial Text-Mining from Twitter – a Feature Space Analysis with a View toward Building Classification in Urban Regions." In: *European Journal of Remote Sensing* 52.sup2 (Aug. 2019), pp. 2–11. ISSN: null. DOI: 10.1080/22797254.2019.1586451.

[115] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. "Advances in Pre-Training Distributed Word Representations." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[116] Fernando Terroso-Saenz and Andrés Muñoz. "Land Use Discovery Based on Volunteer Geographic Information Classification." In: *Expert Systems with Applications* 140 (Feb. 2020), p. 112892. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2019.112892.

[117] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. "YFCC100M: The New Data in Multimedia Research." In: *Communications of the ACM* 59.2 (Jan. 2016), pp. 64–73. ISSN: 0001-0782. DOI: 10.1145/2812802.

[118] Fernando Terroso-Saenz, Andres Muñoz, and Francisco Arcas. "Land-Use Dynamic Discovery Based on Heterogeneous Mobility Sources." In: *International Journal of Intelligent Systems* 36.1 (2021), pp. 478–525. ISSN: 1098-111X. DOI: 10.1002/int.22307.

[119] Gabriel Dax and Martin Werner. “Information-Optimal Abstaining for Reliable Classification of Building Functions.” In: *AGILE: GIScience Series* 2 (June 2021), pp. 1–10. DOI: 10.5194/agile-giss-2-1-2021.

[120] Helen Victoria Roberts. “Using Twitter Data in Urban Green Space Research: A Case Study and Critical Evaluation.” In: *Applied Geography* 81 (Apr. 2017), pp. 13–20. ISSN: 0143-6228. DOI: 10.1016/j.apgeog.2017.02.008.

[121] Zoé A. Hamstead, David Fisher, Rositsa T. Ilieva, Spencer A. Wood, Timon McPhearson, and Peleg Kremer. “Geolocated Social Media as a Rapid Indicator of Park Visitation and Equitable Park Access.” In: *Computers, Environment and Urban Systems* 72 (Nov. 2018), pp. 38–50. ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2018.01.007.

[122] Lewis Mitchell, Morgan R. Frank, Kameron Decker Harris, Peter Sheridan Dodds, and Christopher M. Danforth. “The Geography of Happiness: Connecting Twitter Sentiment and Expression, Demographics, and Objective Characteristics of Place.” In: *PLOS ONE* 8.5 (May 2013), e64417. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0064417.

[123] Linna Li, Michael F. Goodchild, and Bo Xu. “Spatial, Temporal, and Socioeconomic Patterns in the Use of Twitter and Flickr.” In: *Cartography and Geographic Information Science* 40.2 (Mar. 2013), pp. 61–77. ISSN: 1523-0406. DOI: 10.1080/15230406.2013.777139.

[124] Eszter Bokányi, Dániel Kondor, László Dobos, Tamás Sebők, József Stéger, István Csabai, and Gábor Vattay. “Race, Religion and the City: Twitter Word Frequency Patterns Reveal Dominant Demographic Dimensions in the United States.” In: *Palgrave Communications* 2.1 (Apr. 2016), pp. 1–9. ISSN: 2055-1045. DOI: 10.1057/palcomms.2016.10.

[125] Martin J. Kühn et al. “Assessment of Effective Mitigation and Prediction of the Spread of SARS-CoV-2 in Germany Using Demographic Information and Spatial Resolution.” In: *Mathematical Biosciences* 339 (Sept. 2021), p. 108648. ISSN: 0025-5564. DOI: 10.1016/j.mbs.2021.108648.

[126] Joaquín Osorio-Arjona, Jiri Horak, Radek Svoboda, and Yolanda García-Ruíz. “Social Media Semantic Perceptions on Madrid Metro System: Using Twitter Data to Link Complaints to Space.” In: *Sustainable Cities and Society* 64 (Jan. 2021), p. 102530. ISSN: 2210-6707. DOI: 10.1016/j.scs.2020.102530.

[127] Martin Klotz, Michael Wurm, and Hannes Taubenböck. “Der Werkzeugkasten der urbanen Fernerkundung – Daten und Produkte.” In: *Globale Urbanisierung: Perspektive aus dem All.*

Ed. by Hannes Taubenböck, Michael Wurm, Thomas Esch, and Stefan Dech. Berlin, Heidelberg: Springer, 2015, pp. 29–38. ISBN: 978-3-662-44841-0. DOI: 10.1007/978-3-662-44841-0_5.

[128] Qiong Hu, Wenbin Wu, Tian Xia, Qiangyi Yu, Peng Yang, Zhengguo Li, and Qian Song. "Exploring the Use of Google Earth Imagery and Object-Based Methods in Land Use/Cover Mapping." In: *Remote Sensing* 5.11 (Nov. 2013), pp. 6026–6042. ISSN: 2072-4292. DOI: 10.3390/rs5116026.

[129] Yi Yang and Shawn Newsam. "Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification." In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '10. New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 270–279. ISBN: 978-1-4503-0428-3. DOI: 10.1145/1869790.1869829.

[130] Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks." In: *arXiv:1508.00092 [cs]* (Aug. 2015). arXiv: 1508.00092 [cs].

[131] Adrian Albert, Jasleen Kaur, and Marta C. Gonzalez. "Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 1357–1366. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3098070.

[132] *Copernicus Land Monitoring Service - Urban Atlas — European Environment Agency*. https://www.eea.europa.eu/data-and-maps/data/copernicus-land-monitoring-service-urban-atlas. Data.

[133] Shougeng Hu and Le Wang. "Automated Urban Land-Use Classification with Remote Sensing." In: *International Journal of Remote Sensing* 34.3 (Feb. 2013), pp. 790–803. ISSN: 0143-1161. DOI: 10.1080/01431161.2012.714510.

[134] Ivan Elias Ruiz Hernandez and Wenzhong Shi. "A Random Forests Classification Method for Urban Land-Use Mapping Integrating Spatial Metrics and Texture Analysis." In: *International Journal of Remote Sensing* 39.4 (Feb. 2018), pp. 1175–1198. ISSN: 0143-1161. DOI: 10.1080/01431161.2017.1395968.

[135] Gong Cheng, Junwei Han, and Xiaoqiang Lu. "Remote Sensing Image Scene Classification: Benchmark and State of the Art." In: *Proceedings of the IEEE* 105.10 (Oct. 2017), pp. 1865–1883. ISSN: 1558-2256. DOI: 10.1109/JPROC.2017.2675998.

[136] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs." In: *IEEE Transactions on Geoscience and Remote Sensing* 56.5 (May 2018), pp. 2811–2821. ISSN: 1558-0644. DOI: 10.1109/TGRS.2017.2783902.

[137] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. "Deep Metric Learning via Lifted Structured Feature Embedding." In: *arXiv:1511.06452 [cs]* (Nov. 2015). arXiv: 1511.06452 [cs].

[138] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. "Functional Map of the World." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 6172–6180. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00646.

[139] Rodrigo Minetto, Mauricio Pamplona Segundo, and Sudeep Sarkar. "Hydra: An Ensemble of Convolutional Neural Networks for Geospatial Land Classification." In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (Sept. 2019), pp. 6530–6541. ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2019.2906883. arXiv: 1802.03518.

[140] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo, and Gui-Song Xia. "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 3735–3756. ISSN: 2151-1535. DOI: 10.1109/JSTARS.2020.3005403.

[141] Manuel Campos-Taberner, Francisco Javier García-Haro, Beatriz Martínez, Emma Izquierdo-Verdiguier, Clement Atzberger, Gustau Camps-Valls, and María Amparo Gilabert. "Understanding Deep Learning in Land Use Classification Based on Sentinel-2 Time Series." In: *Scientific Reports* 10.1 (Oct. 2020), p. 17188. ISSN: 2045-2322. DOI: 10.1038/s41598-020-74215-5.

[142] Ce Zhang, Isabel Sargent, Xin Pan, Huapeng Li, Andy Gardiner, Jonathon Hare, and Peter M. Atkinson. "Joint Deep Learning for Land Cover and Land Use Classification." In: *Remote Sensing of Environment* 221 (Feb. 2019), pp. 173–187. ISSN: 0034-4257. DOI: 10.1016/j.rse.2018.11.014.

[143] I. D. Stewart and T. R. Oke. "Local Climate Zones for Urban Temperature Studies." In: *Bulletin of the American Meteorological Society* 93.12 (Dec. 2012), pp. 1879–1900. DOI: 10.1175/BAMS-D-11-00019.1.

[144] Chunping Qiu, Lichao Mou, Michael Schmitt, and Xiao Xiang Zhu. “Local Climate Zone-Based Urban Land Cover Classification from Multi-Seasonal Sentinel-2 Images with a Recurrent Residual Network.” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 154 (Aug. 2019), pp. 151–162. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2019.05.004.

[145] Chunping Qiu, Lichao Mou, Michael Schmitt, and Xiao Xiang Zhu. “Fusing Multiseasonal Sentinel-2 Imagery for Urban Land Cover Classification With Multibranch Residual Convolutional Neural Networks.” In: *IEEE Geoscience and Remote Sensing Letters* 17.10 (Oct. 2020), pp. 1787–1791. ISSN: 1558-0571. DOI: 10.1109/LGRS.2019.2953497.

[146] Scott Workman, Menghua Zhai, David J. Crandall, and Nathan Jacobs. “A Unified Model for Near and Remote Sensing.” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 2707–2716. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.293.

[147] Scott Workman, M. Usman Rafique, Hunter Blanton, and Nathan Jacobs. “Revisiting Near/Remote Sensing with Geospatial Attention.” In: *arXiv:2204.01807 [cs]* (Apr. 2022). arXiv: 2204.01807 [cs].

[148] Shivangi Srivastava, John E. Vargas-Muñoz, and Devis Tuia. “Understanding Urban Landuse from the above and Ground Perspectives: A Deep Learning, Multimodal Solution.” In: *Remote Sensing of Environment* 228 (July 2019), pp. 129–143. ISSN: 00344257. DOI: 10.1016/j.rse.2019.04.014.

[149] A. Leichter, D. Wittich, F. Rottensteiner, M. Werner, and M. Sester. “IMPROVED CLASSIFICATION OF SATELLITE IMAGERY USING SPATIAL FEATURE MAPS EXTRACTED FROM SOCIAL MEDIA.” In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLII-4. Copernicus GmbH, Sept. 2018, pp. 335–342. DOI: 10.5194/isprs-archives-XLII-4-335-2018.

[150] Yuan Zhang, Qiangzi Li, Huiping Huang, Wei Wu, Xin Du, and Hongyan Wang. “The Combined Use of Remote Sensing and Social Sensing Data in Fine-Grained Urban Land Use Mapping: A Case Study in Beijing, China.” In: *Remote Sensing* 9.9 (Sept. 2017), p. 865. ISSN: 2072-4292. DOI: 10.3390/rs9090865.

[151] Anqi Lin, Xiaomeng Sun, Hao Wu, Wenting Luo, Danyang Wang, Dantong Zhong, Zhongming Wang, Lanting Zhao, and Jiang Zhu. “Identifying Urban Building Function by Integrating Remote Sensing Imagery and POI Data.” In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 8864–8875. ISSN: 2151-1535. DOI: 10.1109/JSTARS.2021.3107543.

[152] Tawfiq Salem, Scott Workman, and Nathan Jacobs. "Learning a Dynamic Map of Visual Appearance." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 12432–12441. ISBN: 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.01245.

[153] Nathan Jacobs, Nathaniel Roman, and Robert Pless. "Consistent Temporal Variations in Many Outdoor Scenes." In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. June 2007, pp. 1–6. DOI: 10.1109/CVPR.2007.383258.

[154] Nathan Jacobs, Walker Burgin, Nick Fridrich, Austin Abrams, Kylia Miskell, Bobby H. Braswell, Andrew D. Richardson, and Robert Pless. "The Global Network of Outdoor Webcams: Properties and Applications." In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '09. New York, NY, USA: Association for Computing Machinery, Nov. 2009, pp. 111–120. ISBN: 978-1-60558-649-6. DOI: 10.1145/1653771.1653789.

[155] Pedram Ghamisi et al. "Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art." In: *IEEE Geoscience and Remote Sensing Magazine* 7.1 (Mar. 2019), pp. 6–39. ISSN: 2168-6831. DOI: 10.1109/MGRS.2018.2890023.

[156] Xiao Xiang Zhu et al. "So2Sat LCZ42: A Benchmark Data Set for the Classification of Global Local Climate Zones [Software and Data Sets]." In: *IEEE Geoscience and Remote Sensing Magazine* 8.3 (Sept. 2020), pp. 76–89. ISSN: 2168-6831. DOI: 10.1109/MGRS.2020.2964708.

[157] *ERC Starting Grant So2Sat*.

[158] *About OpenStreetMap*. https://wiki.openstreetmap.org/wiki/About_OpenStreetMap.

[159] *OpenStreetMap Statistics*. https://www.openstreetmap.org/stats/data_stats.html.

[160] Mordechai Haklay. "How Good Is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets." In: *Environment and Planning B: Planning and Design* 37.4 (Aug. 2010), pp. 682–703. ISSN: 0265-8135. DOI: 10.1068/b35097.

[161] Yongyang Xu, Zhanlong Chen, Zhong Xie, and Liang Wu. "Quality Assessment of Building Footprint Data Using a Deep Autoencoder Network." In: *International Journal of Geographical Information Science* 31.10 (Oct. 2017), pp. 1929–1951. ISSN: 1365-8816. DOI: 10.1080/13658816.2017.1341632.

[162] Eike Jens Hoffmann, Martin Werner, and Xiao Xiang Zhu. "Quality Assessment of Semantic Tags in OpenStreetMap." In: *IOP Conference Series: Earth and Environmental Science* 509.1 (June 2020), p. 012025. ISSN: 1755-1315. DOI: 10.1088/1755-1315/509/1/012025.

[163] Mara Hvistendahl. "Foreigners Run Afoul of China's Tightening Secrecy Rules." In: *Science* 339.6118 (Jan. 2013), pp. 384–385. DOI: 10.1126/science.339.6118.384.

[164] Steven Delwart. "Sentinel-2 User Handbook." In: 1 (), p. 64.

[165] Siamak Khorram, Cynthia F. van der Wiele, Frank H. Koch, Stacy A. C. Nelson, and Matthew D. Potts. "Data Acquisition." In: *Principles of Applied Remote Sensing*. Ed. by Siamak Khorram, Cynthia F. van der Wiele, Frank H. Koch, Stacy A. C. Nelson, and Matthew D. Potts. Cham: Springer International Publishing, 2016, pp. 21–67. ISBN: 978-3-319-22560-9. DOI: 10.1007/978-3-319-22560-9_2.

[166] Saman Ghaffarian and Salar Ghaffarian. "Automatic Building Detection Based on Purposive FastICA (PFICA) Algorithm Using Monocular High Resolution Google Earth Images." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 97 (Nov. 2014), pp. 152–159. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2014.08.017.

[167] Qinchuan Zhang, Yunhong Wang, Qingjie Liu, Xiangyu Liu, and Wei Wang. "CNN Based Suburban Building Detection Using Monocular High Resolution Google Earth Images." In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2016, pp. 661–664. DOI: 10.1109/IGARSS.2016.7729166.

[168] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. "Google Street View: Capturing the World at Street Level." In: *Computer* 43.6 (June 2010), pp. 32–38. ISSN: 1558-0814. DOI: 10.1109/MC.2010.170.

[169] Sterling Quinn and Luis Alvarez León. "Every Single Street? Rethinking Full Coverage across Street-Level Imagery Platforms." In: *Transactions in GIS* 23.6 (2019), pp. 1251–1272. ISSN: 1467-9671. DOI: 10.1111/tgis.12571.

[170] *Street View Static API Overview | Google Developers*. https://developers.google.com/

[171] *Flickr Blog: What We Accomplished in 2020*. Dec. 2020.

[172] *Flickr Jobs Details*. https://www.flickr.com/jobs.

[173] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data Using T-SNE." In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. ISSN: 1533-7928.

[174] Tobias Leichtle, Tobia Lakes, Xiao Xiang Zhu, and Hannes Taubenböck. "Has Dongying Developed to a Ghost City? - Evidence from Multi-Temporal Population Estimation Based on VHR Remote Sensing and Census Counts." In: *Computers, Environment and Urban Systems* 78 (Nov. 2019), p. 101372. ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2019.101372.

[175] Twitter Inc. *Twitter Global Impact Report*. 2020.

[176] *Twitter: Monthly Active Users Worldwide*. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/.

[177] *POST Statuses/Filter*. https://developer.twitter.com/en/docs/twitter-api/v1/tweets/filter-realtime/api-reference/post-statuses-filter.

[178] TwitterSupport. *Twitter Support Announcing Disabling Geolocations*. https://twitter.com/TwitterSupport/status/1141039841993355264.

[179] Anna Kruspe, Matthias Häberle, Eike Jens Hoffmann, Samyo Rode-Hasinger, Karam Abdulahhad, and Xiao Xiang Zhu. "Changes in Twitter Geolocations: Insights and Suggestions for Future Usage." In: *arXiv:2108.12251 [cs]* (Sept. 2021). arXiv: 2108.12251 [cs].

[180] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. "Do Better ImageNet Models Transfer Better?" In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019, pp. 2656–2666. DOI: 10.1109/CVPR.2019.00277.

[181] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge." In: *International Journal of Computer Vision* 115.3 (Dec. 2015), pp. 211–252. ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y.

[182] Narseo Vallina-Rodriguez, Jon Crowcroft, Alessandro Finamore, Yan Grunenberger, and Konstantina Papagiannaki. "When Assistance Becomes Dependence: Characterizing the Costs and Inefficiencies of A-GPS." In: *ACM SIGMOBILE Mobile Computing and Communications Review* 17.4 (Dec. 2013), pp. 3–14. ISSN: 1559-1662. DOI: 10.1145/2557968.2557970.

[183] Shivangi Srivastava, Sylvain Lobry, Devis Tuia, and John E Vargas. "Land-Use Characterisation Using Google Street View Pictures and OpenStreetMap." In: *21st AGILE Conference on Geographic Information Science (2018)*. Lund, Sweden, 2018, p. 5.

[184] Fei Liu, Yong Wang, Fan-Chuan Wang, Yong-Zheng Zhang, and Jie Lin. "Intelligent and Secure Content-Based Image Retrieval for Mobile Users." In: *IEEE Access* 7 (2019), pp. 119209–119222. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2935222.

[185] Qi Wang, Jingxiang Lai, Kai Xu, Wenyin Liu, and Liang Lei. "Beauty Product Image Retrieval Based on Multi-Feature Fusion and Feature Aggregation." In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 2063–2067. ISBN: 978-1-4503-5665-7. DOI: 10.1145/3240508.3266431.

[186] Inhae Ha, Hongjo Kim, Somin Park, and Hyoungkwan Kim. "Image Retrieval Using BIM and Features from Pretrained VGG Network for Indoor Localization." In: *Building and Environment* 140 (Aug. 2018), pp. 23–31. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.05.026.

[187] Yun Ge, Shunliang Jiang, Qingyong Xu, Changlong Jiang, and Famao Ye. "Exploiting Representations from Pre-Trained Convolutional Neural Networks for High-Resolution Remote Sensing Image Retrieval." In: *Multimedia Tools and Applications* 77.13 (July 2018), pp. 17489–17515. ISSN: 1380-7501. DOI: 10.1007/s11042-017-5314-5.

[188] {{Camera & Imaging Products Association}}. *Exchangeable Image File Formaat for Digital Still Cameras: Exif Version 2.32*. May 2019.

[189] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. "Generalisation in Humans and Deep Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[190] C. E. Shannon. "A Mathematical Theory of Communication." In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[191] Michael Batty. "Spatial Entropy." In: *Geographical Analysis* 6.1 (1974), pp. 1–31. ISSN: 1538-4632. DOI: 10.1111/j.1538-4632.1974.tb01014.x.

[192] Anders Karlström and Vania Ceccato. "A New Information Theoretical Measure of Global and Local Spatial Association." In: *The Review of Regional Research* 22 (2002), pp. 13–40.

[193] Linda Altieri, Daniela Cocchi, and Giulia Roli. "SpatEntropy: Spatial Entropy Measures in R." In: *arXiv:1804.05521 [stat]* (Apr. 2018). arXiv: 1804.05521 [stat].

[194] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springers. ISBN: 978-0-387-31073-2.

[195] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm." In: *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. ICML'96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., July 1996, pp. 148–156. ISBN: 978-1-55860-419-3.

[196] Jerome H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine.” In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. ISSN: 0090-5364.

[197] *Sklearn.Ensemble.GradientBoostingRegressor*. https://scikit-learn/stable/modules/generated/sklear

[198] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. “CatBoost: Unbiased Boosting with Categorical Features.” In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., Dec. 2018, pp. 6639–6649.

[199] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree.” In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 3149–3157. ISBN: 978-1-5108-6096-4.

[200] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785.

[201] Didrik Nielsen. “Tree Boosting With XGBoost - Why Does XGBoost Win "Every" Machine Learning Competition?” PhD thesis. NTNU, 2016.

[202] Francois Chollet. *Deep Learning with Python*. 1st Edition. Shelter Island, New York: Manning, Dec. 2017. ISBN: 978-1-61729-443-3.

[203] Delphis F. Levia et al. “Homogenization of the Terrestrial Water Cycle.” In: *Nature Geoscience* 13.10 (Oct. 2020), pp. 656–658. ISSN: 1752-0908. DOI: 10.1038/s41561-020-0641-y.

[204] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. “Foundations on Imbalanced Classification.” In: *Learning from Imbalanced Data Sets*. Ed. by Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. Cham: Springer International Publishing, 2018, pp. 19–46. ISBN: 978-3-319-98074-4. DOI: 10.1007/978-3-319-98074-4_2.

[205] Dimitrios Marmanis, Mihai Datcu, Thomas Esch, and Uwe Stilla. “Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks.” In: *IEEE Geoscience and Remote Sensing Letters* 13.1 (Jan. 2016), pp. 105–109. ISSN: 1558-0571. DOI: 10.1109/LGRS.2015.2499239.

[206] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. “Transfer Learning from Deep Features for Remote Sensing and Poverty Mapping.” In: *Thirtieth AAAI Conference on Artificial Intelligence*. Mar. 2016.

[207] Mendel Giezen, Stella Balikci, and Rowan Arundel. “Using Remote Sensing to Analyse Net Land-Use Change from Conflicting Sustainability Policies: The Case of Amsterdam.” In: *ISPRS International Journal of Geo-Information* 7.9 (Sept. 2018), p. 381. ISSN: 2220-9964. DOI: 10.3390/ijgi7090381.

[208] Luis Gómez-Chova, Devis Tuia, Gabriele Moser, and Gustau Camps-Valls. “Multimodal Classification of Remote Sensing Images: A Review and Future Directions.” In: *Proceedings of the IEEE* 103.9 (Sept. 2015), pp. 1560–1584. ISSN: 1558-2256. DOI: 10.1109/JPROC.2015.2449668.

[209] Konrad Heidler, Lichao Mou, Di Hu, Pu Jin, Guangyao Li, Chuang Gan, Ji-Rong Wen, and Xiao Xiang Zhu. “Self-Supervised Audiovisual Representation Learning for Remote Sensing Data.” In: *arXiv:2108.00688 [cs]* (Aug. 2021). arXiv: 2108.00688 [cs].

[210] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. “On Calibration of Modern Neural Networks.” In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 1321–1330.

[211] Alexandru Niculescu-Mizil and Rich Caruana. “Predicting Good Probabilities with Supervised Learning.” In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML ’05. New York, NY, USA: Association for Computing Machinery, Aug. 2005, pp. 625–632. ISBN: 978-1-59593-180-1. DOI: 10.1145/1102351.1102430.

[212] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision.” In: *arXiv:2103.00020 [cs]* (Feb. 2021). arXiv: 2103.00020 [cs].

[213] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. “Zero-Shot Text-to-Image Generation.” In: *arXiv:2102.12092 [cs]* (Feb. 2021). arXiv: 2102.12092 [cs].

[214] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. “MAGMA – Multimodal Augmentation of Generative Models through Adapter-based Finetuning.” In: *arXiv:2112.05253 [cs]* (Dec. 2021). arXiv: 2112.05253 [cs].

[215] Yuansheng Hua, Sylvain Lobry, Lichao Mou, Devis Tuia, and Xiao Xiang Zhu. “Learning Multi-Label Aerial Image Classification Under Label Noise: A Regularization Approach Using Word Embeddings.” In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. Sept. 2020, pp. 525–528. DOI: 10.1109/IGARSS39084.2020.9324069.

[216] Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greeff, and Andreas Krause. “Hierarchical Image Classification Using Entailment Cone Embeddings.” In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2020, pp. 3649–3658. DOI: 10.1109/CVPRW50498.2020.00426.

[217] Bin Chen, Bing Xu, and Peng Gong. “Mapping Essential Urban Land Use Categories (EULUC) Using Geospatial Big Data: Progress, Challenges, and Opportunities.” In: *Big Earth Data* 5.3 (July 2021), pp. 410–441. ISSN: 2096-4471. DOI: 10.1080/20964471.2021.1939243.

[218] Hannes Taubenböck, Jeroen Staab, Xiao Xiang Zhu, Christian Geiß, Stefan Dech, and Michael Wurm. “Are the Poor Digitally Left Behind? Indications of Urban Divides Based on Remote Sensing and Twitter Data.” In: *ISPRS International Journal of Geo-Information* 7.8 (Aug. 2018), p. 304. ISSN: 2220-9964. DOI: 10.3390/ijgi7080304.

[219] Jun-Ho Choi and Jong-Seok Lee. “EmbraceNet: A Robust Deep Learning Architecture for Multimodal Classification.” In: *Information Fusion* 51 (Nov. 2019), pp. 259–270. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.02.010.

[220] Ibtihaal M. Hameed, Sadiq H. Abdulhussain, and Basheera M. Mahmmod. “Content-Based Image Retrieval: A Review of Recent Trends.” In: *Cogent Engineering* 8.1 (Jan. 2021). Ed. by D T Pham, p. 1927469. ISSN: null. DOI: 10.1080/23311916.2021.1927469.

[221] Zhichen Zhao, Huimin Ma, and Shaodi You. “Single Image Action Recognition Using Semantic Body Part Actions.” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 3411–3419. DOI: 10.1109/ICCV.2017.367.

[222] Deeptha Girish, Vineeta Singh, and Anca Ralescu. “Understanding Action Recognition in Still Images.” In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2020, pp. 1523–1529. DOI: 10.1109/CVPRW50498.2020.00193.

[223] Michael F. Goodchild. “Citizens as Sensors: The World of Volunteered Geography.” In: *GeoJournal* 69.4 (Aug. 2007), pp. 211–221. ISSN: 1572-9893. DOI: 10.1007/s10708-007-9111-y.

[224] Wojciech Sirko, Sergii Kashubin, Marvin Ritter, Abigail Annkah, Yasser Salah Eddine Bouchareb, Yann Dauphin, Daniel Keysers, Maxim Neumann, Moustapha Cisse, and John Quinn. "Continental-Scale Building Detection from High Resolution Satellite Imagery." In: *arXiv:2107.12283 [cs]* (July 2021). arXiv: 2107.12283 [cs].

[225] Claudio Persello, Jan Dirk Wegner, Ronny Hansch, Devis Tuia, Pedram Ghamisi, Mila Koeva, and Gustau Camps-Valls. "Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current Approaches, Open Challenges, and Future Opportunities." In: *IEEE Geoscience and Remote Sensing Magazine* (2022), pp. 2–30. ISSN: 2168-6831. DOI: 10.1109/MGRS.2021.3136100.

[226] Ivan Henderson V. Gue, Aristotle T. Ubando, Ming-Lang Tseng, and Raymond R. Tan. "Artificial Neural Networks for Sustainable Development: A Critical Review." In: *Clean Technologies and Environmental Policy* 22.7 (Sept. 2020), pp. 1449–1465. ISSN: 1618-9558. DOI: 10.1007/s10098-020-01883-2.

[227] John E. Vargas-Munoz, Shivangi Srivastava, Devis Tuia, and Alexandre X. Falcão. "OpenStreetMap: Challenges and Opportunities in Machine Learning and Remote Sensing." In: *IEEE Geoscience and Remote Sensing Magazine* 9.1 (Mar. 2021), pp. 184–199. ISSN: 2168-6831. DOI: 10.1109/MGRS.2020.2994107.

[228] Emma Dahlin. "Mind the Gap! On the Future of AI Research." In: *Humanities and Social Sciences Communications* 8.1 (Mar. 2021), pp. 1–4. ISSN: 2662-9992. DOI: 10.1057/s41599-021-00750-9.

[229] *Key:Building – OpenStreetMap Wiki*. https://wiki.openstreetmap.org/wiki/Key:build

[230] *Key:Amenity – OpenStreetMap Wiki*. https://wiki.openstreetmap.org/wiki/Key:amer