Geo-information Science and Remote Sensing

Thesis Report GIRS-2018-42

How to build a building classifier:

A data-driven approach to characterize building functions from streetview images with computer vision and machine learning in the city of Amsterdam

David Swinkels

28 October 2018

WAGENINGEN
UNIVERSITY & RESEARCH

Figure on front: City of Amsterdam (2017). Non-residential functions of city center (function map). Retrieved at 10 September 2017 from https://maps.amsterdam.nl/functiekaart/?LANG=en.

How to build a building classifier

A data-driven approach to characterize building functions from streetview images with computer vision and machine learning in the city of Amsterdam

David Tai Wai Swinkels

Registration number 920714820090

Supervisors:

dr. Devis Tuia

MSc Shivangi Srivastava

A thesis submitted in partial fulfillment of the degree of Master of Science
at Wageningen University and Research Centre,
The Netherlands.

28 October 2018

Wageningen, The Netherlands

Course code: GRS-80436

Thesis code: GIRS-2018 -42

Wageningen University and Research Centre

Laboratory of Geo-Information Science and Remote Sensing

Preface

Dear reader,

In this preface, I would like to share my passion for maps and map-making. Historically, maps showed the relative location of landmarks for navigational purposes. Around 1500 Europeans started to regularly travel around on voyages across oceans. These long voyages demanded more accurate maps. Cartography became a science and maps started to have latitude, longitude and a projection. At the end of the 20th century, most parts of the world were mapped, and software engineering transformed the paper map into a digital map. In the digital era maps became accessible to all, 3D (with altitude), 4D (with time), crowdsourced and used imagery from satellites, airplanes and drones. Currently in 2018 humans still use a (digital) map to navigate through the world. However, autonomous vehicles based on artificial intelligence and accurate local sensing are being tested on the road. The sensors from these vehicles will generate loads of spatial data. In this thesis, streetview images will show what functions are located where based on information from cameras on cars. As technology evolves, maps and map-making will evolve too.

I would like to thank my supervisors, dr. Devis Tuia and MSc. Shivangi Srivastava, lecturers at the master Geo-Information Science, fellow students, friends, and family for support during the thesis.

Enjoy reading!

David Swinkels

Abstract

(EN) - More and more information about cities is being captured by satellites, planes, drones, smartphones and recently from cameras on cars. Pictures from cameras on cars can help to solve automatic urban land use mapping, i.e. to detect shops, offices, industrial or residential buildings. Streetview photography platforms, such as Google Street View and Mapillary, have large datasets of ground-level images available. Advances in machine learning and computer vision, i.e. convolutional neural networks, make it possible to classify building functions from streetview images. The research in this thesis showed that building functions can accurately be predicted with streetview images. Also, it was observed that if buildings were more recently built and streetview images had a smaller distance from the camera to the building, the prediction accuracy of the building functions was significantly higher.

(NL) - Méér en méér informatie van steden wordt verzameld door satellieten, vliegtuigen, drones, smartphones en recentelijk ook van camera's op auto's. Foto's van camera's op auto's helpen om automatisch stedelijk landgebruik te bepalen, i.e. het detecteren van winkels, kantoren, industrie of woningen. Streetview fotografie platforms, zoals Google Street View of Mapillary, hebben grote datasets van grondbeelden beschikbaar. Verbeteringen in machine learning en computer vision, i.e. convolutional neural networks, maken het mogelijk om gebouwfuncties te voorspellen op basis van streetview beelden. Onderzoek in deze thesis toont aan dat gebouwfuncties accuraat voorspeld kunnen worden met streetview beelden. Er werd geöbserveerd dat gebouwen welke recenter gebouwd waren en streetview beelden welke een kleinere afstand hadden van camera naar gebouw, significant geassocieerd waren met een hogere accuraatheid van de voorspelde gebouwfunctie.

Table of Contents

List of abbreviations

ANOVA	= Analysis of Variance
API	= Application Program Interface
CNN	= Convolutional Neural Network
CSV	= comma separated value file format
EPSG	= European Petroleum Survey Group publishes a database of coordinate system information plus related documents on map projections (ESRI, 2016)
FOV	= Field of View
GDPR	= General Data Protection Regulation
GPS	= Global Positioning System
GPU	= Graphical Processing Unit
Jpeg/jpg	= Joint Photographic Experts Group file format
OS	= Operating System
ReLU	= Rectified Linear Unit
RGB	= Red Green Blue
SQL	= Structured Query Language

Introduction

Efforts are increasingly being made to automate urban land use mapping by utilizing pictures taken from top views, i.e. by satellite, plane or drone, and recently side views, i.e. cars or smartphone. The new side perspective allows for many new possibilities. Google has been sensing worldwide where house numbers are from cameras on cars (Goodfellow et al., 2013) and are going to sense opening times by increasing the resolution of Google Street View images (Wired, 2017). One problem that was difficult to solve with a top perspective is automatic urban land use mapping (Bechtel et al., 2015). The goal of this research will be to characterize building functions in urban areas from streetview images.

The mapping of building functions requires knowledge of indoor activities. Information about indoor land use, such as a shopping or sports center, often come from surveys, which are expensive and subjective (Frias-Martinez et al., 2012; Jokar et al., 2013). Recently remote sensing was used to predict land use, but remote sensing has difficulty predicting indoor land use with its top view (Leung & Newsam, 2015). Therefore, Leung and Newsam (2015) argue that proximate sensing, which relies on ground level images, could be another source of information to map indoor land use.

Proximate sensing or image-driven mapping discerns information from big datasets of ground-level images. Photo-sharing platforms (e.g. Flickr, Instagram, Panoramio) and streetview photography platforms (e.g. Mapillary, Google Street View) have large datasets of ground-level images publicly available via APIs. An API is the part of a web server that receives requests and sends responses. This data is already being used in research to predict land use or building functions. For example, Sitthi et al. (2016), Tracewski et al. (2017), Workman et al. (2017) and Srivastava et al. (2018a) used images from photo-sharing platforms to make land use maps.

Social media and streetview photography provide a different perspective on the ground. Social media images are generated by users, show individual perspectives and are often located around tourist hotspots (Leung & Newsam, 2015). These images are useful to identify landmarks (Chen et al., 2017), safety perception (Dubey et al., 2016) or city identity (Zhou et al., 2014). Streetview images (e.g. from Google Streetview or Mapillary) are globally available, geographically spread out over the city via the street network, often provide a fixed 360° ground perspective and have been continuously gathered since 2007. Streetview images are used to update place labels on Google Maps (Wired, 2017), to rectify land use of parcels (Pulighe et al., 2015; Verhoeve et al., 2015), to monitor urban appearance over time (Naik et al., 2014) and to map land use and building functions (Workman et al., 2017). Streetview images have a good perspective on buildings to characterize building functions.

Models using machine learning and computer vision can classify objects in images. Machine learning uses statistical techniques to learn patterns from given data to make correct predictions. Computer vision analyses images to produce a high-level understanding of image content. One well-known computer vision model that uses deep learning is the CNN (LeCun et al., 1998). The CNN [Convolutional Neural Network] is successful in various tasks, like image classification, object detection, and object segmentation, and has become the most commonly used image classifier in computer vision (Guo et al., 2016). These advances in machine learning make it possible to classify building functions on an object level from images (see figure 0.1).

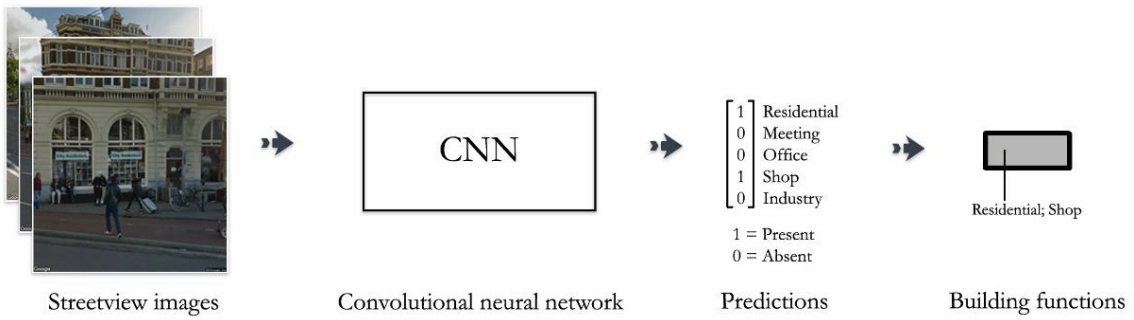


Figure 0.1: Characterizing building functions from streetview images

Research questions

The previous paragraphs about proximate sensing, geo-tagged images, and computer vision gave a context for this research. The purpose of this research is to gain information about building functions from streetview images. By automating the acquisition process of building functions, survey costs can be lowered, and map updates can be faster. Insights from this research will extend academic knowledge by inspecting the importance of building or streetview image characteristics. Furthermore, knowledge is expanded by performing research in a new geography, namely Amsterdam (see figure 0.2). The purpose and case study together lead to the main research question:

How can building functions be characterized by streetview images in Amsterdam?

The main question focuses on the process of streetview image mapping and is split into sub-questions. To clarify the questions, some definitions are given. Streetview images in this research are defined as images, which have a view on streets and buildings. A building function is defined as the functional usage of a building, such as a shop, office, meeting, residential or industrial use. Every building can have multiple functions. Sub-questions are:

- How can streetview images and building functions be acquired in Amsterdam?
- How accurately can streetview images sense buildings in Amsterdam?
- How accurately can convolutional neural networks characterize building functions from streetview images in Amsterdam?
- What building and streetview image characteristics are associated with correct predictions of building functions from streetview images?

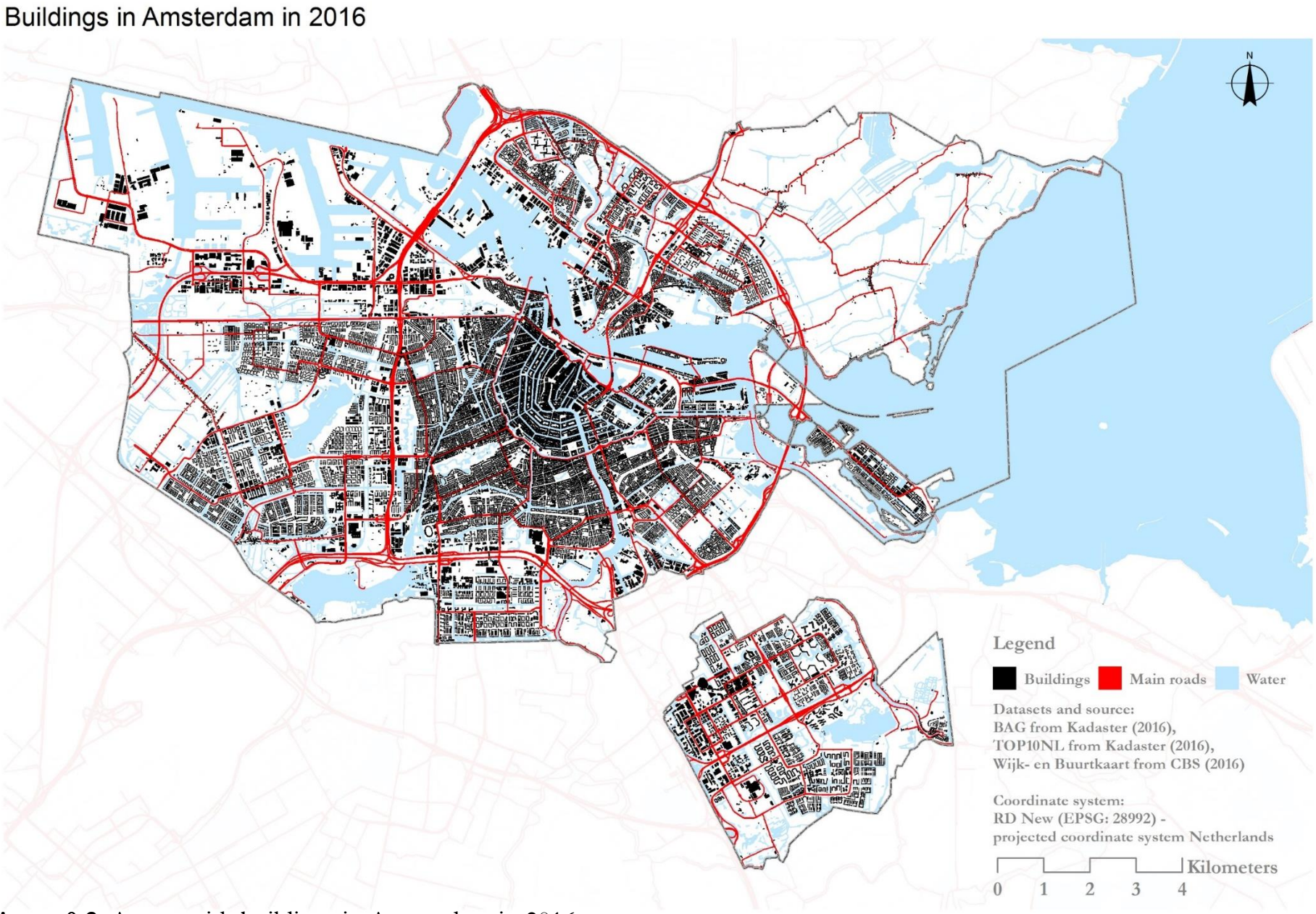


Figure 0.2: A map with buildings in Amsterdam in 2016

Reading guide

The report has several chapters, which follow the introduction, methodology, results and discussion framework (see table 0.1).

Table 0.1: Reading guide and structure of the report							
Chapter name	Introduction	Theory	Methodology	Experiments & Results	Discussion	Conclusion	Appendix
Chapter		1	2	3	4	5	
Content	Research questions	Land use maps	Software setup	Data acquisition	Data acquisition	Research answers	USB Content
	Reading guide	Streetview images	Data acquisition	Data accuracy	Data accuracy	Further research	
		Machine learning and computer vision	Data accuracy	Building classification	Building classification		
			Building classification				

Chapter 1 - Theory

Firstly, an academic understanding is needed. The academic literature will be reviewed for multiple topics: land use maps, streetview images and machine learning for computer vision.

1.1 Land use maps: accuracy and sources

Building functions are an integral part of land use maps. Buildings are the blocks that define how humans use neighborhoods and cities (Steiniger et al., 2008). If there are a lot of shops in a street, the street is a shopping street. If there are multiple shopping streets in proximity, the neighborhood is used for shopping. Besides buildings, land use maps show more information, such as fields, parks or farmland. Land use maps portray all human activities in the environment (Erb et al., 2007; Jokar et al., 2013) and are used in environmental studies, spatial planning and urban management (Jokar et al., 2013). Currently, in urban areas, there is a lack of accurate land use data that is globally available via a consistent methodology (Bechtel et al., 2015). Buildings largely determine land use in urban areas and land use maps are used to improve spatial planning.

Accurate land use maps require accurate data acquisition. Accuracy of land use maps is important for several reasons: environmental management should make decisions on correct data, the errors or uncertainties from land use maps should not propagate in land use models and land use maps as an end product should be accurate (Bechtel et al., 2015; Jokar et al., 2013). For example, land use maps are used in regression models to predict air pollution based on the industrial, residential or infrastructural use of land (Eeftens et al., 2012) or are used as input for climate models (Bechtel et al., 2015). The prediction of pollution or climate should have errors within acceptable limits due to errors in the land use map. Policymakers and researchers depend on the quality of land use data to make decisions for improvement of environmental quality and resource management.

There are several potential sources of information for land use maps in urban areas: census data, remote sensing or proximate sensing. Each method has its advantages and disadvantages.

Firstly, census data about building and parcels is gathered by governmental organizations in the Netherlands (BAG, 2010) and other countries (Erb et al., 2007). Advantages are mapping based on domain knowledge of local experts, high spatial resolution, homogenous cover and often high accuracy. Disadvantages are financial costs and administrative work.

Secondly, volunteered information about land use is gathered by volunteers to create land use maps. Some examples are GeoWiki, OpenStreetMap or Verbeter de kaart (Fritz et al., 2012; Jokar et al., 2013; Kadaster, 2016c). Advantages are fast updates, low expenditure, and local knowledge. Disadvantages are the subjective source and possible vandalism.

Thirdly, satellite images provide information by remotely sensing the environment. Advantages of satellite images are the global scale, high accuracy and free availability (Bechtel et al., 2015). Some disadvantages of deriving land use maps from satellite imagery are clouds (Kovalskyy & Roy, 2013), its top view (Leung & Newsam, 2015) and coarse resolution.

Fourthly, planes sense the natural and built-up environment (Belgiu et al., 2014). Advantages are high spatial resolution and high accuracy. Aerial images have similar limitations as satellite images, but on top, they are costly and not globally available.

Fifthly, streetview images are a source for land use maps by sensing on the ground level what activities happen where in the built-up and natural environment (Kovalskyy & Roy, 2013; Leung & Newsam, 2015). Streetview images have the advantage of ground perspective, notable global availability, high spatial resolution, and high accuracy.

Land use maps can be based on census data, remote sensing, and proximate sensing. Streetview images are a new potential source for land use mapping.

1.2 Streetview images: geo-tagged images, panoramas, and privacy

Streetview images are images, taken from the street, that have a view on streets and buildings. Generally, two types can be distinguished: geo-tagged images from photo-sharing platforms and images from mapping platforms.

Geo-tagged images from photo-sharing platforms (e.g. Flickr, Panoramio or Instagram) are used in academic research to find the perspective of the individual. These platforms have large user bases and provide access to geotagged photos through APIs (Dubey et al., 2016). The geo-tagged images can help determine what human activities are done where. Not everywhere though; photos from photo-sharing platforms are limited to interesting places, such as cities, touristic places, and events (Leung & Newsam, 2015). Supplementary there are more limitations to images from photo-sharing platforms: privacy issues due to visible faces or license plates, no data quality check, non-standardized image quality, and non-standardized GPS. Advantages of images from photo-sharing platforms are individual perspective, high temporal resolution, and potential view inside the building.

Images from mapping platforms, such as Google Street View, are available through large global streetview image archives (Dubey et al., 2016). These image archives keep expanding and have both geo-tagged panoramas and images. Google Street View cars drive around the world every day with cameras and a GPS with a horizontal position accuracy of 2.5 meters (Google, 2018c; Khosla et al., 2014; Zhou et al., 2014). Google combined the data from the gyroscope, magnetometer, and cameras to make 360º panoramas, where the pitch (up-down) and compass heading can be changed. At every streetview location, the camera can be pointed towards an object. Via an API the image can be retrieved. Before Google provided 25,000 free streetview images per day, but Google changed policy and only allows roughly 28,000 free images per month (Google, 2018b). Google shares high-quality Google Street View images via an API.

Another mapping platform, Mapillary, stores and distributes streetview images from users for users. The crowdsourced images from Mapillary have similar limitations as images from photo-sharing platforms, i.e. Mapillary has non-standardized data acquisition. However, Mapillary has the benefit of free availability, higher temporal resolution, and some users of Mapillary (e.g. municipality of Amsterdam) providing high-quality 360° images through Mapillary. In the future, more streetview images are expected to come from autonomous vehicles, who need to sense the environment for navigation (Taneja et al., 2014; Wegner et al., 2016), which could be shared through platforms such as Mapillary or Google Street View.

Privacy issues are a big concern for streetview images. Persons, license plates and home addresses can be identified from images if there are no blurs. Privacy risks are addressed in Europe by national law and European law. Germany and Austria prohibited Google Street View cars to gather any data. European law states in GDPR article 5.1a (EuropeanParliament, 2016): "Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject ('lawfulness, fairness and transparency')". In this research buildings are the data subjects and there is no need to have personal data. Google Street View blurs faces and license plates in streetview images, that were gathered by Google. User-generated images on Google Street View do not have blurs. Other mapping platforms, such as Mapillary, or photo sharing platforms, such as Flickr or Instagram with user-generated images, do not have blurs either. The processing of streetview images requires ethical practices to stay within the national and European law.

1.3 Machine learning and computer vision: building classification

Large image archives can efficiently be analyzed with machine learning algorithms and computer vision. Machine learning algorithms are good at learning non-linear numerical patterns and computer vision has a tradition in identifying objects, edges, gradients or colors in images. A boom in machine learning can be attributed to advances in machine learning algorithms, increased computational power (e.g. GPU, TPU), lowered cost of computing hardware (Bengio et al., 2014; Guo et al., 2016), clustered parallel processing architectures and larger training datasets.

Deep learning is a branch of machine learning and is becoming a standard. Deep learning has multiple layers of information-processing in hierarchical architectures (Deng, 2014), is commonly based on artificial neural networks and has high accuracy. He et al. (2015) developed and trained a deep learning algorithm for a challenging 1000 class image recognition task that beat human level performance of 5.1% in image classification with a top-5 test error rate of 4.94%. With the increasing performance of deep learning models, deep learning became more commonly used in image recognition (Guo et al., 2016) and geoscience (Ghamisi et al., 2017; Zhu et al., 2017).

In deep learning, a non-linear predictive model is trained to fit patterns in the data. During training, there are two stages: forward and backward (Guo et al., 2016; LeCun et al., 2015). In the forward stage, the model predicts the output label from the input data. The backward stage checks the misclassification, i.e. loss, and changes the weights of the model to minimize the loss by back-propagation. This process of changing weights is iterated numerous times to gradually approach to the optimal answer. Therefore, the performance of deep learning models is heavily dependent on the quality of training data.

Convolutional neural networks are a specific subtype of deep learning models, that focus on image recognition and are based on a 2-dimensional convolution with weights shared in the image plane. The CNN architecture has two phases, feature learning and classification:

- Feature learning consists of an input image, convolutional layers, pooling layers, Rectified Linear Unit [ReLU] and an output vector (LeCun et al., 2015)(see figure 2.7). The convolutional layers activate certain features in the images. The ReLu function applies a non-linear function on the convolved feature images by only selecting positive values. The convolution and ReLu function together detect local motifs in colors, blobs or features invariant to the location in the image (LeCun et al., 2015). Pooling layers pass a window over the feature maps and take the average or maximum value of this window. After pooling a filter "sees" a larger part of the image.
- In the classification phase, the output vector is turned into a label via a fully connected neural layer and softmax for multiple labels or sigmoid layer for binary label output (see figure 2.7).

Both phases are learned in a single optimization, unlike traditional machine learning where features are pre-defined. To summarize, feature detection is learned invariant of location in the feature learning phase and in the classification phase, the outcome of detected features is turned into a label.

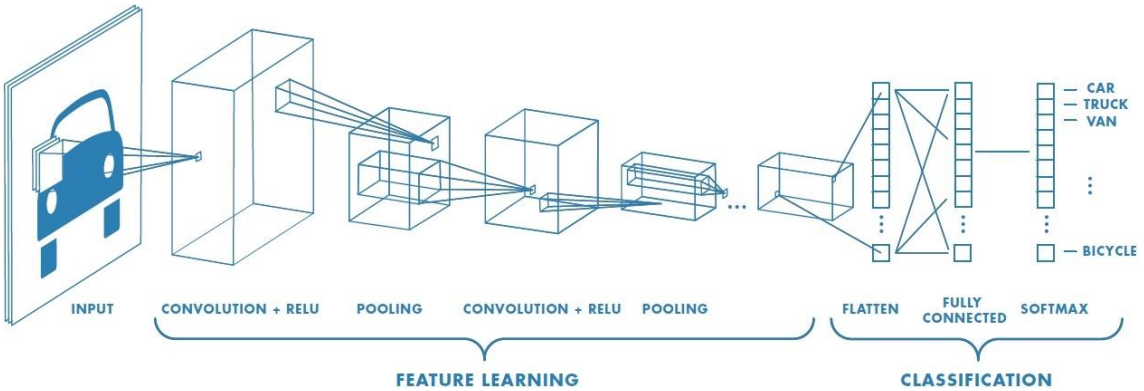


Figure 2.7: Architecture typical convolutional neural network (MathWorks, n.d.)

The architecture and trained weights of convolutional neural networks in the feature learning phase can be re-used. This process is also called re-training or fine-tuning (Guo et al., 2016). Training the weights of the feature learning phase requires a large amount of training data and computation power. To reduce computation time, the feature learning phase of a highly accurate CNN can be re-trained on an image dataset that shares similar types of features. Only the classification phase, specifically the final fully connected neural layer, must be re-trained to link detected features in images to output labels. Some well performing CNN models are Inception-v3 and MobileNetV1, which have been pre-trained on ImageNet (Howard et al., 2017). ImageNet is a collection of 14 million images with more than 1,000 diverse object classes (Deng et al., 2009), and is a popular dataset to train and benchmark computer vision models. It is possible to use CNN architectures trained on ImageNet for building characterization from streetview images because both image datasets share similar image features: RGB images, similar objects and a similar point of view.

1.4 Summarizing the theory

The insights from this literature review on land use maps, streetview images, and machine learning are summarized. Firstly, buildings are part of land use maps, land use maps require accurate information and current land use maps are often based on information from surveys or remote sensing. Secondly, big datasets of images are available from photo-sharing platforms and mapping platforms. Streetview images are a good source, because of the even spread, street perspective, 360º images, accurate position, and fewer privacy constraints. Lastly, machine learning and convolutional neural networks make it possible to characterize what is in streetview images. For all these reasons, this research chose to use streetview images to classify buildings as individual objects with pre-trained CNN models Inception-v3 and MobileNetV1.

Chapter 2 - Methodology

In this chapter, the methodology will be explained. The first section explains how to set up the software environment and three latter sections describe the research methodology.

2.1 Software setup

Software setup will be discussed: data management, hardware, software, and scripting setup.

Data management

Scripts and models have been run in a workspace structure. Small data, reports, scripts, and literature were stored on the M:/Drive (50GB) in a workspace structure. Large geographical datasets and images were stored on an external hard drive (1TB). Back-ups of reports, small data, scripts, and literature were stored on a USB-stick (64GB). Reports, scripts, and workspace were backed up weekly. Scripts were saved on a GitHub repository: https://github.com/Davidswinkels/BuildingCharacterization.

Hardware

The computers in the thesis room of Geo-Information Science at Wageningen University & Research were used to write reports and to make maps on a Windows OS. The re-training of the CNN was performed on a GPU of the author (NVIDIA GeForce GTX760 with 2GB of RAM) on a 64-bit Ubuntu 16.04 OS.

Software

ArcMap 10.5 was used to select spatial data, to inspect spatial data and to visualize maps. Coordinate system during processing was geographic coordinate system WGS84 (EPSG: 4326) because streetview images have WGS84. Final maps were made with the projected coordinate system RD New (EPSG: 28992).
Python 2.7 was used to download images, to classify images, to train models, to characterize land use, to perform statistics and to automate the whole process. The following Python packages were used: urllib, TensorFlow, NumPy, Pandas, and SciPy. Urllib downloaded Google Street View images. TensorFlow was used to do deep learning with a Convolutional Neural Network. TensorFlow utilized CUDA to access the GPU and to perform GPU processing. Numpy was used in array calculations. Pandas handled data frames and data structures in Python. SciPy performed statistics, such as T-test and ANOVA. Python scripts were run from bash terminal and debugged in PyCharm IDE.
R 3.4.0 was used in Rstudio to perform concatenation of building functions.

Scripting setup

Scripts were set up in Python by using Conda. Conda is a combination of virtualenv, which creates virtual environments, and pip, which installs packages. TensorFlow with GPU had special requirements: GPU with CUDA Compute Capability 3.0 or higher, CUDA 9.0 or higher and cuDNN 7.0 or higher. After requirements were done, Bash code created a new virtual environment called tf_py2 with Python=2.7, pip, Pandas, Numpy, urllib3, pillow, SciPy, and TensorFlow (see snippet 2.1).

Snippet 2.1: Bash – creating an environment with Conda to install packages

```
conda create --name tf_py2 python=2.7 pip pandas numpy
conda install --name tf_py2 --channel anaconda urllib3 pillow scipy
source activate tf_py2
pip install tensorflow-gpu
```

2.2 Data acquisition

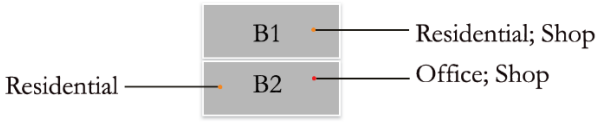
Building data

The research used the building and address dataset (BAG) in Amsterdam of 2016. The building data was available in 2016, was usable for non-commercial purposes and was made by professionals. The BAG dataset stored functional usage information of addresses. These functions have been standardized by urban planners (Bouwbesluitonline, 2012) and were translated (see table 2.1).

Table 2.1: Function classes in BAG (Bouwbesluitonline, 2012)

Function (Dutch)	Function (English)	Description
Woon	Residential	A place for residing.
Bijeenkomst	Meeting	A meeting place for art, culture, religion, communication, catering and watching sports.
Cel	Cell	Cell/prison – a place for compulsory stay.
Gezondheidszorg	Healthcare	A place for medical research, nursery, care or treatment.
Industrie	Industry	A place for professional storing and using of materials and goods, or for agricultural purposes.
Kantoor	Office	A place for administration.
Logies	Accommodation	A place for recreational accommodation or temporary shelter for a person.
Onderwijs	Education	A place for education.
Sport	Sport	A place to exercise sports.
Winkel	Shop	Place for trading in materials, goods or services.
Overige	Other	A place where none of the previous functions could be appointed and where residing of persons is a subservient function.

The BAG dataset stored the geometry and function of all buildings and addresses in the Netherlands. The BAG dataset has buildings with polygon geometries and addresses with point geometries (see figure 2.1). Address data contained information on the street, house number, municipality, and functional usage. Building units, such as apartments, could be stacked on top of each other. A building could have multiple addresses and each address could have multiple functions. The goal was to give buildings a label related to the presence or absence of the function (see figure 2.1).



	B1	B2
Residential	1	1
Meeting	0	0
Healthcare	0	0
Industry	0	0
Office	0	1
Accomodation	0	0
Education	0	0
Sport	0	0
Shop	1	1
Other	0	0

1 = Present
0 = Absent

Figure 2.1: Top view of two buildings (left) and multi-labels buildings (right)

Before combining addresses and buildings, valid data was selected. Firstly, buildings and addresses were selected in the municipality of Amsterdam. Secondly, buildings and addresses that were in use (see snippet 2.2 and 2.3) were selected. Thirdly, buildings spatially containing an address were selected. An example of a building without an address was a storage silo in Amsterdam harbor. Fourthly, buildings and addresses on land were selected. Some buildings and addresses were located on the water because the building was a boat or because the house was built on water. ArcMap was used to select valid buildings in Amsterdam.

Snippet 2.2: SQL query ArcMap - selecting buildings in use

```
EindDatum IS NULL AND ( Status = 'Pand in gebruik' OR Status = 'Pand in gebruik (niet ingemeten)' )
```

Snippet 2.3: SQL query ArcMap - selecting addresses in use

```
EindDatum IS NULL AND ( Status = 'Verblijfsobject gevormd' OR Status = 'Verblijfsobject in gebruik' OR Status = 'Verblijfsobject in gebruik (niet ingemeten)' )
```

Now buildings and addresses were combined. Each building and address got a unique BuildingID or unique AddressID based on the unique ID in the 'Identificatie' column. If the building polygon contained the address, then addresses were spatially joined to buildings with the join operation 'JOIN_ONE_TO_MANY'. A stack of building units (e.g. apartments) was a stack of multiple building polygons. An address was joined to these multiple building polygons because it was unknown to which specific building polygon the address belonged. The downloaded BAG dataset did not have any columns with identifiers to link buildings and addresses directly. Buildings and addresses were spatially joined. The spatial join created multiple building polygons for each address that was added to a building (see table 2.2).

Table 2.2: Example address functions joined to buildings

BuildingID	Function	AddressID
363100012069865	office	363010001026079
363100012069865	office	363010001026080
363100012069865	office	363010001026081
363100012069865	office	363010001026082
363100012070234	office	363010001008860
363100012073393	meeting	363010012076488
363100012073393	meeting	363010012076489
363100012073393	meeting; office	363010012076490

The multiple buildings had to be merged whilst keeping information from multiple addresses and their respective functions. A script in R aggregated all functions per building as one long line of text (see snippet 2.4).

Snippet 2.4: R – aggregate functions for every building

```
buildings_aggr <- aggregate(Function ~ BuildingID, data = Buildings, paste,
                    collapse = ",")
```

The aggregated buildings were assigned with multiple class labels. For every function, a new column was created. The values in the column were set to 1 (= present) or 0 (= absent) based on the presence or absence of the function in the concatenated text (see table 2.3).

Table 2.3: Concatenated building functions per BuildingID

BuildingID	**Function**	Residential	Meeting	Healthcare	Industry	Office	Accommodation	Education	Sport	Shop
363100012061186	industry, meeting	0	1	0	1	0	0	0	0	0
363100012061187	residential, residential, industry	1	0	0	1	0	0	0	0	0
363100012061188	residential	1	0	0	0	0	0	0	0	0
363100012061189	shop, residential, residential	1	0	0	0	0	0	0	0	1

Note: Land use labels are dichotomous present (1) or absent (0)

After labeling the buildings with functions, the functions were added to the geometry and building polygons were transformed into points. Functions were joined from a CSV file to a shapefile in ArcMap based on the unique

BuildingID. The neighborhoodID was spatially joined to the building and then was used to separate the training and testing datasets. Building polygons were converted into centroids (see figure 2.2) because the Google Street View camera needed to be pointed towards one location. Looking in the direction of the centroid gave a good view on the building from any direction.

Figure 2.2: Building polygons and their centers of gravity

Streetview data

After acquiring the locations of the buildings in Amsterdam, the streetview data were downloaded. Out of the multiple sources of streetview data, a choice was made for Google Street View images, because the streetview images from Google were available on most streets in the Netherlands, had 360° horizontal perspective, could be pointed in any direction to look at a building and had good quality camera, GPS, accelerometer and gyroscope.

Streetview images were downloaded with appropriate camera parameters. The heading of the camera was turned towards the building facade (see figure 2.3 and 2.5). The heading is in compass degrees, where 0° is to the North and 90° is to the East. Image zoom levels were defined by the field of view [FOV] in degrees and images were downloaded at FOV = 90°, FOV = 60° and FOV = 30° (see figure 2.4 and 2.5).

Heading = 0°

Heading = 150°

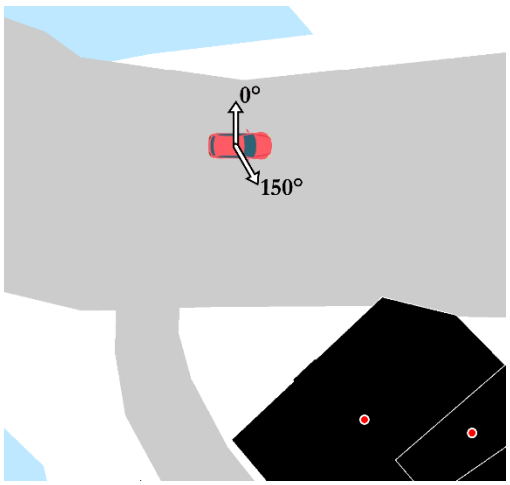
Figure 2.3: Streetview images with changed heading near Amsterdam CS.

FOV = 90°

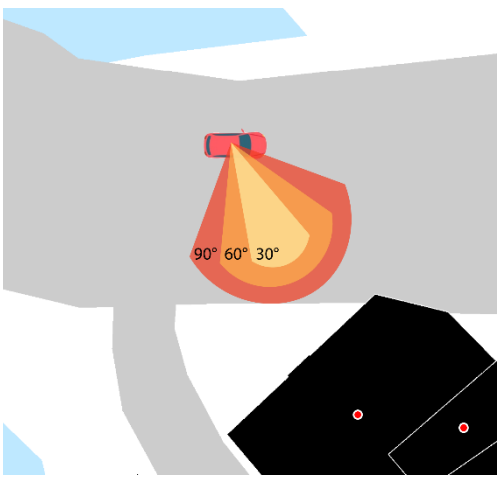
FOV = 60°

FOV = 30°

Figure 2.4: Streetview images with changed field of view near Amsterdam CS.



Changing heading



Changing FOV

Figure 2.5: Streetview car with changing heading and changing field of view from a top perspective. Note: the same streetview location is displayed in all figures on this page.

Downloading the streetview image had several steps. First, all available streetview panoramas were found nearby a location by performing a query to the Google API server (see snippet 2.5). Streetview panoramas were selected roughly 50 meters around the input location. To avoid downloading user images, only panoramas by user "Google" were selected. Then one panorama with the closest date to the date of the building dataset of 2016 was selected. The preferred year was 2016, 2015, 2014, 2013, 2012, 2011 and 2010 in that order. The location of the panoramas was expressed in latitude and longitude with WGS84 as the coordinate system. The resulting selection returned one unique panorama id, location, and date.

Snippet 2.5: Python – find streetview panoramas nearby location

```
url="https://maps.googleapis.com/maps/api/js/GeoPhotoService.SingleI
mageSearch?pb=!1m5!1sapiv3!5sUS!11m2!1m1!1b0!2m4!1m2!3d{0:}!4d{1:}
!2d50!3m10!2m2!1sen!2sGB!9m1!1e2!11m4!1m3!1e2!2b1!3e2!4m10!1e1!1e2!
1e3!1e4!1e8!1e6!5m1!1e2!6m1!1e2&callback=_xdc_._v2mub5 "
url.format(('51.983535', '5.663232'))
requests.get(url)
```

After selecting one panorama, the camera was pointed from panorama towards the centroid of the building. First, the heading from panorama to building centroid was calculated. Then three streetview images 640 pixels wide and high were downloaded at field of view 30, 60 and 90 degrees from one panorama (see snippet 2.6). Each image was stored with a unique filename containing the neighborhoodID, buildingID, panoramaID, FOV in such a way that they could easily be found again manually or automatically with a regular expression.

Snippet 2.6: Python – download streetview image

```
base_url = "https://maps.googleapis.com/maps/api/streetview?size="
url = base_url + size + "&pano=" + pan['panoid'] + "&heading=" +
str(heading) + "&pitch=0&fov=" + str(fov) + "&key=" + api_key
filepath = save_dir + filename + ".jpg"
urllib.urlretrieve(url, filepath)
```

During downloading, a CSV file kept track of download availability, panorama data, and panorama quality. Sometimes no panoramas were available at a given location or given date. The panorama ID, date, latitude, longitude, heading, and distance from panorama to building centroid were all saved to keep track of the download process and image quality. The distance between the two locations in the WGS84 coordinate system was calculated as the great-circle distance with the haversine formula. The download limit of streetview images was 25,000 images per day for a free account. The policy has recently changed to 28,000 images per month (Google, 2018b). Downloading was performed in batches of 24,000 images for 8,000 buildings per day amounting to 325,000 images.

The quality of streetview images was improved by deselecting invalid images.

- Some images were taken from a boat on the water.
- Some images were shot inside buildings, such as the opera hall.
- Some images had inaccurate locations due to bad GPS (see figure 2.6) or the urban canyon effect (Ben-Moshe et al., 2011). Urban canyons block line of sight with GPS satellites and contribute to poor GPS accuracy.
- Some streetview images did not have correct jpg/JFIF format.

All these incorrect streetview images were removed. Invalid streetview images got the value 'invalid' in the column 'valid'.

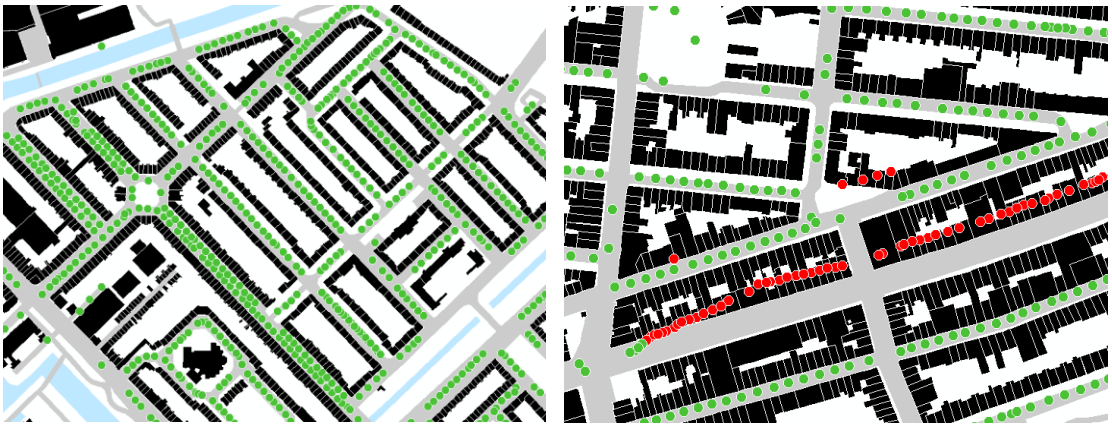


Figure 2.6: Streetview images with correct locations colored green (left) and some streetview images with inaccurate locations colored red (right)

Data selection

The building data of BAG had 11 land use classes with an imbalanced distribution. The residential function was very common; 111,585 buildings out of 124,818 total buildings had a residential function. Other common building functions were other, shop, industry, office and meeting with respectively 8,202, 7,663, 6,280, 4,548 and 4,471 buildings out of the 124,818 buildings. Some uncommon building functions were education, sport, accommodation, healthcare, and cell, with respectively 763, 440, 422, 364 and 8 buildings having that function (see table 2.4).

Table 2.4: Distribution of building functions before invalidation (n = 124,818)

Function	Count	Percentage
	n	%
Residential	111,585	89.4
Other	8,202	6.6
Shop	7,663	6.1
Industry	6,280	5.0
Office	4,548	3.6
Meeting	4,471	3.6
Education	763	0.6
Sport	440	0.4
Accommodation	422	0.3
Healthcare	364	0.2
Cell	8	0.0

Note: Every building can have multiple functions present

Buildings with invalid streetview images were deselected (see table 2.5).

- No streetview images were available near the building in 16,963 cases.
- Streetview images were located on water in 50 cases.
- Streetview images had an invalid location in 1,421 cases. The invalid location was mainly due to inaccurate GPS in urban canyons.
- Sometimes streetview images did not exist after downloading or had a wrong jpg format in 37 cases.

Table 2.5: Buildings with invalid data (n = 18,468)

Reason	Count	Percentage
	n	%
No streetview image download available	16,963	91.8
Streetview image located on water	50	0.3
Streetview image has invalid location (except water)	1,421	7.7
Streetview image does not exist after download or had wrong jpg format	37	0.2
Note: Streetview image can have a wrong location and wrong image format		

The buildings with invalid streetview images were mainly located at the fringes of Amsterdam (see figure 2.7). Some neighborhoods had low percentages of building with valid streetview images because the neighborhoods consisted of recently built buildings (e.g. Rieteiland Oost), had no Google Street View car coverage after 2008/2009 (e.g. Gein Noordoost or Bijlmermuseum), were partly inaccessible by car or had invalid streetview images. Every neighborhood in Amsterdam with buildings still had some buildings with valid streetview images.

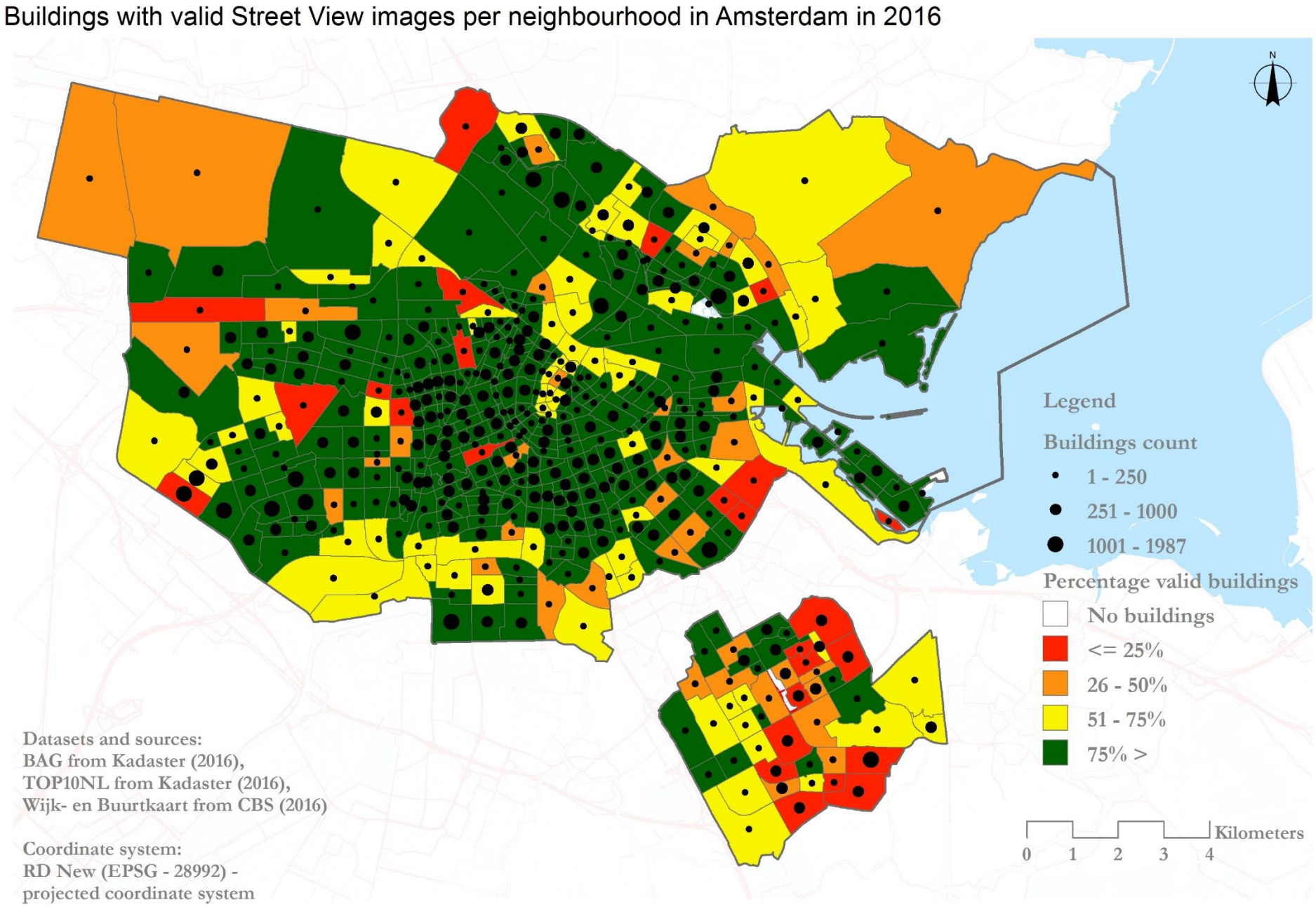


Figure 2.7: A map of buildings with valid Street View images per neighborhood in Amsterdam in 2016

After selecting valid data and usable building functions, a smaller portion of data was left. There were 106,350 buildings left in the dataset. Buildings had residential, shop, industry, office and meeting functions in respectively 95,807, 6,567, 5,394, 3,928, 3,571 out of the 106,350 cases (table 2.6). These five building functions were selected because each class had enough samples to perform training, validation, and testing. The "other" class had enough samples but was deselected. Buildings with "other" function were too varied to be consistently classified as "other" building function. The distribution before and after the selection was similar.

Table 2.6: Distribution of building functions after selecting valid data (n = 106,350)

Function	Count	Percentage
	n	%
Residential	95,807	90.1
Shop	6,567	6.2
Industry	5,394	5.1
Office	3,928	3.7
Meeting	3,571	3.4

Note: Every building can have multiple functions present.

2.3 Data accuracy

The data accuracy determines the prediction accuracy of deep learning models. Therefore, data quality was checked.

Building data

The building data from the BAG is created and maintained by professional organizations, among them the Kadaster and Dutch municipalities, which strive to have very high data quality. The data quality of BAG was found per province by consulting the data quality dashboard developed by Kadaster (Kadaster, 2018a). Amsterdam is in the province of Noord-Holland. The accuracy of the BAG in Noord-Holland was 99.8% for the attributes building age, surface size and building function, and 99.8% for the geometry (Kadaster, 2018a). Furthermore, Kadaster analyzed how building function related to building size to find incorrect building functions (see figure 2.8). The province of Noord-Holland had 99.5% expected correct building functions.

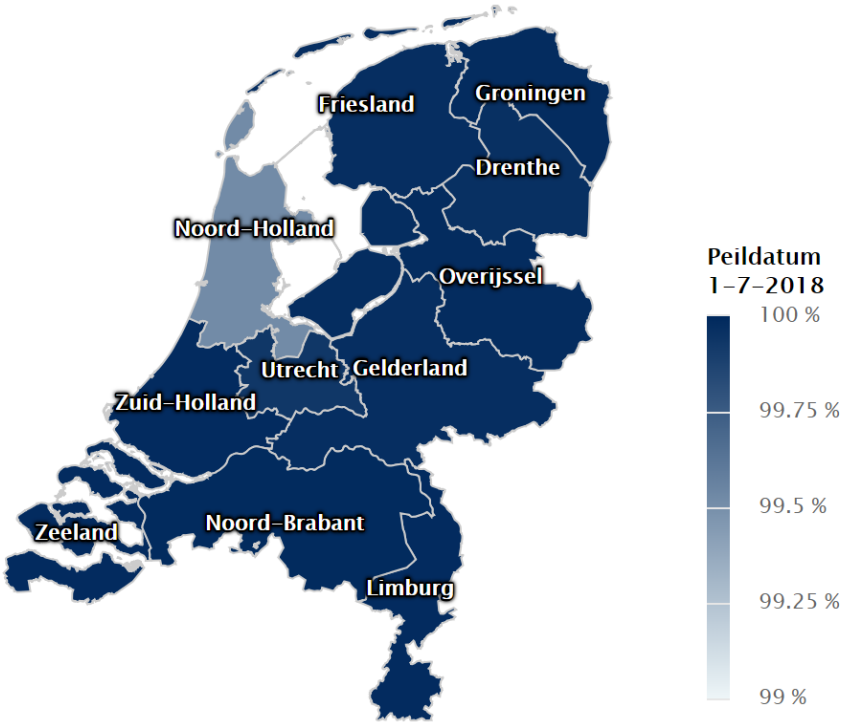


Figure 2.8: Expected correct building functions per province in the BAG based on the building size in the Netherlands (Kadaster, 2018a)

Streetview data

Google strives for high data quality of Google Street View images by having specifications for the Google Street View cameras and reviewing all input Google Street View images (Google, 2018a). Some important specifications are:

- Imagery needs to be ≥8k at 5FPS.
- Imagery needs to have 360° horizontal FOV.
- Imagery needs to have ≥135° contiguous vertical FOV.
- Accelerometer needs to have resolution ≥16 bit.
- Gyroscope needs to have resolution ≥16 bit.
- GPS needs to have a horizontal position accuracy of 2.5 meters.

These specifications show expected data quality and not measured data quality.

Therefore, data quality of downloaded streetview images had to be checked. For 100 randomly selected buildings, their three streetview images were manually checked qualitatively. This sample of 100 buildings was taken from all 124,818 buildings minus 16,963 buildings, who had no streetview image available. The location was checked in ArcMap. The heading was checked in an image viewer by verifying that the middle of the streetview image looks at the correct building based on building information in ArcMap and Google Street View. Occlusion was checked by determining if the building can be seen for each field of view. Some objects, such as vans or greenery, can occlude the view. Furthermore, extra info was stored as to why views are blocked or if the building function could be determined by the streetview image. A manual check-up helped to check data quality of streetview images.

2.4 Building function classification

Building functions were classified by convolutional neural networks with supervised classification. Convolutional neural networks extract features from images. MobileNetV1 and Inception-v3 are ready-made CNN models, which can be fine-tuned to predict new classifications. MobileNetV1 is a shallower CNN with fewer layers and Inception-v3 is a deeper CNN with more layers (Howard et al., 2017). The models are publicly available via TensorFlow and have been pre-trained on ImageNet, which has similar images as streetview images (see section 1.4). MobileNetV1 was chosen for its fast computation time and Inception-v3 for its high accuracy.

Roughly 60% of buildings were used for training, 20% for validating, and 20% for testing. The training dataset trained the models. Validation dataset was used to check progress during the training phase and to perform 4-fold validation by changing training and validation dataset. Each of the five folds was made to have a similar distribution of all types of building functions. Cross-validations helped to see if variation in training data leads to a different test result. The test dataset tested the prediction accuracy of the image classifiers on unseen data. The train and test images were spatially separated to overcome spatial autocorrelation by appointing all buildings in the same neighborhood to the same category (Khosla et al., 2014; Salesses et al., 2013)(see figure 2.9). Invalid streetview images or buildings were not used for training, validating or testing.

Figure 2.9: Separation of training, testing and validation buildings based on neighborhood

By iterating over cross-validation runs, building functions, CNN architectures and field of views 160 CNN models were run (see snippet 2.7). Inception-v3 had input image size of 299 by 299 pixels and MobileNetV1 224 by 224 pixels and both architectures used RGB images. Original streetview images with 640 by 640 pixels were downsized to appropriate image size per architecture. Each CNN was fine-tuned on 60% of the streetview images, had 4,000 training steps, a learning rate of 0.01, training batch size of 100 images and test batch size of one image. The mixed field of view had a random selection of a field of view at 30°, 60° or 90° degrees. In this way a test was performed to check if it matters for the prediction accuracy of the image classifier to have either images with same field of view or images with varying field of view. The CNN with mixed field of view used all 319,050 streetview images for training, validation and testing. The CNNs with single field of view used only 106,350 streetview images for training, validation and testing. Each model predicted binary labels, either absence or presence of the specific building function. Binary classification of multi-labels was used because distribution of building functions was very imbalanced. Per CNN model run the outcome metrics were stored.

Snippet 2.7: Python – loop over cross validations runs, building functions, CNN architectures, field of views to run CNN models in TensorFlow

```
cross_validations = [0, 1, 2, 3]
building_functions = ['Residentia', 'Meeting', 'Industry', 'Office', 'Shop']
architectures = ['inception_v3', 'mobilenet_1.0_224']
fovs = ['F30', 'F60', 'F90', 'F30_60_90']

for cross_validation in cross_validations:
  for building_function in building_functions:
    for architecture in architectures:
      for fov in fovs:
          tf.app.run(main=main, argv=[sys.argv[0]] + unparsed)
```

Note: tf.app.run() selects cross-validation run, building function, field of view and architecture based on the global variable.

The accuracy of predictions was checked by outcome metrics. In the used dataset, labels were very imbalanced. Roughly 90% of buildings had a residential label and 10% had a non-residential label. This imbalance of observations in classes with a binary prediction required multiple outcome statistics, such as overall accuracy, and average accuracy and F1-score (Ghamisi et al., 2017; Powers, 2011; Volpi & Tuia, 2016). Overall accuracy gives an accuracy of all data objects and average accuracy gives accuracy over classes. F1-score is the harmonic mean of the precision and recall. By taking the harmonic mean a high F1-score requires both a high precision and recall to be high itself. Precision is the rate of true positives divided by true positives plus false positives. Recall, also known as sensitivity, is the rate of true positives divided by true positives plus false negatives. As an extra outcome metric, the computation time of the CNN models was tracked, to check how much time each model needed. The F1-score, overall accuracy and average accuracy were the main outcome metrics used to assess the accuracy of the CNN models.

Correct building function predictions by the CNN were collectively analyzed. The analysis showed which and why building functions were hard to predict. Correct predictions were compared for the five building functions: residential, shop, industry, office and meeting. Same for the two architectures: Inception-v3 and MobileNetV1. The association of data characteristics with correct predictions was statistically checked. This was done by checking if certain field of views, distance from streetview image to the building, streetview image age, building age or neighborhood were significantly associated with correct predictions by performing an independent t-test or ANOVA. Data analysis of outcomes was performed to understand why convolutional neural networks do correct predictions.

Chapter 3 - Results

In this chapter, results are shown of data acquisition, data accuracy, and building classification.

3.1 Data acquisition

In figures 3.1–3.13 a selection of streetview images is shown at different fields of view with the multi-label depicting multiple functions per building.

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.1: Streetview images (id: ej9TGTL6wis9yRUGagTumw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012062354)

Table 3.1: Building functions of BuildingID 363100012062354					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	No	No	No

FOV = 30° **FOV = 60°** **FOV = 90°**

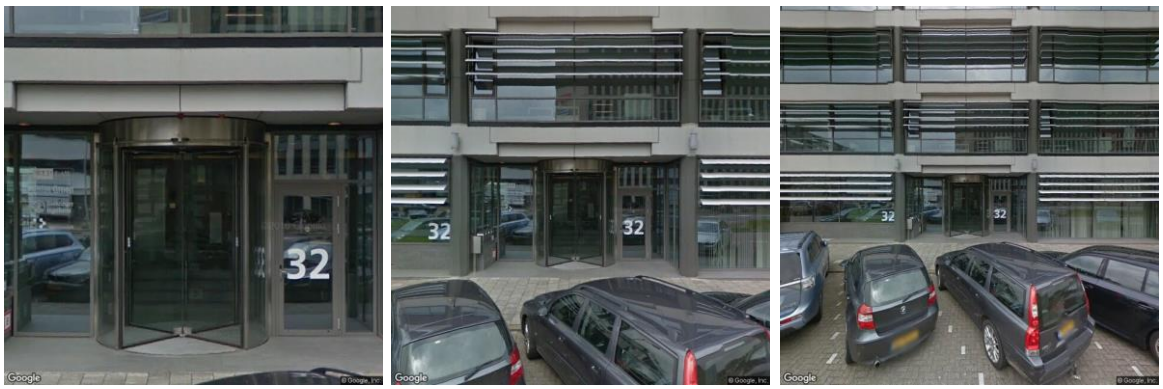
Figure 3.2: Streetview images (id: oX22-GUAt_IubZRNutYHgg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012148038)

Table 3.2: Building functions of BuildingID 363100012148038					
Function	Residential	Meeting	Industry	Office	Shop
Presence	No	Yes	No	No	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.3: Streetview images (id: 4pTxSPl6epjxX1iIkHxJJA) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012089326)

Table 3.3: Building functions of BuildingID 363100012089326					
Function	Residential	Meeting	Industry	Office	Shop
Presence	No	No	Yes	No	No



FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.4: Streetview images (id: MXUz2NUGwQw4MMbSP5SzFg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012079218)

Table 3.4: Building functions of BuildingID 363100012079218					
Function	Residential	Meeting	Industry	Office	Shop
Presence	No	No	No	Yes	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.5: Streetview images (id: v7GP5wMp7GYTuQK0TQEpOw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012165088)

Table 3.5: Building functions of BuildingID 363100012165088					
Function	Residential	Meeting	Industry	Office	Shop
Presence	No	No	No	No	Yes

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.6: Streetview images (id: UqoapBKhfyZYOWMgY_db3w) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012141527)

Table 3.6: Building functions of BuildingID 363100012141527					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	Yes	No	No	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.7: Streetview images (id: Z4NlrlM2uD-TIvctKElKuQ) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012061289)

Table 3.7: Building functions of BuildingID 363100012061289					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	Yes	No	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.8: Streetview images (id: QsmLMrc_Xoodmi8UaDwFgw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012090032)

Table 3.8: Building functions of BuildingID 363100012090032					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	No	Yes	Yes

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.9: Streetview images (id: pKOlhlRTtT5njKGvJlWG1A) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012170947)

Table 3.9: Building functions of BuildingID 363100012170947					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	No	No	Yes

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.10: Streetview images (id: udeOyjSQyqX7P8zbKxLMSg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012128931)

Table 3.10: Building functions of BuildingID 363100012128931					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	Yes	Yes	Yes	Yes

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.11: Streetview images (id: mYyPOHi89deVSd8L25KIkg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012137189)

Table 3.11: Building functions of BuildingID 363100012137189					
Function	Residential	Meeting	Industry	Office	Shop
Presence	No	No	Yes	No	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.12: Streetview images (id: p2gu5KLF7GY8YYHb_jME4Q) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012150001)

Table 3.12: Building functions of BuildingID 363100012150001					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	No	Yes	No

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.13: Streetview images (id: p2gu5KLF7GY8YYHb_jME4Q) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012150001)

Table 3.13: Building functions of BuildingID 363100012150001					
Function	Residential	Meeting	Industry	Office	Shop
Presence	Yes	No	No	Yes	Yes

The streetview images give a good view on buildings and often it is possible to guess the building function correctly based on the facade.

- Residential buildings are often made out bricks, do not have many signs, have a simple door and a simple window (see figure 3.1).
- Hotels, restaurants, and bars (meeting function) often contain tapestry and large signs (see figure 3.2).
- Industrial buildings often have garage doors and metal plates (figure 3.3).
- Offices often have large glass panes and revolving doors (see figure 3.4).
- Shops often have colorful goods, people and signs (see figure 3.9).

Some images show a residential building façade but do have an office or shop function (see figure 3.13). Often it is hard to distinguish all functions of a building with multiple functions. One example is a building that looks like a restaurant but has a residential, meeting, industry, office and shop function (see figure 3.10). There is variation in the images. Streetview images vary due to the architectural style, heading towards building, illumination, urban context, and occlusion. Some images have cars, people, bikes or trees as the urban context.

3.2 Data accuracy

Building data

The building data accuracy was checked by Kadaster (see section 2.3).

Streetview data

A small random selection of 100 downloaded streetview panoramas was validated by checking location, heading and image quality (see table 3.14). The sample was selected directly after downloading and just before invalidating images due to their invalid location.

Table 3.14: Streetview data accuracy (n = 100, N = 107,855)

Function	Count	Percentage
	n	%
Correct image location	98	98.0
Correct image heading	83	83.0
Image unobstructed at FOV 30°	87	87.0
Image unobstructed at FOV 60°	88	88.0
Image unobstructed at FOV 90°	88	88.0

The location was correct for 98 out 100 streetview panorama. One of the two erroneous locations is shown in figure 3.14. The Google Street View car was driving on a road to the South of some buildings, but the GPS registered a position to the North of the buildings. The user was 'Google', timestamp was October 2016 panoID was 'ZDvWozrF0Wvxqx7nIxRdpg', and the location was roughly 35 meters off. The erroneous location was part of a series of erroneous locations. If the location of the streetview image was wrong, the heading also did not point correctly to the building centroid.

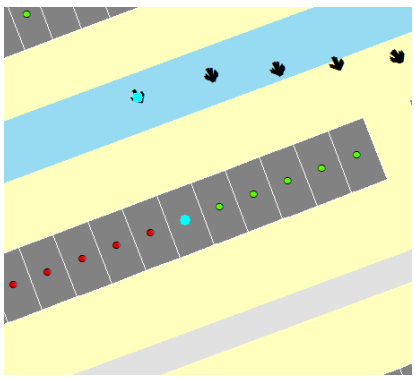


Figure 3.14: Erroneous location of streetview panorama (light blue point) (Pano: ZDvWozrF0Wvxqx7nIxRdpg, BuildingID: B363100012109580)

The heading was correct for 83 out of 100 streetview panoramas. Correct headings pointed straight at the building centroid or had a negligible erroneous heading up to roughly 10 degrees. Incorrect headings had an error of 10 or more degrees, which would often point the camera away from the targeted building. For example, a camera was supposed to look at one building with house number 18 but it looked at buildings with house numbers 17 and 18 (see figure 3.15). Furthermore, the camera did often not have a perpendicular angle towards the facade. This resulted in images that looked at multiple buildings.

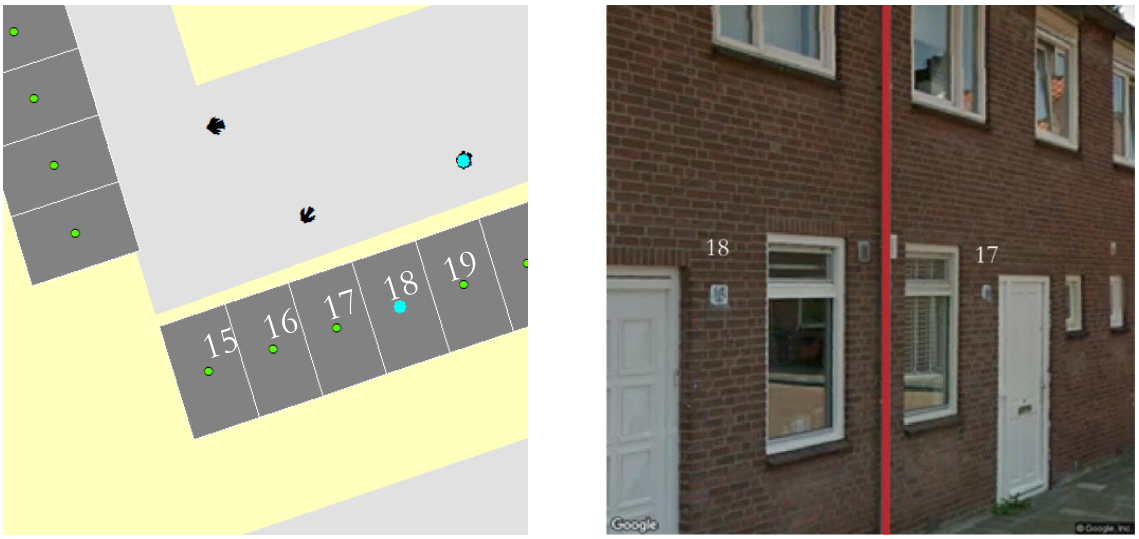


Figure 3.15: Erroneous heading of streetview panorama (light blue point) and streetview image with erroneous heading (red line shows boundary) (PanoID: LJIINN11HDr6ZqenexywLg, BuildingID: 363100012129306)

The labels were checked too. Most images showed a predictable building function. In one case it was hard to see if the building had both the residential and office function (see figure 3.16).

Figure 3.16: Arguable building function of the streetview image (Pano: l4petcWy9cWxjjdQU4hmFQ, BuildingID: 363100012140584)

Images at a FOV of 30, 60 and 90 degrees had an unobstructed line of sight on the building in 263 out of 300 instances. Streetview images that were obstructed at one field of view were mostly blocked at the other two field of views too. When the view on buildings was obstructed, target buildings were occluded in the image by other buildings, trees, large shrubs, busses or caravans (see figure 3.17 and 3.18). When objects like trees or vehicles were in front of the building, often there was still a part of the building that could be seen. When buildings were in front of the target building, often the view on the correct target building was totally obstructed.

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.17: Streetview images (id: Q0b8cyOjOJylpUaNB55myg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012107478) totally occluded by trees

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.18: Streetview images (id: hifeORCkD4G4bLjz9gQZkQ) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012086196) partly occluded by a van

3.3 Building classification

The goal was to characterize building functions from streetview images. The next paragraphs show results of characterizing building functions from unseen streetview images for four different field of views and two different CNN.

Prediction metrics

Residential; the binary CNN classifier based on Inception-v3 architecture predicted residential function with F1-score between 0.90 and 0.92, overall accuracy between 83.60 and 86.59, and average accuracy between 80.00 and 83.01 (see table 3.15). Both Inception-v3 and MobileNetV1 models had slightly higher average accuracy when predicting images with larger field of views. Models predicting from streetview images with a mixed field of view had lower overall and average accuracy than models predicting from images with 90 degrees field of view.

Table 3.15: Outcome statistics of residential function prediction (n = 21,142)

Residential						
	Inception-v3			MobileNetV1		
	F1-score	OA	AA	F1-score	OA	AA
unit	-	%	%	-	%	%
FOV30	0.90 (0.01)	83.60 (2.05)	80.00 (0.14)	0.84 (0.20)	78.00 (22.52)	60.74 (2.60)
FOV60	0.92 (0.01)	85.80 (1.43)	82.23 (0.11)	0.88 (0.13)	81.65 (16.69)	63.95 (8.72)
FOV90	0.92 (0.01)	86.59 (1.73)	83.01 (0.29)	0.76 (0.12)	66.50 (13.50)	76.41 (5.01)
FOVmix	0.91 (0.01)	84.11 (1.30)	81.12 (0.13)	0.94 (0.01)	89.25 (1.82)	74.65 (5.21)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over four cross-validation runs.

Meeting; the binary CNN classifier based on Inception-v3 architecture predicted meeting functions with F1-score between 0.19 and 0.20, overall accuracy between 80.12 and 81.34, and average accuracy between 74.13 and 75.07 (see table 3.16). Models based on Inception-v3 and MobileNetV1 had the highest average accuracy with images with 30 degrees field of view. The model based on MobileNetV1 with images with 90 degrees field of view had a very low overall accuracy of 48.94.

Table 3.16: Outcome statistics of meeting function prediction (n = 21,142)

Meeting						
	Inception-v3			MobileNetV1		
	F1-score	OA	AA	F1-score	OA	AA
unit	-	%	%	-	%	%
FOV30	0.20 (0.02)	81.34 (3.58)	75.07 (0.59)	0.19 (0.03)	86.59 (9.98)	64.69 (5.30)
FOV60	0.20 (0.01)	80.38 (3.18)	74.69 (0.99)	0.17 (0.07)	82.43 (19.50)	62.55 (5.87)
FOV90	0.19 (0.01)	80.12 (1.62)	74.13 (0.42)	0.13 (0.06)	48.94 (27.36)	63.86 (4.44)
FOVmix	0.20 (0.01)	81.24 (0.98)	74.55 (0.18)	0.17 (0.06)	71.21 (25.68)	64.51 (4.84)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over four cross-validation runs.

Industry; the binary CNN classifier based on Inception-v3 architecture predicted industry functions with F1-score between 0.25 and 0.28, overall accuracy between 76.50 and 80.77, and average accuracy between 72.16 and 72.83 (see table 3.17). Both Inception-v3 and MobileNetV1 models did have slightly higher average accuracy when predicting industry functions with images at 60 degrees field of view. The model based on MobileNetV1 and images with 90 degrees field of view had a very high overall accuracy of 92.10, but a lower average accuracy of 65.43.

Table 3.17: Outcome statistics of industry function prediction (n = 21,142)

	Industry					
	Inception-v3			MobileNetV1		
	F1-score	OA	AA	F1-score	OA	AA
unit	-	%	%	-	%	%
FOV30	0.25 (0.03)	76.50 (5.63)	72.16 (0.51)	0.25 (0.06)	77.06 (24.58)	62.84 (3.27)
FOV60	0.28 (0.02)	80.77 (3.30)	72.83 (0.53)	0.29 (0.06)	83.63 (13.02)	66.73 (2.57)
FOV90	0.25 (0.01)	76.75 (1.51)	72.54 (0.47)	0.34 (0.02)	92.10 (1.74)	65.43 (2.35)
FOVmix	0.25 (0.02)	76.38 (5.65)	72.35 (0.42)	0.24 (0.09)	67.41 (26.48)	64.02 (2.94)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over four cross-validation runs.

Office; the binary CNN classifier based on Inception-v3 architecture predicted office functions with F1-score between 0.14 and 0.17, overall accuracy between 66.26 and 77.33, and average accuracy between 68.03 and 70.72 (see table 3.18). Inception-v3 had the highest average accuracy with a field of view of 60 degrees. MobileNetV1 had the highest average accuracy with 90 degrees field of view.

Table 3.18: Outcome statistics of office function prediction (n = 21,142)

	Office					
	Inception-v3			MobileNetV1		
	F1-score	OA	AA	F1-score	OA	AA
unit	-	%	%	-	%	%
FOV30	0.14 (0.01)	66.26 (4.62)	69.23 (0.19)	0.13 (0.03)	71.45 (23.28)	60.90 (5.95)
FOV60	0.16 (0.01)	69.06 (5.13)	70.72 (0.41)	0.17 (0.04)	75.70 (19.90)	63.75 (3.02)
FOV90	0.17 (0.00)	77.33 (2.31)	68.03 (1.21)	0.16 (0.01)	80.31 (8.13)	64.22 (3.75)
FOVmix	0.17 (0.01)	75.03 (4.73)	69.18 (1.04)	0.15 (0.03)	80.73 (18.80)	60.19 (4.05)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over four cross-validation runs.

Shop; the binary CNN classifier based on Inception-v3 architecture predicted shop functions with F1-score between 0.34 and 0.38, overall accuracy between 83.19 and 86.54, and average accuracy between 80.59 and 81.48 (see table 3.19). Inception-v3 had slightly higher average accuracy with a field of view of 30 degrees. MobileNetV1 had the highest average accuracy with images with mixed field of views.

Table 3.19: Outcome statistics of shop function prediction (n = 21,142)

Shop						
	Inception-v3			MobileNetV1		
	F1-score	OA	AA	F1-score	OA	AA
unit	-	%	%	-	%	%
FOV30	0.37 (0.02)	85.68 (1.76)	81.48 (0.32)	0.20 (0.06)	56.96 (16.00)	72.21 (5.54)
FOV60	0.38 (0.03)	86.54 (2.54)	81.39 (0.21)	0.32 (0.15)	71.10 (22.83)	72.13 (2.44)
FOV90	0.34 (0.03)	83.19 (3.01)	80.59 (0.29)	0.34 (0.11)	78.17 (23.05)	71.91 (4.39)
FOVmix	0.36 (0.01)	85.01 (1.07)	81.31 (0.19)	0.20 (0.03)	60.37 (9.82)	74.23 (3.51)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over four cross-validation runs.

The general pattern among the five buildings functions is higher average accuracy and smaller standard deviation for CNN models with Inception-v3 architecture compared to CNN models MobileNetV1 architecture. For the building functions from best to worst average accuracy the order is residential (81.59%), shop (81.19%), meeting (74.61%), industry (72.47%) and office (69.29%). For the field of views from best to worst average accuracy the order is 60 degrees (76.37%), mixed field of view (75.70%), 90 degrees (75.66%) and 30 degrees (75.59%).

Characteristics interpretation

This study found that streetview images with a smaller distance to building centroid had statistically significantly more correct building function predictions (t=-28.54, p=0.00)(see table 3.20). For mixed field of view, the average distance for correct predictions was smallest out of four field of view types. Especially for residential, meeting and office functions the distance from panorama to the building was smaller for correct predictions than for incorrect predictions.

Table 3.20: Average distance of correct and incorrect predictions for residential, meeting, industry, office and shop functions (model = Inception-v3; cross-validation run = 0; n = 21,142)

	Residential		Meeting		Industry	
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction
unit	meter	meter	meter	meter	meter	meter
FOV30	16.24 (6.91)	19.29 (9.59)	16.39 (7.17)	18.98 (9.07)	16.56 (7.05)	17.44 (8.88)
FOV60	16.16 (6.88)	19.98 (9.72)	16.21 (7.04)	18.67 (8.79)	16.52 (7.12)	17.96 (9.18)
FOV90	16.23 (6.93)	20.72 (10.21)	16.22 (6.99)	19.43 (9.33)	16.5 (7.02)	17.78 (9.16)
FOVmix	15.94 (6.69)	18.78 (8.95)	15.92 (6.71)	18.68 (8.84)	16.32 (6.82)	17.58 (8.63)

	Office		Shop		Total
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Total prediction
unit	meter	meter	meter	meter	meter
FOV30	16.32 (7.32)	17.5 (7.83)	16.67 (7.46)	17.56 (8.06)	16.77 (7.54)
FOV60	16.09 (7.01)	18.25 (8.40)	16.68 (7.52)	17.52 (7.67)	16.77 (7.54)
FOV90	16.26 (7.14)	18.33 (8.46)	16.70 (7.65)	17.08 (7.06)	16.77 (7.54)
FOVmix	15.92 (6.96)	17.81 (8.08)	16.68 (7.58)	17.09 (7.41)	16.77 (7.54)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over the subset.

This study found that more recently built buildings had statistically significantly more correct building function predictions (t=-35.31, p=0.00)(see table 3.21). For the mixed field of view, the building age for correct predictions was smallest out of four fields of view types. Especially the correct predictions of office function had a low average building age out of all building functions.

Table 3.21: Average building age of correct and incorrect predictions for residential, meeting, industry, office and shop functions (model = Inception-v3; cross-validation run = 0; n = 21,142)

	Residential		Meeting		Industry	
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction
unit	year	year	year	year	year	year
FOV30	115.93 (208.52)	165.80 (302.12)	105.87 (194.11)	232.19 (349.03)	95.93 (168.31)	215.59 (341.94)
FOV60	118.76 (213.96)	155.64 (291.29)	101.50 (185.96)	203.12 (322.43)	103.30 (185.18)	225.62 (352.92)
FOV90	122.18 (220.68)	142.91 (278.55)	109.42 (200.86)	197.77 (320.41)	100.13 (178.24)	215.92 (343.07)
FOVmix	115.38 (203.42)	146.94 (278.38)	95.94 (174.53)	188.99 (308.12)	90.31 (153.33)	186.4 (312.84)

	Office		Shop		Total
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Total prediction
unit	year	year	year	year	year
FOV30	88.15 (154.18)	184.04 (304.59)	109.68 (202.63)	236.40 (348.20)	124,67 (228,51)
FOV60	96.43 (173.82)	185.77 (307.20)	109.82 (203.34)	238.31 (347.57)	124,67 (228,51)
FOV90	98.89 (177.79)	203.00 (326.62)	103.58 (193.41)	215.18 (325.10)	124,67 (228,51)
FOVmix	82.53 (138.64)	176.27 (296.29)	98.59 (183.13)	211.65 (323.57)	124,67 (228,51)

Note: The first number is an average and the second number between brackets is the standard deviation. Both were calculated over the subset.

This study found that more recently shot streetview images had statistically significantly less correct building function predictions (t=30.11, p=0.00)(see table 3.22). For the meeting, industry, office, and shop function the mixed field of view had the oldest average image age. On average streetview images in the prediction are less than a year old.

Table 3.22: Average image age of correct and incorrect predictions for residential, meeting, industry, office and shop functions (model = Inception-v3; cross-validation run = 0; n = 21,142)

	Residential		Meeting		Industry	
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction
unit	days	days	days	days	days	days
FOV30	337.4 (350.63)	353.69 (428.72)	362.48 (365.58)	213.11 (338.53)	365.25 (362.22)	261.17 (364.86)
FOV60	335.44 (348.78)	365.46 (442.49)	372.5 (360.03)	231.08 (363.08)	358.14 (358.59)	255.76 (385.84)
FOV90	331.97 (346.59)	400.79 (477.87)	361.92 (357.69)	236.43 (384.65)	359.15 (352.43)	269.98 (403.07)
FOVmix	333.02 (341.34)	357.59 (417.47)	382.42 (355.38)	245.87 (370.4)	375.26 (348.77)	277.36 (386.07)

	Office		Shop		Total
subset	Correct prediction	Incorrect prediction	Correct prediction	Incorrect prediction	Total prediction
unit	days	days	days	days	days
FOV30	398.32 (366.44)	245.86 (343.83)	358.96 (367.85)	200.81 (314.86)	340,25 (365,56)
FOV60	390.68 (366.19)	231.15 (339.49)	359.31 (366.26)	194.48 (325.18)	340,25 (365,56)
FOV90	379.07 (361.25)	222.34 (353.13)	373.79 (370.12)	196.29 (306.16)	340,25 (365,56)
FOVmix	421.85 (358.67)	240.34 (348.76)	379.05 (369.87)	210.85 (318.34)	340,25 (365,56)

Note: The image age is determined in days from panorama creation day until 1st of November 2016. The first number is an average and the second number between brackets is the standard deviation. Both were calculated over the subset.

There was a statistically significant difference in the number of correct building function predictions between neighborhoods as determined by one-way ANOVA (f=122.89, p=0.00)(see figure 3.19). Only 20% of the neighborhoods were used to test the accuracy of correct predictions and these neighborhoods have a color. Other 80% of neighborhoods with no color on the map were used for either training or validation. In "Elsenhagen Zuid" and "Bijlmermuseum Zuid" correct predictions were very low. In the city center of Amsterdam, there were some neighborhoods with correct building prediction percentage around 25-50%. In other neighborhoods, the average percentage of correctly predicted building functions was significantly higher around 50 percent up to 95 percent.

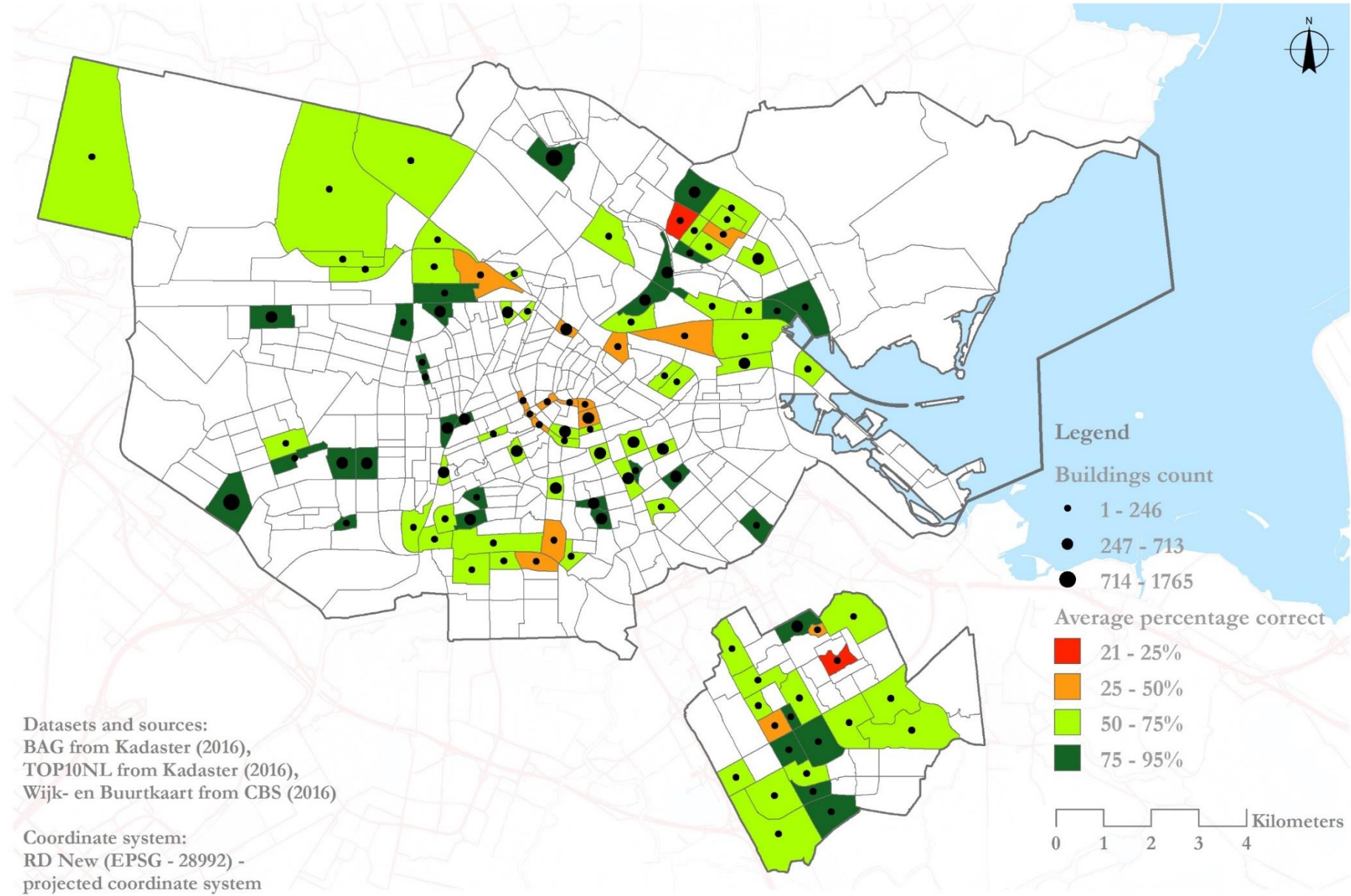


Figure 3.19: Average percentage correctly predicted building function per neighborhood in Amsterdam in 2016

Image interpretation

Buildings with their three streetview images at field of view 30, 60 and 90 degrees are shown together with ground truth label and predicted labels. The predicted labels show how many correct building function predictions the four cross-validation runs had for each building function and field of view with Inception-v3.

Residential: The image classifiers were able to correctly predict that buildings, such as residential houses or flats, had a residential function (see table 3.23-3.25). Residential buildings often have doors, windows, garage doors, pavement, trees, a balcony, and simple architectural style (see figure 3.20-3.23). Some streetview images are occluded by a tree, shrubs, cars or a crane. For a residential flat, the image classifiers were not able to predict anything correctly due to a crane obstructing the view (see table 3.26; see figure 3.24).

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.20: Streetview images (id: _gb8okkq7EtFcW95v1o0bg) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012245197)

Table 3.23: Image ground truth label and predicted label for building with ID 363100012245197 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	0	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	4/4
	FOV60	4/4	4/4	4/4	4/4	4/4
	FOV90	4/4	4/4	4/4	4/4	4/4
	FOVmix	4/4	4/4	4/4	4/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.21: Streetview images (id: F7lgKE4yR4VnDj5EUZ3Cug) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012097872)

Table 3.24: Image ground truth label and predicted label for building with ID 363100012097872 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	1	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	0/4	4/4
	FOV60	4/4	4/4	4/4	1/4	4/4
	FOV90	4/4	4/4	4/4	0/4	4/4
	FOVmix	4/4	4/4	3/4	0/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.22: Streetview images (id: XQDU-gHl6zMKe6feD1faLQ) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012242951)

Table 3.25: Image ground truth label and predicted label for building with ID 363100012242951 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	1	0	1	1
Correct prediction (out of 4 iterations)	FOV30	4/4	3/4	4/4	4/4	3/4
	FOV60	4/4	0/4	4/4	0/4	4/4
	FOV90	4/4	4/4	4/4	4/4	4/4
	FOVmix	4/4	1/4	4/4	0/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.23: Streetview images (id: DmxCFPS2ct9tqIPae6wn8Q) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012245239)

Table 3.26: Image ground truth label and predicted label for building with ID 363100012245239 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	0	0
Correct prediction (out of 4 iterations)	FOV30	0/4	0/4	0/4	1/4	0/4
	FOV60	0/4	0/4	0/4	0/4	0/4
	FOV90	0/4	0/4	0/4	0/4	0/4
	FOVmix	0/4	0/4	0/4	0/4	0/4

Meeting: The image classifiers correctly predicted the presence of meeting function for some bars and a restaurant (see table 3.27-3.29). Building with meeting functions had chairs, glass panes, benches, a bar sign, sunshades, an ad, people, doors, large windows, a restaurant sign, canopies, decorations, and multiple stories (see figure 3.24-3.26). For an image of a bar which solely showed a white wall at field of view 30 degrees, the image classifiers were not able to predict the presence of meeting function (see table 3.29; see figure 3.26). When using the zoomed out streetview images, the image classifiers were able to see the bar function correctly (see table 3.29).

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.24: Streetview images (id: tjbbemVaqN1XSKPDgjAl9g) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012171127)

Table 3.27: Image ground truth label and predicted label for building with ID 363100012171127 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		0	1	0	1	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	0/4
	FOV60	4/4	4/4	3/4	4/4	0/4
	FOV90	4/4	4/4	4/4	4/4	0/4
	FOVmix	4/4	4/4	3/4	4/4	0/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.25: Streetview images (id: 8Yt8KWOwIjrYUS5kqdj1oQ) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012179633)

Table 3.28: Image ground truth label and predicted label for building with ID 363100012179633 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	1	0	0	1
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	4/4
	FOV60	4/4	4/4	0/4	2/4	4/4
	FOV90	4/4	4/4	0/4	4/4	4/4
	FOVmix	4/4	4/4	0/4	2/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.26: Streetview images (id: UIVt_hP0foI5nBnnSKdQEQ) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012182863)

Table 3.29: Image ground truth label and predicted label for building with ID 363100012182863 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	1	0	0	0
Correct prediction (out of 4 iterations)	FOV30	0/4	1/4	0/4	0/4	0/4
	FOV60	0/4	4/4	0/4	0/4	0/4
	FOV90	0/4	4/4	0/4	0/4	0/4
	FOVmix	0/4	4/4	0/4	0/4	0/4

Industry: For an industrial shed and maintenance corridor, the image classifiers were able to correctly predict the presence of industrial function (see table 3.30-3.31). The images of the industrial building consist of garage doors, corrugated plates, dilapidated wood, pavement, a metal balcony, brick wall, small windows, a metal plate door, and balconies (see figure 3.27-3.28). For an image of a lunch bar, the image classifiers were unable to predict the presence of industrial function and predicted meeting, office and shop function (see table 3.32). The lunch bar was in front of the targeted industrial building.

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.27: Streetview images (id: 5XyO41t3gPWqgZGXV6qIHw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012127675)

Table 3.30: Image ground truth label and predicted label for building with ID 363100012127675 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		0	0	1	0	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	4/4
	FOV60	4/4	4/4	4/4	4/4	4/4
	FOV90	4/4	4/4	4/4	4/4	4/4
	FOVmix	4/4	4/4	4/4	4/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.28: Streetview images (id: EHWib7ZcQF-yabzmnAPpSA) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012099153)

Table 3.31: Image ground truth label and predicted label for building with ID 363100012099153 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	1	0	0
Correct prediction (out of 4 iterations)	FOV30	4/4	3/4	4/4	0/4	4/4
	FOV60	4/4	4/4	4/4	4/4	4/4
	FOV90	4/4	4/4	3/4	4/4	4/4
	FOVmix	4/4	4/4	4/4	1/4	4/4

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.29: Streetview images (id: bCcY6oKWoWExbfH9t2Vg8A) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012168480)

Table 3.32: Image ground truth label and predicted label for building with ID 363100012168480 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	1	0	0
Correct prediction (out of 4 iterations)	FOV30	0/4	0/4	4/4	0/4	0/4
	FOV60	1/4	0/4	0/4	0/4	0/4
	FOV90	1/4	0/4	0/4	0/4	0/4
	FOVmix	0/4	0/4	0/4	0/4	0/4

Office: The image classifiers were able to predict the office function correctly most of the time (see table 3.33-3.35). Office images consisted of a lawn, cars, trees, multi-story buildings, blinded windows, a road, brick walls, a golden office sign, a sign of a shop, a person sitting at a desk, a monumental stair, large high windows, a door, a lamp, small balconies and pavement (figure 3.30-3.32). For one building residential, shop, meeting, and industry functions were poorly predicted (see table 3.35) due to an office in a residential area and the image looking at too much context (figure 3.32).

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.30: Streetview images (id: HN21u8xn6T9o4O5YkMiXRA) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012133996)

Table 3.33: Image ground truth label and predicted label for building with ID 363100012133996 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		0	0	0	1	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	2/4	4/4	4/4
	FOV60	2/4	4/4	4/4	4/4	4/4
	FOV90	4/4	4/4	4/4	3/4	4/4
	FOVmix	4/4	4/4	3/4	1/4	4/4



FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.31: Streetview images (id: INkn7TWKbTWgwLPGubK7jw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012171669)

Table 3.34: Image ground truth label and predicted label for building with ID 363100012171669 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	1	0
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	4/4
	FOV60	4/4	4/4	4/4	4/4	4/4
	FOV90	0/4	3/4	4/4	4/4	4/4
	FOVmix	3/4	4/4	4/4	3/4	4/4

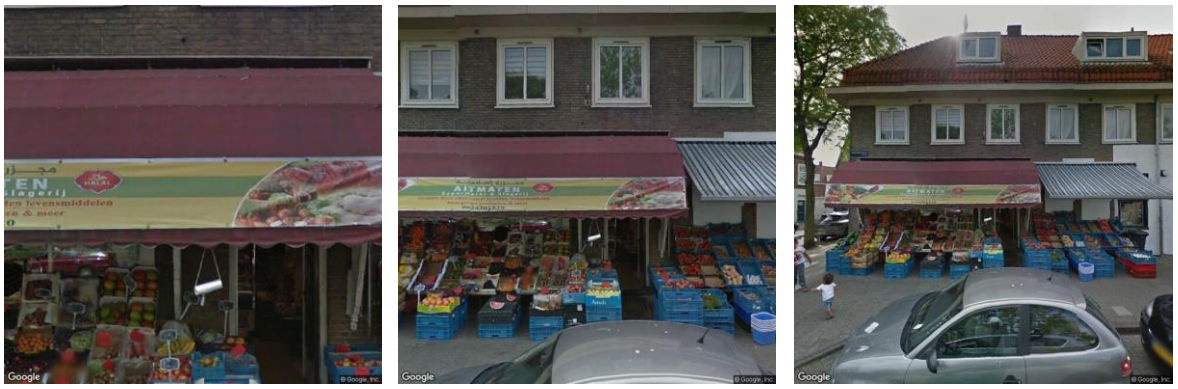

FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.32: Streetview images (id: LsKOXf1Q0fTd6nVcWN_M6g) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012074703)

Table 3.35: Image ground truth label and predicted label for building with ID 363100012074703 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		0	0	0	1	0
Correct prediction (out of 4 iterations)	FOV30	0/4	0/4	0/4	4/4	0/4
	FOV60	0/4	0/4	0/4	4/4	0/4
	FOV90	0/4	0/4	0/4	2/4	0/4
	FOVmix	0/4	0/4	0/4	3/4	0/4

Shop: For a grocery store, the image classifiers were able to correctly predict the presence of shopping function (see table 3.36). The grocery store had a canopy, shop sign, brick wall, groceries, windows, a car and a 2-story brick building (figure 3.33). For other shops, such as a shop in a residential area and a closed shop, the image classifiers had a very hard time to distinguish the shopping function (see table 3.37-3.38; see figure 3.34-3.35). If the car is far from the building and the building parcel is narrow, the streetview image shows the targeted building plus a lot of context at field of view 60 or 90 degrees (see figure 3.34).



FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.33: Streetview images (id: uuSv2M4SgVlE4KasWMt58w) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012066068)

Table 3.36: Image ground truth label and predicted label for building with ID 363100012066068 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	0	1
Correct prediction (out of 4 iterations)	FOV30	4/4	0/4	4/4	4/4	4/4
	FOV60	4/4	0/4	4/4	4/4	4/4
	FOV90	4/4	0/4	4/4	4/4	4/4
	FOVmix	0/4	0/4	4/4	4/4	4/4



FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.34: Streetview images (id: zYP2p066z-YqpHGuqXEsGw) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012166972)

Table 3.37: Image ground truth label and predicted label for building with ID 363100012166972 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	0	1
Correct prediction (out of 4 iterations)	FOV30	4/4	4/4	4/4	4/4	0/4
	FOV60	4/4	4/4	4/4	4/4	3/4
	FOV90	4/4	4/4	4/4	4/4	0/4
	FOVmix	4/4	4/4	4/4	4/4	0/4



FOV = 30° **FOV = 60°** **FOV = 90°**

Figure 3.35 : Streetview images (id: YkycpnsUo8iqVHtAvSqyog) looking with field of view of 30, 60 and 90 degrees at building (id: 363100012091039)

Table 3.38: Image ground truth label and predicted label for building with ID 363100012091039 (model = Inception-v3)

		Residential	Meeting	Industry	Office	Shop
Ground truth (1 =present; 0=absent)		1	0	0	0	1
Correct prediction (out of 4 iterations)	FOV30	1/4	4/4	0/4	0/4	0/4
	FOV60	4/4	3/4	0/4	0/4	0/4
	FOV90	0/4	0/4	0/4	1/4	0/4
	FOVmix	0/4	0/4	0/4	0/4	0/4

Chapter 4 - Discussion

This chapter discusses insights from the data acquisition, data accuracy, and building classification. It gives practical improvements, theoretical embedding and some indications for further research.

4.1 Data acquisition

Data acquisition took quite some time and it could have gone more efficiently. Therefore, these paragraphs on building and streetview data acquisition will focus mainly on practical improvements.

Building data

The BAG dataset is a high-quality dataset that holds different types of data, such as buildings and addresses. In this research, the building geometry and address information of the BAG needed to be combined. In the downloaded geodatabase, accessible via ArcMap, the different building and address data were connected based on their unique IDs via a relationship class. However, the relationship class functionality did not work in ArcMap. Therefore, a spatial join had to be performed and information from all floors was added to a 2D representation of the building. Furthermore, the functional usage column in address could have multiple usages separated by a semicolon in the same value box. Combining the relational data would have been easier in a NoSQL database or triple store. Kadaster has been working on an ontology to map their building data from relational database management systems [RDBMS] to triples (Kadaster, 2016a) and already provides the data via a SPARQL endpoint (Kadaster, 2018b).

The building data did not contain enough samples of all building functions. Few buildings had education, sport, accommodation, healthcare or cell functions. Therefore, the image classifier could not be trained on these building functions. By having a larger study area, such as a province, the classifier would

have enough samples to learn to predict these functions. Another solution would be to enlarge the data by applying data augmentation (Taylor & Nitschke, 2017). Data augmentation is a regularization scheme that artificially inflates the data-set by using label preserving transformations to add more invariant examples. These extra images are created by varying the original images: changing saturation, adding Gaussian noise, flipping, rotating, scaling, cropping or translating images. The inflated training data permits prediction of more building functions since the sample is larger. Data augmentation helps to inflate the data, which enables predicting all types of building functions.

Besides the Dutch building dataset, it could be nice to use OpenStreetMap or New York building data (PLUTO). Especially, OpenStreetMap is interesting since it provides a global dataset and it enables predicting building functions on a global scale (Srivastava et al., 2018a). More building datasets are available to perform classification on.

Streetview data

Streetview image data are available globally and accessible via an API. The downloading of Google Street View images were restricted to 25,000 free images per day per key. This took nearly three weeks. The download process could be sped up by using multiple keys or having more credits. Recently, Google started billing the downloading of Google Street View images and only allows roughly 28,000 free Google Street View downloads per month (Google, 2018b). Therefore, it might be interesting to download user-generated images from Mapillary or Flickr and do a comparison of streetview images.

Streetview images are positioned mainly in urban environments. They are sampled relatively sparse (Taneja et al., 2014) and miss views on inaccessible outdoor areas or indoor places (Zhou et al., 2014). For rural areas with low road density, streetview images are not the best source to make a complete land use map. Combining streetview data with aerial data can improve the coverage and accuracy of land use mapping (Workman et al., 2017).

The streetview images can be enhanced by looking at the building's façade instead of the building centroid. In this research, the camera was turned towards the building centroid with a static field of view with either 30, 60 or 90 degrees. It would have been better to dynamically set the heading and field of view to look only at the target building's façade (see figure 4.1). There is no data on building façades. The building façade would have to be determined by taking the edge of the building polygon closest to the road.

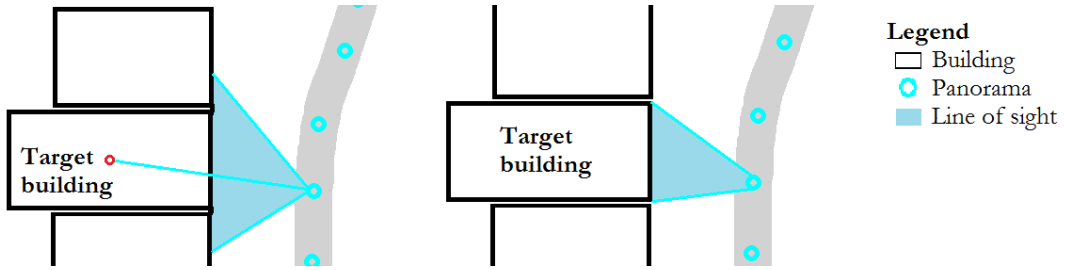


Figure 4.1: Setting heading to building centroid with a static field of view (left) and setting heading and field of view to building façade dynamically (right)

The line of sight from panorama to targeted building was often obstructed by objects, such as another building, a tree or a truck. In some situations, the obstructed line of sight could have been avoided. There are multiple ways to get to this solution. The first method would be to filter out obstructed views by having an image classifier decide if the panorama is looking at the building or at objects that obstruct the view on the building. After checking which image views on the target have obstructions (e.g. trees, cars, busses), the image view with the least obstruction could be selected. The second method would be to avoid looking at incorrect objects by using an accurate topographical map. A topographical map shows where certain objects, such as trees and other buildings, are located. Then the panorama with the least obstructed view can be picked (see figure 4.2). If there is no good view of the building, the building and image should simply be discarded from the sample to avoid inaccurate data. The approach with the topographical map does not work for mobile objects, such as cars, vans, and bikes. Therefore, panoramas with least obstructed view could be derived by using both an image classifier to detect occlusion and a topographical map to detect objects.

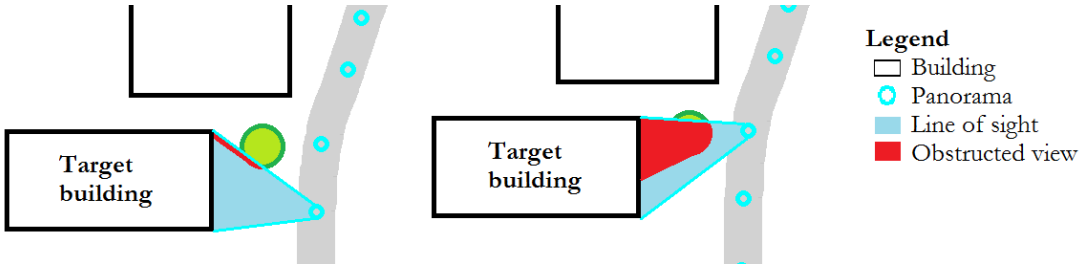


Figure 4.2: Panorama selection based on the unobstructed view

4.2 Data accuracy

The data accuracy is very important for deep learning models. Therefore, the following paragraphs will go into detail how accurate the data is. There are some practical tips to improve data accuracy and some theoretical implementations.

Building data

The label of the building function was quite accurate but did not always reflect what was visible in the downloaded streetview image. This is mainly for four reasons: wrong heading, occlusion, building information from multiple floors and indistinguishable building function. The heading and occlusion were discussed in section 4.1. During data acquisition, the building functions of multiple addresses over multiple floors were all joined to a 2D representation of the building. Unfortunately, the address data does not have information on the floor level or any other 3D information yet, but the Kadaster is taking small steps to go to 3D building registrations (Kadaster, 2016b). Another issue is the indistinguishable building function. Most of the time a human could guess the building functions from the image, but sometimes the building function was not visible or very hard to distinguish from the image. A solution would be to combine data from social media, taxi data (Niu et al., 2017) or the business registry, named Kamer van Koophandel, with streetview images to determine the building function. Building function data could have improved if building functions were available per floor and if building data was combined with data from social media, taxi data or business registries.

Streetview data

The positional accuracy of streetview panoramas was affected by the urban canyon effect and perhaps other sources. The urban canyon effect can be partly resolved by using a dual constellation system GNSS integrated with other low-cost complementary sensors (Li et al., 2017). To further improve the positional accuracy, Google should check if streetview images from cameras on cars are correctly located on the street and not on other topographical objects, such as buildings, grassland or water.

The heading was sometimes inaccurate, but it is unclear exactly why. There are several options why the heading towards the building centroid sometimes was erroneous: bad aim at building centroid instead of building the facade, faulty sensor equipment due to magnetic fields, bad gyroscope, bad heading calculation or unbeknownst reasons. Some initial test measurements in Amsterdam with a compass did not give closure to this question. It is likely that specialized compass equipment is needed to measure headings in urban environments. Finding out if the heading is correct of streetview images, why sometimes it is faulty and how to improve heading accuracy, can be part of another research or thesis.

The image view on target building was correct most of the time, but sometimes objects obstructed the view, viewpoints were different and building images varied. Obstructed views and different viewpoints can be solved by earlier solutions mentioned in section 4.1 – streetview data acquisition. The streetview images varied a lot due to illumination, weather, the shape of the building, the material of the building, and image context (Wegner et al., 2016). Image quality could be improved by looking at the building's façade and using data augmentation to overcome variation scarcity.

4.3 Building classification

The architectures Inception-v3 and MobileNetV1 were used to predict building functions, but recently better CNN architectures have been developed. The priority of choosing the CNN architecture for building function classification should be to have high prediction accuracy. Therefore, MobileNetV1 with lower prediction accuracy and computation time (Howard et al., 2017) was not suitable. Even better feature extraction architectures than Inception-v3 have been developed. AmoebaNet, Inception-v4, Inception Resnet v2 and MobileNetV2 have been developed by researchers at Google (Real et al., 2018; Sandler et al., 2018; Szegedy et al., 2017) and they outperform most other deep neural networks while keeping the number of operations reasonably low (Canziani et al., 2016; Huang et al., 2017; Wong, 2018)(see figure 4.3). The development of these new deep learning architectures is very fast. If one wants to have the best performing deep neural network, one needs to keep up to date with state-of-the-art deep learning algorithms and be open-minded to switch algorithms during longer research or industry projects.

The pre-trained CNN architectures, Inception-v3 and MobileNetV1, have predefined weights in the feature learning phase. These weights make the CNN detect certain features, gradients, and colors based on the 14 million images of the thousand generic classes of ImageNet. In this research, the weights of the feature learning phase were not changed. By training the weights of the feature learning phase, the CNN can adapt the type of features it recognizes. The CNN could have trained itself to recognize specific types of building features in the streetview image dataset. This has the potential to increase prediction accuracy but does require a large training dataset and is very computationally intensive. Pre-trained CNN architectures make highly accurate CNN quickly available but end-to-end training can offer higher prediction accuracy.

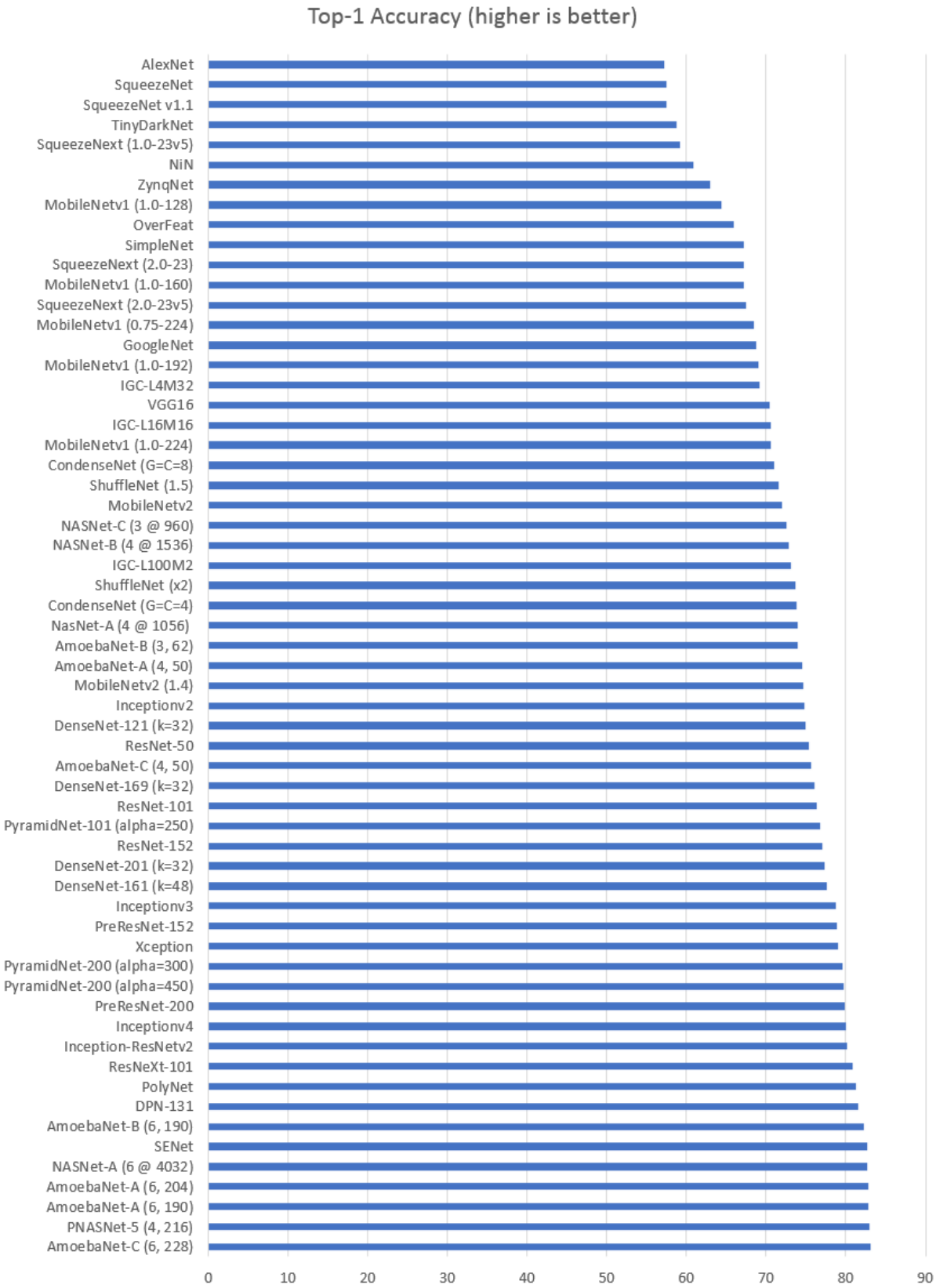


Figure 4.3: Top-1 accuracy across 60 different deep convolutional neural networks for the ILSVRC 2012 dataset (Wong, 2018)

There was a huge imbalance in the building function samples. Most buildings had only residential function and none of the other building functions. It was very uncommon to have buildings with only of these other functions. The imbalance is bad for efficiency and quality of CNN model because the model is trained on easy negatives that do not improve or even worsen the model (Lin et al., 2018) and the model trains to predict only the dominant class. There are several solutions to overcome this imbalanced sample problem: balanced cross entropy loss, focal loss, model initialization, two-stage detectors (Lin et al., 2018), balanced samples and hard negative mining. In this research, the simple balanced samples approach was used. During training of the CNN classification phase, a batch of 100 images was randomly picked from both the presence and absence of the building function. Therefore, the model was able to overcome the imbalanced problem of a dominant class, but the model did train on easy negatives or easy positives which could worsen the model. A focal loss could have improved the model by letting the model learn more from difficult predictions. Focal loss reshapes cross entropy loss such that it decreases the amount it learns from well-classified examples (Lin et al., 2018). With an imbalanced sample, multiple approaches can be tested to overcome imbalance and to find optimal balancing strategy during training of the CNN.

The image classifiers in this research were using binary classification on a multi-label problem. Every building may have any of the five building functions present or absent. The building functions are not mutually exclusive. The same building can have residential and shop function. Moreover, there is an interdependency between the building functions, which can boost accuracy if correctly used in prediction. If a building has an industry function, it is less likely to have a meeting or residential function too. This interdependency could not be predicted by a multi-class approach, because there were not enough samples per unique combination of building functions. A multi-label approach would solve the multi-attribute classification problem, where a building can have

multiple building functions as an attribute and takes interdependency into account (Gong et al., 2013; Karalas et al., 2015; Niu et al., 2017; Yu et al., 2017). Further research with the acquired data in this thesis, showed that the multi-label approach increased prediction accuracy (Srivastava et al., 2018b).

The image classifiers were using single streetview images with coarse resolution as input and could be improved by using a combination of data. Firstly, Huang et al. (2017) found out that increasing resolution of images increases mean average prediction because both small and large image features are considered. This research downloaded images with 640x640 pixels, which were resized to images with 299x299 pixels as input for Inception-v3. If the CNN and GPU could handle larger image size, the prediction accuracy could increase. Secondly, combining information of multiple streetview images, aerial images, social media or business registries could help to improve classification (Niu et al., 2017; Srivastava et al., 2018b; Wegner et al., 2016; Workman et al., 2017). In figure 4.4 an end-to-end framework is shown that combines the output of three streetview image classifiers in a ground-level feature map together with the output of an overhead image classifier to produce a building function prediction.

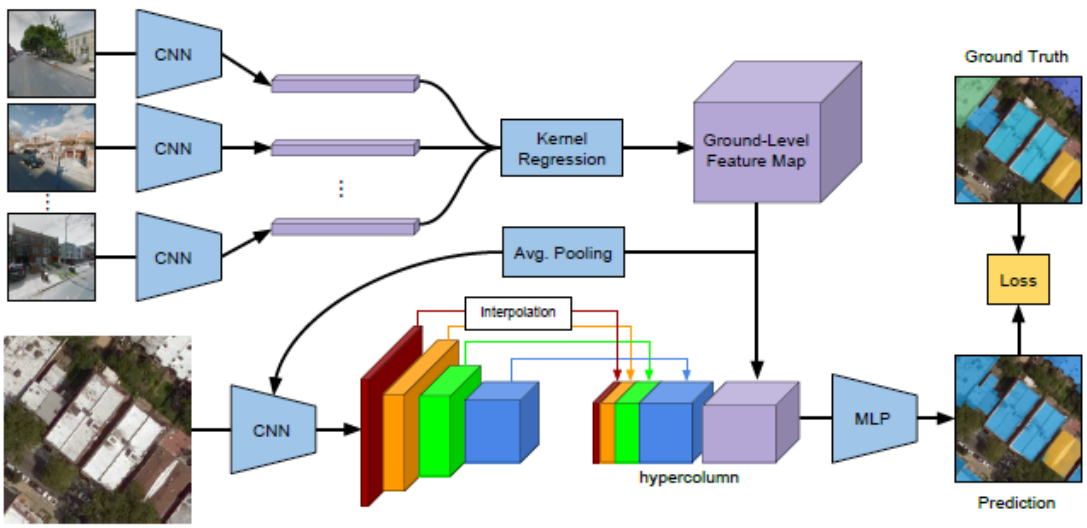


Figure 4.4: Overview of network architecture to combine multiple streetview images and an aerial image (Workman et al., 2017)

The deployed methods could drastically be improved, but results show there is valuable information visible in streetview images on building functions. Inception-v3 with its deep neural network gave high mean average prediction accuracies with a small standard deviation when predicting any of the five building functions. It had average prediction accuracies of 81% for the shop and residential function. For meeting, industry, and office the mean average prediction accuracies respectively were around 75%, 72%, and 69% percent. MobileNetV1 with its shallower neural network gave relatively low mean average prediction accuracy with a larger standard deviation. For residential, meeting, industry, office and shop function the mean average prediction accuracy respectively was 69%, 64%, 65%, 62%, and 73%. The CNN models were able to characterize building functions from the streetview images and performed better with a deeper neural network architecture.

Convolutional neural networks predicting meeting, industry, office, and shop function did have tolerable average accuracy but had low F1-scores. Residential function prediction had a high F1-score of around 0.90. Meeting, industry, office, and shop function predictions respectively had lower F1-scores of 0.20, 0.26, 0.16 and 0.36. These low F1-scores can be attributed to very low precision, due to a lot of false positives (see table 4.1 and 4.2). The recall was high due to a small set of false negatives. The streetview images of false positives were difficult to distinguish from streetview images that were true negatives.

Table 4.1: Confusion matrix office prediction
FOV = 30; Model = Inception-v3; Cross validation run 0;

		Actual		
		Office	Non-Office	Total
Predicted	Office	631	7858	8489
	Non-Office	194	12459	12653
	Total	825	20317	21142

Table 4.2: Statistical outcome measures

Sensitivity	0.764848
Specificity	0.613230
Precision	0.074331
Recall	0.764848
F1-score	0.135495
Overall accuracy	0.619147
Average accuracy	0.689039

Varying the zoom level of the streetview images lets the image classifier detect smaller or larger features of the building better. For residential and industry function a larger field of view, i.e. zooming out, gave a slightly higher mean average prediction accuracy. An assumption is that for residential and industry functions the view on the building block is important to detect the building function. For meeting and shop function a small field of view, i.e. zooming in, gave a slightly higher mean average prediction accuracy. A meeting and shop often have small features, such as labels, signs, chairs, goods, that identify the building as a meeting or shop function. Therefore, zooming in on shops and meeting places can help to improve performance. The field of view with mixed zoom levels did not perform better or worse than the other field of views. This is likely due to the training and prediction dataset not always having the same field of view. Multiple field of views of streetview images hold information on building functions.

To understand the model predictions, the streetview characteristics and building characteristics of correct and incorrect predictions were compared for all building functions and field of views. Firstly, correct predictions had a significantly smaller distance from streetview camera to building centroid than incorrect predictions for all building functions. Near things are easier to see. Secondly, the building age of correct predictions was significantly lower than the building age of incorrect predictions for all building functions. Older buildings have a very similar archaic architectural style with bricks, windows and a door irrespective of building function. Thirdly, the streetview image age of correct predictions was significantly lower than that of incorrect predictions for residential function. However, for meeting, industry, office and shop function the streetview image age was higher for correct predictions. One could argue that less time between streetview image measurement and building label measurement would lead to more correct predictions, but for some reason, this is not the case. Fourthly, there was a significant difference in correct predictions

between all neighborhoods. In some neighborhoods with tall residential flats, such as “Elsenhagen Zuid” and “Bijlmermuseum Zuid”, the prediction accuracy was very bad. It is unbeknownst why. A further understanding of CNN predictions is needed (Zeiler & Fergus, 2014), which can be achieved by looking at saliency maps, intermediate features or instance segmentations. The characteristics of the streetview images and buildings can be used to understand predictions and to select high-quality data to train the model.

A brief recap, streetview images were downloaded from Google Street View, have good quality and can be used to predict building functions with convolutional neural networks. In theory, the results are promising, but for practical applications, the prediction accuracy still needs to increase. Various data and methodological improvements have been discussed to improve prediction accuracy. Ordered by priority the improvements are: applying data augmentation, switching to better CNN architecture, solving view obstructions, targeting view on the object only, solving class imbalance, predicting multi-labels, understanding CNN predictions, combining street and aerial views, increasing image resolution and comparing multiple data sources.

Chapter 5 - Conclusion

To conclude the research on characterizing building functions from streetview images, the sub-questions and main questions are answered. Furthermore, recommendations are given how to improve prediction of building functions and how to continue research.

5.1 Research answers

Q1: How can streetview images and building functions be acquired in Amsterdam?

A good way to acquire functions of buildings in Amsterdam is to download the high-quality buildings dataset BAG, relate the functions with buildings spatially and perform a many to one join. Streetview images of buildings can be acquired from Google Street View by pointing the camera from streetview panorama location towards the building centroid with a certain zoom level and by downloading the streetview images.

Q2: How accurately can streetview images sense buildings in Amsterdam?

The accuracy of the location, heading, and image of streetview images was determined by checking streetview images belonging to 100 random buildings in Amsterdam. Firstly, the location of streetview images sensing buildings in Amsterdam is highly accurate. Validation of 100 random streetview images gave 98 correct streetview locations out of 100. Secondly, the heading of streetview images sensing buildings in Amsterdam is accurate. The heading was correct in 83 out of 100 images. Thirdly, streetview images sensing buildings in Amsterdam had an unobstructed view on the targeted building in 263 out of 300 images at field of view 30, 60 and 90 degrees. The expected accuracy of the location, heading, and image of streetview images sensing buildings in Amsterdam respectively are 98%, 83%, and 87.7%.

Q3: How accurately can convolutional neural networks characterize building functions from streetview images in Amsterdam?

Convolutional neural networks with Inception-v3 architecture gave high mean average prediction accuracies with a small standard deviation when predicting any of the five building functions. It had average accuracies of 81% for residential and shop function. For meeting, industry, and office function the mean average prediction accuracies respectively were 75%, 72%, and 69% percent. F1 scores were high for residential function with 0.90 but low for other building functions - 0.14-0.38 F1-scores - due to low precision and high recall.

Q4: What building and streetview image characteristics are associated with correct predictions of building functions from streetview images?

Characteristics, such as distance from streetview image to building, building age, streetview image age and neighborhood were significantly associated with correct predictions. Zooming out was associated with more correct predictions for residential and industrial function and zooming in was related to more correct predictions for shop and meeting functions. If the building age and distance are smaller, then the model can predict more correctly. If the streetview image age is larger, then the model can predict more correctly. Certain neighborhoods, such as inner urban areas or areas with high residential flats, were associated with less correct predictions.

Main question: How can building functions be characterized by streetview images in Amsterdam?

Building functions can be characterized accurately with streetview images by downloading streetview images and building data, selecting good quality data and using a convolutional neural network to predict building functions. If buildings are more recent and streetview images have a smaller distance to a building, the prediction accuracy of building functions improves.

5.2 Further research

Further research helps to increase knowledge and improve methods to monitor objects in the urban, rural or natural environment with streetview images.

- Research and industry should focus on improving methods of detecting obstructed views on the target object (see section 4.1)
- Research and industry should focus on improving methods to focus the camera on target objects without too much context (see section 4.1). This requires very accurate streetview camera position and heading. Therefore, quantitative research is required to determine the accuracy of position and heading.
- More research can be done how to undertake the temporal monitoring of objects with streetview images. Naik et al. (2017) have an interesting approach to monitor scenes through time with streetview images but monitoring objects can be more relevant. Google Street View cars only drive by once per year and it might be interesting to use Mapillary or other sources that have a faster revisit time.
- More research can be done how to combine top view and side view imagery with even more sources and these methods can be shared. Some good research has been done on combining top view imagery, side view images and social media data (Niu et al., 2017; Wegner et al., 2016; Workman et al., 2017), but their methods are not easy to put into practice on itself nor to be combined with other data sources.

In industry, there is still a lot of opportunities to improve the use and methods of monitoring with streetview images. Platforms, such as Mapillary and Google Street View, are great in making volunteered streetview images quickly available. This data can be used. If a municipality bans shared bicycle schemes, the location of illegal bicycles can be monitored with streetview images. A municipality can monitor demand and supply in certain areas for car parking

spots, bike stands, benches, pavement et cetera. Then municipalities can adapt the physical environment to the needs of the public better. The quality of the urban infrastructure, such as roads, can be monitored for holes in the road, traffic signs, decaying trees, type of tree or even if trees are pollinating. For autonomous vehicles, it is interesting to check with streetview images if roads have correct lines on side of the road. These are just some of the opportunities, but there will be many more applications as data will become more readily available.

References

BAG. (2010). *Kwaliteit van de basisregistraties adressen en gebouwen*. Den Haag: VROM, Ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieubeheer.

Bechtel, B., Alexander, P., Böhner, J., Ching, J., Conrad, O., Feddema, J., . . . Stewart, I. (2015). Mapping Local Climate Zones for a Worldwide Database of the Form and Function of Cities. *ISPRS International Journal of Geo-Information, 4*(1), 199-219.

Belgiu, M., Tomljenovic, I., Lampoltshammer, T., Blaschke, T., & Höfle, B. (2014). Ontology-Based Classification of Building Types Detected from Airborne Laser Scanning Data. *Remote Sensing, 6*(2), 1347-1366.

Ben-Moshe, B., Elkin, E., Levi, H., & Weissman, A. (2011). *Improving Accuracy of GNSS Devices in Urban Canyons*. Paper presented at the Canadian Conference on Computational Geometry, Toronto.

Bengio, Y., Courville, A., & Vincent, P. (2014). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798-1828.

Bouwbesluitonline. (2012). Bouwbesluit 2012 - 1.1 Algemeen. Retrieved from https://www.bouwbesluitonline.nl/Inhoud/docs/wet/bb2012/hfd1/par1-1

Canziani, A., Paszke, A., & Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Implications. *CoRR, abs/1605.07678*.

Chen, Y., Parkins, J. R., & Sherren, K. (2017). Using geo-tagged Instagram posts to reveal landscape values around current and proposed hydroelectric dams and their reservoirs. *Landscape and Urban Planning, 170*(1), 283-292.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). *ImageNet: A large-scale hierarchical image database*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, Miami.

Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing, 3*, 1-29.

Dubey, A., Naik, N., Parikh, D., Raskar, R., & Hidalgo, C. (2016). *Deep Learning the City: Quantifying Urban Perception at a Global Scale*. Paper presented at the European Conference on Computer Vision, Amsterdam.

Eeftens, M., Beelen, R., de Hoogh, K., Bellander, T., Cesaroni, G., Cirach, M., . . . Hoek, G. (2012). Development of Land Use Regression models for PM(2.5), PM(2.5) absorbance, PM(10) and PM(coarse) in 20 European study areas; results of the ESCAPE project. *Environmental Science and Technology, 46*(20), 11195-11205.

Erb, K.-H., Gaube, V., Krausmann, F., Plutzar, C., Bondeau, A., & Haberl, H. (2007). A comprehensive global 5 min resolution land-use data set for the year 2000 consistent with national census data. *Journal of Land Use Science, 2*(3), 191-224.

ESRI. (2016). FAQ: What do the acronyms EPSG and POSC stand for? Retrieved from https://support.esri.com/en/technical-article/000002814

EuropeanParliament. (2016). REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union, 119*(1), 1-88.

Frias-Martinez, V., Soto, V., Hohwald, H., & Frias-Martinez, E. (2012). *Characterizing Urban Landscapes using Geolocated Tweets*. Paper presented at the Privacy, Security, Risk and Trust, Amsterdam.

Fritz, S., McCallum, I., Schill, C., Perger, C., See, L., Schepaschenko, D., . . . Obersteiner, M. (2012). Geo-Wiki: An online platform for improving global land cover. *Environmental Modelling and Software, 31*(1), 110-123.

Ghamisi, P., Plaza, J., Chen, Y., Li, J., & Plaza, A. J. (2017). Advanced Spectral Classifiers for Hyperspectral Images: A review. *IEEE Geoscience and Remote Sensing Magazine, 5*(1), 8-32.

Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2013). Deep Convolutional Ranking for Multilabel Image Annotation. *CoRR, abs/1312.4894*.

References

Goodfellow, I., Bulatov, Y., Arnoud, S., & Shet, V. (2013). Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. *CoRR, abs/1312.6082*.

Google. (2018a). Installing Tensorflow on Ubuntu. Retrieved from https://www.tensorflow.org/install/install_linux

Google. (2018b). Street View API Usage and Billing. Retrieved from https://developers.google.com/maps/documentation/streetview/usage-and-billing

Google. (2018c). Street View auto ready specifications. Retrieved from https://developers.google.com/streetview/ready/specs-prograde

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing, 187*(1), 27-48.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. Paper presented at the IEEE International Conference on Computer Vision, Santiago.

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR, abs/1704.04861*.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., . . . Murphy, K. (2017). *Speed/accuracy trade-offs for modern convolutional object detectors*. Paper presented at the Computer Vision and Pattern Recognition, Honolulu.

Jokar, J., Helbich, M., Bakillah, M., Hagenauer, J., & Zipf, A. (2013). Toward mapping land-use patterns from volunteered geographic information. *International Journal of Geographical Information Science, 27*(12), 2264-2278.

Kadaster. (2016a). Primeur: linked open data van het Kadaster. Retrieved from https://www.kadaster.nl/primeur-linked-open-data-van-het-kadaster

Kadaster. (2016b). Spoorzone Delft allereerste akte met 3d-weergave van rechten. Retrieved from https://www.kadaster.nl/spoorzone-delft-allereerste-akte-met-3d-weergave-van-rechten

Kadaster. (2016c). Verbeter de kaart. Retrieved from https://www.verbeterdekaart.nl

Kadaster. (2018a). BAG kwaliteitsdashboard voor afnemers. Retrieved from https://www.kadaster.nl/bag-kwaliteitsdashboard-voor-afnemers/dashboard?theme=BAGBOG&category=GOW&view=province&province=&community=

Kadaster. (2018b). SPARQL Endpoint. Retrieved from https://data.pdok.nl/sparql#

Karalas, K., Tsagkatakis, G., Zervakis, M., & Taskalides, P. (2015). Deep Learning for Multi-Label Land Cover Classification. *Image and Signal Processing for Remote Sensing XXI, 9463*, 1-14.

Khosla, A., An, B., Lim, J., & Torralba, A. (2014). *Looking Beyond the Visible Scene*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, Columbus.

Kovalskyy, V., & Roy, D. P. (2013). The global availability of Landsat 5 TM and Landsat 7 ETM+ land surface observations and implications for global 30m Landsat data product generation. *Remote Sensing of Environment, 130*(1), 280-293.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceeding of the IEEE, 86*(11), 2278-2323.

Leung, D., & Newsam, S. (2015). Land cover classification using geo-referenced photos. *Multimedia Tools and Applications, 74*(24), 11741-11761.

Li, X., Jiang, R., Song, X., & Li, B. (2017). A Tightly Coupled Positioning Solution for Land Vehicles in Urban Canyons. *Journal of Sensors, 2017*, 1-11.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal Loss for Dense Object Detection. *Transactions on Pattern Analysis and Machine Intelligence*, 1-1.

MathWorks. (n.d., 16-04-2018). Convolutional Neural Network. Retrieved from https://www.mathworks.com/discovery/convolutional-neural-network.html

Naik, N., Kominers, S. D., Raskar, R., Glaeser, E. L., & Hidalgo, C. A. (2017). Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences of the United States of America, 114*(29), 7571-7576.

References

Naik, N., Philipoom, J., Raskar, R., & Hidalgo, C. (2014). *Streetscore - Predicting the Perceived Safety of One Million Streetscapes*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus.

Niu, N., Liu, X., Jin, H., Ye, X., Liu, Y., Li, X., . . . Li, S. (2017). Integrating multi-source big data to infer building functions. *International Journal of Geographical Information Science, 31*(9), 1871-1890.

Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Pulighe, G., Baiocchi, V., & Lupia, F. (2015). Horizontal accuracy assessment of very high resolution Google Earth images in the city of Rome, Italy. *International Journal of Digital Earth, 9*(4), 342-362.

Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2018). Aging Evolution for Image Classifier Architecture Search. *CoRR, abs/1802.01548v5*, 1-14.

Salesses, P., Schechtner, K., & Hidalgo, C. A. (2013). The collaborative image of the city: mapping the inequality of urban perception. *PLoS One, 8*(7), 1-22.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *CoRR, abs/1801.04381*, 1-14.

Sitthi, A., Nagai, M., Dailey, M., & Ninsawat, S. (2016). Exploring Land Use and Land Cover of Geotagged Social-Sensing Images Using Naive Bayes Classifier. *Sustainability, 8*(9), 921-943.

Srivastava, S., Lobry, S., Tuia, D., & Vargas-Muñoz, J. (2018a). *Land-use characterization using Google Street View pictures and OpenStreetMap*. Paper presented at the Association of Geographic Information Laboratories in Europe, Lund.

Srivastava, S., Vargas-Muñoz, J., Swinkels, D., & Tuia, D. (2018b). *Multi-label Building Functions Classification from Ground Pictures using Convolutional Neural Networks*. Paper presented at the GeoAI, Seattle.

Steiniger, S., Lange, T., Burghardt, D., & Weibel, R. (2008). An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques. *Transaction in GIS, 12*(1), 31-59.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. Paper presented at the Artifical Intelligence, San Fransisco.

Taneja, A., Ballan, L., & Pollefeys, M. (2014). *Never Get Lost Again: Vision Based Navigation Using StreetView Images*. Paper presented at the Asian Conference on Computer Vision, Singapore.

Taylor, L., & Nitschke, G. (2017). Improving Deep Learning using Generic Data Augmentation. *CoRR, abs/1708.06020*.

Tracewski, L., Bastin, L., & Fonte, C. C. (2017). Repurposing a deep learning network to filter and classify volunteered photographs for land cover and land use characterization. *Geo-spatial Information Science, 20*(3), 252-268.

Verhoeve, A., Dewaelheyns, V., Kerselaers, E., Rogge, E., & Gulinck, H. (2015). Virtual farmland: Grasping the occupation of agricultural land by non-agricultural land uses. *Land Use Policy, 42*(1), 547-556.

Volpi, M., & Tuia, D. (2016). *Semantic labelling of aerial images by learning class-specific object proposals*. Paper presented at the IEEE International Geoscience and Remote Sensing Symposium, Beijing.

Wegner, J., Branson, S., Hall, D., Schindler, K., & Perona, P. (2016). *Cataloging Public Objects Using Aerial and Street-Level Images – Urban Trees*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas.

Wired. (2017). Google's new street view cameras will help algorithms index the real world. Retrieved from https://www.wired.com/story/googles-new-street-view-cameras-will-help-algorithms-index-the-real-world/

Wong, A. (2018). NetScore: Towards Universal Metrics for Large-scale Performance Analysis of Deep Neural Networks for Practical On-Device Edge Usage. *CoRR, abs/1806.05512*, 1-9.

Workman, S., Zhai, M., Crandall, D., & Jacobs, N. (2017). *A Unified Model for Near and Remote Sensing*. Paper presented at the International Conference on Computer Vision, Venice.

Yu, Q., Wang, J., Zhang, S., Gong, Y., & Zhao, J. (2017). Combining local and global hypotheses in deep neural network for multi-label image classification. *Neurocomputing, 235*, 38-45.

Zeiler, M., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision - ECCV 2014.* (Vol. 8689). Cham: Springer.

Zhou, B., Liu, L., Oliva, A., & Torralba, A. (2014). *Recognizing City Identity via Attribute Analysis of Geo-tagged Images*. Paper presented at the European Conference on Computer Vision, Zurich.

Zhu, X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine, 5*(4), 8-36.

Appendix

Appendix 1: table of content of the USB that accompanies the thesis report

- Report (Word, PDF)
- Midterm & Final presentation (PPTX)
- Datasets used and created
- Scripts
- Literature