Imperial College London

Imperial College London

School of Public Health

Measuring Social and Health Inequalities using Machine Learning and Object Detection with Street View Imagery

CID:

01603194

Author:

Antje Barbara Metzler

Word count:

10,282

Supervisor:

Prof. Majid Ezzati

A thesis submitted for the degree of

MSc Health Data Analytics and Machine Learning

August 30, 2019

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout this project. First, I would like to thank my supervisor Prof. Majid Ezzati for the constant support and constructive criticism throughout the project. I am also very thankful for the guidance of Emily Muller, Dr. Esra Suel, Dr. Ricky Nathvani and Dr. James Bennett, and their invaluably advice during the project work. Additionally, I'd like thank my parents for the opportunities they have offered me, as well as their unquestioned support. Finally, thank you to my sisters for being the best role models I could wish for and I also thank my friends for their passion, inspiration and making London feel like home.

Contents

1 Introduction **2**

1.1 Background . . . 2

1.2 Research Objective . . . 3

1.3 Research Question . . . 4

2 Literature and Data Review **5**

2.1 Literature Review . . . 5

2.2 Datasets . . . 7

2.2.1 Imagery Data . . . 8

2.2.2 Outcome (Label) Data . . . 8

3 Research Methods **12**

3.1 Research Approach . . . 12

3.2 Research Ethics . . . 12

3.3 Two-Step Modelling . . . 13

3.3.1 Step one: Deep Learning and Object Detection . . . 13

3.3.2 Data Sampling and Preparation . . . 17

3.3.3 Step two: Machine Learning Classifiers . . . 18

3.4 Evaluation and Validity . . . 20

3.4.1 Classifier Evaluation . . . 20

3.5 Software, Hardware, Data and Code Management . . . 22

3.5.1 Software and Hardware . . . 22

3.5.2 Data and Code Management . . . 22

4 Results **24**

4.1 Step one: Object Detection . . . 24

4.1.1 Inequalities in Measures of Income and Wellbeing in London . . . 29

4.2 Step two: Machine Learning Classifier 30
4.2.1 10 Most Frequent Objects Detected 30
4.2.2 Individual Objects 32

5 Discussion **35**

6 Conclusion **38**
6.1 Strengths and Limitations 38
6.2 Recommendations for Further Study 39
6.3 Conclusion 41
Appendix 42

Bibliography **50**

Acronyms

AI Artificial Intelligence. 3–6, 41

AP Average Precision. 16

API Application Programming Interface. 8, 23

CAM Class Activation Map. 36, 39–41

CNN Convolutional Neural Network. 3, 13, 15, 16, 38

COCO Common Objects in Context. 13–17, 24, 27, 36, 40

LMIC Low- and Middle-Income Countries. 2, 3

LSOA Lower Layer Super Output Area. 8–10, 12, 13, 17, 18, 25–27, 32, 33, 37, 39

MAE Mean Absolute Error. 18, 21, 30, 32, 34, 35

mAP mean Average Precision. 15–17, 38

R-CNN Region convolutional neural network. 15, 16

RBF Radial Basis Function. 19

RDS Research Data Store. 22, 36

RFC Random Forest Classifier. 13, 19, 24, 30, 32, 34, 38

RPN Region Proposal Network. 16

SVC Support Vector Classification. 13, 19, 24, 30, 33, 34, 38

ULEZ Ultra Low Emission Zone. 26

UN United Nations. 2

VGG Visual Geometry Group. 16

YOLO You Only Look Once. 15

List of Figures

2.1 Example of a Google Street View panoid image of London sliced in equally sized image cut-outs. . . . 8
2.2 Mean Income . . . 9
2.3 Liv. Environment . . . 9
2.4 Health . . . 9
2.5 Mean Income . . . 9
2.6 Liv. Environment . . . 9
2.7 Health . . . 9

3.1 Example of the Mask R-CNN object detection algorithm run on a test image of the streets of London. . . . 14
3.2 High level diagram of the detection meta-architectures of the Faster R-CNN algorithm [1](p.3). . . . 17
3.3 Total number of panoids per LSOA. . . . 18
3.4 Schematic structure of grid search with 5-fold cross-validation algorithm [2]. . . . 21

4.1 Total number of top 15 objects detected in all LSOAs. . . . 25
4.2 Sum of top 10 objects detected per panoid per LSOA. . . . 25
4.3 Number of cars detected per panoid per LSOA. . . . 26
4.4 Number of people detected per panoid per LSOA. . . . 27
4.5 Number of specific objects detected per panoid per LSOA. . . . 28
4.6 Correlation matrix: pairwise correlation of the 10 object classes. . . . 29
4.7 Correlation matrix heatmap of the SVC for each inequality outcome. . . . 31
4.8 Correlation matrix heatmap of the RFC for each inequality outcome. . . . 31
4.9 Negative linear trend: Boxplots of the living environment deprivation decile and the number of people detected. . . . 33

List of Tables

4.1 Results of SVC run on 10 most commonly detected objects and the inequality measures. 31

4.2 Results of RFC run on 10 most commonly detected objects and the inequality measures. 31

4.3 Results of SVM run on number of cars detected and the inequality measures . . 32

4.4 Results of RFC run on number of cars detected and the inequality measures . . 32

4.5 Results of SVM run on number of people detected and the inequality measures. 33

4.6 Results of RFC run on number of people detected and the inequality measures. . 33

1 Total counts of objects detected 42

2 Hyperparameters for SVC 43

3 Hyperparameters for RFC 43

Abstract

Currently, it is estimated that half of the world's population lives in cities, and this figure is predicted to double by 2050. Worldwide, these urban populations are facing growing social and health inequalities with gaps of up to ten years in life expectancy in London. Urban inequalities are multidimensional, with inequality outcomes varying remarkably in time and space across a city. However, measuring demographic factors at a high temporal and spatial resolution to analyse urban inequalities is time-consuming and labour-intensive. Considering that digital imagery data is increasingly collected and processed, Street View data analysis could serve as viable and cost-effective alternative to current strategies. Additionally, recent advances in high-throughput computing allow the application of deep learning techniques to large datasets, such as urban imagery data.

This Master's thesis explored the so-called 'meet-in-the-middle' approach, where objects were used as intermediate markers between images and outcomes. For this purpose, a pre-trained network was used to identify specific pre-defined visual markers of social and health related status with an object detection algorithm (e.g. detecting cars, people and bicycles). Further, the relationship between the detected objects and the inequality outcomes, namely mean income, living environment deprivation and health deprivation were examined with two different machine learning classifiers, a support vector classifier and a random forest classifier. The living environment deprivation scored the highest prediction performance, followed by mean income and health deprivation. The 'meet-in-the-middle' approach could not achieve an allocation performance as high as the end-to-end solution, which is often considered as a black box. Yet, the results suggests that introducing heuristics, such as objects, into a model can improve the models' interpretability while also reducing the data and computing power required. Further research could encompass including additional information to the classifiers and the analysis of class activation maps to detect added evidence on the importance of objects in imagery data.

Chapter 1

Introduction

1.1 Background

Today more than half of the world's population lives in cities. The United Nations (UN) estimates that by 2045 more than six billion people will be living in urban areas, with the largest growth expected in cities in low- and middle-income countries [3]. Although overall living standards in cities are higher compared to those in rural areas, inequalities among city dwellers are large and increasing. Massive disparities in life expectancy and income exist over short distances, even in cities such as London [4, 5]. Ezzati [5] points out that urbanisation could be used as an opportunity to not only improve urban public health, but also to raise cities as 'nodes in a [...] global network' to advance overall population health on country level (p.1). To take up this opportunity of directing urban development, it is important to understand urban dynamics and health inequalities. By finding ways to capture urban inequalities, the impact of urban policies and programmes can be modelled in greater detail and thus aid in selecting the most efficient way towards health equity. But which features collectively make certain parts of a city more liveable and healthier than others?

Various social and economic inequalities persist in London, such as wealth, access to health care and recreation services, and the living environment, to name a few. These inequalities are multidimensional and vary in space and time. Understanding and analysing these heterogeneous data can be challenging, mostly because the data that captures environmental variation and dynamics are high dimensional and need to be measured at high temporal and spatial resolution. Up to the present, inequality measures are largely presented as statistics gathered by state authorities or agencies based on empirical evidence. This census data is not only unavailable in most low- and middle-income countries Low- and Middle-Income Countries

(LMIC) but also not regularly updated [6], even in a city like London. Recent advances in high-throughput computing have revealed the potential of using imagery data to analyse complex urban environments, which provide a new opportunity for studying cities.

Today's machine learning techniques, such as deep neural networks, reduce the dimensionality of large-scale data without compromising underlying complex patterns, and can therefore identify patterns in eclectic datasets. Imagery data, such as Google Street View images, are available for cities of different countries and can capture urban variations, encompassing information about living conditions, vicinity to roads or services, and green spaces that may not be available through other means. Deep learning has become an increasingly popular tool in disciplines not previously associated with the technology, thus there have been a number of empirical investigations into the potential of Artificial Intelligence (AI) and imagery data in the context of environmental health. Nonetheless, in most publications neural networks are trained on a full dataset, which requires a sufficient amount of data and computational resources.

This thesis builds upon research conducted by the 'Pathways to Equitable Healthy Cities' group, especially a paper by Suel et al. [7], in which a Convolutional Neural Network (CNN) is used to predict inequality outcomes directly from raw images. Contrary to her approach, this thesis investigated whether introducing heuristics to the model can produce similar results and how this method could be used as an alternate, less computationally expensive approach to model inequalities in cities. This heuristics approach also adds an element of interpretation, which allows for more intuitive and intelligible modelling.

1.2 Research Objective

This thesis aimed to investigate the opportunities and challenges of deep learning with imagery data in an environmental global health context in the Greater London area. It builds upon the research by Suel [7], who hypothesises that street level images can be representative of spatial distributions of income, unemployment, health, education and crime. This thesis explored the question of whether certain heuristics, such as objects detected in the pictures, could be used to predict inequality measures instead of relying on a whole image dataset and an end-to-end training method. The Google Street View images are run through a pre-trained Convolutional Neural Network (CNN) to obtain objects detected within each pictures. Finally, the relationship of the objects detected and the inequality outcomes were analysed. Additionally, it was discussed whether heuristics can help to simplify a model, and if the discriminative attributes of the CNNs can be identified as objects in the image, rather than more loosely defined visual

compositions.

1.3 Research Question

The thesis approached the problem of using deep learning with imagery and object detection to explore the inequalities in cities with following question:

To what extent can heuristics in imagery data - objects detected in an urban environment - predict measures of urban inequalities, such as mean income, living environment and health deprivation, using deep learning and supervised machine learning classifiers by the example of Greater London?

This research question lead me through the machine learning and statistical modelling of inequalities with imagery data. Further, I explored if a 'meet-in-the-middle' approach delivers similar results as the end-to-end solution. Answering the research question also improved the understanding of creating more interpretable models to analyse urban inequalities.

The thesis is structured into five chapters with additional information, such as tables and charts, added in the appendix. I firstly explain the background and research objective. Secondly, I review relevant literature on the topic of using AI to understand environments and further introduce the imagery and outcome label datasets employed. Thirdly, I describe the methods of the research, which are split into object detection for imagery data of London and machine learning classifiers. Next, I present the results and their interpretation. Finally, I conclude the thesis by identifying the strengths as well as the limitations to this research and point out recommended further study.

Chapter 2

Literature and Data Review

In this chapter, I firstly review literature around the conceptual idea behind this thesis: understanding cities with imagery data. Secondly, I take a closer look at methodological papers, mostly concerning the use of AI to study urban environments. Furthermore, I also examine the concept of transfer learning and machine learning classifiers and how they are applied to answer socioeconomic questions. In addition, I introduce Suel's [7] research, in which she demonstrates how street imagery has the ability of capturing inequality measures. Finally, I explain the imagery and outcome label datasets used and how they were acquired.

2.1 Literature Review

Do People Shape Cites, or Do Cities Shape People?

The idea of inferring socioeconomic status from visual inspections is not a novel one. In 1902, Charles Booth published a book in which he coloured the streets in maps of London based on poverty and wealth status that were inferred from the state of streets, homes and clothing of residents [8]. His premise is based on the idea that wealth has visual correlations in housing, street setup and vehicles. In line with Booth's work, several researchers have tried to deduce information from visual inspections and the image representation of such. Rundle [9], for example, offers evidence that Google Street View imagery can be used instead of in-person auditing of neighbourhood environments. The author further states that images bear the potential of revealing public health and behaviour. Additionally, Steele [6] demonstrates that satellite imagery and mobile phone data can also be utilised to model socioeconomic outcomes. In this paper the author created hierarchical Bayesian geostatistical models to map poverty rates. Overcoming data shortages in direct measurement is a key advantage of using imagery

data for modelling social, health and economic outcomes.

Salesses [10] argues that for the most part there are two different narratives to the research of cities. Salesses defines them as either 'emphasis on a city's built environment' or the 'connection between demographic and economic variables, with the physical appearance [...] playing little or no role' (p.3). In his paper the author tries to investigate the relationship of both narratives by analysing geo-tagged images in multiple cities and rating them by perception of safety, class as well as uniqueness. The research team finds a significant correlation between the perception of safety of the images and the number of homicides (controlling for income, population and age). On the contrary, crime predictions based on observations, such as described in the 'broken windows theory' by Kelling [11], have recently been challenged by O'Brien [12]. The author suggests that broken windows, soiled surfaces and streets, as well as graffiti only increase the perception of crime and not necessarily the crime rate itself. Sampson et al. [13] further imply that observed disorder can predict the disorder perceived, however economic and social background are of greater importance.

Deep Learning using Imagery Data for Understanding Environments

Increasing computing power has unlocked the potential of applying deep learning techniques to large datasets to extract and interpret complex information. Only in the last few years there has been some discussion on the use of imagery data, such as Google Street View and its potential to aid the comprehension of environments in a public health context. Weichenthal [14] describes in his paper how AI can aid exposure science and environmental health research in the future. The article discusses how various spatially-correlated exposures can potentially be captured by an image as a representative measure. Additionally, it points out that deep learning techniques will help to understand environmental impacts on health and 'may allow for analyses to be efficiently scaled for broad coverage' (p.1). Further, Helbich [15] successfully utilises AI in his research to investigate the relationship of green or blue view with geriatric depression in Beijing. In the paper he describes how deep learning is used to quantify and analyse green and blue spaces in images of the city. Helbich also states that Street View images combined with deep learning methods 'provide a valuable tool for automated environmental assessments of physical streetscapes, applicable to large epidemiological studies' (p.115). Moreover, researchers have tried to map urban greenery and landscapes with Google Street View imagery and computer vision. Richards, Li and Seiferling utilised deep learning image segmentation techniques to quantify and model urban landscapes to measure and analyse urban dynamics [16, 17, 18].

This idea is picked up by Suel et al., who investigates in her article 'Measuring social, environmental and health inequalities using deep learning and street imagery' [7] whether deep learning approaches can capture inequality measures, such as unemployment and income, using imagery data. The author applies a deep learning approach to Google Street View images of London to create a model that predicts the outcomes previously mentioned and how these predictions compare for these different outcomes. She further explores the transferability of the model by testing the network on other cities in the United Kingdom.

Visual Heuristics

This thesis will introduce certain heuristics captured by an object detection algorithm, such as cars, people and potted plants, to predict multiple inequality measures. In that way, I explored the question of whether known indicators of wealth can act as heuristics in an inequality model and thus create similar results as the end-to-end predictions. Only a small number of researchers have tried to use visual heuristics to explain non-visual attributes. In order to quantify the visual changes of a city, for example, Naik [19] attempted to introduce a metric for change in appearance with computer vision in his paper. Moreover, Arietta et al. [20] successfully identifies visual city characteristics that can predict non-visual city attributes, such as crime statistics, housing prices, population density, with non-linear Support Vector Regression. The authors thus offer a way to make the investigation of urban socioeconomic attributes more intuitive. Gebru [21] reported the correlation between make and model of cars, which were detected in Google Street View images, with the election results in the United States in her research. The research team suggests that imagery data - together with visual heuristics contained in the pictures - could be used to complement more labour-intensive approaches to detect socioeconomic trends with fine spatial and temporal resolution.

2.2 Datasets

In order to answer the research question posed, various datasets, imagery data and inequality measures, from several data sources were required. The following section explains how the data was collected, pre-processed and how it is presented.

2.2.1 Imagery Data

The imagery dataset is composed of street level images and defined by the Greater London area. For direct comparison purposes, this thesis adopted the identical imagery dataset as used in Suel's research paper [7]. The images were collected with the Google Street View Application Programming Interface (API) [22]. Furthermore, the author obtained the 181,150 postcodes assigned to the 33 local authority districts of the Greater London administrative area from the Office for National Statistics Postcode Directory for the United Kingdom [23]. At each postcode the API returned the identifier of the nearest available panorama image that was most recently taken by Google. The timestamps of the so-called 'panoids' range from 2008 to 2017. In Greater London panoids were available for 156,58 postcodes. In order to obtain the images that cover a 360 degree panorama, the 131,465 panoids were saved as four equally sized image cut-outs. Thus the researchers obtained in total a number of 525,860 images representative of 156,581 postcodes in the city. [7] As depicted in Figure 2.1, the panoids were sliced so that each cut-out corresponds to the left, front, right and rear view of the image capturing vehicle. Throughout this thesis, the term panoid will be used to refer to each image id, which contains a set of four images.

Figure 2.1: Example of a Google Street View panoid image of London sliced in equally sized image cut-outs.

2.2.2 Outcome (Label) Data

Similarly to the imagery data, parts of the outcome label data that was obtained for Suel's publication has been used for this thesis [7]. Suel acquired various datasets on human health and wealth, including mean household income [24], living environment deprivation [25] as well as health deprivation [25]. All outcomes were measured on Lower Layer Super Output Area (LSOA) level and saved for 4,838 LSOAs of London. The LSOA is a geographical code used by

the Office of National Statistics to describe an area with a population of approximately 1500 [26]. Further, the outcomes were converted into deciles by LSOA, with decile 1 indicating worst-off and 10 indicating the best-off decile. However, it is important to consider that deprivation indices, such as living environment and health deprivation deciles, do not necessarily linearly capture how LSOAs differ in relation to wealth, as a result of a focus on the lower well-being outcomes.

In the following section I describe each outcome label and the distribution of the outcome label datasets. Figures 2.2 to 2.4 show that the distribution of deciles varies per panoid, in comparison to an equal distribution on the LSOA level. Figures 2.5 to 2.7 show the distribution of the deciles on the map of Greater London.

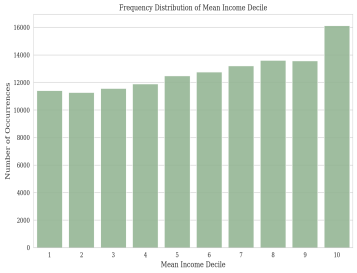

Figure 2.2: Mean Income

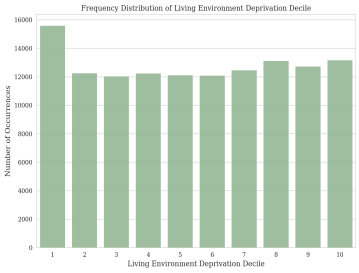

Figure 2.3: Liv. Environment

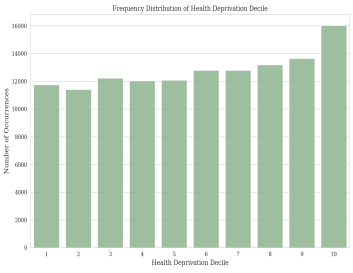

Figure 2.4: Health

Figures 2.2 - 2.4 Frequency distribution of outcome deciles.

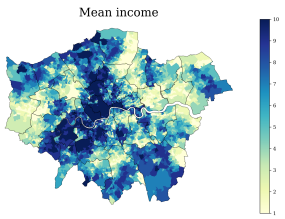

Figure 2.5: Mean Income

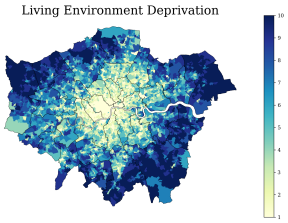

Figure 2.6: Liv. Environment

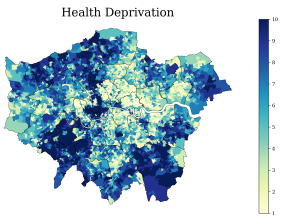

Figure 2.7: Health

Figures 2.5 - 2.7 Maps of outcome deciles.

Mean Income Decile

The mean income data was retrieved from the London Datastore published by the Greater London Authority [24]. The dataset consists of continuous values from the year 2012 and 2013. The mean income is further divided into deciles in relationship to Greater London, ranging from 1 to 10, with 10 indicating the most-affluent LSOAs. Figure 2.2 depicts the distribution of the

mean income deciles per image id, with 2 being the least frequent decile, and 10 being the most frequent decile featured in the dataset. Figure 2.5 illustrates the geographical representation of the LSOAs and its respective mean income deciles. The inner city of London, as well as some parts in the north, such as Hampstead, and parts in the southwest, such as Richmond and Twickenham, are represented with the highest mean income decile. The economically less fortunate areas of London can be found in the very west, such as Hayes, and northeast parts of the city.

Living Environment Deprivation Decile

The living environment deprivation decile was acquired from the Office for National Statistics open database in 2015 and is computed by taking multiple estimates into account [27]. Those factors include road accident rates per 1000 inhabitants, poor housing conditions as defined in not meeting the 'Decent Homes' UK standard and houses without central heating indicating high heating costs. The underlying data was collected by the UK Department of Transport, UK Air Information Resource and the Office for National Statistics [27, 7].

Figure 2.3 depicts the distribution of the living environment deprivation deciles, with 3 being the least frequent decile, and 1 being the most frequent decile found in the dataset. Figure 2.6 illustrates the geographical representation of the living environment deprivation deciles per LSOA. The figure indicates that the best-off areas of London in terms of living environment are all located in the outskirts of the city. The less fortunate areas of London in terms of living environment are found in the inner city of London and its surroundings.

Health Deprivation Decile

Similarly to the living environment deprivation decile, the health deprivation decile has also been obtained from the Office for National Statistics open database [27]. The index is calculated by taking mortality, morbidity and hospital admissions rates into account. The Office for National Statistics defines this index as indicator for 'risk of premature death and impairment through poor physical or mental health' (p.2) [7].

Figure 2.4 depicts the distribution of the health deprivation deciles, with 2 being the least frequent decile, and 10 being the most frequent decile. Figure 2.7 delineates the geographical representation of the health deprivation deciles per LSOA, indicating that the deprivation deciles are not as evenly spread across the city. The higher deciles are found in the inner city, as well as Kensington in the west and parts of the southeast and southwest of the city.

Generally, the most health deprived people are located in the east of London, such as Stratford.

Chapter 3

Research Methods

3.1 Research Approach

This thesis combined several research methods from different disciplines, such as statistics, machine learning as well as epidemiology. This highly cross-disciplinary research enables the generation of novel insights, but also faces various challenges. While I drew research methods from different disciplines, my overall research approach took an inductive manner. The two-step modelling approach was inspired by Chadeau's publication on 'meet-in-the-middle' [28] modelling for metabolic profiling. The paper describes how 'identifying the overlap between markers of exposure and predictive markers of disease outcome' (p. 84) has the potential of discovering new predictive features and thus could play an important role in disease prevention. Although my application was rather different from genetic modelling, the idea that certain assumptions or previously chosen factors can help to improve a model, is the same in both cases. Contrary to the 'meet-in-the-middle' approach, an end-to-end solution, such as a neural network, acts more like a black box, which makes the model harder to interpret.

Firstly, I obtain the objects detected from the deep learning model, which I then carry out further machine learning analysis on. The second step consists of a machine learning classifier that examines the relationship between the objects detected on each LSOA level and the outcome label.

3.2 Research Ethics

The research carried out in this thesis does not handle any data that could be repatriated to an individual. Furthermore, the imagery data published by Google is anonymised such that any

pedestrians faces or license plates are masked by defocus. There has been some controversy around Google recording imagery data at dwelling places of vulnerable people, such as women's centres. Google users, however, now have the option of flagging potentially invasive pictures that are then removed if found appropriate by the company [29]. Finally, all outcome data used is cited and publicly available.

3.3 Two-Step Modelling

Above all, I explain the first step of the methods, in which objects are detected in the imagery dataset with a Convolutional Neural Network (CNN). This includes describing the concept of transfer learning and how its application is relevant to my research. In this context, I also briefly outline the Common Objects in Context (COCO) dataset. Furthermore, I delineate some of the background of CNNs and my final choice of the architecture of the network. Secondly, I expound the indexing of the objects detected by LSOA and how each panoid is matched to the corresponding LSOA. Finally, I elaborate on the machine learning classifiers, namely Support Vector Classification (SVC) and Random Forest Classifier (RFC), and how these classifiers are fine-tuned and evaluated.

3.3.1 Step one: Deep Learning and Object Detection

The concept of deep learning was inspired by the structure of the human brain, where information is processed in sequence by specific areas, yet seemingly analysed 'at a glance' [30] (p.454). Even though the first of the so-called artificial neural networks was introduced in the 1990s, increasing computational power has only recently unlocked the full potential of the technology. 'Deep' networks, which are networks with more than three (non)linear layers, are usually trained with backpropagation and can be comprised of various architectures [30].

Neural networks can be used as a type of machine learning classifiers, which are algorithms that attempt to predict classes or labels of data points. Typically, they are presented as a set of n training objects $X = x_1, x_2.., x_n$, which can be expressed as a vector with dimension D. For each data point there is a label y_n, with the labels usually expressed as integer, $y_n \in [1, 2, 3, ..., C]$. [31] The classification task can thus be seen as the approximation of a mapping function ($y = f(X)$) from input variables (X) to specific label variables (y).

Neural networks are a high dimensional implementation of machine learning algorithms, with multiple parameters and hyper-parameters to tune.

Figure 3.1: Example of the Mask R-CNN object detection algorithm run on a test image of the streets of London.

Figure 3.1 shows an object detection algorithm trained on the COCO dataset, run on an example image of the streets of London. The algorithm recognises objects such as cars, persons, and traffic light and adds an accuracy score to the prediction.

Object Detection with Transfer Learning

Although most of deep learning algorithms, such as neural networks, are trained to carry out one particular task and are thus only run on a single dataset, their application does not have to be limited. In her book, Torrey [32] defines the idea of transfer learning as the transferal of knowledge from one machine learning task to another. In practice this implies that a neural network is trained on one dataset, but applied to another. In this thesis, I used a pre-trained neural network on an existing imagery dataset that includes labels relevant to the Street View imagery. Using a pre-trained network for this research has multiple advantages. Firstly, it saved a lot of resources, temporal as well as computational. In order to create a highly accurate network, one needs a great amount of labelled ground-truth training data. The labelling itself is time-consuming and requires rigorous manual labelling processes. Openly available labelled datasets thus help to make object detection appropriate for various applications. There are two datasets, the Common Objects in Context (COCO) and the OpenImages dataset, that are in wide use for object detection purposes. The COCO dataset contains 80 labels, which were trained with more instances per category than other common open-source datasets. The labels

in the dataset include but are not limited to objects that are typically found in an outside street environment, such as traffic lights, cars, trucks, bicycles, benches and people [33]. The Open Images dataset, on the other hand, contains a lot more labels, however most of them are irrelevant in an outdoor environment and are therefore not useful for this particular problem [34]. Almost all labels that are included in the COCO dataset are also found in the Open Images dataset. For this reason, I have chosen to use a neural network that has been pre-trained on the 330,000 images of the COCO dataset [33]. Since the dataset is widely used, a lot of research has been published on the performance of different network architectures, trained on this particular dataset. This is a further advantage of employing a transfer learning approach, considering that the choice of network structure can easily be based on systemic evidence.

Choosing a Suitable Convolutional Neural Network

When working with imagery data CNNs are the most commonly chosen type of neural network. CNNs differ from regular artificial neural networks by a few key points. Most importantly, CNNs are locally connected and the neurons are not necessarily attached to all outputs of the following layer [30]. Sharing weights is a way of adding reasonable assumptions to the model and in that way decrease the number of hyper-parameters. A typical CNN design consists of two parts that vary in their functionality. Firstly, the feature extraction part, which includes convolution layers as well as sub-sampling layers. Secondly, the classification part, which is mostly the only fully connected section of the network [30]. There are several design choices that can be made for both parts and thus choosing the suitable network architecture is a crucial step in the modelling process.

Meta-architectures and Feature Extraction Architectures

Today, the most commonly used CNN meta-architectures for object detection, including bounding box and mask detection, are the Fast or Faster Region convolutional neural network (R-CNN) [35, 36], the Mask R-CNN [37], and YOLO algorithm [38], which are all (with the exception of YOLO) available in the Tensorflow detection model zoo for the COCO dataset [39]. The Tensorflow detection model zoo offers several off-the-shelf models, which have been trained on several prevailing datasets that are useful for out-of-the-box inference [39].

Research has shown that the Faster R-CNN, when trained on the COCO dataset, tends to give the highest prediction performance results in form of the mean Average Precision (mAP) [40, 41]. Both papers, however, state that the precision is related to increased memory and

training time. Ren et al. [42] build upon the Fast R-CNN by Girshick [35] with a CNN that proposes regions - a so-called Region Proposal Network (RPN). The Faster R-CNN is a combined network, which means that region proposal and object detection are jointly run by one CNN. Furthermore, the Faster R-CNN is, as one might guess by its name, about ten times faster than the original Fast R-CNN, however without any loss in prediction ability. Since real-time prediction is not required in this thesis, speed is also not much of an issue.

The Tensorflow detection model zoo also uses different CNN feature extractors to train on object datasets, such as the COCO and Open Images [34] publicly available datasets. The ResNet [43] architecture is the most recently published feature extraction architecture that also shows the lowest error rate [44]. When comparing it to other architectural configurations, such as the Visual Geometry Group (VGG) [45] created by researchers in Oxford or Inception (v2 or v3) [46, 47], the ResNet holds a higher mAP value compared to the other two configurations [1]. Furthermore, the Tensorflow detection model zoo also indicates a mAP of 32 with a speed of 106 ms per $600x600$ image for the ResNet architecture trained on the COCO dataset [39]. Figure 3.2 depicts the meta-architectures of the Faster R-CNN algorithm. A feature extractor, such as the ResNet that is used in this thesis, extracts fragments of the image, which are subsequently classified and the bounding boxes are refined.

The mAP is defined by Shanmugamani as 'the product of precision and recall of the detected bounding boxes' and is commonly used as measure of quality to compare classification networks [48] (p.111). It is computed by determining the Average Precision (AP) for each set separately and further averaging over the classes. The cut-off value for a true positive detection is at 0.5.

The mathematical notation of the mAP is defined as

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

where Q is the number or queries in the set and AP(q) is the average precision for a given query q. The AP is calculated by finding the area under the precision-recall curve with

$$AP = \int_{1}^{0} p(r)dr \tag{3.1}$$

where p is the precision and r is the recall. [48]

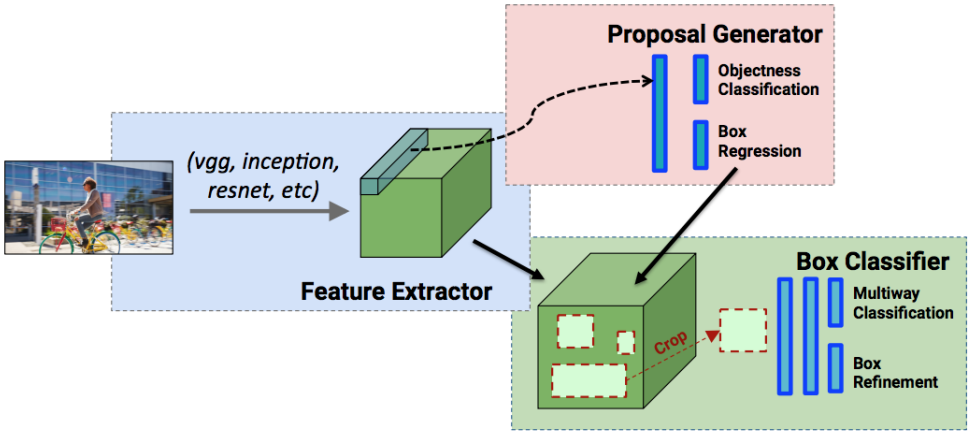

Figure 3.2: High level diagram of the detection meta-architectures of the Faster R-CNN algorithm [1](p.3).

In summary it can be said, therefore, that I have chosen to use a Faster R-CNN network with ResNet architecture that has been pre-trained on the 330,000 images in the COCO dataset [33] for its high mAP score as well as efficiency in prediction.

3.3.2 Data Sampling and Preparation

The information of objects detected at every panoid - image a, b, c and d - is further aggregated and saved as information for each image id point. Since the machine learning analysis is implemented on a LSOA level, all image indices had to be matched to the corresponding LSOA. The LSOA that corresponds to each image id was identified by postcode and later paired with its LSOA. Not every panoid could be matched to a LSOA, which decreased the dataset from 131,465 to 127,975 panoids. Furthermore, the number of panoids that were taken in each LSOA varies between 1 to 211, with a median of 25 panoids per LSOA. For that reason the number of objects detected had to be 'normalised' by dividing all objects detected in one LSOA by the number of panoids that were recorded in the corresponding LSOA. Figure 3.3 depicts the overall distribution of the number of panoids per LSOA. Finally, there is no actual normalisation of the outcome dataset performed, since the classifiers chosen do not require a normal distribution of the data.

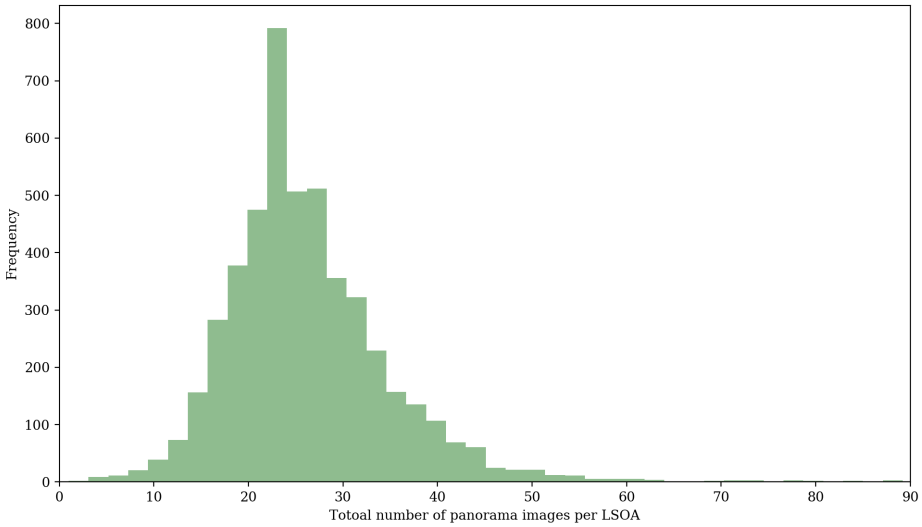

Figure 3.3: Total number of panoids per LSOA.

3.3.3 Step two: Machine Learning Classifiers

In the second part of the methods, I present the relationship between the ten most commonly detected objects per LSOA per panoid and the outcome data described in chapter 2. Subsequently, I examine the relationship between single objects detected, such as cars and people, and the measures of inequality. I use two different machine learning classifiers to detect how well the objects are representative of an outcome measure. Finally, by comparing the prediction performance of the classifier, one can come to know the relationship between the dependent and independent variables.

Ultimately there is no right or wrong choice of machine learning classifier. It is rather a quest of finding the most suitable method for the available dataset. Other factors, such as the type of data, training size, scoring function and number of features or labels, are also of importance [49]. Raschka further points out that the performance metric should be chosen before selecting the actual algorithm [49]. In order to compare the results of the thesis with Suel's predictions, I have chosen to use the same prediction performance measure as in her paper. For these reasons, I have decided on the performance metric for the classifiers to be the Mean Absolute Error (MAE). Additionally, the MAE is a good measure of how much predictions and the 'truth' deviate from one another. The numerical nature of the dataset, as well as the associated integer numerical labels allow the use of most of the classifiers available on scikit-learn, a software machine learning library for Python [50]. In addition, time and computing power were not decisive factors for the choice of the algorithm. If there are not any

concerns about the dataset size, scikit-learn suggests the use of a SVC [51]. The Support Vector Classification (SVC) is described by Singh as very high in accuracy and as a flexible algorithm, because kernel functions can include nonlinear solutions [52]. Furthermore, ensemble methods have been increasingly applied in various fields and are among the methods used in multiple winning 'Kaggle' competitions, indicating great prediction accuracy [53]. Ensemble methods, such as the Random Forest Classifier (RFC), help to improve the predictive performance, however often at the expense of interpretability [54]. For the reasons stated above, I have chosen to utilise both the interpretable and versatile SVC as well as the novel but less interpretable RFC as classifiers.

Each classifier has a set of hyper-parameters that determine the prediction performance. Thus each classifier has to be fine-tuned to reach the optimal setup for the dataset provided. In the following section, I explain the classifiers as well as the corresponding hyper-parameters that they were tested on.

C-Support Vector Classification (SVC)

Support vector machine algorithms try to maximise the margin between two points - the support vectors - to find the best hyperplane that divides the categories. Kernel functions can be included in order to transform the hyperplane into a feature space, and in that way find non-linear solutions for the the optimal hyperplane. [55] In this thesis I shall use the SVC implementation of the scikit-learn Python library [56]. The scikit-learn SVC implementation uses an one-vs-one approach to handle multi-class labelling [56]. This means that each label is classified against all other labels and finally aggregated. The hyper-parameters tested for the support vector classification include a linear and a non linear solution, which are the following:

Radial Basis Function (RBF) Kernel

The RBF function is defined as $\exp(-\gamma\|x - x'\|^2)$ and tested with $\gamma \in [10^{-3}, 10^{-4}]$ and penalty parameters $C \in [1, 10, 100, 1000]$.

Linear Kernel

The linear kernel is defined as $\langle x, x' \rangle$ and evaluated with the penalty parameters $C \in [1, 10, 100, 1000]$.

Random Forest Classifier (RFC)

The second machine learning classifier employed in this thesis is a commonly applied ensemble method. The Random Forest Classifier (RFC) consists of a number of classification tree pre-

dictors that are all trained on a randomly selected subset - with the same distribution for all trees - of the dataset. The votes of each decision tree are then aggregated, in order to decide the final outcome class of the classifier [57]. The hyper-parameters tested for the random forest classification in the thesis are the following:

Number of trees

The number of trees in the forest is defined as n estimators $\in [10, 100, 500, 1000, 2000]$

Number of features

The number of features to consider when looking for the best split max features $\in [1, 2, 5, 10]$

Quality of split

The function to measure the quality of a split is either defined as 'Gini impurity' or 'entropy'.

3.4 Evaluation and Validity

3.4.1 Classifier Evaluation

The dataset is split into 80 per cent training and 20 per cent testing data. A stratified approach was chosen, in order to ensure that all labels were equally present in the evaluation. The model is fine-tuned with a grid-search stratified 5-fold cross-validation method with the negative mean absolute error as score. The scikit-learn implementation 'GridSearchCV' tests every combination of the above defined hyper-parameters in a five-fold stratified shuffle split [2]. This means that each combination of hyper-parameters is tested on five different parts, which in total make up the full training set. Figure 3.4 illustrates the schematic structure of the 5-fold cross-validation, which is carried out on the 80 per cent test set.

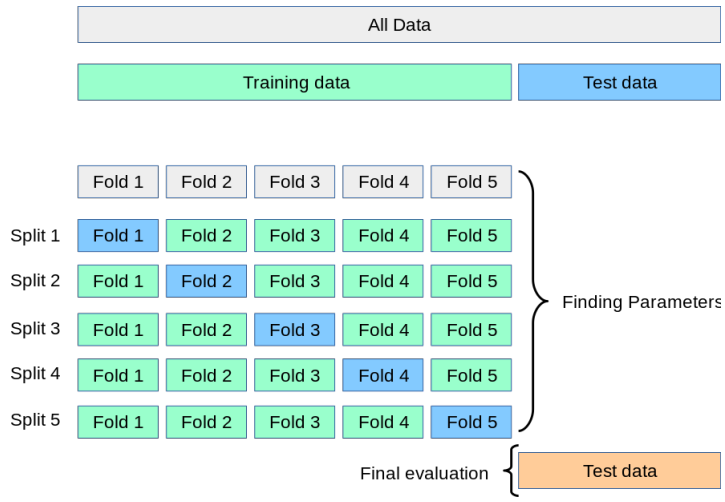

Figure 3.4: Schematic structure of grid search with 5-fold cross-validation algorithm [2].

In addition to the scoring metric, the Mean Absolute Error (MAE), other evaluation metrics were used to assess the performance of the classifiers. Each performance metric is briefly defined below.

Mean Absolute Error (MAE)

The MAE is the average vertical distance between the identity line and each point. Usually it is applied in a regression setting. However, due to the hierarchical nature of the deciles, this measure of accuracy proves to be appropriate. The MAE is defined as

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| \tag{3.2}$$

where $\hat{y}_i$ is the predicted value of sample i, y_i is the true corresponding value and n the number of samples included [58].

Kendall's tau coefficient (τ)

The Kendall's tau coefficient takes the ordering of the results into account. It is a non-parametric metric of association and is based on 'concordances and discordances in paired observations' that ranges from 0 to 1, with 1 indicating a perfect relationship [59]. The tau coefficient is calculated as

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) * (P + Q + U)}} \tag{3.3}$$

where P and Q are the numbers of concordant and discordant pairs respectively. The variable T is defined as the number of ties in x, and U the number of ties in y [60]. When a tie occurs in both x and y it is not recorded in T or U [60].

Cohen's kappa coefficient (κ)

The Cohen's kappa coefficient is a measure of correlation that ranges from -1 to 1 and is mostly used to measure interrater reliability. Interrater reliability assesses to what extent the collected data is representative of the measured variable. [61] The coefficient is calculated as

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3.4}$$

where p_o is the observed agreement ratio, which is the empirical probability of the label being assigned to any sample. And p_e is defined as the expected agreement if labels are assigned randomly by the annotators. p_e is calculated by 'using a per-annotator empirical prior over the class labels' [62].

Pearson's correlation coefficient (r)

The Pearson's correlation coefficient ranges from -1 to 1 and describes the strength of the linear relationship between two variables. A zero value indicates no relationship, -1 and 1 describe a negative or a positive relationship respectively. [63] The correlation coefficient is calculated as

$$r = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(x_i - m_x)^2 \sum_{i=1}^{n}(y_i - m_y)^2}} \tag{3.5}$$

where m_x defined as the mean of the vector x and m_y is the mean of the vector y [64].

3.5 Software, Hardware, Data and Code Management

3.5.1 Software and Hardware

The model was run on a HP machine with an Ubuntu 18.04 operating system and a GeForce RTX 2080 Ti graphics card by NVIDIA. The object detection algorithm is run within a docker environment. A docker container allows the bundling of software, libraries and configuration files. The containers require less computing power than virtual machines, because all containers can be run by a single kernel of an operating-system [65].

3.5.2 Data and Code Management

The imagery data, outcome label data and indexing file were retrieved from the Imperial College shared Research Data Store (RDS) [66]. The Python code for the data pre-processing, classification as well as the visualisations is stored within a GitHub repository and can be retrieved

at `github.com/baerbelblume/urban-inequalities`. Further Python code for the running of the object detection algorithm with the Tensorflow API was adapted from a GitHub repository by Muller and Nathvani [67].

Chapter 4

Results

I present my findings of both modelling steps in the results chapter of this thesis. First, I reveal summary statistics of the object detection algorithm. Furthermore, I elaborate on the most frequently detected objects and display selected maps that depict the results. Secondly, I expound the results of the machine learning classifiers and their fine-tuning. I have used the SVC and RFC classifiers to determine the relationship between the ten most commonly detected objects and the different outcome deciles, namely mean income, living environment and health deprivation decile. Furthermore, I examine the prediction with simply one variable, such as cars detected, to find out how much importance each variable carries by itself.

4.1 Step one: Object Detection

The object detection algorithm could identify a total number of 2,943,194 objects in the 525,860 images. The most frequently detected objects in the imagery dataset of London were, in decreasing order, cars, followed by persons, trucks, potted plants, benches, busses, motorcycles, bicycles and chairs. The exact object counts can be found in the appendix 1.

Top 10 objects detected

Figure 4.1 shows the total number of the top 15 most commonly detected objects in the Greater London area. The ten most frequent objects detected make up about 97.3 per cent of total number of objects detected (2,864,242). A large proportion of the labels included in the COCO dataset are not relevant in an outdoor urban environment and thus only detected in very low numbers. For this reason I have chosen to solely utilise the ten most commonly detected objects as dependent variables for the machine learning classifier.

Figure 4.2 depicts the density of the top 10 objects detected per panoid per LSOA on a map of Greater London. There is an average of about 20 objects detected per image id per LSOA. The density of objects detected was quite balanced, with a higher amount of objects observed in the more central parts of the city. The lowest number of objects detected per panoid per LSOA was found at the boarder around Greater London, with around five to 15 objects identified.

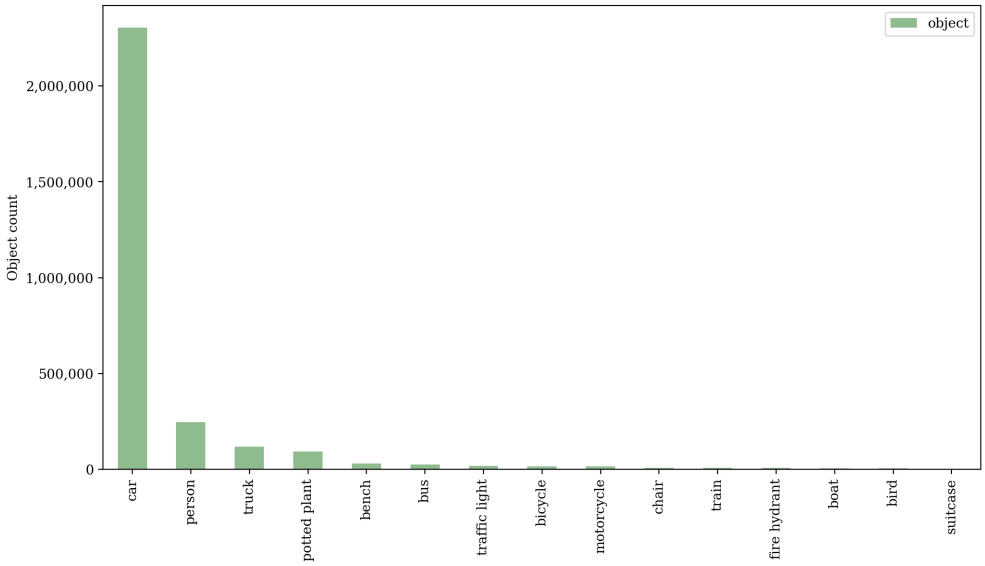

Figure 4.1: Total number of top 15 objects detected in all LSOAs.

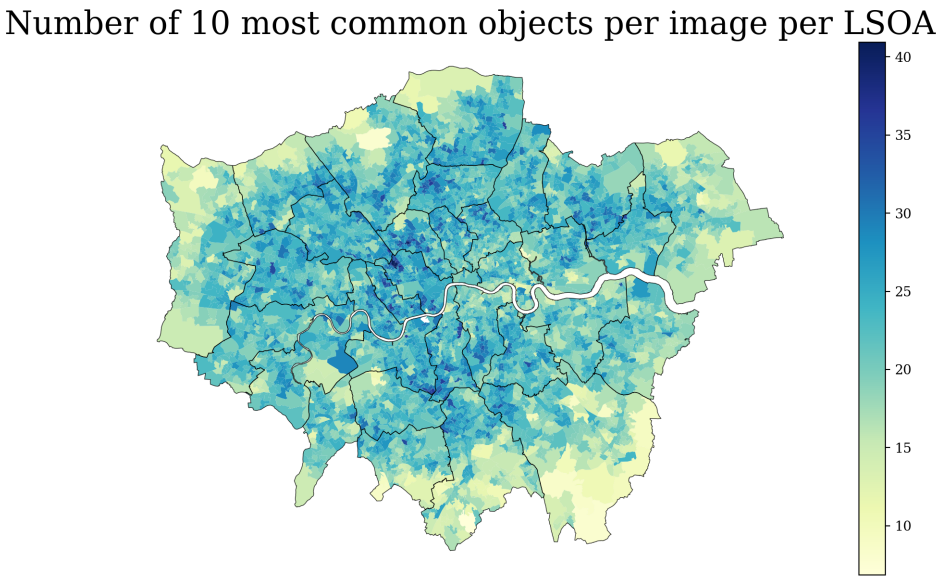

Figure 4.2: Sum of top 10 objects detected per panoid per LSOA.

Car

The car was by far the most common object detected with an average number of 18 cars detected per panoid per LSOA. Figure 4.3 depicts the number of cars detected per panoid per LSOA on a map of London. Contrary to one's first intuition, the City of London had a rather low car per panoid per LSOA count. This might be due to the fact that there are usually no available parking spaces on sides of the road in the inner city. This means that there are generally only cars detected in front or behind the Google Street View vehicle. Yet, in the outskirts of the city there are often cars parked on the sides of the street. One could argue, however, that in the suburban areas there are probably not as many cars in front or behind the Google Street View vehicle. Additionally, the 'congestion charge zone' appointed by Transport for London that has become the Ultra Low Emission Zone (ULEZ) in April 2019 covers exactly those parts of the inner city that have low car counts. This could be taken as an indicator for an effective policy to reduce the number of cars in urban environments.

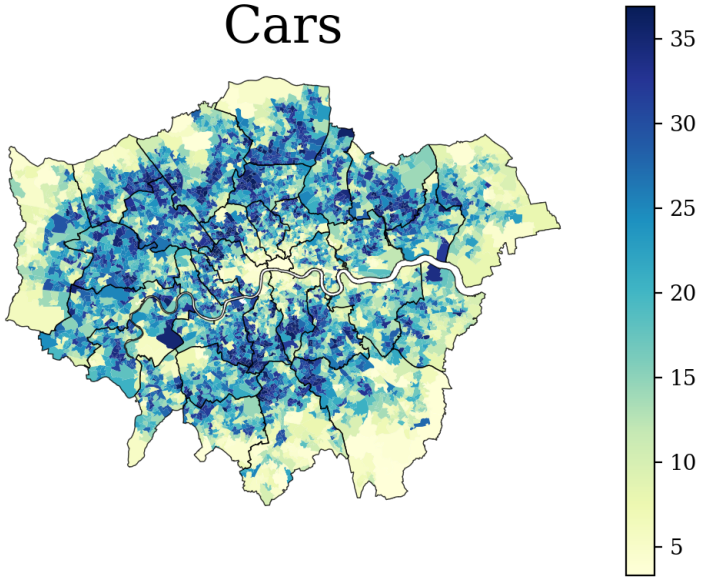

Figure 4.3: Number of cars detected per panoid per LSOA.

People

The second most frequently detected object was a person, with an average number of two persons detected per panoid per LSOA. The map of the number of people detected per image id per LSOA, figure 4.4, shows quite an opposite pattern to the previous map depicting the

frequency of cars per panoid per LSOA. The LSOAs with the highest number of people are all located in the centre of the city with up to 18 people per image id per LSOA. The lowest number of people detected was around the outer boarder of London, which touches at most parts the M25 motorway.

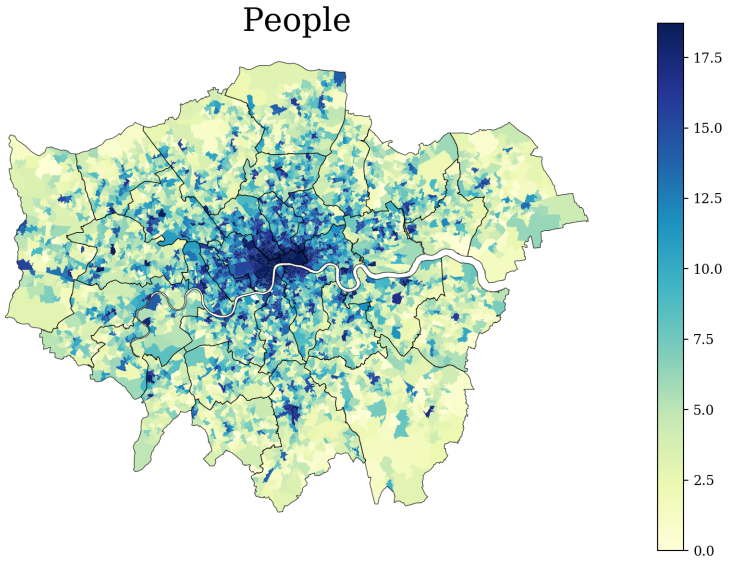

Figure 4.4: Number of people detected per panoid per LSOA.

Potted Plants, Bicycles and Other Objects Detected

Figure 4.5 depicts the other most frequently detected objects in the London dataset. On average, there were about 0.9 trucks and 0.7 potted plants detected per panoid per LSOA. It seems striking that the trucks detected do not show the same pattern as the cars detected in figure 4.3. The highest number of trucks was detected outside the city centre. The COCO dataset does not include a label for plants in general, so any plant detected are classified as 'potted plant'. Meanwhile, most of the potted plants observed are found near parks, for example Hyde Park in the centre of London. This reveals that possibly trees or bushes were also detected as potted plants. The recognised benches and busses seem to be evenly spread across the whole city, however, the number of traffic lights were more speckled across the map. Moreover, the number of bicycles and motorcycles found per panoid per LSOA showed a clear spatial trend towards the city centre. This was quite surprising, because cycling in the city is dangerous, with an overall decrease from 22 to 11 per cent of weekly cyclists in the borough of City of London [68]. Yet, this number also includes bikes on the side of the road, and - depending

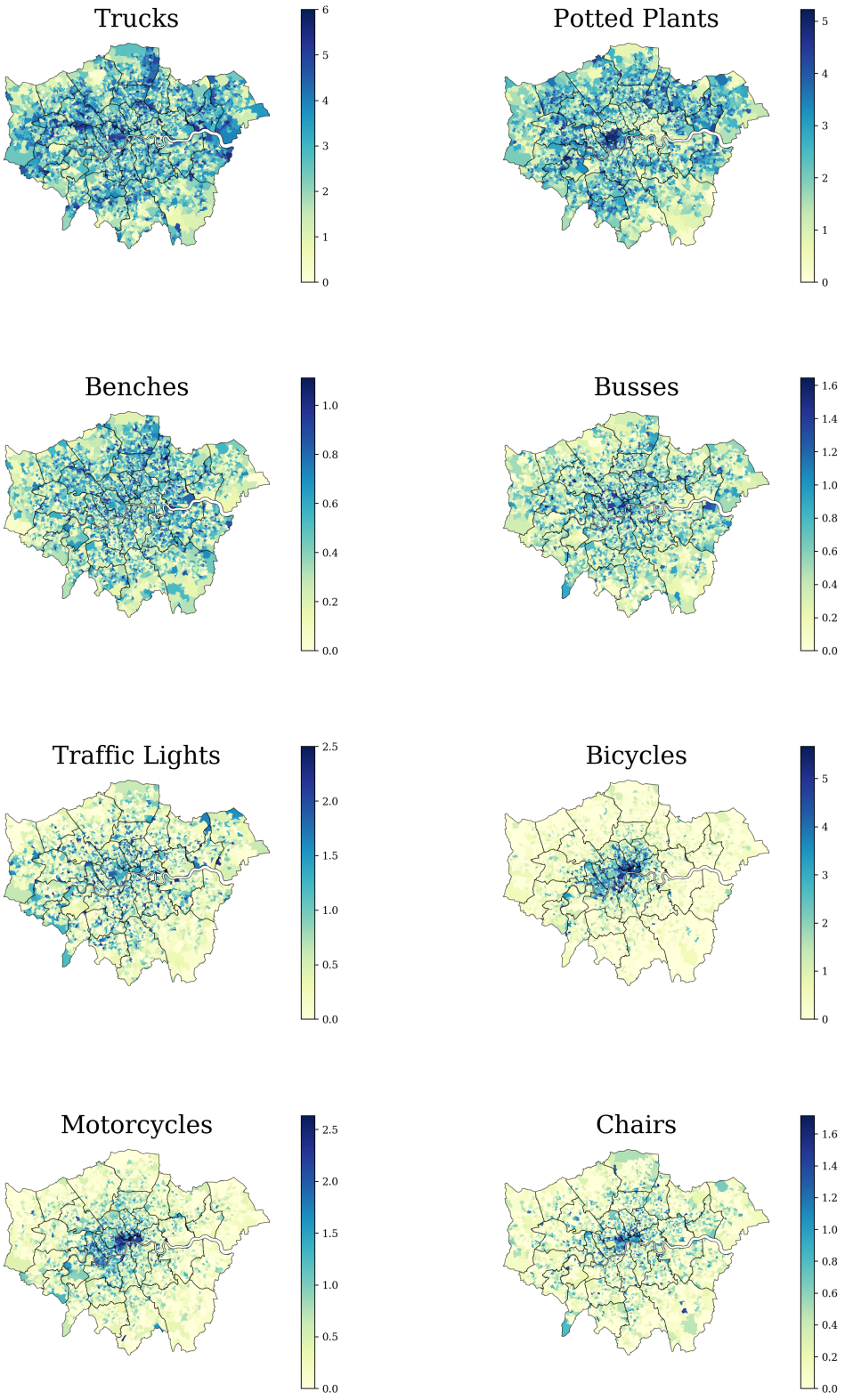

Figure 4.5: Number of specific objects detected per panoid per LSOA.

on the time of the day - the cycles detected will mostly include the bikes of commuters into the city centre. Lastly, it seemed quite astonishing that the chair was among the top ten most frequently detected objects. When inspecting the map, however, one can identify a high number of chairs in the city centre and around parks. This number most likely represents lively areas with cafes and restaurants that encompass outside sitting areas.

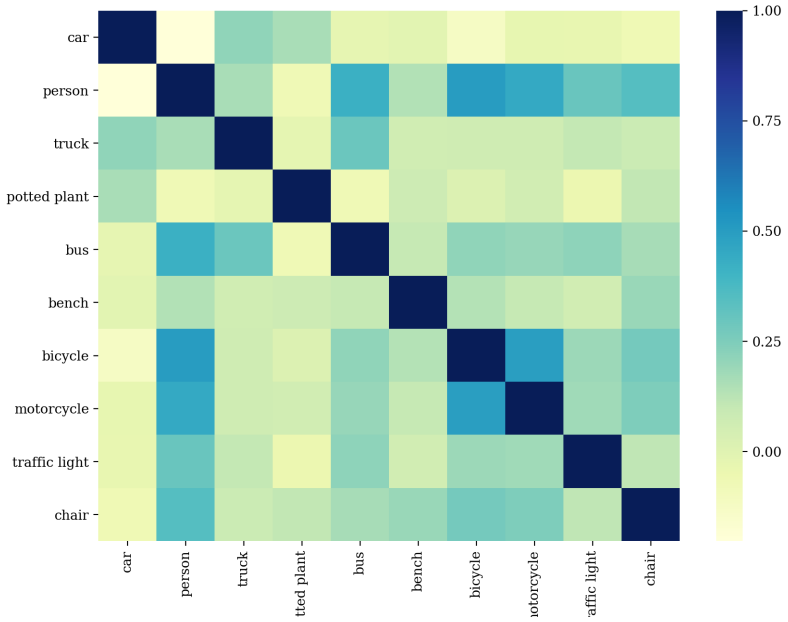

Figure 4.6: Correlation matrix: pairwise correlation of the 10 object classes.

Figure 4.6 depicts the Pearson correlation between the ten different objects detected. The matrix indicates a high correlation between the bicycles and motorcycles variables. Other weaker correlations can be found between persons and trucks or motorcycles and traffic lights.

4.1.1 Inequalities in Measures of Income and Wellbeing in London

The objects detected and described in the first part of the results chapter are further analysed with three different inequality measures.

In summary, these observed wellbeing outcomes, as depicted in the data review chapter (Figures 2.5, 2.6 and 2.7) revealed that London's most affluent areas are the boroughs of City of London, Chelsea, Kensington and Westminster. The least well-off people live in the east, southeast and northeast, as well as the outskirts in the west of the city. The east parts of London are increasingly gentrified and show hubs of well-off areas. Similar to other in other megacities, overcrowding is of concern. Areas where space is limited, such as the city centre, are particularly vulnerable to this development [25]. Living environment, which is measured

by rating housing quality, air pollution and road safety, is worse in the city centre, also in the wealthy areas. None of the outcomes in this thesis, namely mean income, health and living environment deprivation, show a strong correlation between each other.

4.2 Step two: Machine Learning Classifier

The second part of the analysis expounds the results of the machine learning classifiers, namely the SVC and the RFC. The classifiers have been trained with the ten most common objects and the MAE, Kenall's tau rank, Pearson's correlation coefficient and Cohen's kappa coefficient of the different outcomes, which I have described above, are reported. Furthermore, the relationship between the most common objects - cars and people - are examined one by one. In that way one can find out whether those single variables could act as strong predictors by themselves.

4.2.1 10 Most Frequent Objects Detected

The MAE for both classifiers, when run on the ten most common objects to predict the mean income, living environment deprivation and the health deprivation decile showed similar results, with a MAE ranging from 2.3 to 2.5. The living environment deprivation decile was most accurately predicted with a MAE of about 1.8. One explanation for this high prediction performance was revealed when looking only at the people as independent variable, as explained in section 4.2.2 below.

Figures 4.7 and 4.8 display the heatmap of the correlation matrices of each inequality outcome. The figures reveal that the highest and lowest decile were predicted with the highest accuracy. This pattern was mirrored in each prediction for the different inequality measures and is evident in both classifiers.

Additionally, the results of the fine-tuning of each classifier for every outcome can be found in the appendix 2 and 3.

SVC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	2.307135	0.375403	0.472569	0.137082
Health Deprivation Decile	2.499483	0.347516	0.450387	0.088522
Living Environment Deprivation Decile	1.838676	0.525123	0.662777	0.148247

Table 4.1: Results of SVC run on 10 most commonly detected objects and the inequality measures.

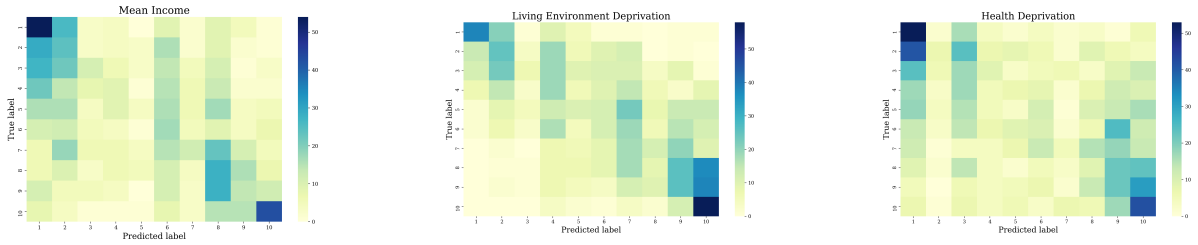

Figure 4.7: Correlation matrix heatmap of the SVC for each inequality outcome.

RFC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	2.36091	0.345481	0.447642	0.115639
Health Deprivation Decile	2.471562	0.322984	0.424278	0.076063
Living Environment Deprivation Decile	1.778697	0.523298	0.663637	0.156371

Table 4.2: Results of RFC run on 10 most commonly detected objects and the inequality measures.

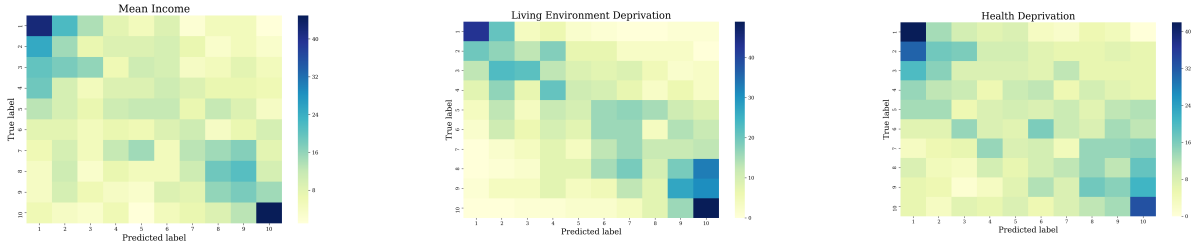

Figure 4.8: Correlation matrix heatmap of the RFC for each inequality outcome.

4.2.2 Individual Objects

Cars

The number of cars per panoid per LSOA predicted all the different outcomes almost equally well, with a MAE of 3.1 to 3.4 (average of 3.26). The RFC delivered a little better results, however, cars in general did not seem to be a good heuristic to explain any of the three outcomes. While the Kendall's tau coefficient and the Pearson's correlation coefficient both indicated almost no correlation, the prediction performance was still higher than random.

SVC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	3.436401241	0.095077326	0.121200591	0.027176795
Health Deprivation Decile	3.2885212	0.087771534	0.11011377	0.038206281
Living Environment Deprivation Decile	3.153050672	0.095916608	0.097594685	0.036753146

Table 4.3: Results of SVM run on number of cars detected and the inequality measures

RFC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	3.244053775	0.038630424	0.052542181	-0.002546176
Health Deprivation Decile	3.305067218	0.002669834	-0.003496618	-0.007069181
Living Environment Deprivation Decile	3.111685626	0.036561188	0.050809764	-0.002719585

Table 4.4: Results of RFC run on number of cars detected and the inequality measures

People

Both classifiers showed that people are a slightly better predictor of the outcomes than cars. Especially, the living environment deprivation with a MAE of about 2.1 or 2.4 could be predicted quite well by this independent variable. This is not too surprising when we compare the density map of people detected in the city (Figure 4.4) to the living environment deprivation map in chapter 2 (Figure 2.7). Both maps show similar patterns, with the highest density of people per panoid per LSOA and the worst-off living environment deprivation deciles in the centre of the city. For this reason I have plotted the number of people and the living environment deprivation decile against each other in ten boxplots, as shown in figure 4.9. The plot corroborates the assumed relationship by depicting a weak negative linear correlation between the two variables. Both, Kendall's tau coefficient and the Pearson's correlation coefficient of

the SVC, confirmed this, with the former indicating a correlation of almost 0.5 and the latter of 0.6. The other outcomes, mean income and health deprivation, showed practically no or no significant correlations with the number of people detected per panoid per LSOA.

SVC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	2.915201655	0.052428292	0.066593904	0.022302977
Health Deprivation Decile	3.171664943	0.212042341	0.253544664	0.062363608
Living Environment Deprivation Decile	2.187176836	0.480123503	0.601165	0.124121986

Table 4.5: Results of SVM run on number of people detected and the inequality measures.

RFC	**MAE**	τ	**Pearson's r**	κ
Mean Income Decile	3.071354705	0.069144023	0.092535494	0.0179791
Health Deprivation Decile	3.012409514	0.096523986	0.129693994	0.015819914
Living Environment Deprivation Decile	2.420889349	0.300594258	0.408011732	0.041203167

Table 4.6: Results of RFC run on number of people detected and the inequality measures.

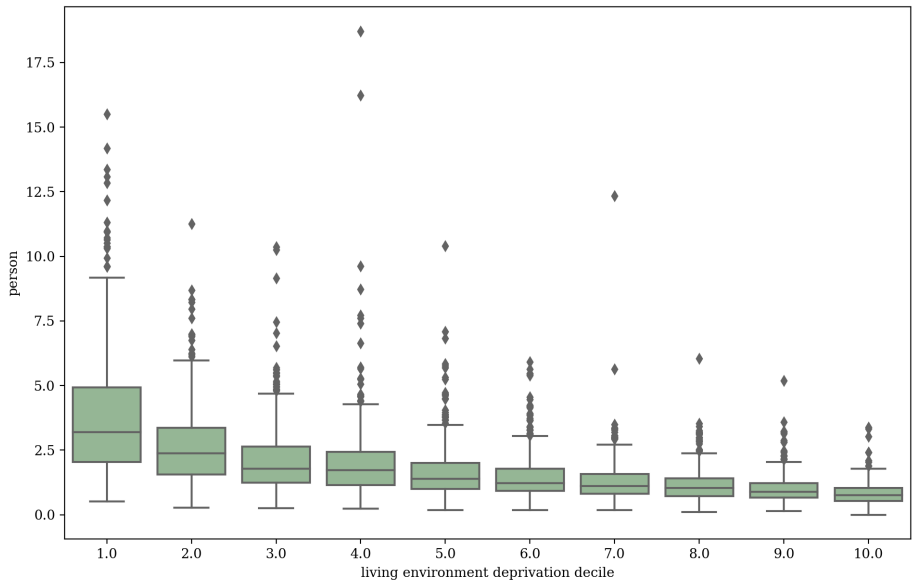

Figure 4.9: Negative linear trend: Boxplots of the living environment deprivation decile and the number of people detected.

In summary, both classifiers, the SVC and the RFC, predicted the outcomes almost equally well. The ten most frequently detected objects predicted the outcomes with an average MAE of 2.2. The best prediction performance was obtained for the living environment decile, which consists of observed data on housing quality, air pollution and road safety.

Chapter 5

Discussion

In this chapter of my thesis, I first discuss the results on their own, and subsequently explore them in the context of Suel's research paper [7]. Finally, I expound some of the main challenges faced when conducting the research and how these problems were tackled.

When predicting multiple social and health outcomes in a two-step approach using imagery data and machine learning classifiers, some inequalities are better predicted than others. Living environment deprivation was most accurately detected, which suggests that the pre-defined objects within the images mirror the pollution (sources), quality of housing and road safety to some extent. Inspecting the variables, I found that the number of people detected indicates the strongest correlation to the living environment inequality measure. The prediction performance for the mean income decile was not as high as the living environment prediction with an average 2.3 as MAE score for the ten most frequent objects as predictor. This seemed surprising, since one might assume that objects, such as cars and potted plants, are an exemplar of wealth and exhibit a stronger visual correlation. Moreover, the prediction of the health deprivation revealed the weakest allocation performance. This could imply that the objects detected are weaker proxies for mortality, morbidity and hospital admissions rates. Using single variables as predictors, cars and people, also showed varied results. While people seemed to be a strong predictor for living environment in particular, cars did not show any strong correlation between either of the outcome labels. In addition, both classifiers could detect the worst-off and best-off decile of any outcome with the highest accuracy, as shown in figures 4.7 and 4.8. This might be due to the fact that the deciles are an arbitrary way of labelling the data. Other ways of categorising the outcome labels would possibly improve the predictions substantially. This alternate labelling could be composed of dividing the outcome labels simply into 'best-off' and 'worst-off'.

Contrary to my findings, the prediction performance of Suel's classifier did get similar results for mean income as well as living environment deprivation. This suggests that the labels chosen from the COCO dataset and thus the objects detected, are more suitable to predict living environment than the mean income decile. Additionally, it could suggest that the mean income is better predicted by visual changes in the environment that are not actually defined objects, such as a specific condition of an object or housing facades. In computer vision, especially in object detection, the output can mostly be categorised into either 'things' and 'stuff' [69]. The difference is described by Forsyth et al. [70] as

> The distinction between materials — "stuff" — and objects — "things" — is particularly important. A material is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape. An object has a specific size and shape.

This prompts a rather fundamental question of whether the 'stuff' detected in an environment, such as facades and greenery, represents and mirrors social and health inequalities more successfully than the 'things' that I have used in this thesis. These questions could be further explored by analysing the Class Activation Map (CAM)s of a neural network that has been fully trained on the same imagery dataset, which I will explain in the final chapter.

While the prediction of inequality measures admits of improvement, it is a fact that inequalities in cities persist and are increasing. Imagery data holds a lot of valuable information when collected in a reliable and meticulous manner. Although London has realised the problem of growing social instability and security through increasing social and economic inequalities [71, 72, 73], yet insufficient focus has been placed on creating sustainable solutions. As imagery data can ultimately feed into modelling policy scenarios that tackle growing inequalities, it is important that this resource is not solely collected and used by the private sector, such as companies like Google. Cities have the possibility to gather their own data that could predict social and health inequalities. Nevertheless, providing the evidence for social and health inequalities is only the first step on a long way to a healthier, more equitable city life.

Challenges faced

Lastly, there were several challenges that I had to overcome in order to lay out the research as planned. First of all, the sheer size of the imagery dataset (64.5GB tar file) had proven to be challenging in itself. The download from the RDS failed several times due to expiration of the connection. In addition, the object detection algorithm also skipped some images, which were

retraced and added to the findings. Furthermore, the different indexing of the data sources also brought some issues. The metadata file, which mapped the image id to the appropriate LSOA, was not as comprehensive as expected. As a result, the number of objects detected had to be 'normalised' by LSOA. Furthermore, some philosophical questions arose when selecting the evaluation method. While it is feasible for Suel [7] to test her network on all LSOAs, it is not so much so for the classifiers in this thesis. This is due to the fact that I also fine-tuned the machine learning classifiers with a five-fold cross-validation procedure. Thus testing on all LSOAs would imply a fine-tuning step for each test fold, which again overfits the classifier.

Chapter 6

Conclusion

In the final chapter of this thesis, I depict the strengths as well as the limitations of the model and research design. I further make suggestions on how this thesis could be extended and ascertain whether the research question was answered sufficiently.

6.1 Strengths and Limitations

Several advantages to the selected research methods could be identified. Firstly, the imagery as well as outcome label dataset was openly available and easily reproducible. By using the same imagery data for all outcomes, the prediction performance could also be readily compared. Secondly, the models required less computational power, since an 'off-the-shelf' fully trained network was used for object detection. It further showed that transfer learning is an efficient and more interpretable way to extract information from imagery data and create a model that predicts inequality measures. Moreover, the classifiers chosen to predict the inequality deciles were robust, since they have proven to deliver similar prediction results. Finally, the model is more intuitive and interpretable than a end-to-end solution, which is sometimes considered a black box. This research thus helped to envision and potentially initiates further research in an area, which has not been explored sufficiently.

Due to its constrained scope, this thesis, however, also exhibits some limitations. Firstly, since working with 'off-the-shelf' CNNs for object detection, some features were not ideally tuned for their application. There are multiple ways of finding the 'most suitable' network architectures, and the mAP, which I have used as a determining factor is only one of them. Secondly, the choice of classifier and tuning hyper-parameters was also very determining of the outcome. For this reason I have chosen two different classifiers, SVC and RFC, both classifiers that have proven to deliver stable results. Further, I have limited this research to a choice

of three inequality measures, namely mean income, health deprivation and living environment deprivation. Selecting a different set of outcome data would have changed the relationship found between the objects detected and the inequalities profoundly. Furthermore, calculating deciles from the outcome data introduced a further simplification to the model. The conversion to deciles, however, was not only due to simplification but also to better compare the results to Suel's [7] research. Moreover, the ordinal nature of the outcome data was not taken into account during the classification process. In addition, some pictures with objects detected were not used in the classifier, because of the sampling of the images by image id. The metadata file did not include the LSOA for every image id, and thus some objects were left out. Moreover, there was a difference in recording year of the outcomes and the images due to the nature of how the data was collected. Although the overall street environment is not assumed to change that quickly, it is possible that wrong associations were made between the two. Additionally, the imagery data itself could show some disparities that could change the number of objects detected. Those disparities include different conditions when the image was taken, such as variation in season, weather, time of recording, as well as general obstructions or construction work. Finally, the test design of the classifier could be criticised. The final test set that was withheld until the end represents only 20 per cent of the LSOAs. One could argue that the reported prediction metrics are thus not representative of the whole dataset.

6.2 Recommendations for Further Study

Due to the limited scope of this thesis and the multiple design-choices that have been made along the way, there are several recommendations for further study. First, I describe how further data of the objects detected could have been included in the classifier. In order to examine if the 'meet-in-the-middle' approach of using objects as model heuristics is successful, one could further investigate the Class Activation Map (CAM)s of the neural network trained on the London image dataset. Furthermore, the transferability of the model could be tested with image data from other locations.

Additional Information

Additional information on object size and confidence of object detection could have been included to the classification. One could argue that size of an object might indicate the importance of it in an image. As for example, when a parked car is detected in the background of the image, this could indicate that it has already been detected at a previous instance. Including the size

of the object detected would thus ensure that closer objects are weighted more and in that way carry more decisive semantic information at that specific location. The same can also be said for the confidence of the detection. The object detection algorithm outputs confidence scores for occurrence of an object. This information could also be used to improve the classifier by weighting objects with lower confidence less and therefore also placing importance on the more confident detections.

One major downfall of the COCO and Open Images dataset is that they do not include greenery, such as trees or bushes, as objects that can be detected. The COCO dataset contains potted plants as a label, however, there is no other label that represents objects in green spaces, such as trees. There is substantial evidence that urban planting impacts the quality of life [15, 18]. Potentially rich information is disregarded by missing out on identifying parks and trees by the side of the road. In this regard making use of a more comprehensive label dataset would possibly improve the results.

Class Activation Maps

Analysing the Class Activation Map (CAM)s of neural networks are a way of finding the implicit attention of the neural network on an image [74]. By examining the maps from the networks that were trained by Suel [7], one could identify whether the labels, which have been included in the object detection, are representative and valuable heuristics. Furthermore, the CAMs could act as a way to define new labels, which then could improve the machine learning classifiers.

Transferability

Moreover, the transferability of the model to other cities could be investigated. Suel [7] successfully runs her trained network on imagery data of the West Midlands, Greater Manchester and West Yorkshire. This suggests that there is some structural overlap that could also be detected by the 'meet-in-the-middle' approach of this thesis. A further step could therefore be to test two-step approach with the imagery data of other cities in the UK. In that way one could study the transferability of the modelling and compare the predictions with Suel's research.

6.3 Conclusion

To conclude, this thesis aimed to study the potential of a 'meet-in-the-middle' approach with imagery data by using deep learning object detection and machine learning classifiers to predict inequality outcomes. Increasing computing power has only recently unlocked the potential of applying AI to understand complex structures and dynamics through imagery data. There is some research in the field of AI predicting non-visual outcomes, such as socioeconomic factors, based on visual changes in the environment. Even though an end-to-end solution, such as a fully trained network, delivers more accurate predictions, the two-step approach helped to understand how and what kind of objects matter more than others in inequality predictions. For instance, results from this thesis imply that there is a linear relationship between the number of people and the living environment deprivation decile in London. Further, mean income predictions achieved higher allocation performance as the living environment deprivation. Comparing the results to Suel's research [7], it becomes apparent that some objects are more representative of the inequality measures than others, depending on object as well as outcome label. The two-step approach helps to simplify and make the model more interpretable, allowing faster modelling, however at the price of prediction performance. Yet, there are several ways how the classifier could be improved. Further study could include the size and confidence of detection to the classifier. Additionally, the CAMs of Suel's [7] network could be analysed to gather information on the implicit attention of the images and whether they overlap with the objects detected. The results of this thesis help to understand the use of a 'meet-in-the-middle' approach in a global health context. They further reveal the importance of research in urban inequalities. Finally, the research question was sufficiently answered by highlighting the potential that AI holds for understanding urban environments with imagery data. It further explains to what extent heuristics can predict inequality outcomes and delineates the benefits of this approach.

Appendix

Object	**Total Count**
car	2,304,015
person	246,451
truck	118,416
potted plant	92,924
bench	29,930
bus	25,056
traffic light	17,136
bicycle	15,978
motorcycle	14,336
chair	8,455
train	7,988
fire hydrant	6,946
boat	4,802
bird	4,279
suitcase	4,175
parking meter	3,570
backpack	3,335
kite	3,319
stop sign	3,291
clock	2,922

Table 1: Total counts of objects detected

Outcome	**Hyperparameters**
Mean income	C: 1000, gamma: 0.001, kernel: rbf
Health deprivation	C: 1000, gamma: 0.001, kernel: rbf
Living environment deprivation	C: 1000, kernel: linear

Table 2: Hyperparameters for SVC

Outcome	**Hyperparameters**
Mean income	criterion: gini, max features: 5, n estimators: 1000
Health deprivation	criterion: gini, max features: 10, n estimators: 500
Living environment deprivation	criterion: gini, max features: 2, n estimators: 1000

Table 3: Hyperparameters for RFC

Bibliography

[1] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[2] scikit learn. Scikit-learn documentation: 3.1. cross-validation: Evaluating estimator performance. `https://scikit-learn.org/stable/modules/cross_validation.html`, 2019. Online; accessed 12-July-2019.

[3] United Nations. World Urbanization Prospects: The 2018 revision. https://esa.un.org/unpd/wup/Publications/Files/WUP2018-KeyFacts.pdf, 2018.

[4] Mark Anderson and Achilleas Galatsidas. Urban population boom poses massive challenges for Africa and Asia. *the Guardian*, 2014.

[5] Majid Ezzati, Christopher J Webster, Yvonne G Doyle, Sabina Rashid, George Owusu, and Gabriel M Leung. Cities for global health. *Bmj*, 363:k3794, 2018.

[6] Jessica E Steele, Pål Roe Sundsøy, Carla Pezzulo, Victor A Alegana, Tomas J Bird, Joshua Blumenstock, Johannes Bjelland, Kenth Engø-Monsen, Yves-Alexandre de Montjoye, Asif M Iqbal, et al. Mapping poverty using mobile phone and satellite data. *Journal of The Royal Society Interface*, 14(127):20160690, 2017.

[7] Esra Suel, John Polak, James Bennett, and Majid Ezzati. Measuring social, environmental and health inequalities using deep learning and street imagery. *Scientific Reports*, 9, 12 2019.

[8] Charles Booth. *Life and Labour of the People in London: The city of London and the West End*, volume 3. Macmillan, 1902.

[9] Andrew G. Rundle, Michael D.M. Bader, Catherine A. Richards, Kathryn M. Neckerman, and Julien O. Teitler. Using Google Street View to audit neighborhood environments. *American Journal of Preventive Medicine*, 40(1):94 – 100, 2011.

[10] Philip Salesses, Katja Schechtner, and César A Hidalgo. The collaborative image of the city: mapping the inequality of urban perception. *PloS one*, 8(7):e68400, 2013.

[11] James Q. Wilson George L. Kelling. Broken Windows. `https://www.theatlantic.com/magazine/archive/1982/03/broken-windows/304465/`, 1982.

[12] Daniel Tumminelli O'Brien and Robert J Sampson. Public and private spheres of neighborhood disorder: Assessing pathways to violence using large-scale digital records. *Journal of research in crime and delinquency*, 52(4):486–510, 2015.

[13] Robert J. Sampson and Stephen W. Raudenbush. Seeing disorder: Neighborhood stigma and the social construction of "broken windows". *Social Psychology Quarterly*, 67(4):319–342, 2004.

[14] Scott Weichenthal, Marianne Hatzopoulou, and Michael Brauer. A picture tells a thousand... exposures: opportunities and challenges of deep learning image analyses in exposure science and environmental epidemiology. *Environment international*, 122:3–10, 2019.

[15] Marco Helbich, Yao Yao, Ye Liu, Jinbao Zhang, Penghua Liu, and Ruoyu Wang. Using deep learning to examine street view green and blue spaces and their associations with geriatric depression in Beijing, China. *Environment international*, 126:107–117, 2019.

[16] Daniel R Richards and Peter J Edwards. Quantifying street tree regulating ecosystem services using Google Street View. *Ecological indicators*, 77:31–40, 2017.

[17] Xiaojiang Li, Carlo Ratti, and Ian Seiferling. Mapping urban landscapes along streets using Google Street View. In *International cartographic conference*, pages 341–356. Springer, 2017.

[18] Ian Seiferling, Nikhil Naik, Carlo Ratti, and Raphäel Proulx. Green streets - Quantifying and mapping urban trees with street-level imagery and computer vision. *Landscape and Urban Planning*, 165:93–101, 2017.

[19] Nikhil Naik, Scott Duke Kominers, Ramesh Raskar, Edward L. Glaeser, and César A. Hidalgo. Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences*, 114(29):7571–7576, 2017.

[20] Sean M Arietta, Alexei A Efros, Ravi Ramamoorthi, and Maneesh Agrawala. City forensics: Using visual elements to predict non-visual city attributes. *IEEE transactions on visualization and computer graphics*, 20(12):2624–2633, 2014.

[21] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *Proceedings of the National Academy of Sciences*, 114(50):13108–13113, 2017.

[22] Google. Street view static api. `https://developers.google.com/maps/documentation/streetview/intro`, 2019. Online; accessed 6-June-2019.

[23] Office for National Statistics. Ons postcode directory (august 2017). `https://ons.maps.arcgis.com/home/item.html?id=1e4a246b91c34178a55aab047413f29b`, 2019. Online; accessed 6-June-2019.

[24] Greater London Authority. Household income estimates for small areas, 2015.

[25] Communities Local Government Ministry of Housing. English indices of deprivation 2015, 2015.

[26] NHS. Nhs business definitions: Lower layer super output area. `https://www.datadictionary.nhs.uk/data_dictionary/nhs_business_definitions/l/lower_layer_super_output_area_de.asp?shownav=1`, 2019. Online; accessed 12-June-2019.

[27] National Statistics. English indices of deprivation 2015. `https://www.gov.uk/government/statistics/english-indices-of-deprivation-2015`, September 2015.

[28] Marc Chadeau-Hyam, Toby J. Athersuch, Hector C. Keun, Maria De Iorio, Timothy M.D. Ebbels, Mazda Jenab, Carlotta Sacerdote, Stephen J Bruce, Elaine Holmes, and Paolo Vineis. Meeting-in-the-middle using metabolic profiling – a strategy for the identification of intermediate biomarkers in cohort studies. *Biomarkers*, 16(1):83–88, 2011. PMID: 21114379.

[29] Google. Google-contributed street view imagery policy. `https://www.google.com/intl/en/streetview/policy/`, 2019. Online; accessed 12-July-2019.

[30] E. R. Davies. *Computer Vision: Principles, Algorithms, Applications, Learning*. Academic Press, 5 edition, 2018.

[31] Simon Rogers and Mark Girolami. *A First Course in Machine Learning.* Chapman & Hall/CRC, 2nd edition, 2016.

[32] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global, 2010.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[34] Google. `https://storage.googleapis.com/openimages/web/index.html`, May 2019. Online; accessed 11-July-2019.

[35] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[36] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[37] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[38] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[39] Tensorflow. Tensorflow detection model zoo. `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md`, 2019.

[40] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 2019.

[41] Nikhil Yadav and Utkarsh Binay. Comparative study of object detection algorithms. *International Research Journal of Engineering and Technology (IRJET)*, 4(11), 2017.

[42] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[44] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *arXiv preprint arXiv:1901.06032*, 2019.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[48] Rajalingappaa Shanmugamani. *Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[49] Sebastian Raschka and Vahid Mirjalili. *Python machine learning*. Packt Publishing Ltd, 2017.

[50] Gramfort Thirion Grisel Blondel Prettenhofer Weiss R. Pedregosa, Varoquaux, Passos Cournapeau D. Dubourg, Vanderplas, M. Brucher, and Duchesnay Perrot. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[51] scikit learn. Choosing the right estimator. `https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html`. Online; accessed 10-July-2019.

[52] Narina Thakur Amanpreet Singh and Aakanksha Sharma. A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315, March 2016.

[53] William Vorhies. Want to win competitions? pay attention to your ensembles. `https://www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles`, May 2016. Online; accessed 10-July-2019.

[54] Seth Flaxman. January 2019.

[55] Jason Bell. Support vector machines. In *Machine Learning*, pages 139–160. John Wiley Sons, Inc, Indianapolis, IN, USA, 2015.

[56] scikit learn. Scikit-learn documentation: C-support vector classification. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html, 2019. Online; accessed 12-July-2019.

[57] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[58] scikit learn. Scikit-learn documentation: 3.3. model evaluation: quantifying the quality of predictions. https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error, 2019. Online; accessed 12-July-2019.

[59] PennState STAT509. 18.3 - kendall tau-b correlation coefficient, 2018.

[60] SciPy. Scipy documentation: scipy.stats.kendalltau. https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.kendalltau.html, 2019. Online; accessed 12-July-2019.

[61] Mary L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2015.

[62] scikit learn. Scikit-learn documentation: sklearn.metrics.cohen_kappa_score. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html, 2019. Online; accessed 12-July-2019.

[63] David M. Lane. Online statistics education: A multimedia course of study. https://onlinestatbook.com/, 2014. Online; accessed 11-July-2019.

[64] SciPy. Scikit-learn documentation: scipy.stats.pearsonr. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html, 2019. Online; accessed 11-July-2019.

[65] Docker. https://www.docker.com/resources/what-container, 2019. Online; accessed 11-July-2019.

[66] Imperial College London. https://www.imperial.ac.uk/admin-services/ict/self-service/research-support/rcs/rds/, 2019. Online; accessed 10-July-2019.

[67] Ricky Nathvani Emily Muller. Object detection. https://github.com/emilymuller1991/object_detection, 2019.

[68] Sophia Sleigh. Cyclist numbers fall across england as roads are 'too dangerous' to ride. https://www.standard.co.uk/news/uk/cyclist-numbers-fall-across-england-as-roads-are-too-dangerous-to-ride-a4202151.html, 2019. Online; accessed 12-July-2019.

[69] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.

[70] David A Forsyth, Jitendra Malik, Margaret M Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding pictures of objects in large collections of images. In *International workshop on object representation in computer vision*, pages 335–360. Springer, 1996.

[71] Greater London Authority. Better health for all Londoners - Consultation on the London Health Inequalities strategy. Technical report, 8 2017.

[72] Greater London Authority. The London Health Inequalities Strategy. Technical report, 9 2018.

[73] Greater London Authority. Inclusive london: The mayor's equality, diversity, and inclusion strategy. Technical report, 5 2018.

[74] A. Lapedriza A. Oliva A. Torralba B. Zhou, A. Khosla. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.