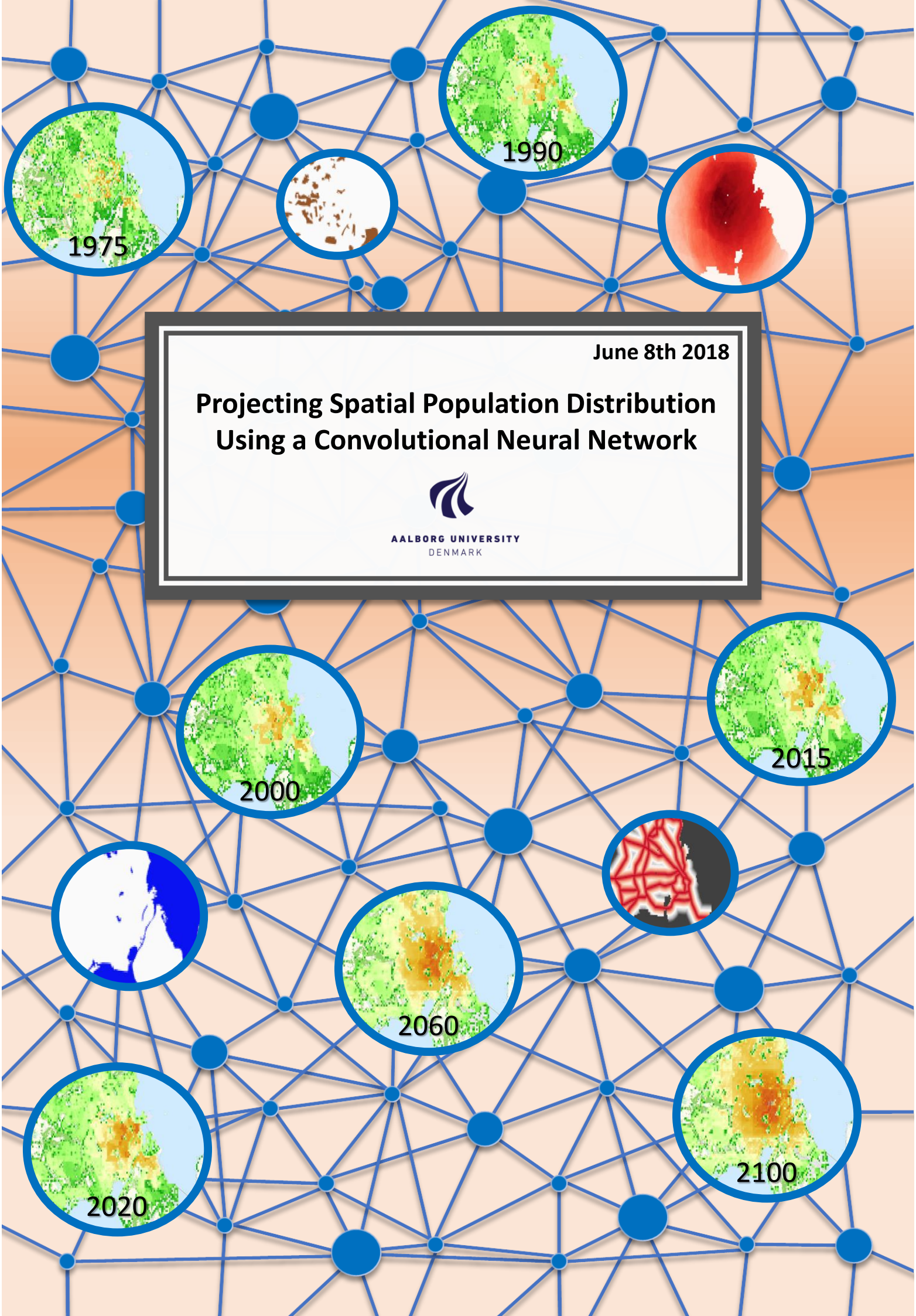

June 8th 2018

Projecting Spatial Population Distribution Using a Convolutional Neural Network

AALBORG UNIVERSITY
DENMARK

AALBORG UNIVERSITY
DENMARK

Department of Development and Planning (Geoinformatics)
A.C. Meyers Vænge 15
2450 Copenhagen SW, DK
Study secretary: Janni Rise Frellsen
Phone number: +45 9940 2535
email: jrl@plan.aau.dk

Study Program and Semester:
Geoinformatics 10th semester

Project Title:
Projecting Spatial Population Distribution Using a Convolutional Neural Network

Project Period:
February 1st, 2018 - June 8th, 2018

Semester Theme:
Master's Thesis

Supervisor:
Carsten Kessler

Written by:
Hans Skaarup Larsen
Niels Bach-Sørensen
Thomas Breilev Lindgreen

Number of Pages:
121

Abstract
Being able to project future spatial population distributions is an important tool to tackle challenges following continuous global population growth and possible future climate challenges. This report presents an answer to how a convolutional neural network can be used to project future spatial population distribution and what results can be achieved by using this approach. This is investigated by designing a convolutional neural network, PopNet. PopNet identifies complex spatial patterns in historical data, on a 250 meter resolution grid and projects the future spatial population distribution. From this architecture, two future scenarios are simulated based on IIASA SSP2 population projections for Denmark and France respectively and the results evaluated. While the neural network method does have flaws, the results prove how, and that, convolutional neural networks can be used to project future population distribution. A number of key challenges, strengths and weaknesses are found and further alterations are proposed that could improve PopNet precision and applicability for future use.

Hans Skaarup Larsen | Niels Bach-Sørensen | Thomas Breilev Lindgreen

Hans Skaarup Larsen | Niels Bach-Sørensen | Thomas Breilev Lindgreen

Preface

This report is our master's thesis in Geoinformatics at Aalborg University. The project-period has elapsed from February 1st to June 8th 2018. The centre of interest in the thesis is to improve how future spatial population distribution can be projected using a convolutional neural network.

Reading Instructions

We are referring to sources in the guidelines given by the Harvard reference method. This means a citation will appear as [Surname, Year, page number] in the text, while details on it is available in the bibliography. We have created all figures and tables that does not have a citation.

Software

The results achieved in the project has been reached with several resources among those are: programming languages, applications and libraries. The most notable of those are listed below. All code related to this project can be accessed through `https://github.com/knasti/PopNet`.

- **Languages**
  - PostgreSQL (with PostGIS) and Python
- **Applications**
  - pgAdmin III, PyCharm and QGIS
- **Libraries**
  - GDAL, NumPy, psycopg2, scikit-learn and Tensorflow

Hardware

The hardware is a premise for the possibilities and limits of what can be done within the field of neural networks. This have had an essential impact on the report as all processes have been run on the available laptop computers. An overview of the used laptops, their CPU's and GPU's can be seen in table 0.1. These are the most vital parts used in training and application of the convolutional neural network created.

	PC One	PC Two	PC Three
GPU	NVIDIA GeForce GTX 960M	NVIDIA GeForce GTX 980M	NVIDIA Quadro M2200
GPU Dedicated Memory	2019 MB	8153 MB	4062 MB
CPU	Intel(R) Core(TM) i7-6700HQ CPU	Intel(R) Core(TM) i7-6700HQ CPU	Intel(R) Core(TM) i7-7700HQ CPU
CPU Performance	2.60GHz (8 CPUs), ~2.6GHz	2.60GHz (8 CPUs), ~2.6GHz	2.80GHz (8 CPUs), ~2.8GHz

Table 0.1: Hardware used for processing and running PopNet

Abbreviations

In table 0.2 the abbreviations used in this project are listed.

Abbreviation	Explanation
ADAM	Adaptive Moment Estimation
CIESIN	Center for International Earth Science Information Network
CNN	Convolutional Neural Network
EEA	European Environment Agency
GPW	Gridded Population of the World
GADM	Global Administrative Areas
GHSL	Global Human Settlement Layer
IIASA	International Institute for Applied Systems Analysis
LRN	Local Reponse Normalization
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
SEDAC	Socioeconomic Data and Applications Center
SSP	Shared Socio-economic Pathways
UN	United Nations
UN DESA	United Nations, Department of Economic and Social Affairs

Table 0.2: Abbreviation overview

In addition to the abbreviations listed above, we are using **PopNet**, short for Population Neural Network, as a name for the final neural network architecture presented in the project.

The term **geosimulation**, which in its essence refer to a stochastic type of prediction model, will in this report be used as a broader catch-all term. As such the term will cover both stochastic models, rule based models and deterministic models, that do not use machine learning to project future spatial population distribution.

The resolution of geographical grids mentioned in this report is the resolution that is applicable at equator.

Contents

List of Figures

List of Tables

Part I

Introduction

Problem Area 1

Projecting the future population is an important piece of information to tackle future demands for food, water and energy supply. The projections also tell policy-makers and experts about major trends that could have national and global effects, and thus enables decision-makers can take action based on different future scenarios. This can be in the form of adapting policies to the scenarios, but also through investments that will help accommodate the changes to maintain, or improve, a sustainable society whether that is economically, socially or environmentally [Population Reference Bureau, 2001; Ahn et al., 2005].

Population projections are usually done at country level, and this project focuses on distributing those numbers to a spatial grid to achieve a high resolution spatial population distribution. This will give opportunity to more accurately adapt to future local changes and tell where possible challenges will appear and react to those in time. The question of where, that is addressed by such a population distribution, allows for further analysis that can decide the right action to take for the different communities. The knowledge of where has also become increasingly important with global warming impacting the world. Consequences ranges from sea-level rise that could leave large populated areas uninhabitable, droughts that kills off crops and makes animal husbandry impossible, to heat waves that make for unbearable living conditions [Carson et al., 2016; Patz et al., 2005]. Being able to match projections of these phenomena with population may prove to be important to avoid disasters through knowledge-based decision- and policy-making.

Today there are models that predict spatial population distribution using geosimulations. These models have to be kept simple due to manually created programming rules and computational scalability. This means that the complexity of the population cannot be modelled on a local scale and might miss important correlations between different variables [Jones and O'Neill, 2016; Keßler and Marcotullio, 2017]. To improve on this we apply the use of a neural network. This field of research has matured along with new computational capabilities that allow for creating very accurate classifications in domains such as image recognition based on analyses of complex patterns in the data [Goodfellow et al., 2016]. That matureness of machine learning is sought to be used and implemented into the modelling of future spatial population distribution. Namely convolutional neural networks (CNNs) are of interest as they have previously been used to recognise spatial patterns, and we believe they can also be used on historical spatial data [Castelluccio et al., 2015]. By that, complex contexts between population and spatial data can be obtained to build a function that estimates the future spatial population distribution.

Thereby, improvements to the spatial population prediction capabilities can improve and

support planning and decision-making for the future which in turn could help shape a sustainable future.

Research Questions 2

This section introduces the main research question and a number of sub-research questions that will serve as a supporting structure for the report and a guide to how the main research question is answered.

In light of the gains and benefits of knowing where how many people live in the future, the objective of this report is to expand the knowledge of solutions within the field of predicting future population distributions. The specific purpose of this report is to investigate how it is possible to use a CNN to estimate the future spatial population distribution. From this purpose the following research question has been formulated:

How can a convolutional neural network be used to project future spatial population distribution and what results can be achieved?

To further guide and support the answering of this question a number of sub-research question have been formulated. The purpose of these questions is to ensure a thorough answer for the main research question, in a structured manner.

The sub-research questions are presented below along with an elaborating explanation of their purpose and how they are sought answered.

1. *What projections of future population growth are applicable to project future spatial distribution?*

   The purpose of this questions is to ensure that the research done is up to date within the field of population projections. That we use applicable and reliable data as a basis for the spatial population distribution. In this respect, answering this question also ensure a foundation of knowledge for understanding possible impacts on results, that using a certain population projection compared to another could have.

   This will be answered by doing a literature review of available population projections and current research, especially in regard to spatially explicit projections. Doing so enables us to determine their differences, methods and if there are clear indicators on how acknowledged they are within their field.

2. *What high resolution spatial and historical population data exists and what geographical features affect future spatial population distribution?*

   This question ensures we are up to date on current research regarding to what geographically related elements are generally accepted to impact population distribution and what spatially explicit historical population data exists. By answering this question, the possible sources of population data we have to train

our CNN are uncovered. This is because the neural networks' ability to predict the future spatial distribution of population is based on historical changes.

To answer this question a literature review of relevant and related research, as well as a study of existing spatially explicit population data is done. As such this question is answered by a review and presentation of existing data and of the discovered relevant geographical elements, that can have an influence on future spatial distribution. This is both done in relation to how data is produced and its reliability in terms of possible effects for further use, but also in relation to metadata such as scale, frequency, age and precision. The data covered is thereby only a review of existing and used data within the field, and not an exhaustive list of all data that could be applied.

3. *What current models and technical approaches are, or can be, used to project future spatial population distributions?*

   Answering this question ensures we are up to date on the field of projecting spatial population distribution and ensure that the research done by this report is not redundant or have already been conducted by others. It also ensures a basic introduction and understanding of CNNs as an approach to solving the main research question.

   It will be answered by a literature review of existing research on the topic of geosimulations and spatial projection. This will be followed by a technical walkthrough of CNNs and how this technology are likely to be able to answer the main research question in terms of possible challenges and solutions.

4. *How can the CNN setup for projecting future spatial population distribution be constructed and how can the used geospatial data be processed and prepared to work with it?*

   The purpose of this question is to ensure a thorough description of how the CNN architecture and practical design is created and why. It also covers what preparation and processing is done to the used geospatial data and the choices made within this process and why.

   By answering this question, transparency into the process, data preparation, choices and construction of the CNN is ensured. This is important as the structure of the CNN and the choices made during preparation of the geospatial data are essential, and have fundamental impact on the results the model gives.

5. *How well does the model predict future population distribution?*

   This question focuses on the results and evaluation of how the model performs when run. Answering it ensures a thorough analysis of the performance and results of the model so that patterns, tendencies, strengths and weaknesses can be found.

   The question is answered by analysing the resulting projections made by the model in order to find tendencies, patterns as well as to estimate how well the model projects future population distribution.

6. *What knowledge can be gained from the results, experiments and tests?*

Answering this question will widen the scope of findings from the results, in order to provide a wider context on how they can contribute both to the field of CNNs as well as spatial population projections in general. Doing so opens for a discussion of how the research done compare to other methods within the field and discuss strengths, weaknesses and new knowledge gained through the entire process.

The questions are therefore answered by discussing the findings of the results, tests and experiments. This is done in a broader context of research already done within the field of projecting future population projections as well as discussing concepts such as causality and uncertainty. As such, the discussion will focus to a lesser degree on the actual prediction results, but more on the methods and perspectives on using a CNN to project spatial population distribution.

The next chapter will briefly introduce the methodologies used and outline how the sub-reasearch questions are answered in relation to the overall report structure.

Methodology 3

This chapter presents the neural network practical methodology and how it is applied in the report. The chapter also contains an overview of the report in the shape of a figure. This figure shows how and where the both the practical methodology is applied but also highlight the use of literature reviews in relation to what sections, and what sub-research questions they support answering.

This means that the methodology presented in this chapter is not exhaustive but only presents the formal methodology that is not commonly applied in research. As such the literature review is not explained in greater detail here. In respect to the evaluation chapter no overall formal methodology for evaluating this type of results exist. As a result the individual methods applied have been considered in terms of their ability to explain and evaluate the results.

3.1 Neural Network Practical Methodology

This section contains the considerations and approach to using and evaluating the neural network. It also points out the necessity of building and testing a neural network in praxis rather than discussing the possibility of applying a CNN.

Within the field of combined neural networks and geoscience there are no immediate fixed methodologies on how to evaluate the applicability of a neural network in regard to a certain task or objective. This means the possibilities of use cannot necessarily be fully understood beforehand, but has to be continually tested, changed and applied to a certain setting to understand it, and its applicability [Goodfellow et al., 2016]. So while the first step of applying a machine learning approach is choosing an algorithm and understanding the principles behind it, the second is to be able to monitor and use the feedback to adapt the model [Goodfellow et al., 2016, p. 416]. In the scope of answering the research question, the first step has been chosen in advance, as the CNN is incorporated in the research question.

What this means in relation to our approach of using a neural network for prediction of future spatial distribution of population, is that to fully understand and find out if it is doable, it is needed to build and test it. This aligns well with the normal praxis on the area of neural networks, where development and science happen in close proximity with building and testing the projects in question. It also means that it is not possible to fully understand the possibilities of applying a neural network in regards to the research question without creating one, testing it and adapting it [Nielsen, 2017].

In relation to the neural network built for the purpose of this report, an overall practical

methodology by Goodfellow et al. [2016] were followed as a guideline to the process and approach. The individual four points can be seen presented below along with an explanation of how this translates into the report [Goodfellow et al., 2016, pp. 416-417].

1. *Determine your goals — what error metric to use, and your target value for this error metric. These goals and error metrics should be driven by the problem that the application is intended to solve.*

   In the terms of predicting future population distribution our error metric is logically the number of population and the goal is to have a realistic projection of where people will likely live.

2. *Establish a working end-to-end pipeline as soon as possible, including the estimation of the appropriate performance metrics.*

   This sets the practical first goal of establishing an initial working end-to-end pipeline, meaning a CNN that is capable of having an input in the correct format and outputting a result. In this case our first step is therefore to establish a simple neural network capable of using a basic spatial population layer and based on this predict and output a result in a useful format such as an image or raster.

3. *Instrument the system well to determine bottlenecks in performance. Diagnose which components are performing worse than expected and whether poor performance is due to overfitting, underfitting, or a defect in the data or software.*

   Point number three refers to the complete design and structure of the code. It should be straight forward to complete the incremental changes mentioned in point number four and see the effects of such changes. However, this point will also be revisited for adjustments incrementally as better ways to design the architecture might come up.

4. *Repeatedly make incremental changes such as gathering new data, adjusting hyperparameters, or changing algorithms, based on specific findings from your instrumentation.*

   When the base has been successfully established as described in bullet point two, more ancillary data can be added through an iterative process. In the case of the neural network established here, the ancillary layers are linked to spatial phenomenons and attain the same format as the input data. In addition to adding data, a number of changes are likely to be made and evaluated, both as a result of the instrumentation and the cost-function, but also as a result of testing the capabilities of the neural network on different scenarios.

In addition to the practical methodology presented, the architecture have to be considered more closely. This is because the neural network have to be adapted to fit the research question and have the capability to estimate, rather than categorize. What is meant by this is that the CNN have to use population projection numbers and therefore these have to be integrated into the architecture. This process is covered in greater detail in chapter 8.

While the effects of following this methodology cannot necessarily be readily seen from the report, as the process of building the neural network itself cannot be observed, some effects can be seen from the report structure and the process of testing, evaluating and changing the network (cf. section 8.3 and 8.4).

3.2 Report Overview

This section contains a simple overview of the report and to what chapters the different methodologies have been applied to and what research question is answered by each part of the report.

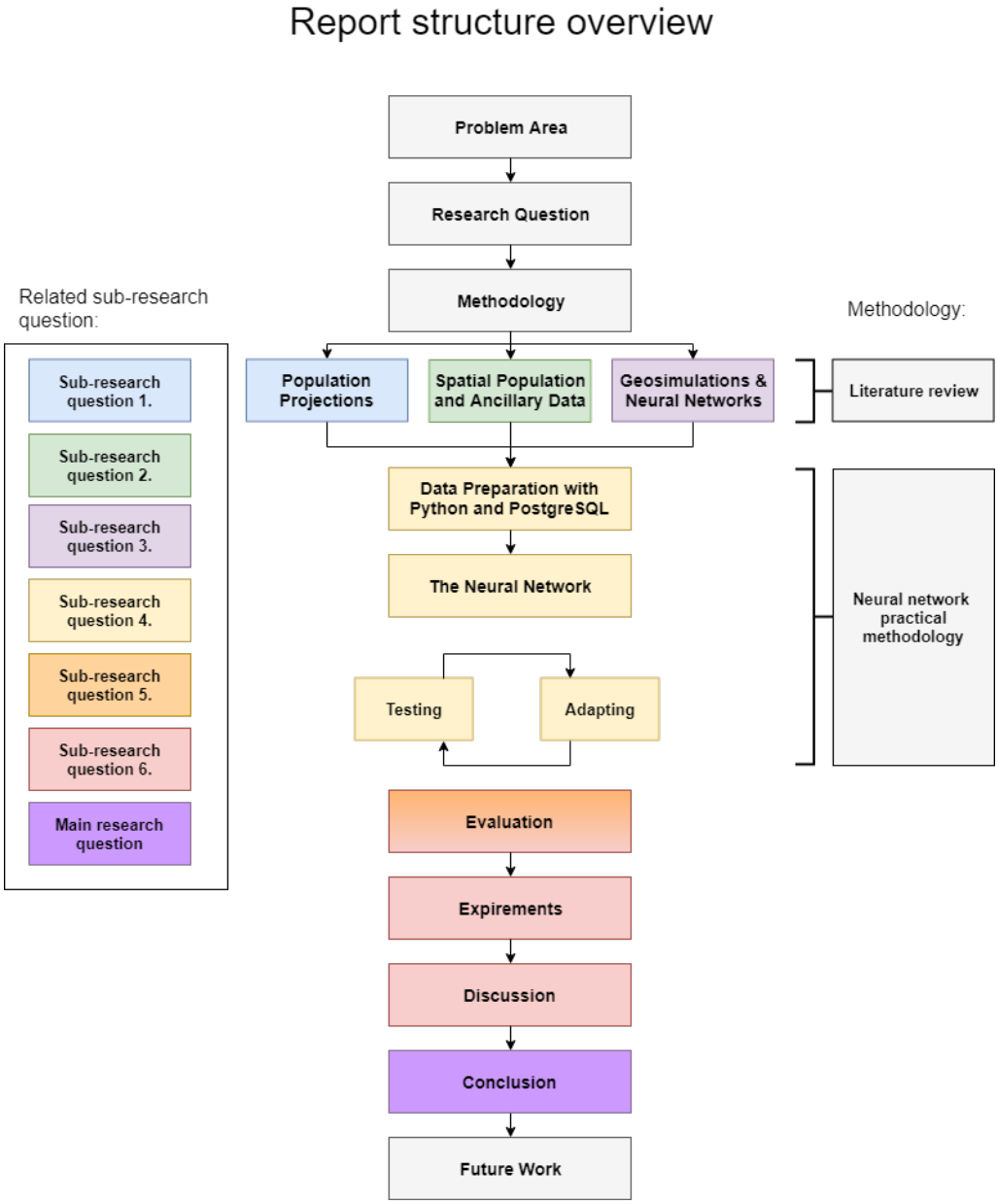

Figure 3.1: Overview of report structure, research-questions and methodology application

In relation to the conclusion, the main research question is not answered by the conclusion

itself but by the answers to- and findings of the sub-research questions which is summarized in the conclusion - thereby answering the main research question.

The next part will examine existing research through the literature review. The first chapter of the next part will introduce population projections created by different organisations.

Part II

Literature Review

Population Projections 4

This section introduces the concept of population projections and the context of them in the report. The two main organizations and their respective population projections are introduced and loosely compared in order to highlight existing differences, both in terms of methodology and resulting numbers. The section ends with a short discussion that compares the different projections to the report purpose and their use in current research of spatially explicit population projections.

4.1 Global Population Projections

There is an increasing number of humans on the planet and this number has been steadily increasing for thousands of years. While the early stages of increasing population was largely linked to the invention and adoption of agriculture, it was mainly from the 19th century and onwards that human population experienced increasing growth. This is linked to a number of inventions such as fresh water supply, improvements within health, nutrition and medicine that resulted in lower death rates. Later the introduction of medicine such as antibiotics once again drove death rates down further and while these changes occurred relatively fast, the more cultural and tradition based norms affecting the number of children a family got, changed comparatively far slower [KC and Lutz, 2017, p. 181]. These previously described advancements along with further improvements within medicine, vaccines and healthcare led to a population increase from roughly 2.5 billion in the 1950's to 7.5 billion in 2017 [United Nations, 2017b]. Some research however, suggests that we are likely to see a peak in world population during the next 100 years due to a number of things such as decreasing birth rate [IIASA, 2014]. Yet if this will come true is still debatable, as other sources, for example the United Nations (UN), forecasts continuous population growth at least until the year 2100 [United Nations, 2017a].

For the purpose of this report we rely on existing future population projections and therefore do not present changes, new models or projections. It is nonetheless important to understand the scope and limits of the projections used. Global population projections rely on predicting human behaviour and knowledge of social and economic trends, that depend on how humans act in the future. Predicting the future dealing with human behaviour cannot be done without uncertainty [Lutz and Samir, 2010, p. 2783]. This inherent uncertainty of the projections must therefore be considered in relation to the results in this report.

4.2 Different Population Projections

Creating projections about the future world population is bound to different dynamics, population drivers and assumptions about how the world develops and how this impacts the population growth and trends. A number of individual organizations create varying projections about future world population, based on different methodologies and assumptions. This section will introduce a number of different, available projections of future world population which can be used as a basis for determining spatial distribution of future population.

The United Nations, Department of Economic and Social Affairs (UN DESA) publishes a set of world population projections and currently revises the assessments every second year, the last revision published in 2017. The UN population projections present estimates for 233 countries and areas based on both official data from the individual countries as well as data gathered by other survey programs and departments of the UN such as United Nations High Commissioner for Refugees on for example international migration patterns [United Nations, 2018].

The UN world population data contains a historical time series from 1950 and currently projects world population till the year 2100. The basis of the UN population projection is a medium variant projection which corresponds to the median of several thousand projected trajectories of specific demographic components. There is however a total of nine different projection variants in the 2017 revision, which differ in relation to assumptions on the level of fertility as well as mortality and migration, the difference between the individual projections can be seen from table 4.1 [United Nations, 2017c, p. 30]. These differ however based on the medium variant, for example as the high fertility variant is projected as 0.5 births above the medium variant and the low fertility projection as 0.5 births below [United Nations, 2017c, p. 16].

	Assumptions		
Projection variants	*Fertility*	*Mortality*	*International migration*
Low fertility	Low	Normal	Normal
Medium fertility	Medium	Normal	Normal
High fertility	High	Normal	Normal
Constant-fertility	Constant as of 2010-2015	Normal	Normal
Instant-replacement-fertility	Instant-replacement as of 2015-2020	Normal	Normal
Momentum	Instant-replacement as of 2015-2020	Constant as of 2010-2015	Zero as of 2015-2020
Constant-mortality	Medium	Constant as of 2010-2015	Normal
No change	Constant as of 2010-2015	Constant as of 2010-2015	Normal
Zero-migration	Medium	Normal	Zero as of 2015-2020

Table 4.1: Projection variants in terms of assumptions for fertility, mortality and international migration [United Nations, 2017c, p. 31]

In more recent times compared to when the UN started publishing population predictions, the International Institute for Applied Systems Analysis (IIASA) started publishing long-range global population projections in 1994. These projections were centered around 13

regions and three different scenarios [O'Neill et al., 2001, p. 209]. They are currently contributing to a larger research project called the Shared Socioeconomic Pathways (SSPs). The human core, which is what the population projections of the shared socioeconomic pathways is, is a joint project between the IIASA and other members of the Wittgenstein Centre of Demography and Global Human Capital [Lutz et al., 2014a, p. 4].

The SSPs are part of a scenario based framework, established by the climate change research community in order to facilitate the integrated analysis of future climate impacts, vulnerabilities, adaptation, and mitigation. The pathways were developed over the last years as a joint community effort and describe plausible major global developments that together, would lead to different challenges for mitigation and adaptation to climate change, in the future [van Vuuren et al., 2017, p. 153].

The SSP scenarios consist of five scenarios that are built upon five different qualitative predictions of possible future world developments. While the SSPs main purpose is to be used in relation to climate change adaption and mitigation, the scenarios are constructed based on socioeconomic and environmental elements, challenges and development that have been judged to be determinants for adaption and mitigation [O'Neill et al., 2017, p. 171].

The five different scenarios can be described briefly as;

The first scenario, SSP1, is a positive sustainability scenario that describe a gradual change toward a more balanced and sustainable path with shared global resources, decreased inequality and increasing education and health worldwide. The second, SSP2, is the middle of the road scenario where world development continues based on historical patterns with slow improvements to inequality and slow improvements to sustainability. The third, SSP3, is a regional rivalry scenario that puts emphasis on a fragmented and nationalist world which leads to decreasing focus on equality, sustainability, health and education. The fourth, SSP4, is a scenario with focus on social inequality both within and between countries leading to a gap between groups of people with lower education and labor intensive jobs and an internationally connected group with higher education and economic wealth. The fifth and last SSP5 focuses on a rapid fossil-fueled development future, in which faith is put in the market economy and technological progress. This development leads to increased investment in health and education, but also relies upon the faith in successful mitigation of possible environmental problems [van Vuuren et al., 2017, p. 157].

The population projections are one of the key SSP scenario quantitative drivers, of the overall SSP framework and the main focus in relation to this report. The individual population projections and how the SSP scenarios have been translated into population projection parameters of fertility, mortality, migration and education based on certain country grouping. The country groupings consist of high fertility countries, low fertility countries and low fertility countries that are part of OECD. This can be seen from table 4.2 presented below. Projections are done on a country level for all countries of the world, and have projections stretching to the year 2100.

Country groupings	Fertility	Mortality	Migration	Education
SSP1				
HiFert	Low	Low	Medium	High (FT-GET)
LoFert	Low	Low	Medium	High (FT-GET)
Rich-OECD	Medium	Low	Medium	High (FT-GET)
SSP2				
HiFert	Medium	Medium	Medium	Medium (GET)
LoFert	Medium	Medium	Medium	Medium (GET)
Rich-OECD	Medium	Medium	Medium	Medium (GET)
SSP3				
HiFert	High	High	Low	Low (CER)
LoFert	High	High	Low	Low (CER)
Rich-OECD	Low	High	Low	Low (CER)
SSP4				
HiFert	High	High	Medium	CER-10%/GET
LoFert	Low	Medium	Medium	CER-10%/GET
Rich-OECD	Low	Medium	Medium	CER/CER-20%
SSP5				
HiFert	Low	Low	High	High (FT-GET)
LoFert	Low	Low	High	High (FT-GET)
Rich-OECD	High	Low	High	High (FT-GET)

Table 4.2: Shared Socioeconomic Pathways definitions for the demographic and human capital component [KC and Lutz, 2017, p. 184]

This highlights one key difference between the UN projections and the IIASA SSP projections. The IIASA addresses the role of education level in global population trends.

Lutz et al. [2014a] argue that by adding education as a key component to the conventional demographic projection elements of age and sex, this substantially alters the resulting future population projections. The argument is that education serves as a clear, if not the single most important source of population heterogeneity, as higher education leads to lower mortality and fertility [Lutz and Skirbekk, 2017, p. 2]. Thereby the addition of the education element serves as a supporting quality dimension to predicting future population numbers. It is however stressed by Lutz et al. [2014a] that it is nearly impossible to proof causality between these factors under all circumstances and cultures, yet they argue that there are good reasons for assuming that the assumption would hold over the projection period [KC and Lutz, 2017, p. 182].

4.3 Comparing Projections

The UN data is the most used population projection but the IIASA is also widely used. These are therefore the main focus in terms of possible predictions for creating future spatial distribution of population [Lutz et al., 2014a, p. 528].

As described previously the UN and SSP population projections do not match up completely. As can be seen from figure 4.1, showing the scenarios as graphs of projected

world population by the UN and SSP, these are quite different.

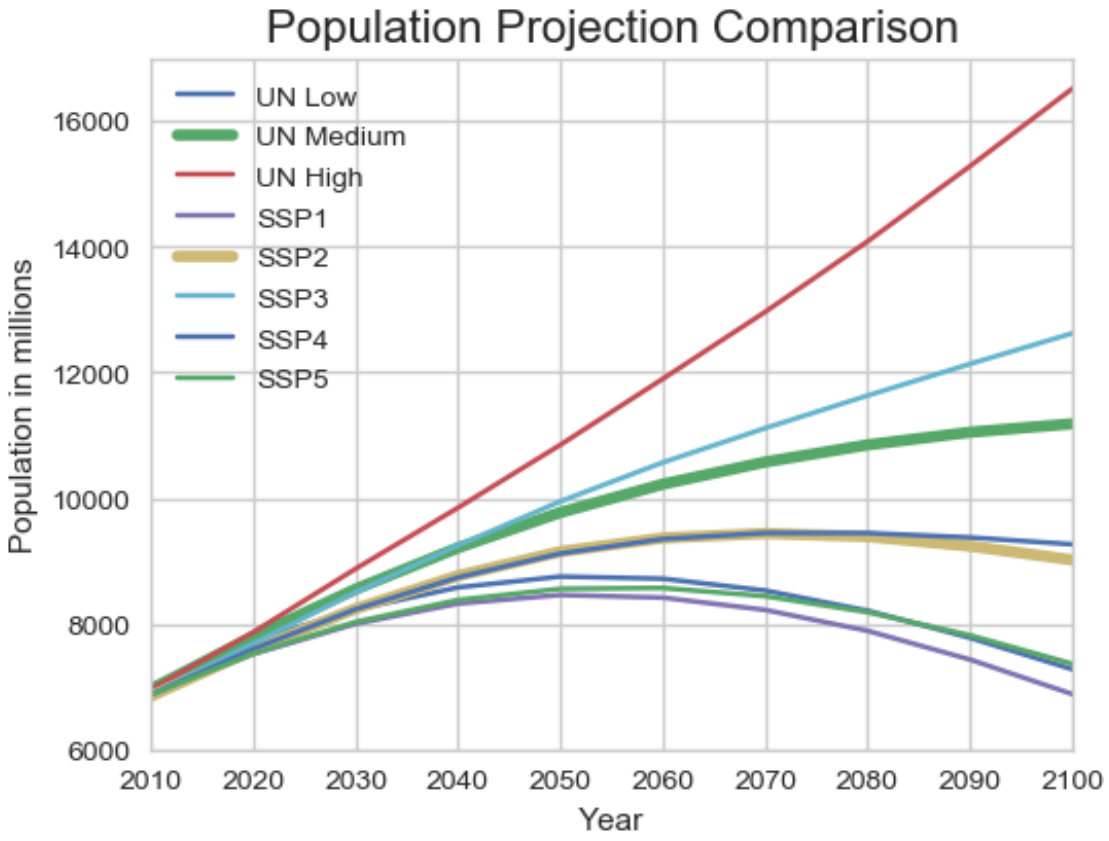

Figure 4.1: Comparison of UN and IIASA medium population projection scenarios [United Nations, 2017b; IIASA, 2018]

From these we can see that the UN population projection predicts a vastly higher amount of population growth for both the medium variant compared to the SSP2 scenario and the high Variant compared to the SSP3 scenario. On the medium scenario the UN projects a growth towards a world population of approximately 11.2 billion people by the year 2100 compared to the SSP2 projection which projects more than two billion less people at approximately 9 billion people by end of the decade.

One reason for this is as previously touched upon that the UN and IIASA population projections use different assumptions for how to calculate future fertility and mortality trajectories which result in different population projection results. However, there are two other factors that impact this. The first is the fact that IIASA adds a level of education as an impacting factor on especially fertility. When this arguably important source of population heterogeneity factor is taken into account it can impact projections significantly. An example of this can be seen from Nigeria. Here the UN projects a population increase from 160 million to 914 million from 2010 to 2100 compared to the IIASA, who using education as a factor, projects an increase to 576 million by 2100 [Lutz et al., 2014b]. Unrelated to that, the second factor is that the UN and the IIASA have different approaches to current fertility in China and Africa. In the example Nigeria from 2014, the UN uses an assumption that a woman gets an average of six children, which

has been steady for the previous 10 year period. Data from 2013 however, point toward an average of 5.5 children per women in Nigeria, which the UN projects to hit around 2020-2025. As a result IIASA research argue that the UN is slow to adjust to new data and changes because of their statistical approach [Lutz et al., 2014b].

In the scope of this report, the problem lies in projecting where people will live in the future and in that respect we build upon, and therefore inherit the same challenges and uncertainties as displayed from the different population projection choices.

In relation to spatial population geosimulations and spatial demographic projection research, there are examples of both the UN and the SSP projections being used. Keßler and Marcotullio [2017], uses the data supplied by the UN DESA as that data simultaneously distinguish between rural and urban population per country which can support the spatial distribution of the population numbers. In comparison, Jones and O'Neill [2016] uses the SSP population projections in conjunction with the matching SSP urbanization projections to generate an outcome that matches the SSP prediction.

As such there are examples of contemporary research relying on either of the two datasets. While the demographic projections that the IIASA SSP projections seem the most inclusive as they are based on both a qualitative, as well as a quantitative research process and include more demographic elements in the form of the education parameter.

The data used for the spatial projections in this report will mainly rely on the IIASA SSP medium scenario. This is done to be able to create a result consisting of comparable spatial projections for multiple countries or regions based on the same population projection. These can then in the future be compared to results based on different population projections to see how this affects the projected future spatial distribution.

The differences between projections also highlight that every result that builds upon the population projections, and thereby the projected spatial distribution of the projected population, must be regarded with a higher risk and uncertainty the further it goes into the future. This is unavoidable due to the nature of predicting an aspect of the future that depend on human behaviour which apply to both projecting population development but also to the projection of the spatial distribution of said population.

This section answers the first sub-question of; *What projections of future population growth are applicable to project future spatial distribution?*. Both the UN and the IIASA SSP population projections which are introduced are highly used and projects population growth by country until 2100. The IIASA SSP projections seem to have a slight edge as it is uses more elements in form of education which arguably have a high impact on future population growth. In the scope of applicability in regards to spatial population distribution, there are examples of both being used in current research and the numbers projected are the main difference between the two. The numbers however do not affect the spatial projection method, but only the results.

The next chapter will investigate available data that is applicable to recognise population patterns, and thereby can be used to make a gridded population distribution prediction.

Spatial Population and Ancillary Data 5

In recent years there have been a substantial development in relation to available satellite imagery and spatial datasets, that are relevant for global human population estimation and mapping. This section introduces available data that can be used within a machine learning environment to model future spatial population distribution in different countries. Furthermore, it will investigate the data in terms of its characteristics and the methods used to produce the data.

5.1 NASA Socioeconomic Data and Applications Center

Gridded Population of the World (GPW) is a global dataset developed by Center for International Earth Science Information Network (CIESIN). The first version of GPW was released in 1995, and the current GPW version 4 (GPWv4) was released in July 2016 and updated in November 2017. GPW models the human distribution on Earth, on a continuous global raster surface with different target years. The latest GPWv4 consists of the following layers and target years [CIESIN, 2016]:

- Population Count (2000, 2005, 2010, 2015, 2020)
- Population Density (2000, 2005, 2010, 2015, 2020)
- UN WPP-Adjusted Population Count (2000, 2005, 2010, 2015, 2020)
- UN WPP-Adjusted Population Density (2000, 2005, 2010, 2015, 2020)
- Data Quality Indicators (2010)
- Land and Water Area (2010)
- Administrative Unit Center Points with Population Estimates (2000, 2005, 2010, 2015, 2020)
- National Identifier Grid (2010)
- Basic Demographic Characteristics (2010)

Different resolutions and formats are available for the GPWv4 dataset. The native and most detailed resolution is a global gridded output of approximately one kilometer at equator in reference system GCS WGS 1984. The grid has also been aggregated to other lower resolutions of approximately 5, 30, 55 and 110 kilometers. All layers are available in GeoTiff, ASCII and five of the layers are also available in NetCDF format. However, the NetCDF format is not available in a 1 kilometer resolution [CIESIN, 2017, p. 4].

Development Method
The GPWv4 uses an areal-weighting method combined with census data to produce a gridded raster surface covering the Earth. The development consists of the following steps:

1. Locate tabular population counts
2. Match population counts to geographic boundaries (census or administrative)
3. If needed, adjust boundaries to the global framework
4. Estimate population for target years
5. Adjust population to UN estimates
6. Estimate population by age and sex
7. Transform to raster

The GPWv4 is developed by collecting tabular population data from national statistical offices and organisations. This data is then matched to spatial boundary data from national agencies such as statistical offices, mapping agencies and planning agencies. To ensure alignment between boundary data, it is then matched to the Global Administrative Areas (GADM) data layer [Hijmans et al., 2015]. Thereafter, the population is estimated and adjusted for all census or administrative areas, and adjusted to the UN estimate if needed. The estimated population within each administrative boundary area is then distributed using an areal-weighting method to a one kilometer grid [CIESIN, 2017, pp. 6-11].

Accuracy and Limits
Because of the way the GPWv4 dataset is developed using areal-weighting, it may have certain characteristics that the user needs to keep in mind. By not using additional data for allocating the population to the grid other than census data within the administrative boundaries, and by utilizing a simple areal-weighting approach, the data keeps the fidelity of the input data. However, the areal-weighting method is affecting the precision of the population within each cell in the grid, as the precision and accuracy of a given cell is a direct function of the size of the input administrative area. This means, that in countries or areas, where the input administrative areas are quite large, the precision within each grid cell will be degraded [Lloyd et al., 2017b, p. 2]. Therefore, study areas that are smaller than the average size of the administrative units, should not be used. As such, the data is most suited for larger study areas and will only be applicable for local analysis in certain places. Furthermore, the data also contains artificially high population densities along coastlines next to high populated areas, where actual land area within a given cell may be very small. This is again related to the areal-weighting method [CIESIN, 2017, pp. 24-25].

5.2 European Commission – Global Human Settlement Layer

The Global Human Settlement Layer (GHSL) project is supported by the Joint Research Centre European Commission which contributes to the project together with CIESIN, Columbia University. The project produces several datasets that contain information about the spatial population distribution on earth. These data layers consist of built-up areas, population density maps and settlement maps. The information for these layers is derived from data mining technologies and evidence-based analytics of global archives with satellite imagery, census data and volunteered geographic information. The data is automatically and systematically processed to generate information about the presence of population, settlements and infrastructure. The dataset consists of the following layers and target years [European Commission, 2015]:

- GHS Built-Up Grid (1975, 1990, 2000, 2015)
- GHS Built-Up Quality (1975, 1990, 2000, 2015)
- GHS Population Grid (1975, 1990, 2000, 2015)
- GHS Settlement Grid (1975, 1990, 2000, 2015)

Depending on the data layer, the data is available in different resolutions. The layer is either available in a 38, 250 or a 1000 meter grid in reference system GCS WGS 1984. Built-Up Grid is available in all resolutions. Built-Up Quality is only available in a 38 meter grid. Population is available in 250 and 1000 meter grids and Settlement is only available in a 1000 meter grid. The data is available in raster format as TIFF files together with OVR (pyramids) files [European Commission, 2016].

Development Method
The GHSL project builds upon the GPWv4 project, to produce a multi-temporal population grid with better resolution than GPWv4. This is done by combining the GPWv4 population estimates from CIESIN with GHS Built-Up presence for the years 1975, 1990, 2000 and 2015 to a 250 meter grid. Population estimates are produced and provided by CIESIN for the target years. The population estimates are then matched to administrative boundaries and checked for bordering issues, because different countries or administrative areas can have different surveying techniques. To correct these issues, the GADM data layer is used. Furthermore, the GHS Built-Up layer, which is derived from LANDSAT imagery for the target years, has been aggregated from the native 38 meter grid to a 250 meter grid, which describes the proportion of built-up area within each cell. The final population grid is produced through raster based dasymetric mapping, where the Built-Up layer is used to allocate and restrict the distribution of the population [Freire et al., 2016, pp. 1-3].

Accuracy and Limits
A validation was performed by IRC and CIESIN on a sample of the data to ensure that all of the input population was disaggregated, and that the totals from each administrative area was preserved. This was performed through a correlation analysis on a sample of 18 European countries resulting in an r-value of 0.83, which means that there is a strong correlation between the input population and the disaggregated result from the dataset [Freire et al., 2016, p. 4].

5.3 WorldPop

The WorldPop project started in 2013 to combine the continent based studies of Afripop, Asiapop and Ameripop. The aim with the project is to produce detailed population distribution data for the whole of Central and South America, Africa and Asia that can be used to measure the future impact of population growth and facilitate planning. The resolution of this dataset is a 100×100 meter grid [WorldPop, 2016a]. The coordinate system varies, but usually UTM is used for country level data with people per hectare or a country specific grid. Datasets with people per pixel is usually projected in GCS WGS 1984. The temporal resolution is however varying between two or three target years [WorldPop, 2016b].

Production Method
Like GHSL, WorldPop also uses a semi-automated dasymetric modeling approach, where population census data, satellite images, administrative areas and ancillary data used as a weight layer, are combined in a 100 meter grid to estimate the population distribution for different years. The approach is similar to GHSL, but Worldpop also uses the ancillary data which are comprised of layers that describe the land cover and subsequently have a relation to the presence of people. These land cover data are among others; urban area, water bodies, trees, industrial area, slope and protected areas [WorldPop, 2016c].

5.4 Ancillary Datasets

In dasymetric modelling, it is common to include ancillary data layers, that can have an influence on the population distribution. This is done to improve the detail and precision of the estimates and can be done by utilizing the relationship between the ancillary data and the presence of, or lack off, people.

This can be seen in the population grids introduced previously, for example the WorldPop project use them for estimating spatial population distribution based on historical data [Tatem, 2017]. In both previous and current research, there are examples of using ancillary data to estimate current and future land use or urban growth, and to support estimation of future spatial population distribution.

Herold et al. [2003] use slope, roads network, lakes, ocean, parks and natural preserves as ancillary data to describe and improve estimation of future urban growth. What this points toward is that there is a correlation between for example the slope of a given area or distance to nearest road and future urbanization.

The same point is made by Pijanowski et al. [2013], as they use ancillary data such as distance to existing urban area, distance to stream, distance to primary road and more, to support a simulation urban growth. In this report which focuses on US data, they discover that highways in this case had the highest positive contribution to the goodness of fit of the model [Pijanowski et al., 2013, p. 262].

While these examples of literature do not directly link the associated ancillary data with population, they do link it to urban land use. This logically leads to the conclusion that if urbanization is linked to these factors, so is population, as urbanization usually follow increasing population numbers. There is of course exceptions related to industrial areas where people do not live. In the same way, areas with water bodies are likely linked as a limiting factor to urban land use and can thereby also be used as a logical inhibitor to human habitation in general.

These links are also supported by the research on future population distribution. Keßler and Marcotullio [2017] uses water bodies to ensure that the population estimation is only done on viable areas and urban extends as this is a key part of their model. In the same way, Jones and O'Neill [2016] uses slope, surface water and mandate for protection to narrow down where future urbanization and thereby population increases are likely to occur.

As such, existing research underline that ancillary data is a key component in improving the estimation of future spatial distribution of population. Which ancillary data to use depends on what is being modelled and what the researcher is trying to estimate. The most commonly used datasets describe areas with; slope, major roads and various land cover types like permanent water bodies, wetlands, forests and protected nature types [Nagle et al., 2014, p. 3].

Ancillary Data

The following subsection will investigate the availability and coverage of possible ancillary datasets that can have either a negative or a positive correlation with population and settlement distribution.

The Copernicus Programme

Copernicus Global Land Service is part of the Land Monitoring Core Service. The Copernicus Programme monitors the earth to provide a wide range of services by combining data from satellites and ground censors. The pan-European component of Copernicus coordinated by the European Environment Agency (EEA) provides land cover and land use data which covers Europe. Potential ancillary data from Copernicus and coverage years are listed below [Copernicus, 2018]

- **Corine Land Cover** (1990, 2000. 2006, 2012)
  Corine land cover consists of 44 different classes of land cover types such as; water bodies, sea and ocean, forests, green urban areas, continuous urban fabric etc. available as vector data.
- **EU-DEM Slope** (2012)
  Slope is available as a 25 meter resolution raster grid
- **Forests** (2012)
  Forests is available in a 20 or 100 meter resolution raster grid
- **Permanent Water Bodies** (2012)
  Permanent Water Bodies is available in a 20 or 100 meter resolution raster grid
- **Wetlands** (2012)
  Wetlands is available in a 20 or 100 meter resolution raster grid

Other Datasets

As mentioned, major roads are also an indicator for the presence of people and for expansion of settlement which always branches out from existing roads. Sources with road data that has a good coverage are found to be OpenStreetMap and Socioeconomic Data and Applications Center (SEDAC), which is a department under NASA. Sources with available road data with global coverage and the layer names are therefore;

- **OpenStreetMap** - Roads
- **SEDAC** - Global Roads Open Access Data Set

EEA is also a large provider of different environmental datasets. One of the datasets provided contains lakes and rivers available as vector data [European Environment Agency, 2012], which can be useful for calculating coverage. OpenDataSoft also provides different free datasets, among those are European train stations [Capitaine Train, 2015] which likely correlate with the presence of population.

- **EEA** - European Lakes
- **OpenDataSoft** - European train stations

This section has touched on topics in relation to the second sub-question from the research question section:
What high resolution spatial and historical population data exists and what geographical features affect future spatial population distribution?

Upon researching what spatially explicit population data that exists, it is apparent that the availability of such data and its temporal resolution is fairly limited. When estimating the future population for a given country or region, there is a need for a wide range of ancillary and population data for as many target years as possible, which leaves a wider temporal resolution as something to be desired in the current datasets. As it can be seen in the summary data table 5.1, three sources for temporal population data was found, them being SEDAC, European Commission and WorldPop. Of these three data sources, the European Commission data seems to be a somewhat precise and detailed dataset in terms of its grid size of 250 meters and its population estimate in each grid cell. SEDAC has a larger grid size and does not incorporate built-up areas when estimating the population in each grid cell, from the administrative areas. WorldPop has less coverage and with a temporal resolution of only two to three target years.

The next chapter is investigating research and approaches that have been used to create spatial population distributions, while examining it in a machine learning context.

Source	**Layer Name**	**Data Format** **Grid Sizes** **meters = m,** **kilometers = km**	**Temporal Resolution** **(Target Years)**	**Coverage**
GPWv4	Population Count	Raster 1, 5, 30, 55 and 110 km	2000, 2005, 2010, 2015, 2020	World
GPWv4	Population Density	Raster 1, 5, 30, 55 and 110 km	2000, 2005, 2010, 2015, 2020	World
GPWv4	UN WPP-Adjusted Population Count	Raster 1, 5, 30, 55 and 110 km	2000, 2005, 2010, 2015, 2020	World
GPWv4	UN WPP-Adjusted Population Density	Raster 1, 5, 30, 55 and 110 km	2000, 2005, 2010, 2015, 2020	World
GPWv4	Data Quality Indicators	Raster 1, 5, 30, 55 and 110 km	2010	World
GPWv4	Land and Water Area	Raster 1, 5, 30, 55 and 110 km	2010	World
GPWv4	Administrative Unit Center Points with Population Estimates	Vector (Point)	2000, 2005, 2010, 2015, 2020	World
GPWv4	National Identifier Grid	Raster 1, 5, 30, 55 and 110 km	2010	World
GPWv4	Basic Demographic Characteristics	Raster 1, 5, 30, 55 and 110 km	2010	World
GHSL	GHS Built-Up Grid	Raster 38 m, 250 m and 1 km	1975, 1990, 2000, 2015	World
GHSL	GHS Built-Up Quality	Raster 38 m	1975, 1990, 2000, 2015	World
GHSL	GHS Population Grid	Raster 250 m and 1 km	1975, 1990, 2000, 2015	World
GHSL	GHS Settlement Grid	Raster 1 km	1975, 1990, 2000, 2015	World
WorldPop	Population	100 m	Varies between 2010 to 2020 with two or three target years.	Central and South America, Africa and Asia
Copernicus	Corine Land Cover	Vector	1990, 2000, 2006, 2012	Europe
Copernicus	EU-DEM Slope	Raster 25 m	2012	Europe
Copernicus	Forests	Raster 20 m and 100 m	2012	Europe
Copernicus	Permanent Water Bodies	Raster 20 m and 100 m	2012	Europe
Copernicus	Wetlands	Raster 20 m and 100 m	2012	Europe
OpenStreetMap	Roads	Vector	Varies	World
SEDAC	Global Roads Open Access Data Set	Vector	Varies depending on country from 1980 and 2010	World
EEA	European Lakes	Vector	2012	Europe
OpenDataSoft	European Train Stations	Vector	2015	Europe

Table 5.1: Summary table of sources and available data

Geosimulations & Neural Networks 6

The SSP and UN scenarios predict an increase in the human population throughout the 21st century (cf. chapter 4). To meet the challenges that arise with such an increase, it is essential to know the future spatial distribution of people [Keßler and Marcotullio, 2017]. This will allow governments and organizations to assess resource allocation, transport and urban planning, poverty mapping and environmental impacts among other, which thereby allow them to act proactively [Lloyd et al., 2017a].

The idea of projecting future population can be traced back to 1798, where Malthus [1798] put focus on issues arising with population increase. More recently, studies have been trying to find methods to predict population growth on a country basis, but also in higher spatial resolution with grids from 1.000×1.000 meter to 100×100 meter. The addition of geosimulation has played a major role in achieving such high spatial resolution.

Torrens and Benenson [2005] introduced the concept of geosimulation, which is based on Cellular Automata and Multi-Agent Systems, combined referred to as Geographic Automata Systems. In such a system automatons change states in time-steps according to predetermined rules based on internal and external information. In general an automaton's neighbourhood impacts the change in states the most. However, the automatons can also change location allowing for simulating moving objects such as migrating households, that gives the opportunity to model complex phenomena. Geosimulations were initially used in urban modeling, but have many other use cases in a spatial context, for example distribution of population over time. Benenson and Torrens [2004] have implemented such a geosimulation approximating the future global population.

Keßler and Marcotullio [2017] have recently published an article that introduces a population geosimulation. The simulation estimates the future spatial population distribution globally on a grid with 1.000×1.000 meter resolution up to year 2100. The approach in this paper is based on rules that randomly relocates people into cells so that they match the population projection of UN DESA. In addition country-specific thresholds on the cells' maximum number of people are used to determine how the population is distributed. The simulation runs the rules in ten year increments [Keßler and Marcotullio, 2017, p. 2]. This, despite its simplicity, produces a seemingly realistic scenario on a global scale. However, to capture the future spatial population distribution accurately on a local scale, more complex simulations will be needed. This could be in the form of a model with more complex rules based on statistics as proposed by the authors, but we believe this problem is potentially better suited for a machine learning approach [Keßler and Marcotullio, 2017, p. 3].

Machine learning, and especially neural networks, can pick up patterns that are hard for humans to discover, and even then are extremely hard to capture in a programming context. This is due to its ability to mimic complex phenomena through layers of neurons that are interconnected with weights and biases, as shown in figure 6.1. The input layers represents the features that are used to predict the output, where the hidden layers are used to find patterns in the input features. While traditional machine learning techniques like Linear Regression, K-means, Decision Trees and Random Forest are generally easier to set up and trains faster, they tend to perform worse than neural networks when predicting complex phenomena. This is evident from, among other, the ImageNet competition, where neural networks outperforms other machine learning algorithms [Goodfellow et al., 2016, pp. 18-25]. This is the reason why we are exploring the usage of neural networks, and not other machine learning techniques, to estimate future population, which is a very complex phenomenon as it depends on numerous factors and their interaction with each other.

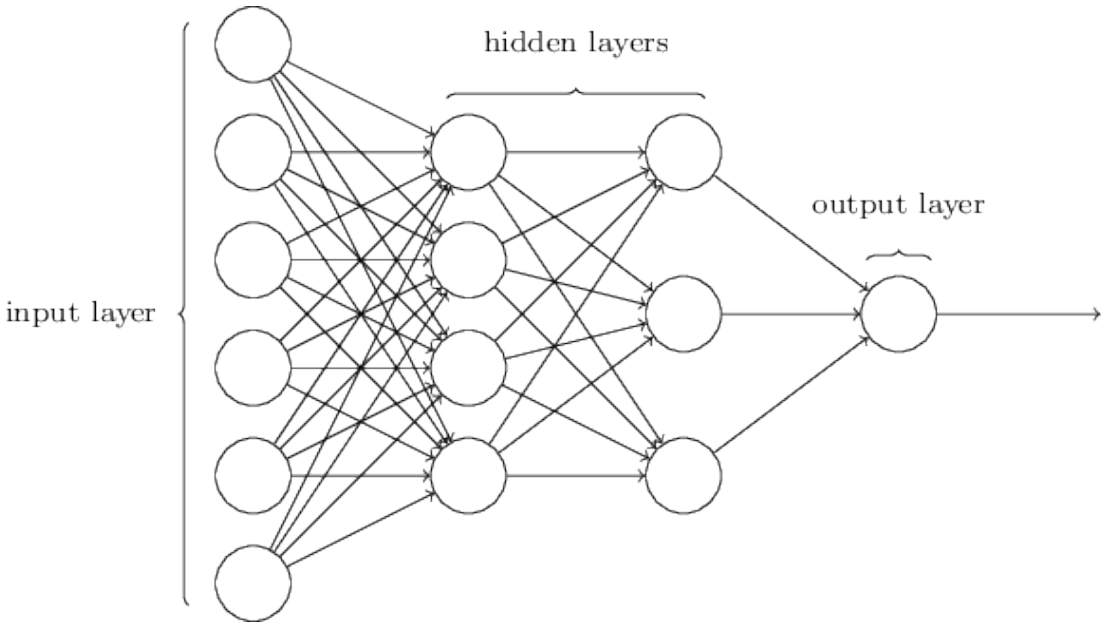

Figure 6.1: Simple neural network showing the connections between neuron-layers [Nielsen, 2017]

Furthermore, historical population data is now available in structured grids as found in chapter 5. This fits neural networks' requirement about having highly regular data input [Qi et al., 2016, p. 1].

There are three paradigms within machine learning; supervised learning, unsupervised learning and reinforcement learning [Sutton and Barto, 2017, pp. 1-2].

Supervised learning is used to predict phenomena by training a model fed with input and labeled data. The model creates a function so the model's output has as big of an resemblance as possible of the labeled data. When this process, called training, is finished it is possible to use new input data, but without the labeled data, to predict the phenomenon [Learned-Miller, 2014]. This is used in different fields such as speech and image recognition that has a wide range of application uses.

Unsupervised learning is used to find hidden structures in unlabeled data [Sutton and Barto, 2017, p. 2]. A simple example is height and weight differences between men and women, where an unsupervised learning algorithm can separate such data points into two clusters. A more useful example is to use it for data security, where an unsupervised learning algorithm can detect anomalies and act on those [Eskin et al., 2002]. This paradigm has a lot of potential as there is no need for creating labels manually, which can be a hindrance as it takes time and are not necessarily simulating the correct behavior [Hinton and Sejnowski, 1999].

Reinforcement learning trains agents to take actions in an environment towards a specific goal [Sutton and Barto, 2017, pp. 1-2]. The agent is rewarded for desired actions leading it closer to the goal, thus making it go through a trial-and-error process to determine the best actions. This means that this technology is excelling, when it is possible to define an isolated environment and define rules and a goal within it [Sutton and Barto, 2017, pp. 2-4]. This is for instance used in games, where the goal is to win, where the likes of AlphaGo have beat the best human players in Go [Sutton and Barto, 2017; Churchland and Sejnowski, 2016, pp. 365-372].

The supervised learning paradigm is the most suitable for our project. We have historical data that can be used as input and labels. In addition, this data has undergone data processing making it suitable for this project (cf. chapter 5). At the same time there is no apparent way to create an environment with rules and a goal, to resemble the required conditions in reinforcement learning. The neural network method, CNN, that is part of the research question and typically used for supervised learning is explained below.

CNNs emerged in 1998 with Yann LeCun's neural network (LeNet-5) inspired by the research made by Hubel and Wiesel [1962] on the visual cortex and receptive field [LeCun et al., 1998, pp. 2284-2285]. The concept of CNNs builds upon construction of complex patterns, that can be used to analyze images, based on local features derived from neighboring pixels. This is based on the assumption that neighboring pixels are more correlated than others, which corresponds well with Tobler's first law of geography *"everything is related to everything else, but near things are more related than distant things"* [Tobler, 1970, p. 236]. Thus, it should be possible to use a CNN similarly to analyze spatial phenomena, such as the focus in this project; spatial population distribution. At the same time neural networks have become more feasible due to the amount of data available and increasing computing power. The data makes neural networks able to train on large amounts of historical data to learn, and the computational improvements makes it possible to do it within a reasonable time frame. Krizhevsky et al. [2012] showed the potential of neural networks in 2012 in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), where he achieved the highest accuracy with his CNN AlexNet. Most recently the CNNs GoogLeNet and ResNet have achieved even higher accuracy in the competition, which just stresses the continuous improvement in the field [Szegedy et al., 2015; He et al., 2016]. There are other types of neural networks such as the recurrent neural networks that memorizes data used in speech, but they are not nearly as efficient in recognizing spatial and gridded patterns [Goodfellow et al., 2016, pp. 326-335].

A simplified describtion of CNN's is that they operate by finding patterns in image format data. An example of this could be an input image of $224 \times 224 \times 3$ pixels. In this case

the first two numbers refer to the height and width of the image and the last number refers to the depth which for regular images is three from the red, green, blue image layers (RGB). The basics of the CNN is that the computer reads the input pixel as a number corresponding to for example the RGB 255 color scale in pictures. This can then be used to find and identify patterns by applying supervised machine learning like CNN [Goodfellow et al., 2016].

The CNN trains on the image through convolutional layers. In the first convolutional layer, the filters applied looks for a certain amount of overall patterns such as orientation, lines, colors etc. An example of such a trained layer can be seen visualized below in figure 6.2.

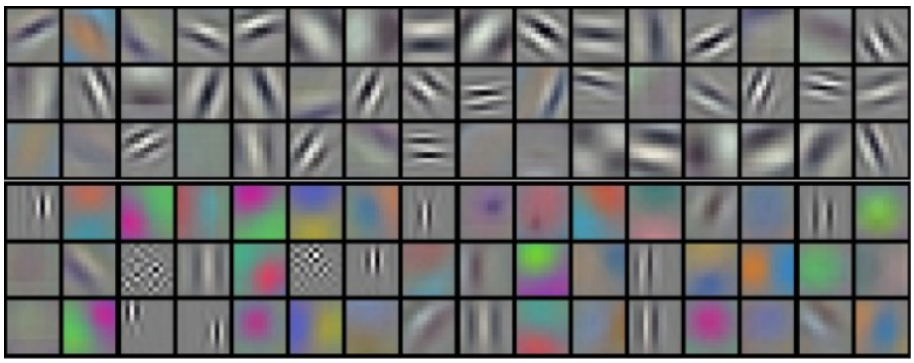

Figure 6.2: Example visualisation of an initial convolutional layer trained on AlexNet [Krizhevsky et al., 2012, p. 6]

A second convolutional layer then applies the findings from the first one and combines them to create new and more complex set of patterns than the first convolutional layer. An example of a visualization of a second convolutional layer can be seen in figure 6.3 where elements such as lines, circles and curves are combined.

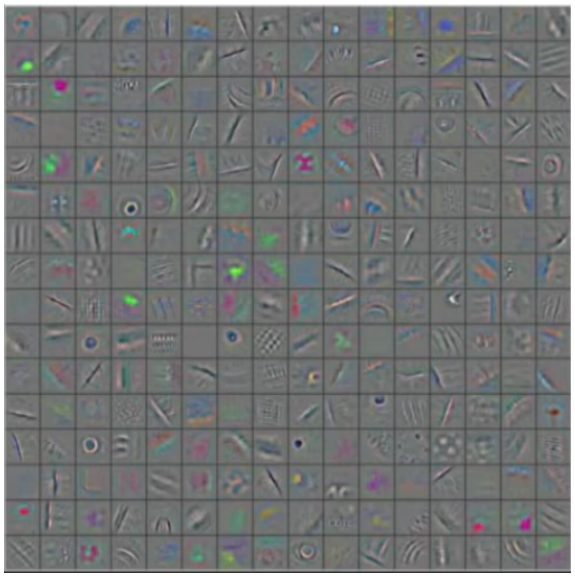

Figure 6.3: Example visualisation of a second convolutional layer [Zeiler, 2015]

In short, the more correctly applied layers the more complex and unique shapes the neural

network is capable of identifying. This also depends on the data that the filters are trained on, as in most cases it is trained as a supervised learning from which it learns the filters as visualised in figure 6.2, 6.3 and 6.4.

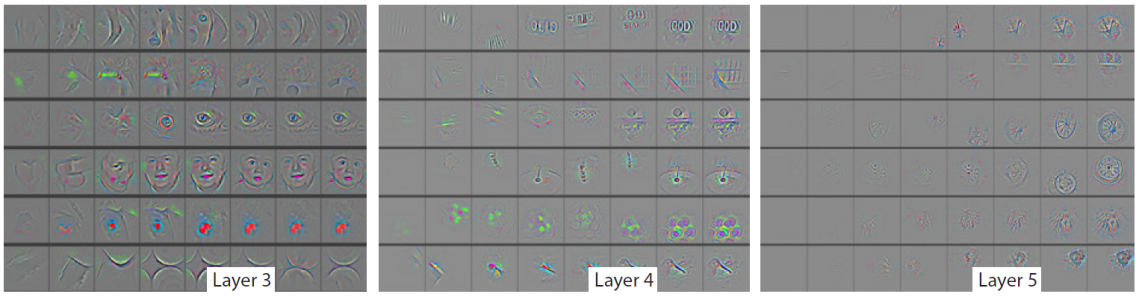

Figure 6.4: Example visualisation of a third, fourth and fifth convolutional layer [Zeiler and Fergus, 2013]

When the model with all trained layers is applied it will ideally recognize the pattern of the image to be somewhat similar to the images trained on and classify the content accordingly. While this is a very simplified way of describing the more advanced CNNs, since they utilize a number of advanced functions to change values, decrease size by combining values to, for example, decrease processing time and improve recognition ability. This is also based on the fact that most CNNs are, as previously noted, used to classify image content such as a certain object like a car, boat or horse.

The use of neural networks is not entirely new in geographic science as Tang et al. [2006] back in 2006 proposed to use neural network models to improve population estimates.

In 2013, Pijanowski et al. [2013] made use of a CNN in a geographical context, where the Stuttgart Neural Network Simulator was used. The neural network they created performs a geosimulation that predicts urban growth on a national scale. The geosimulation is based on The Land Transformation Model, which uses historical data to learn about spatial patterns. The model's results are validated against independent datasets from another source or year, as there are no ground truth to validate against.

Another study, authored by Robinson et al. [2017] uses a CNN on satellite imagery to estimate population in a 0,01° × 0,01° grid. Differently from our study, the authors are trying to compliment the existing gridded population data such as the once mentioned in chapter 5 (WorldPop, SEDAC and GHSL), rather than predicting the future population distribution. The neural network is trained on input from LANDSAT and US Census population counts from year 2000 and validated on the same data from year 2010. They have treated the problem as a classification problem and based the architecture on the VGG-models in the high-level neural network API, Keras [Simonyan and Zisserman, 2014b]. This means that rather than finding a specific number of people within a grid cell they have classifications such as; no people, few people and many people.

The text above answers the sub-question; *What current models and technical approaches are, or can be, used to project spatial population distributions?* and touches on the other sub-question; *How well does the model predict future population distribution?*.

In regard to the first of those questions, it is apparent that a lot of models that deal with spatial change already exist. Those models have been changing over time, and have gotten better. Currently geosimulations have been used to simulate and estimate similar challenges to that of our research objective. However, this approach become increasingly harder the more complex the model, while also suffering from performance issues. A handful of studies have also used neural networks to estimate and predict spatial change, but all as classification problems. We want to implement a CNN that solves a regression problem, e.g. the number of people within a given cell. This will be done by creating a network architecture that captures the spatial patterns in spatial population development based on historical population data paired with features such as slope, height, water and infrastructure.

Validation of population estimates can be tricky as there is no ground truth to compare the results to. This is also applicable for regular historical population counts or the population grids mentioned in chapter 5, as they all have uncertainties attached to them. Nonetheless, those datasets can be assumed to be somewhat close to reality and thus also be used as validation datasets, while keeping their uncertainties in mind. This means it could be possible to train the model on a number of years and then leave a year out for validation like Pijanowski et al. [2013] and Robinson et al. [2017] do. Another approach is to train the model on a country and use that model on another country in the same region, that can be expected to have similar population distribution patterns.

This chapter marks the end of the Literature Review, and the next focus will be on the preparation and implementation of spatial population distribution predictions using a CNN.

Part III

Preparation and Implementation

Data Preparation with Python and PostgreSQL

7

This chapter will answer the secondary part of the fourth sub-question;
How can the CNN setup for projecting future spatial population distribution be constructed and how can the used geospatial data be processed and prepared to work with it?
After acquiring and downloading the necessary raw data from different portals and data suppliers listed in chapter 5, it needs to be prepared and formatted correctly, so that it can be used by the CNN. The input format for the CNN is normally in the form of images. This means that it is based on reading a set of pixels where every pixel contains one value. This is the same principle as with raster geodata. The effect of this is that every set of data needs to be formatted and contained in an image-like format.

In this case a TIFF image file format has been chosen. The TIFF file can be geocoded as a GeoTIFF which means that its geospatial location can be saved and read from the metadata. This is however not useful in relation to using the data in a CNN environment - rather the geolocation known in the GeoTIFF metadata is merely the outermost X and Y coordinates of the entire array of pixels which is useful for later visualization. Each TIFF file can however contain multiple bands which can be described as different sets of pixel values, which means that every set, both population data and ancillary data, can be saved into the same file in different bands. Before the data is saved as different bands in the TIFF file format, the needed calculations and preparation has to be done. In the case of this project the preparations are done partially by python scripting, tools such as a PostgreSQL/PostGIS database as well as GDAL and OGR2OGR tools.

This chapter contains a presentation of how the raw data from chapter 5, consisting of population and ancillary data is processed through Python and PostgreSQL to prepare it for use with the CNN. The whole preparation process is illustrated in figure 7.1.

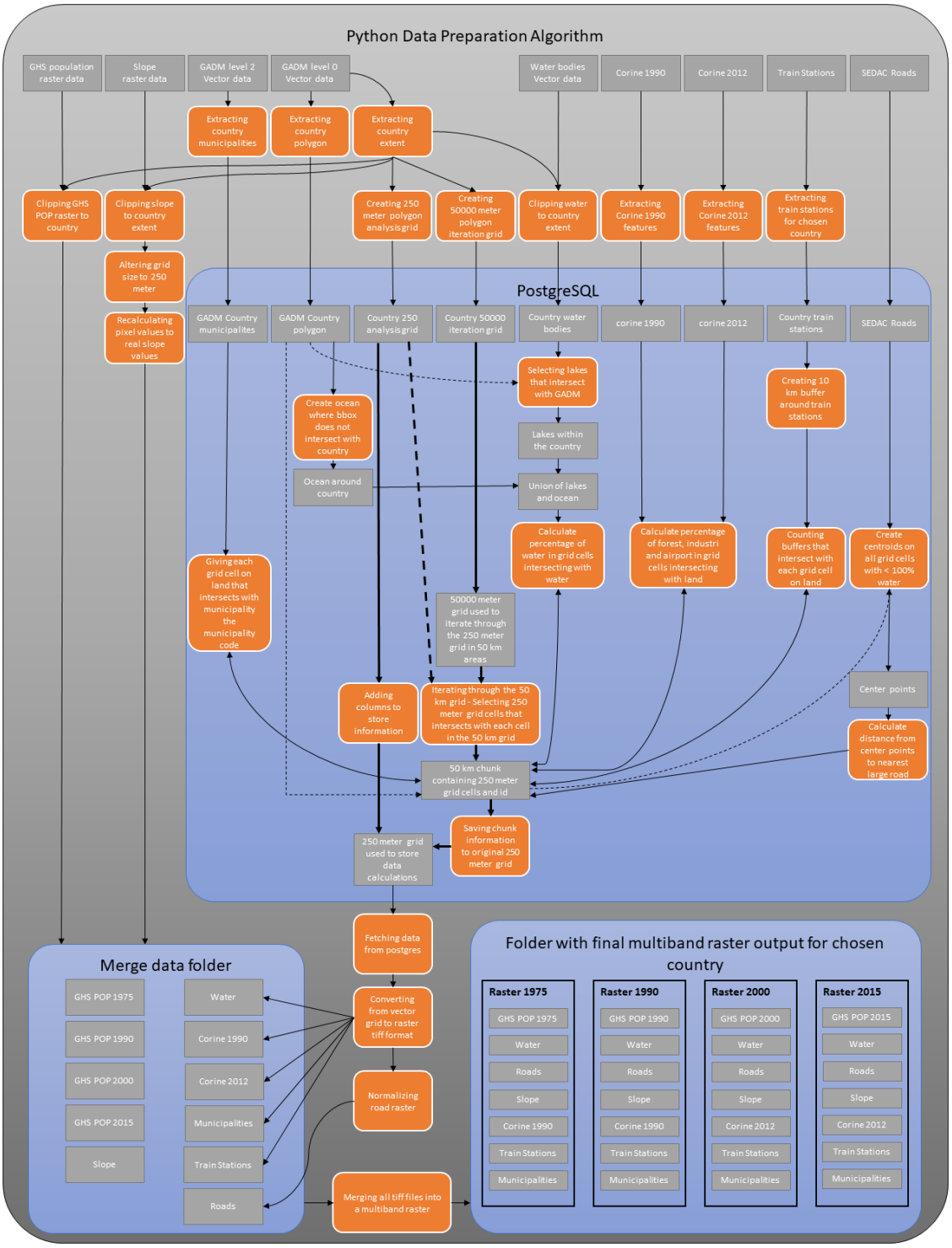

Figure 7.1: The data preparation process in Python and PostgreSQL

7.1 Initial Data Preparation

The data preparation is handled with the use of Python programming, different libraries for example GDAL, and PostgreSQL to prepare the raw data for use with the CNN. In this project it was decided to automate the preparation process to be able to process data for one country at a time when needed. Several approaches were investigated, especially

how to optimize the time consuming spatial queries in PostgreSQL. In the initial setup, the data was processed in PostgreSQL on a country level, which led to queries taking days to calculate. A solution for this was found by combining Python loops with a grid and through an iterative process calculate smaller sections of the map at a time. The whole data preparation process illustrated in figure 7.1 consists of several scripts and is explained in subsequent text.

The whole algorithm is divided into several scripts and functions:

- main.py - main script, choose country, different options and run the data preparation
- process.py - script for initial data preparation and import of functions from other scripts
- import_to_postgres.py - contains a function to import data to PostgreSQL
- postgres_queries.py - contains the PostgreSQL query setup
- postgres_to_raster.py - contains functions for extraction of data from PostgreSQL and conversion to raster format.
- rast_to_vec_grid.py - contains a function for converting raster to vector grid

The initial part of the data processing is handled in Python with the use of GDAL. This process takes care of limiting and extracting the data that matches the chosen country and prepare it for import to PostgreSQL. This is done to limit the data and optimize calculation time, by only operating on data within the spatial extend of the chosen country.

The chosen country is first extracted with GDAL from the GADM adm0 dataset containing country polygons and an extent or bounding box layer is created. The extent is then used to clip the country out of the slope and GHS population rasters. The end result of this, is four population rasters for the years 1975, 1990, 2000, and 2015, which are saved in a merge folder awaiting merging with other data. The slope raster, which is in a grid of 25 meters, is altered to 250 meters with gdalwarp by using the extent of one of the GHS population raster files through a subprocess call.

```
1 cmds = 'gdalwarp -s_srs EPSG:54009 -tr 250 250 -te {0} {1} {2} {3} -cutline {4} -srcnodata 255
      -dstnodata 0 {5} {6}'\
2 .format(minx, miny, maxx, maxy, cutlinefile, srcfile, dstfile)
3 subprocess.call(cmds, shell=True)
```

We are using the world Mollweide equal area projection, EPSG 54009. The reason why this projection is used is because it is an global equal-area projection which aligns with the idea of being able to make spatial population predictions for any country in the world or even on a global scale. It is also the same format that the GHS population data is in, which makes it easy to use this as the basis and adapt other data to it. The drawback of using a pseudocylindrical projection is that the scale becomes less accurate the farther away from the 40:44 N and 40:44 S standard parallels. For this research it means that this should be considered when preparing data for certain locations such as far from the center of the projection - and under certain circumstances such as calculating distances to a linestring with few vertexes, that have been reprojected from another projection.

Afterwards, the slope pixel values, which initially have values between 0 and 255, is recalculated with an equation provided with the dataset, to give the real slope value between 0 and 90 degrees, before the slope raster also is saved in the merge folder.

```
cmds = 'python {0}\gdal_calc.py -A {1} --outfile={2}
    --calc="numpy.arcsin((250-(A))/250)*180/numpy.pi" --NoDataValue=0'\
.format(python_scripts_folder_path, dstfile, outfile)
```

Next, two vector grid layers are created with the function "rasttovecgrid". This is a 50 km iteration grid and a 250 meter analysis grid, which are used in PostgreSQL to speed up the query calculation time and to store data. The spatial extent is taken from one of the existing population rasters and grid cell size is set to 250 meters.

```
rasttovecgrid(out_file name and path, minx, maxx, miny, maxy, size_x, size_y)
```

The rest of the data, which is water, municipalities, train, Corine cover and roads, are prepared by either clipping to country extent or extracting necessary features with ogr2ogr before importing them to PostgreSQL together with the iteration- and analysis grid.

7.2 Calculation in PostgreSQL

After the data is loaded into PostgreSQL the main calculations are done based on area. The main concept of this, is that there is a need to process the vector data and other information into an input, such as an integer, that can be interpreted by the CNN and output it in a pixel based format. The data handling with Python in PostgreSQL is built up around the 50 km iteration grid and communication with PostgreSQL is handled through Psycopg2, which is a Python package for handling comminucation with PostgreSQL. This section contains a walk through example of the processing done and the queries used, which is illustrated in figure 7.1 in the box labelled PostgreSQL. All coverage calculations follow the same procedure, which is described below.

The first step in the data process is to get the id numbers of all the cells in the 50 km iteration grid covering the chosen country, which are saved in a list called ids with the following code:

```
# getting id number of sections within the iteration grid covering the country
ids = []
cur.execute("SELECT gid FROM {0}_iteration_grid;".format(country))
section_id = cur.fetchall()

# saving ids to list
for id in section_id:
ids.append(id[0])
```

The list of ids is then used to iterate through the 50 kilometer grid, selecting one 50 kilometer section at a time. For each section, a check is initially performed, this check is implemented to speed up the overall query time, as the check does not cost a lot of time. For the Corine 1990 layer, the check is performed by confirming, if the 50 kilometer section intersects with any Corine data. If it does not intersect, there is no need to calculate anything as the coverage column in the 250 meter analysis grid is initially set to 0 by default. Therefore, the script proceeds to the next section in the 50 kilometer grid. If the section does intersect with the Corine 1990 data layer, the section is saved as a table containing the 250 meter grid cells' geometry and ID's for further processing, as seen below in the code.

```
for section in ids:
```

```
        # start single section query time timer
        t0 = time.time()

        # Check if section intersects with corine cover layer
        cur.execute("SELECT {0}_iteration_grid.gid \
                        FROM {0}_iteration_grid, subdivided_{0}_corine90 \
                        WHERE ST_Intersects({0}_iteration_grid.geom, subdivided_{0}_corine90.geom) \
                        AND {0}_iteration_grid.gid = {1};".format(country, section))
        result_check = cur.rowcount

        if result_check == 0:
                print("Section number: {0} \ {1} is empty, moving to next section".format(section,
                    len(ids)))

        else:
                print("Section number: {0} \ {1} is not empty, Processing...".format(section,
                    len(ids)))

                # select cells that is within each section and create a new table
                cur.execute("CREATE TABLE section_nr{1} AS (SELECT {0}_cover_analysis.id,
                    {0}_cover_analysis.geom \
                                        FROM {0}_cover_analysis, {0}_iteration_grid \
                                        WHERE {0}_iteration_grid.gid = {1} \
                                        AND ST_Intersects({0}_cover_analysis.geom,
                                            {0}_iteration_grid.geom));".format(country, section))
                conn.commit()
```

The last part of the code is the actual calculation of the Corine coverage. The query selects all the ID's in the section, which is actually the ID's of the 250 meter grid cells within the section and calculates the intersection area between a 250 meter cell and Corine 1990 cover. The result is then saved in the 250 meter analysis grid, where the ID's match between the section ID and analysis grid ID. Lastly, the section table is dropped, before moving on to the next section in the 50 kilometer grid.

```
# calculating corine 1990 coverage
cur.execute("WITH a AS (SELECT section_nr{1}.id, sum(ST_AREA(ST_INTERSECTION(section_nr{1}.geom,
    subdivided_{0}_corine90.geom))/62500*100) as corinecover \
                    FROM section_nr{1}, subdivided_{0}_corine90 WHERE
                        ST_intersects(section_nr{1}.geom, subdivided_{0}_corine90.geom) \
                    GROUP BY id) \
                    UPDATE {0}_cover_analysis SET corine_cover90 = corinecover from a WHERE a.id
                        = {0}_cover_analysis.id;".format(country, section))
conn.commit()
```

7.3 Post Data Preparation

When all the data, consisting of water, Corine 1990, Corine 2012, train, roads, municipalities has been calculated and saved in separate columns in the 250 meter analysis grid. It is then exported to different shapefiles by the use of ogr2ogr as seen in examples below.

```
cmd = 'ogr2ogr -f "ESRI Shapefile" {0} PG:"host={1} user={2} dbname={3} password={4}" \
-sql "SELECT id, corine_cover, geom FROM {5}_cover_analysis"'.format(path, pghost, pguser,
    pgdatabase,
pgpassword, country)
subprocess.call(cmd, shell=True)
```

Once all the data has been exported, the shapefiles are then turned into rasters to match the GHS population raster with gdal_rasterize as seen below. These files are then saved

in the merge folder, where the slope and population raster files also are located.

```
cmd = '{0}\gdal_rasterize.exe -a CORINE_COV -te {1} {2} {3} {4} -tr {5} {6} {7} {8}' \
.format(gdal_rasterize_path, minx, miny, maxx, maxy, xres, yres, src_file, dst_file)
subprocess.call(cmd, shell=True)
```

The last operation is merging the raster files into single multiband raster files for the years 1975, 1990, 2000 and 2015. This is done with gdal_merge.py. The input for this script is an outfile path and name, and a list of TIFF files that needs to be specified in the same order as the bands they should be saved in. In this process the population rasters are divided into the four years and a merge with the ancillary data is performed for each of the four years. Furthermore, the Corine layer also has different years, 1990 and 2012. It was decided to include Corine 1990 for the years 1975 and 1990, and Corine 2012 for the years 2000 and 2015.

```
outfile = country_path + "\{0}.tif".format(1975)
original_tif_pop = merge_folder_path + "\GHS_POP_1975_{0}.tif".format(country)
water = merge_folder_path + "\{0}_water_cover.tif".format(country)
road_dist = merge_folder_path + "\{0}_roads.tif".format(country)
slope = merge_folder_path + "\slope_{0}_finished_vers.tif".format(country)
corine = merge_folder_path + "\{0}_corine1990.tif".format(country)
train = merge_folder_path + "\{0}_train_stations.tif".format(country)
municipal = merge_folder_path + "\{0}_municipality.tif".format(country)
cmd_tif_merge = "python {0}\gdal_merge.py -o {1} -separate {2} {3} {4} {5} {6} {7} {8}"\
.format(python_scripts_folder_path, outfile, original_tif_pop,
water, road_dist, slope, corine, train, municipal)
subprocess.call(cmd_tif_merge, shell=False)
```

The merged TIFF files one for each year is then saved in a final folder, ready to be used in the CNN, as seen in the bottom of figure 7.1.

In the next chapter the convolutional neural network architecture is described, tested and adapted. This will then fully answer the fourth research question.

The Neural Network 8

This chapter shows how we have been using machine learning to answer the first part of the sub-question;
How can the CNN setup for projecting future spatial population distribution be constructed and how can the used geospatial data be processed and prepared to work with it? The second part of the question will also be addressed in section 8.2 as chapter 7 does not cover it completely.

The machine learning framework used is Google's Tensorflow which provides functions and methods to support a neural network architecture. This also means that Python is used, as it is the main language that Tensorflow is directed towards [Google, 2018a].

8.1 Programming Structure

The structure of the project's machine learning code is important to ensure simplicity that helps with understanding, using the code and following the principles of the practical methodology from section 3.1, that refers to the design and structure of the code's ability to adapt to changes. Our structure is heavily inspired by Gemy's GitHub project Tensorflow-Project-Template, while also picking up elements from Qi's PointNet [Gemy, 2018; Qi et al., 2016].

The structure allows for changing model hyperparameters in a configuration file. his means that parameters such as learning rate, batch size etc. can be controlled and changed in one place when training and testing different scenarios. In addition, the different models can be saved in specific folders, indicated in the configuration file, which makes it possible to have log data, models and output for multiple countries and regions at once, in a simple and controlled way. The code design is object-oriented, thus allowing for easier debugging and avoidance of redundant code among other.

Gemy's structure is well accepted on GitHub, meaning that our structure should be natural to navigate for developers and researchers familiar with machine learning. This will allow for easy replication of this project's setup and methodologies.

The structure can be seen in figure 8.1 below. It consists of the following folders: *base*, *configs*, *data*, *data_loader*, *data_scripts*, *experiments*, *mains*, *models*, *trainers* and *utils*. The *data*-folder contains the input data that was created in chapter 7 and *data_scripts* have the code from that chapter.

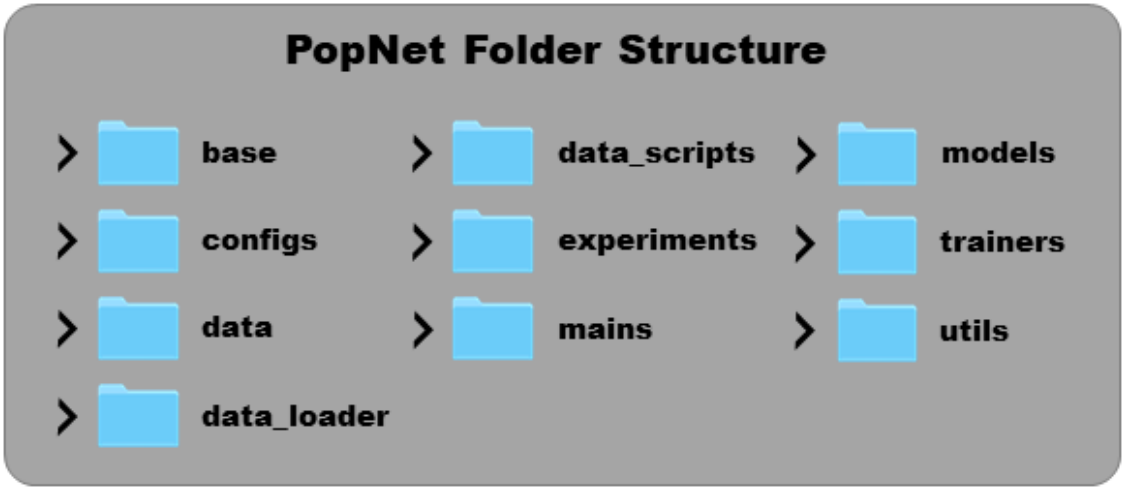

Figure 8.1: Folder structure used in the programming structure

The folder, which contains the classes for base-model and -trainer, is *base*. The base-classes ensure that different models and trainers inherit the same methods, thus making a change to a model or trainer easier.

When the base-classes are created, the model(s) and trainer(s) which are tailored to our solution, can be used. The models and trainers are available in the *models*- and *trainers*-folder respectively.

The utilities folder, *utils*, contains scripts that logs results, parse the configuration file and creates new folders in *experiments* based on the country or region that has been chosen.

Before training can be started, data needs to be processed and loaded so it fits the model's data placeholders' shape, in Tensorflow this is where the input data will be assigned upon training, testing and use. This is done with scripts in the *data_loader* folder, which loads the previously prepared data from *data*, and also generates arrays that fits the before-mentioned placeholders. Furthermore, the user needs to specify parameters for the configuration-file, in the *configs* folder, as mentioned above.

The *mains* folder contains the main file, which utilizes and gathers the elements from the other folders. This means that when running this file it will automatically load in the data chosen in the configuration, load in the model and start training and testing as well as logging the process.

Trained models are saved on the fly so that it is possible to have access to models even though the training has not yet finished. In addition, a model will be created when the last epoch of the training has been run. This folder also contains the script, that uses a trained model to generate an output e.g. a prediction for the the future spatial distribution in the given area. Outputs, logging and models are saved within the *experiments* folder's sub-folders such as country- or region-names like Denmark, India and Northern-Africa.

8.2 Neural Network Preparation

In chapter 7 we create a multiband TIFF that can be imported as a 3D-Tensor, which has the shape; [height, width, features]. This tensor will have to be processed further for it to be used in the neural network. In theory we could feed in the whole tensor at once, but this creates several issues. One is that it takes a lot of memory to run all the data simultaneously, while only allowing the weights to update once per epoch, which is inefficient in itself. In addition, some optimizers have hard times escaping local minima, if the data is not noisy, which is not the case when the whole dataset has been processed. This results in an even more inefficient training that might get stuck [Bottou, 2010; Ge et al., 2015]. Furthermore, this will only allow the model to be used for equally sized tensors limiting the use to the country it was trained on. Thus, it will remove the possibility to use the model for parts of the country or other similar countries. The alternative is to use batches, where a batch contains a small sample of the dataset. In models that train on the renowned MNIST-dataset which consists of 70,000 images of handwritten digits, a batch is typically set to 100 images [Google, 2018b]. However, our dataset is one grid and not a bunch of images like the MNIST-dataset or many of the other datasets used in CNNs (cf. chapter 6). To cope with this, the grid is divided into mini grids, in this report called chunks, that each basically resembles an image as shown in figure 8.2. Hereafter we can divide the chunks into batches like image-data.

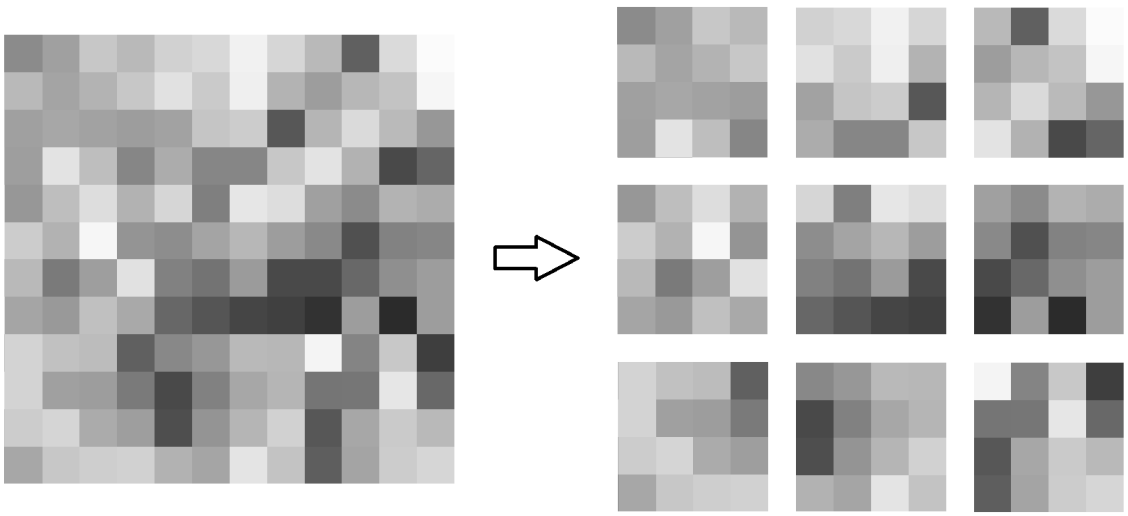

Figure 8.2: Visualization of the transformation from grid to chunks

The size of the chunks are important, as they need to be so big that geographic patterns and their developments such as edges of settlements can be learned by the model. At the same time their size should not get too big as the issues with memory and efficiency, mentioned above, will re-emerge. It is also worth noting that the countries tend not to fit the placeholders meaning we have to create null-cells, to keep the tensor-shape valid. This means that we have to add null-cells based on the remainder between chunk size and the input tensor's height and width as shown in equation 8.1.

$$null\_cells = chunk\_size - input\_tensor \ \ mod \ \ chunk\_size \tag{8.1}$$

An example is shown below, where chunk size is 16×16 and the input tensor's height and width is 31×27, which shows that there should be added one row and five columns of null-cells, making it a 32×32 tensor. The values carried by the null-cells are corresponding to the value that are expected to have the least appeal to people, meaning that they are filled up with water, far away from roads and has no population etc.

$$(16, 16) - (31, 27) \ \ mod \ \ (16, 16) = (1, 5)$$

Despite the irregularity created by the null-cells, the model should learn that people cannot live in a null-cell and thus not distribute people there, and in any case the final output gets clipped to its original width and height. Figure 8.3 below shows the concept of null-cells.

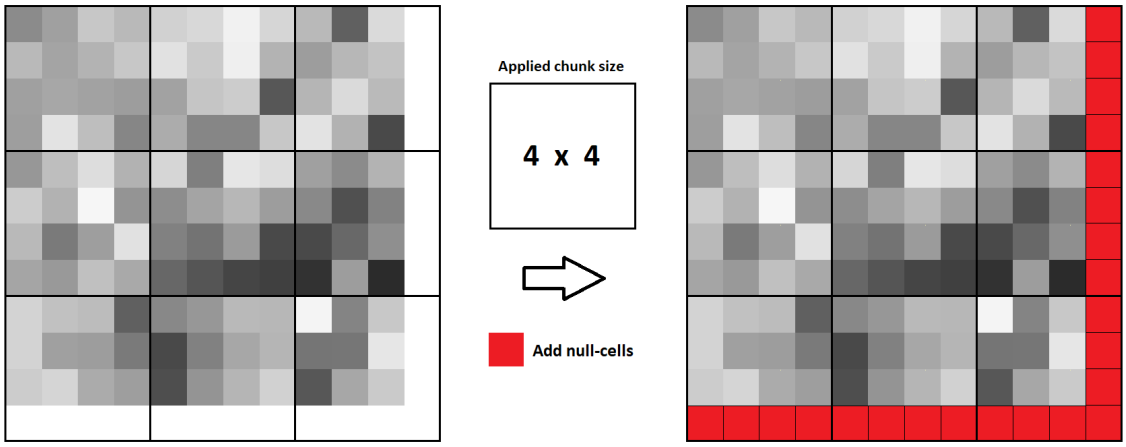

Figure 8.3: Addition of null-cells to make tensor valid

With the chunks and null-cells we have a data-structure that is similar to that of regular images, which as previously mentioned is known to work well with CNNs. Therefore, we can split the dataset randomly into a train- and test-set. The training data will directly impact the model's weights and biases and that way continuously improve it until convergence. After each epoch of training the model is tested on the test data, to check for overfitting, and importantly the results are not used to update weights and biases, to avoid the model being correlated to that data. Because the model is tested for every epoch it is possible to see the cost function's progress over time, thus determining if the training is going well. The data can also be divided into a third category, evaluation data. This data works like the test data, but where the test data can be used to adjust the hyperparameters such as number of epochs, learning rate and the architecture of the model itself, evaluation data is only used in the very end for a final evaluation. The concept of dividing the data into train, test and eval data is shown on figure 8.4.

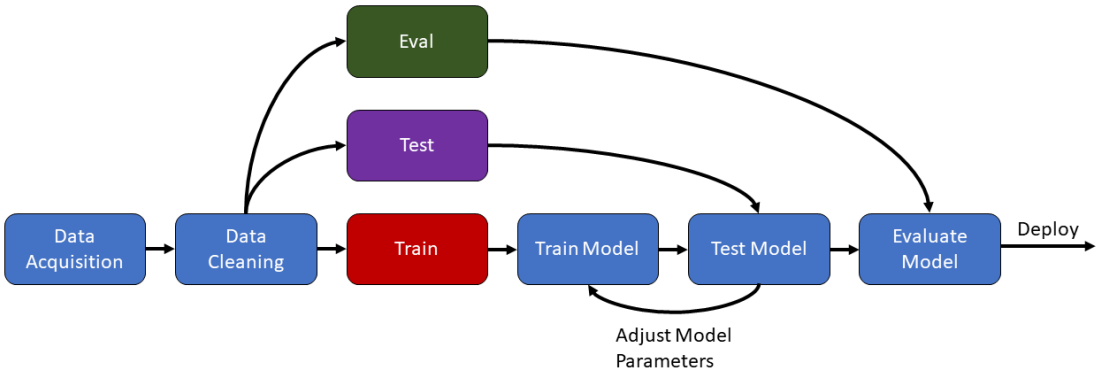

Figure 8.4: Flow-chart that shows the usage of train, test and evaluation data

The randomization happening in the split of the data is important as the model can then otherwise have problems converging. This will happen if the model is fed the same type of chunks over and over again, leading it to, wrongfully, believe that this type is the true value of all chunks [Bengio, 2012].

We have opted not to normalize all the data because we are interested in the actual population values and their development. It is hard to capture future development, when all data is between zero and one. In addition, there is no obvious way to obtain the original unit, which makes it harder to define thresholds based on population projection numbers from SSP. The disadvantage is that the training will take longer to converge because the initial values are so different, not all between zero and one, thus making it harder for the weights to adjust.

The training will be done chronological for each epoch so 1975 - 1990 is used first, then 1990 - 2000 and finally 2000 - 2015, where the last year works as label and the first as input. It could potentially also be viable to train on all of the dataset on equal terms, but we believe the most recent years should have the most impact on the model, thus training on them last makes sense. After dividing the data it is now ready to enter the neural network.

This finishes the answer for the last part of sub-question *How can the CNN setup for projecting future spatial population distribution be constructed and how can the used geospatial data be processed and prepared to work with it?*, that partly have been covered in the previous chapter as well as in this section. When data is prepared for a CNN like the one in this project it requires multiple processing steps, and can be quite an extensive task. This goes from handling big TIFF-files, spatial operations and merging rasters as initial actions to create a multiband raster. This raster needs further processing in the form of null-cells, randomization among others to match the neural network's placeholders and general criteria. Some of the preparation is essential for the neural network, while other are best-practice approaches to obtain good results in a machine learning context.

The next section will present the neural network architecture used and the reasoning behind it.

8.3 Neural Network Architecture

A supervised machine learning approach with CNN is well suited for finding patterns in a gridded geographic dataset like ours, as mentioned in chapter 6. Therefore the architecture is built up around this.

We are using the Rectified Linear Unit (ReLU), $f(x) = max(0, x)$, activation function to determine whether a neuron should fire or not. This is based on research showing that ReLU trains much faster than traditionally activation functions like Tanh and Sigmoid without suffering from accuracy issues [Krizhevsky et al., 2012, p. 3]. In this project we are also interested in values that are above one, ReLU can do exactly that.

Most studies using CNNs uses Cross-Entropy as their cost function, this includes the majority of the CNNs mentioned in chapter 6. However, Cross-Entropy is designed for categorical data in classification problems, and thus is not applicable for the problem we are trying to solve. Instead we need a cost function that can evaluate continuous data by measuring how far off it is from the correct continuous values. Mean Absolute Error (MAE), defined as equation 8.2 below, can do exactly this, and is our choice of cost function.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8.2}$$

The choice is based on the preservation of unit, in this case population numbers, making it easier to relate the error to the actual problem. This means an error of one, can be directly translated into an average deviation of one person per cell. Other regression functions like Root Mean Squared Error (RMSE) and Mean Squared Error punishes bigger errors as it squares the errors, which can be beneficial in some cases. While RMSE seems to also preserve unit Willmott and Matsuura [2005] argues that it is an inappropriate and ambiguous measure of average error. Their argument is that RMSE has three characteristics; average, power and square and thus is biased from the power and square components compared to MAE that only has average. This further establishes our choice of cost function. MAE keeps track of how well each cell matches up with the label cell, but we do also need to track and train on the overall population progress to implement the projection numbers from SSP2 (cf. chapter 4).

To do this we introduce a complimentary cost function that takes the difference between the predicted sum for a chunk and its label. The label is defined as, $pop_{proj}\frac{popchunk_{cur}}{pop_{cur}}$, where pop_{proj} is the total projected number for the population, pop_{cur} is the total current (training data year) population number and $popchunk_{cur}$ is the current population in the given chunk. Thereby we try to retain some similarity to the percentage of population that is within the chunk, while applying the projection number and guiding the development. The weight of each of the cost functions are set to 80 procent and 20 procent for MAE and the complimentary cost function respectively based on results from the testing (cf. section 8.4). The target value for the combined cost function is one, based on the criteria from the practical methodology, which might seem low. But in Denmark alone there are 2.6 million cells meaning this target error could at worst be off with 2.6 million persons, if all of the errors are either below or above the label value. The higher the population density of the

country or region the less this error means relatively.

We are using the Adaptive Moment Estimation (ADAM) optimizer to minimize our cost function. ADAM has advantages over the traditional Gradient Descents optimizers, as it progressively adapts and decreases the learning rate on the fly. This gives the option to use a larger learning rate in the beginning, thus allowing for fast convergence. This also means that we do not have to spend much time tuning the learning rate. ADAM, alongside most optimizers in machine learning, is using backpropagation to determine which weights and biases have to be adjusted to minimize the cost function [Kingma and Ba, 2014; Bengio, 2012; Walia, 2017].

We want the output tensor shape to be equal to the input's shape height and width, as the algorithm needs to predict the future population for the area represented by the input tensor. This means that we have to design the architecture so we get this shape. The output shape after going through a convolutional layer is given in equation 8.3, where the following parameters are included; filter size (F), input size (W), padding (P) and stride (S). Filter size is the tensor shape of the filter used for convolution and input size is the tensor shape of the input chunk. Padding is extra cells that are added around the input chunk to make sure that the output of the convolutions match the desired shape. The last parameter, stride, represents how many cells the filter moves on the input chunk. All the parameters are visualized on figure 8.5.

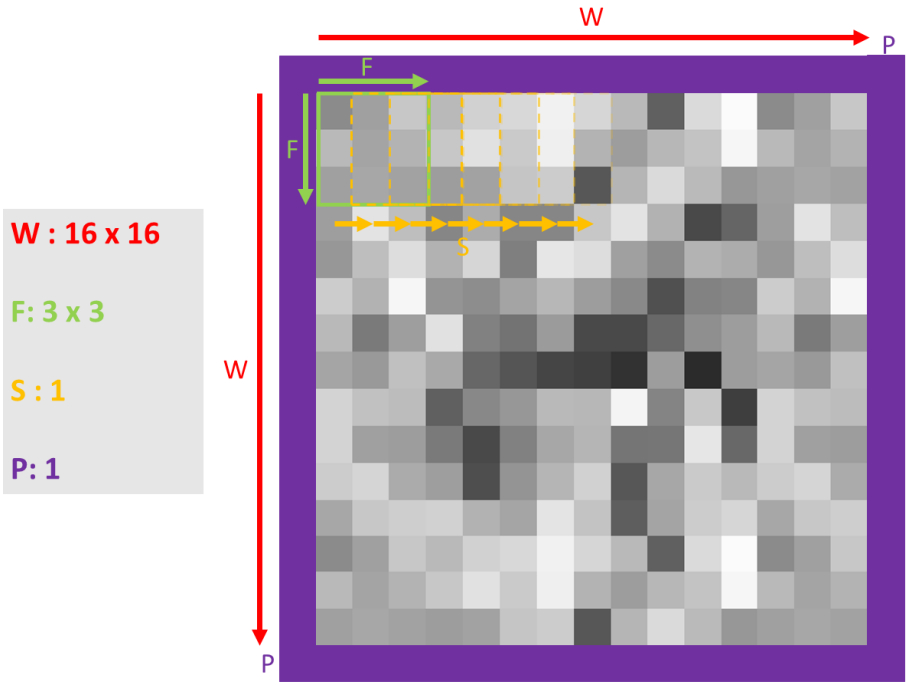

Figure 8.5: Illustration of convolutional layers - filter size (F), input size(W), padding (P) and stride (S)

$$output\_shape = \frac{W - F + 2P}{S} + 1 \tag{8.3}$$

For this equation to maintain the same shape we need to add a certain amount of padding

depending on the filter size and stride. Equation 8.4 is applicable for getting the same shape when stride is one. Maintaining the shape of the first three dimensions (batch size, height, width) is desired in this project because we want the output width and height to be equal to that of the input shape's. The padding used is symmetric, which prevents the model from predicting sharp edges between chunks.

$$padding = \frac{F-1}{2} \tag{8.4}$$

When multiple filters are used, each produces a data point in the last dimension. In our project this means a 4D-tensor input as [batch size, chunk height, chunk width, no. features] will get the shape [batch size, chunk height, chunk width, no. filters]. Cutting edge neural networks in image recognition are using downsampling techniques like pooling, larger strides and dilated convolutions. The downsampled results represent different features in an image whether that is hair, a nose or some other object and thus does not necessarily need all of the values from the original input image leading to a computational performance improvement [Springenberg et al., 2014]. In this project we will not use downsampling, as we are interested in population in all cells, and can therefore not afford to lose or generalize information in neighbouring cells. While we do not downsample the grid size, we are downsampling the number of features to just one, representing population.

The final architecture, named PopNet, illustrated in figure 8.6, consists of three consecutive convolutional layers each using a filter size of 3×3 and having 256 filters. This is based on the technique used by VGGnet [Simonyan and Zisserman, 2014a]. The idea behind the consecutive 3×3 convolutional layers is that they add up so that the second convolution of 3×3 would effectively cover an 5×5 area, as it is based on the results of an initial 3×3 convolution and so on [Simonyan and Zisserman, 2014a, pp. 2-3]. In addition to the convolutional layers, there are two dense layers, the first one having 512 neurons and the second one at the end connecting all the neurons into one output prediction.

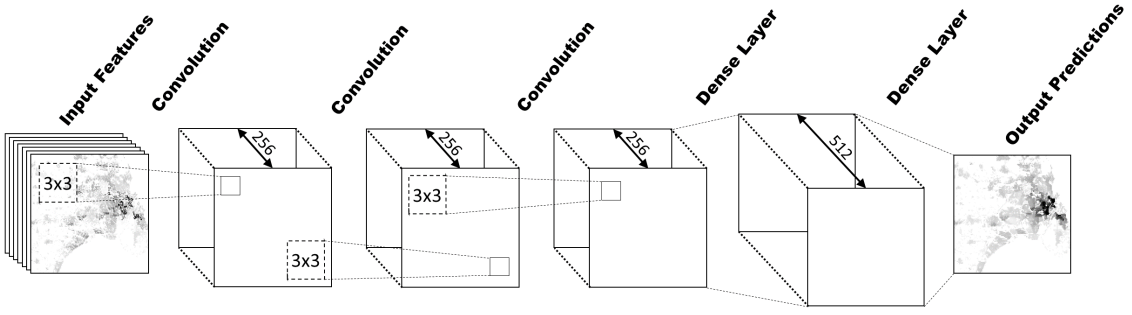

Figure 8.6: Cnn Structure

This architecture represents one way to set up a CNN for projecting spatial population distribution.

The architecture has continuously been adapted and improved through the testing covered in the following section.

8.4 Testing

The architecture has been adjusted based on a number of tests and incremental changes, based on the practical methodology in section 3.1, which has been used to understand the functions, input and processes in relation to the output. This gives an insight into how to affect the output toward a realistic prediction function. Tensorboard's graphs, log files and visualizations of the output are used to evaluate the changes.

To do this in a structured manner the different hyperparameters must be tested and assessed if not individually then methodological. There are multiple aspects that need to be tested and assessed, which cannot be done in a reasonable amount of time if they are tested for all possible scenarios. As an example we do not test the impact of changing the chunk size twice just because there has been changes to other hyperparameters or the architecture, unless we have a suspicion that it will have a different impact than earlier. As such, there are overall multiple aspects to changing and adjusting hyperparameters.

Some of the tests have shown obscurities in the trained model due to input layers' values. For example the default value of the distance to roads of water covered tiles was set to a value of one million meters. This resulted in the removal of population within entire chunks that had a lake in it, as shown in figure 8.7. While this is not realistic, it does show that the neural network is capable of restricting areas that it weights unsuitable for population. The solution to the issue was adjusting the distance to road value initially from one million to fifty thousand.

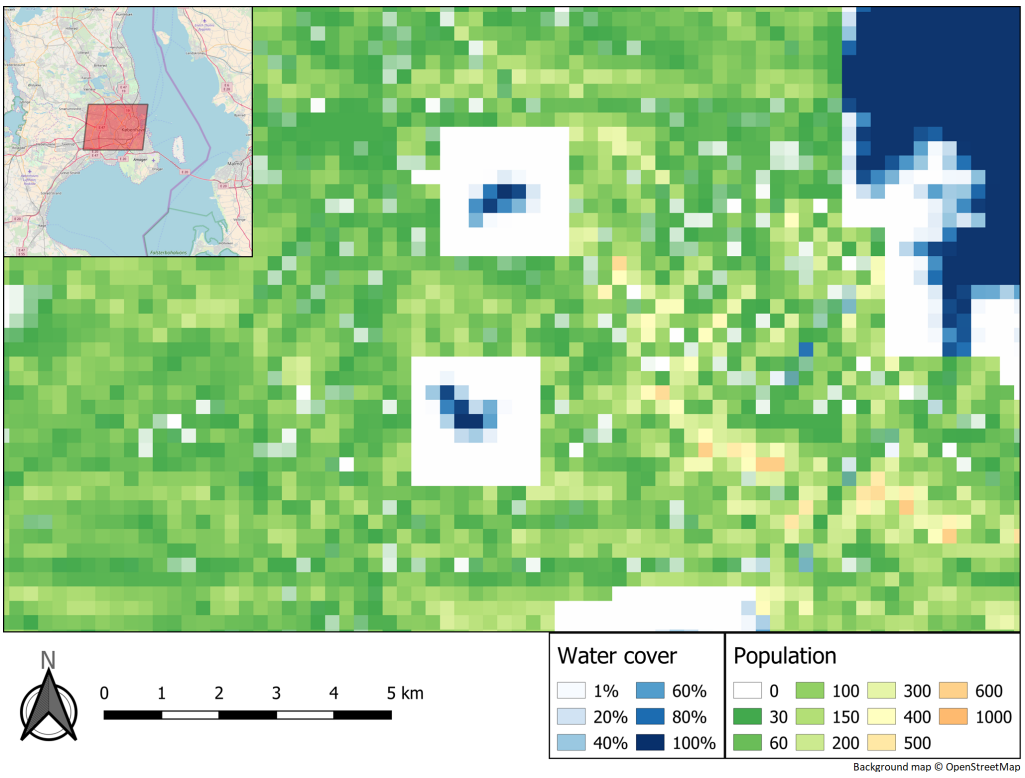

Figure 8.7: Two chunks in the middle of Copenhagen completely emptied for people due to lakes in them

Throughout the testing it also became apparent that the algorithm could not recognize unique patterns around larger cities, which we believed to be a result of lack of input features. This resulted in the algorithm having an upper bound of around 200 people per cell, which lead to a severe underestimation of the amount of people in larger Danish cities like Copenhagen and Aarhus, where there are cells that should have more than 3,000 people. So despite having a cost function that meets the target value and most cells are predicted correctly as shown from the histogram in figure 8.8, the model gives an unrealistic picture of the future spatial population distribution. Furthermore this model strengthens its own patterns from iteration to iteration as it aggressively spreads larger cities boundaries to meet the projected population number for the country in question. At the same time it captures the pattern of urbanization, leading to towns and villages being abandoned, however it is seemingly exaggerated making the prediction less realistic.

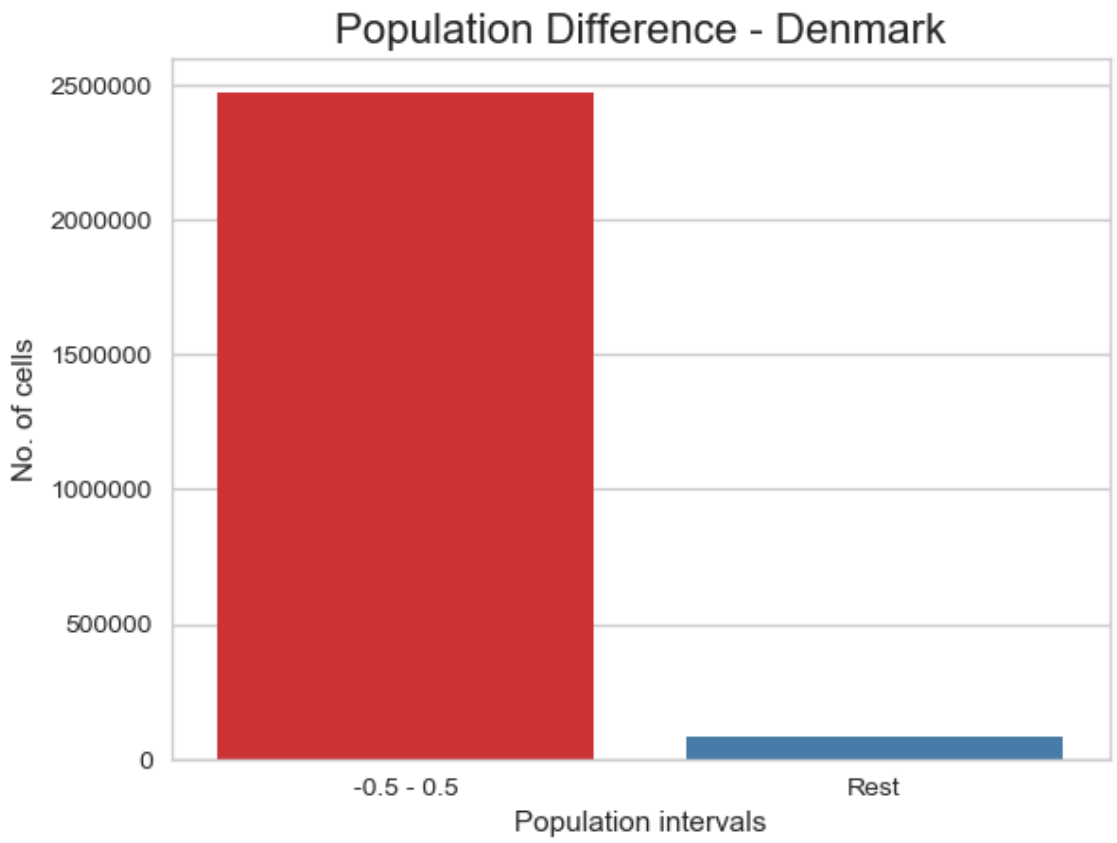

Figure 8.8: Population difference cells from 2015 to a predicted population distribution in 2020

For the model to be able to recognize the patterns in the cities we tried to feed it more data. For this the number of train stations within a distance from each cell has been used, as this should likely correlate with how large and dense a city is. Despite adding this data it did not help the underlying problem, which was caused be the architecture itself. The network architecture used Local Response Normalization (LRN) layers after each convolution as done in AlexNet [Krizhevsky et al., 2012]. The implementation of the LRN layers did improve the speed at which convergence occurred, but as the name suggest impacted the output values as these became normalized in the network, and thus

had lower maximum values than intended. This use of LRN layers was thereby removed from the architecture of PopNet.

We have also been trying to add administrative boundaries to the algorithm. The goal was to recognize patterns that divides urban and rural municipalities and through that improve the accuracy. However, categorical data needs to be embedded, in a one-hot encoding or similar, to make sense in a machine learning context. Because CNNs are mostly used for image-recognition Tensorflow is not supporting categorical data since it only makes sense if the data represents temporal or spatial data that relates to the image. Therefore, we have not tested this feature, but suspect that it could provide improvements to the final model.

Besides the data, the variables related to the neural network setup needs to be tested. An assessment is needed of the effects of the following hyperparameters and functions.

- Batch size
- Chunk size
- Learning rate
- Number of epochs
- Population projection
- Cost function
- Optimization function
- Activation function
- Depth and width

Changing chunk size has a big impact on the cost function as well as the nature of the output. Small chunk sizes from 8×8 to 32×32 lowers the cost function over time and leads to converging, but the size itself limits those chunks. This limitation manifests when the model allocates people to a populated chunk iteratively resulting in the chunk getting filled with population, as pictured in the timeline in figure 8.9. This issue stems from the fact that the model evaluates and predicts each chunk individually and thus cannot distribute people into neighbouring chunks. This issue could possibly be solved by use of overlap between chunks or by relating the given chunk to its surroundings, so it is aware of the pattern in neighbouring chunks and can adapt accordingly. These ideas are illustrated in figure 8.10 and 8.11, but has not been implemented because of its complexity and likely corresponding time requirements. Zhang et al. [2018] have recently used self-attention in a CNN, which is similar to the surroundings idea and has shown to do well in capturing long-range dependencies. This presents an actual way of taking cells that are not inside the immediate neighbourhood into account.

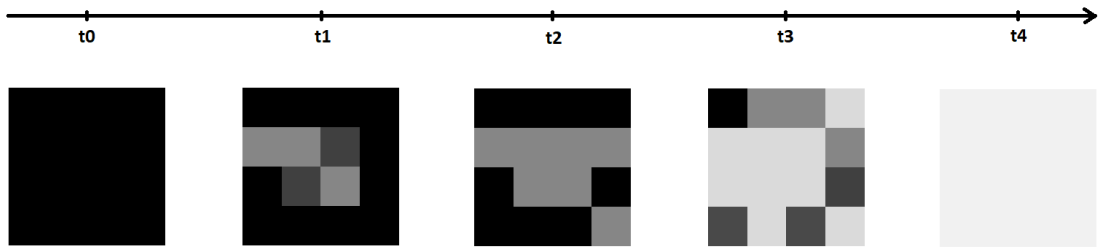

Figure 8.9: Chunk being filled by population symbolized by the white colour

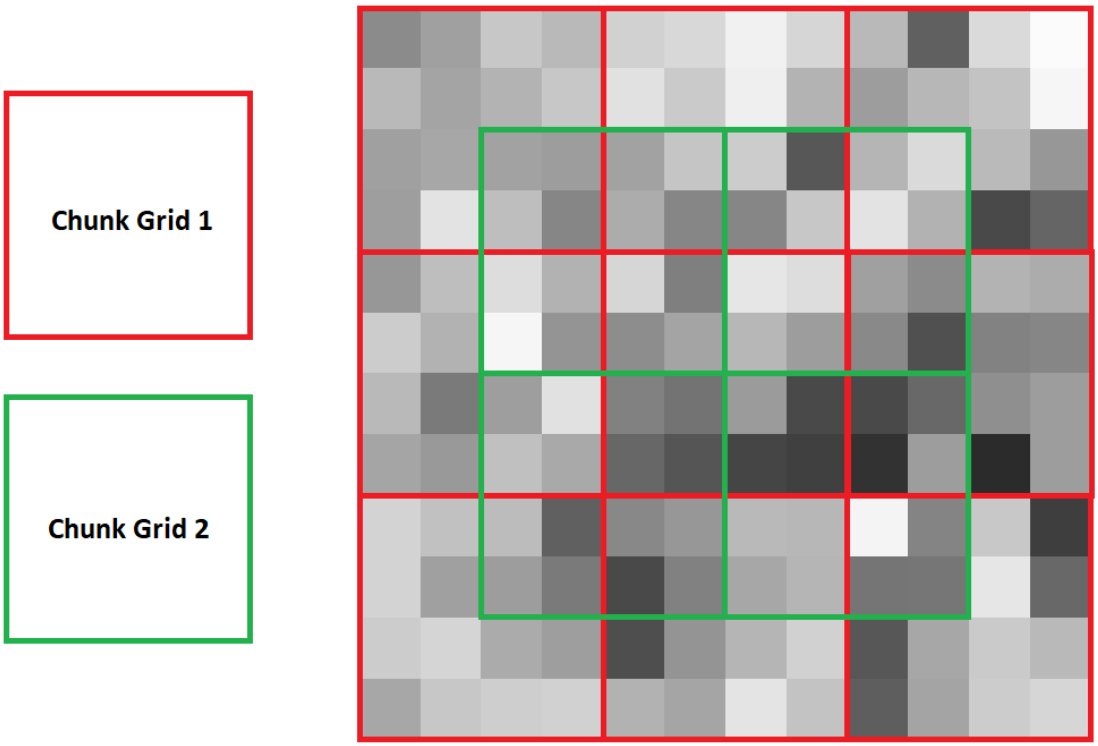

Figure 8.10: Chunks showing the overlapping idea

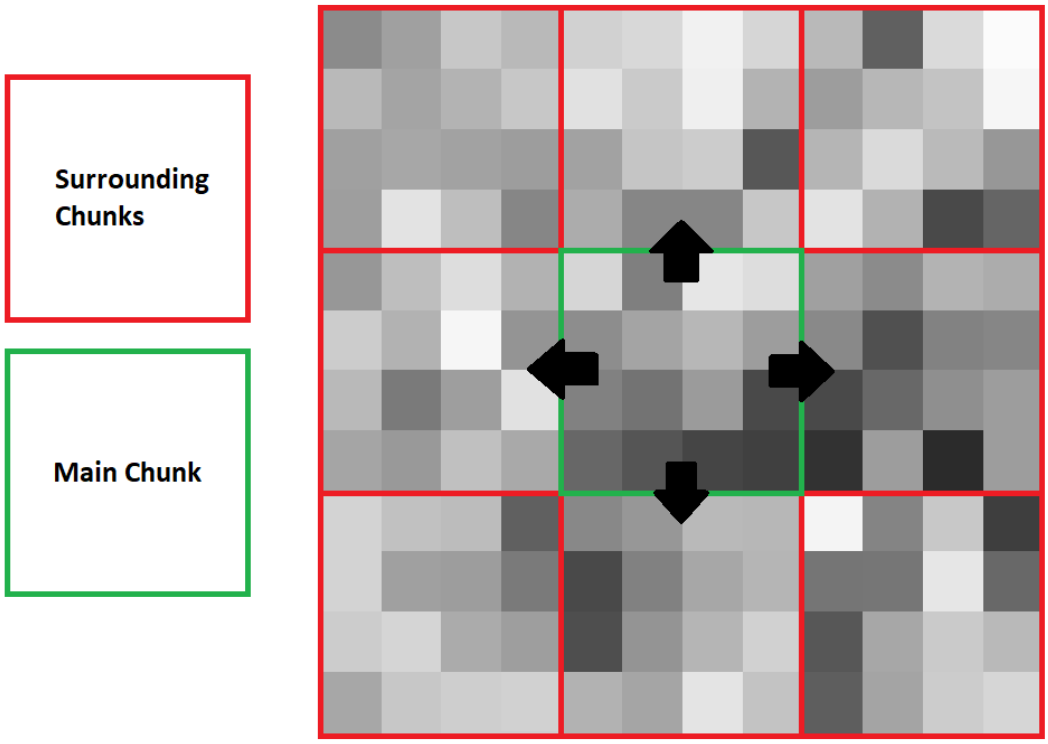

Figure 8.11: Chunks showing the surroundings idea

To lower the impact of the issue mentioned above, it is possible to create larger chunks from 64×64 to 128×128 as they will be filled up less frequently. However, those chunks tend to not improve as the training goes on. We believe this is a result of the bounding box of Denmark containing a lot of water, which drowns the other features, meaning that the algorithm has a hard time distinguishing small parts of a large chunk from each other, that should have a relative large impact. This results in almost all population cells being set to zero and despite how unrealistic this scenario is it does in fact produce a relative good cost function at around two because so many cells in the ocean do indeed have zero population. So choosing a chunk size at around 16×16 currently makes for the best results. In addition we are shifting the chunks, as shown in figure 8.12. This shifting is an addition of a random number of null cells to the top and left side of the array, between zero and half the chunk size. This removes or at least makes the chunk edges less visible, that was an issue caused by the chunk being filled up.

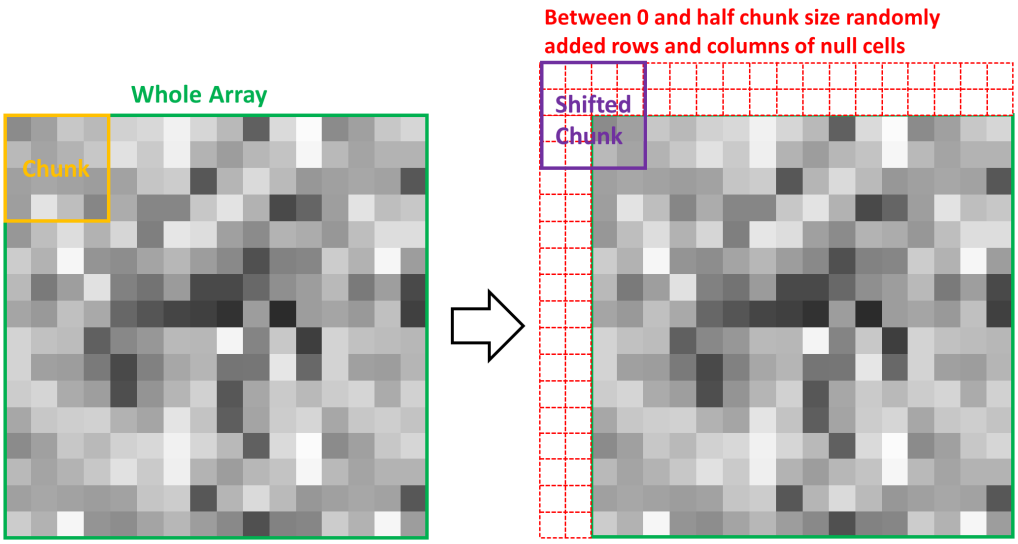

Figure 8.12: Illustration of chunk shift implementation

Changing the learning rate should not have a big impact, as the ADAM-optimizer is updating it along the way. It was however found during tests that the initial learning rate was essential to avoid the ADAM-optimizer to initially fail under certain circumstances due to what we suspect is testing all filter weights as 0 with good results, resulting in it finding a local minima it could not escape. This tendency can be seen visualized through Tensorboard in figure 8.13 of the training from where it can be seen that the cost function did not decrease beyond the initial improvement even after running 400,000 batches. Due to this we lowered the learning rate tenfold to 0.0001 which seemingly allows the optimizer to find different improvements and avoid getting stuck, allowing the ADAM-optimizer to then set an appropriate learning rate itself.

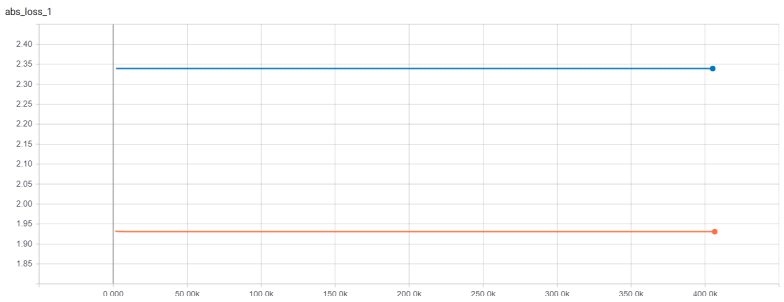

Figure 8.13: Cost function improvement over continuous training - The orange graph shows training- and the blue shows test progression

Testing other optimizers like gradient descent showed worse results than ADAM, and required much more fine-tuning of the learning rate to obtain a reasonable result.

Test of alternative activation functions to ReLU, Sigmoid and Tanh, also does not have any notable impact on the cost function value. Nonetheless Sigmoid and Tanh function does slightly affect the computation speed negatively, which was expected as papers have documented that ReLU reduces the likelihood of vanishing gradient [He et al., 2016].

The number of epochs required to run before convergence depends on the chunk size. Small chunks requires fewer epochs to obtain good results, around 40 to 200 epochs depending on country size, while large chunks improves slower. However, we have not run the model long enough to observe an actual converging, as it keeps improving, even though it diminishes over time.

The batch size determines how often the neural network's weights are updated. We have observed that the batch size generally does not impact the result. However, Keskar et al. [2016] have shown that large batch sizes performs worse which also relates to the point made about running the whole dataset at once in section 8.2. Therefore, we are using batch sizes of 16 or below depending on the chosen chunk size.

The cost function evaluates on the chunks as well as the cells, as described in section 8.3. MAE, that is used for evaluating the cells have the biggest impact on reducing the cost function. This has led to better results, when the cost function that evaluates the individual cell population change, have a higher weight and the chunk evaluation a comparatively lower weight. The result of this is that the weighting in the final model is distributed 80/20 percent to individual evaluation and chunk evaluation respectively.

Depth and width of a CNN refers to the number of convolutional layers, depth, and the number of filters applied within each layer, width.

As were covered previously in section 8.3 our current applied neural network consists of three individual convolutional layers consisting of 256, 256 and 256 filters respectively.

Regarding the number of convolutional layers chosen for the CNN, multiple things were revealed by testing. In theory, more convolutional layers and a larger number of filters can more precisely recognize and identify unique patterns as the amount, and the complexity of both, increases with a higher number (cf. chapter 6). This is supported by the findings and trends of CNN research such as Goodfellow et al. [2013] which found that increased CNN depth improved performance in street number recognition. The trend also seems clear when looking at the CNN contestants of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) over the years where the number of convolutional layers applied tend to increase from year to year.

If you compare the network created for the purpose of this report to well known CNNs such as AlexNet (2012) or VGG Net (2014), one can see that both the number of filters and convolutional layers are higher. AlexNet applies five convolutional layers with between 48 and 192 filters [Krizhevsky et al., 2012]. In comparison VGG Net has a more special structure applying only 3×3 filter dimensions and applied up to 16 convolutional layers with between 64 and 512 filters [Simonyan and Zisserman, 2014a]

The subject of convolutional depth and width however also relates heavily to the question of hardware requirements, and in the scope of this report, limits. While a larger and more complex network is in essence better the computing requirements do increase substantially which can be seen again from AlexNet or VGG Net as they before their appliance in the ILSVRC were trained respectively by two GTX580 GPU's for five to six days and four Nvidia Titan Black GPU's for two to three weeks [Krizhevsky et al., 2012] [Simonyan and Zisserman, 2014a]. This of course cannot be compared one-to-one with what we test and

strive to achieve, but still serves as a clear example of the possible computing resources that can be applied.

With these considerations factored into our tests the goal was to achieve the highest number of convolutions within the scope of available computational resources. During the tests it became clear that without applying pooling layers, which we are unable to do due to the needed output as mentioned in section 8.3, the increase from three convolutional layers to four, made the neural network's ability to converge too slow with the given resources. An example of these tests is visualized from Tensorboard in figure 8.14, showing a network with three convolutional layers. Here the network starts to improve and converge after only a few epochs. While the same applies to the network with four likewise convolutions, each chunk and epoch takes increased time to train equalling a decrease from 36 batches per second on three convolutional layers, to 26 batches per second with four convolutional layers.

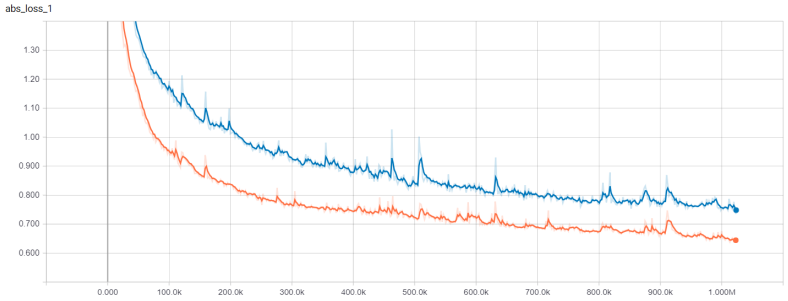

Figure 8.14: Cost function improvement over continuous training with three convolutional layers - The orange graph shows training- and the blue shows test progression

As a result of this, the number of three convolutional layers were chosen even though a deeper network could possibly support better pattern recognition and therefore more precise predictions of spatial population distribution.

In terms of network width, the number of filters applied by the convolutional layers, there seems to be an agreement that a wider network generally improves accuracy to a certain degree. Tests done on CaffeNet, which is a copy implementation of AlexNet in terms of setup, pointed toward that having a narrow network leads to a significant decrease to accuracy. Continuously increasing width however, only improves accuracy to a certain point, at which it starts to overfit [Mishkin, 2016]. The same was pointed out again by Mishkin et al. [2016] who found that network widths contribution gets increasingly lower as can be seen from their tests on CaffeNet shown in figure 8.15, that shows saturation at about three times the original width.

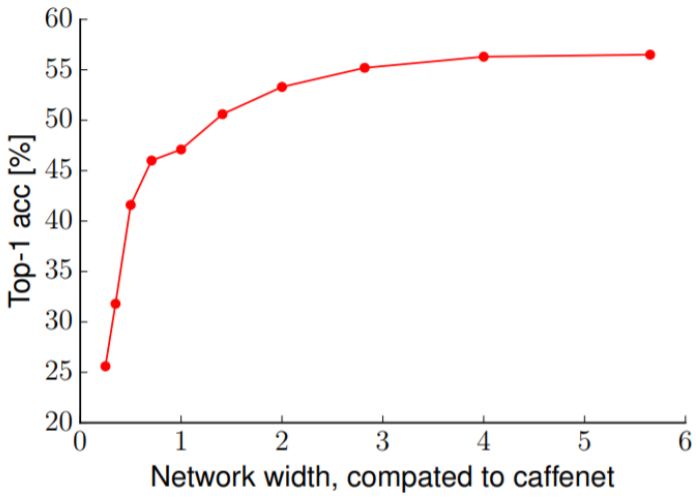

Figure 8.15: CaffeNet image classification performance with different width settings [Mishkin et al., 2016, p. 13].

What these tests highlight is that network width can have an impact on a larger CNN such as AlexNet/CaffeNet, especially if the network width is lower than ideal. There is however no way of calculating or knowing what the right settings to apply to a given CNN is, outside of testing it extensively. Increasing the width of the network also increases he computational requirements and thereby the time it takes to train.

In relation to tests done on our CNN, with the purpose of predicting population distribution, results of changing the width does also have an impact. It is however harder to test as a prediction cannot be immediately verified as true or false nor as a result of an improved cost function.

To test the effect of the width of our network a number of tests have been done with different settings. These were done on the version of PopNet that still utilized the LRN and as such the exemplified results should be seen in this perspective.

One example is a model trained on three convolutions of 128, 128 and 256 respectively, this has an improving cost function over time. The output value sum that should equal population were always between 4.9 and 5.5 million fpr every iteration from the year 2020 to 2100. While the projected numbers haven't been precisely hit in any test, this result is millions off from the projected increase from 6.8 to 7.4 million. The number however is only part of the problem as based on inspection clear irregularities seem clear.in Comparison the model trained on three convolutions of 64, 64 and 64 act differently. This model also have issues with predicting the correct amount of population and has an increasing amount starting at 6.4 million and ending at close to 12 million. This is thereby also far from the current or projected population of Denmark. it does however predict city growth in a pattern that fits the expectation of growing big cities due to factors such as being trained on data showing historical urbanization.

Looking at the tests done on that version of PopNet in relation to filter width, there is as exemplified through above examples, no clear pattern of improvement and change

seems sporadic. This is likely because other hyperparameters are more essential and that the performance of filter width are hard to evaluate when we use the CNN as we do. Because of this we rely on the theoretical and practical knowledge that numbers between 128 to multiple thousands are likely a good choice [Mishkin, 2016]. This is balanced with a consideration for the increased computational requirements that follow a wider network and a middle way between them was chosen. The result of this is a network of three convolutional layers all consisting of 256 filters.

Chapter 7 and 8 thereby answers the fourth sub-research question; *How can the CNN setup for projecting future spatial population distribution be constructed and how can the used geospatial data be processed and prepared to work with it?*. Results and experiments produced with PopNet for Denmark and France will be analysed, evaluated and discussed in the next part.

Part IV

Results

Evaluation 9

This chapter contains the results produced by the final models trained on the neural network architecture and the corresponding evaluation of those results. This specifically relates to the sub-research question *How well does the model predict future population distribution?* and partially to *What knowledge can be gained from the results, experiments and tests?*.

The evaluation is based on predictions made by PopNet with the settings previously covered in section 8.3 and 8.4. The CNN models used are one made for Denmark and one made for France, these are introduced, described and evaluated concurrently. This means that both will be initially introduced and then evaluated based on different methods that gives insight into the results and model.

A loose methodological approach is thereby used for the evaluation. This is chosen because there are no previously created overall methods for evaluating the workings and results that we would like to achieve. The findings and results are then discussed further in the discussion chapter.

The training parameters for Denmark and France are listed in table 9.1 and 9.2. The learning rate, batch and chunk size are chosen based on the tests described in the previous section. The French model has been trained significantly longer, 17 hours, compared to the Danish model, 9 hours. When examining the cost function for the two models it is apparent that the Danish model obtains a lower value for the cost function. We believe that this is because Denmark has many water cells that are relatively easy to predict for the neural network, and thus lowers the cost function artificially. This means that based on the cost function value, we cannot conclude that one model is better than the other, let alone conclude whether the models are realistic. However, the cost function does indicate that the results are somewhat good as they are close to the target value of one, but this will be investigated later in the evaluation. Another interesting observation is that the French test has achieved a lower cost function than the training, this is unexpected as the neural network has not adjusted weights and biases based on this data. But we believe this is because of the chunks that have been randomly put into the test dataset coincidently have been easier to predict similar to the water cells in Denmark. Otherwise this just indicates that the model is not overfitted to the training data.

Danish Model	Learning rate	Chunk size	Batch size	No. batches	No. epochs	Time	Cost function
Train	0.0001	16×16	16	218,000 (70 %)	115	09h00m	0.78
Test				93,000 (30 %)			0.94

Table 9.1: Training and testing of the Danish Model

French Model	Learning rate	Chunk size	Batch size	No. batches	No. epochs	Time	Cost function
Train	0.0001	16×16	16	455,000 (70 %)	44	17h15m	1.35
Test				195,000 (30 %)			1.23

Table 9.2: Training and testing of the French Model

To evaluate the final trained model for Denmark and France, a test is conducted on a period of time for which we have available data on what change have occurred, and how the historical population distribution is. To do this, the CNN is set to predict the spatial population distribution change from the year 2000 to 2015 for both countries. The results will then be evaluated against the historical 2015 datasets for Denmark and France. The model is thereby trying to predict data that has been used for training which is not ideal, but as historical data is sparse, using a period outside of training is simply not reasonable. The effect of this is however negligible as the model is equally trained on the change between the four periods of time, and thereby cannot simply recreate the progress between the chosen periods. This can thereby indicate how well the trained models perform, when trying to predict the future distribution. In the following evaluation for Denmark and France, the population and mean absolute error in tables 9.3 and 9.4 is based on all cells and their corresponding values. The remaining presented results of the historical comparison, only takes cells with a values that are greater than or equal to one into account. The reason for this is that by excluding the large amount of cells with values below one, such as water cells and large empty stretches of land, the difference in the populated areas that are hard for the model to predict, become more clear.

The results for Denmark can be seen in table 9.3 and figures 9.1 and 9.2. When comparing the prediction to the historical 2015 data, it is evident that the model is slightly off in its ability to predict the total population as it predicts 103,856 people less than the historical population total from 2015. This could among other things be due to the model not being trained enough and it might benefit from longer training. The MAE of 0.9, which resembles the value of the cost function, indicates that the prediction in general on average is off by 0.9 per cell. The mean and standard deviation of the difference between the prediction and historical data is negative 3.73 and positive 42.37 people respectively. The negative mean indicates that the model in general is underpredicting by a small fraction and the data varies around the mean with an average of 42.47, as seen in table 9.3.

Denmark	Predicted Population	Difference in Population	Mean Absolute Error	Mean Population Difference	Standard Deviation of Difference
Prediction	5,565,141	103,856	0.9	-3.73	42.37
Historical	5,668,997				

Table 9.3: Denmark - population difference between predicted 2015 and historical 2015

Figure 9.1 shows a comparison between the predicted values and the historical values in a cumulative histogram plot. The histogram plot shows the number of cells per population value and is divided into four subplots with population per cell; 0 - 100, 100 - 250, 250 - 500 and 500 - 4000. These show that the Danish model underestimates the number of cells in the 0 - 15 population per cell range, but catches up to the historical data and predicts very well in the low population areas with around 15 - 80 population per cell. However, it starts to underestimate again at around 80 population per cell, which gets worse in the

range from 100 - 300 population per cell indicated by the blue colour being above the purple overlap. From 300 and onwards it catches up to the historical data again, which indicates that the model is actually overestimating. This is also pointed out by the red colour rising above the purple in the 800 - 2000 population per cell range. In general the model performs well in its estimation and the difference in the distribution is only around 1000 cells at its largest difference.

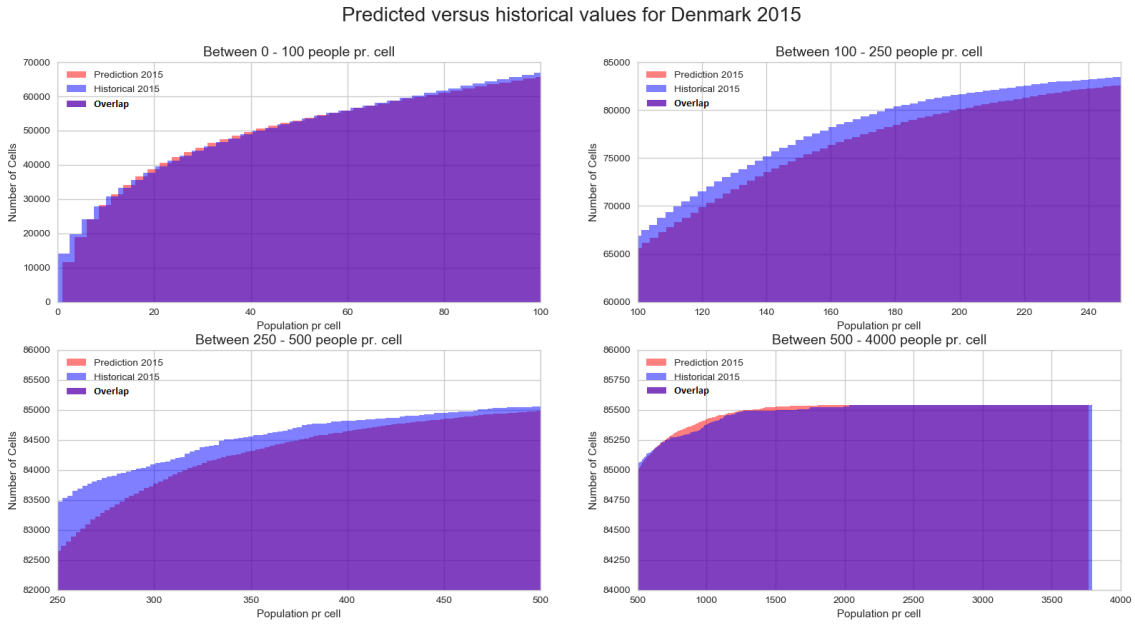

Figure 9.1: Denmark - Cumulative comparison

However, the cumulative only says something about the overall distribution. It does not say how well the trained model can predict the cell values spatially, as it only gives an overall picture of the relationship between the number of cells and their values. Figure 9.2 exemplifies how well the model predicts spatially by illustrating, how much it is over- or under predicting the value in a given cell in four major cities in Denmark. The map shows that the model both over- and underestimates the values compared to the historical 2015 data. In lesser populated areas outside of the major cities it predicts the values quite well, where the cells tend to be gray. However, the majority of cells in the cities tend to be green, meaning that the prediction is overestimating in these cells, which corresponds well with the findings from the previous plots. Showing that the model has a tendency to over predict in more populated areas.

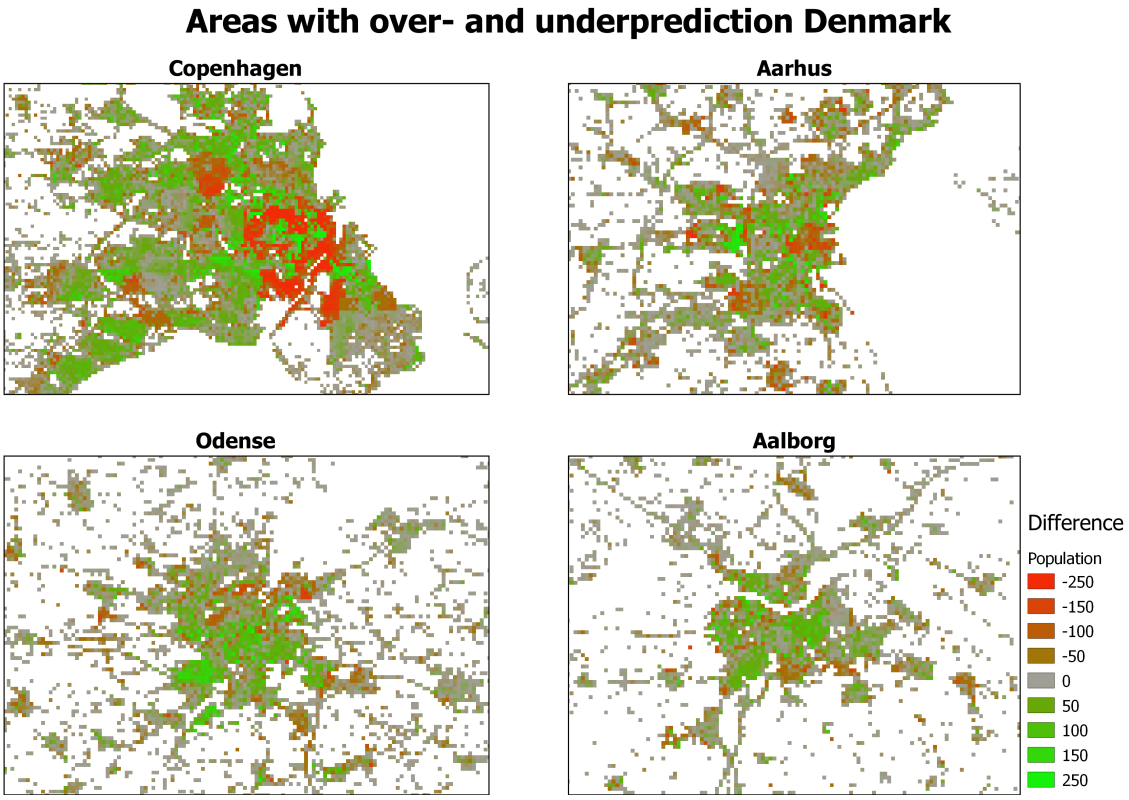

Figure 9.2: Areas in major cities with over- and underprediction

The comparison between the prediction values and the historical data for France in 2015 shows that the model is predicting the total population better, and is off by 30,732 people, as shown in table 9.4. The mean absolute error of 1.08, which resembles the value from the cost function, indicates that the prediction in general on average is off by 1.08 per cell. The mean and standard deviation of the difference between the prediction and historical data, shows that the mean is negative 2,54. This means that the model in general is underestimating by a small fraction and the standard deviation is slightly smaller at 29.81, indicating that the data values have less variance than Denmark.

France	Predicted Population	Difference in Population	Mean Absolute Error	Mean Population Difference	Standard Deviation of Difference
Prediction	64,364,168	30,732	1.08	-2.54	29.81
Historical	64,394,900				

Table 9.4: France - population difference between predicted 2015 and historical 2015

Examining the data shown in figure 9.3 also indicates that the model for France is underestimating slightly in the lower values between 0 and 15 population per cell, but is estimating very well in the 15 - 110 population per cell range. It then starts to underestimate, indicated by the blue colour in figure 9.3 above the purple overlap becoming thicker. It starts to overestimate again in the 350 - 1000 population per cell range, where the blue color becomes thinner, but does not quite catch up to the historical distribution. In general the model performs quite well in its estimation of the distribution per cell value, where the difference in distribution is only around 6500 cells at its largest difference.

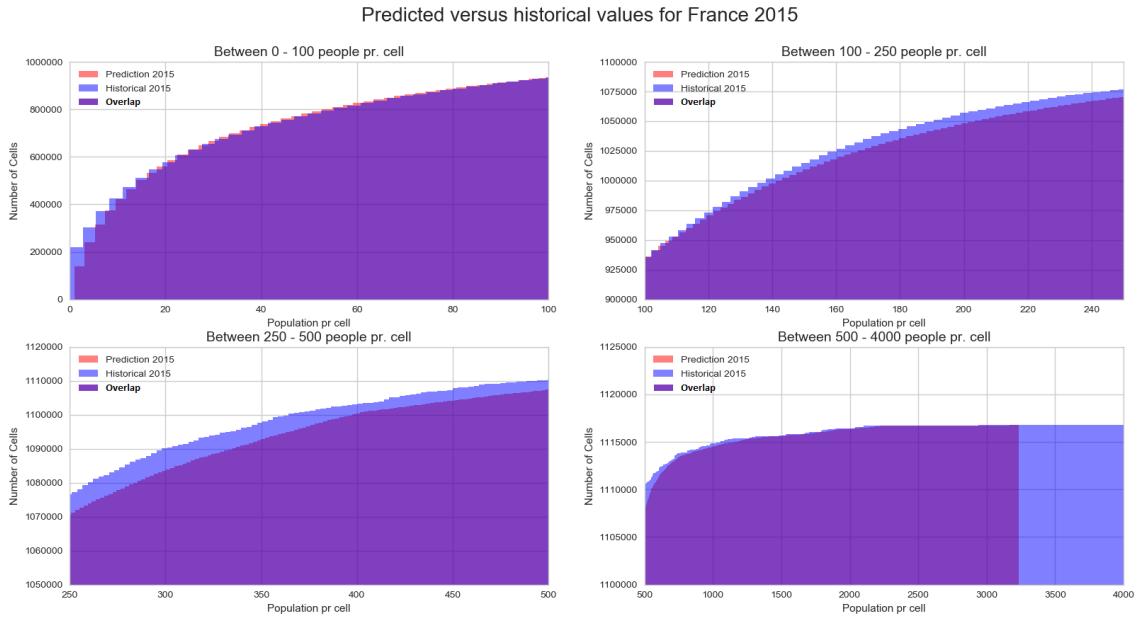

Figure 9.3: France - Cumulative comparison

Examining the distribution for four major French cities in figure 9.4 it is clear, that the French model seem to not over- or underpredict as much as the Danish model in cities, indicated by more cells being gray. It is, however, still moving towards the green, meaning that it over predicts the cell values a bit compared to the historical 2015 data in more populated areas.

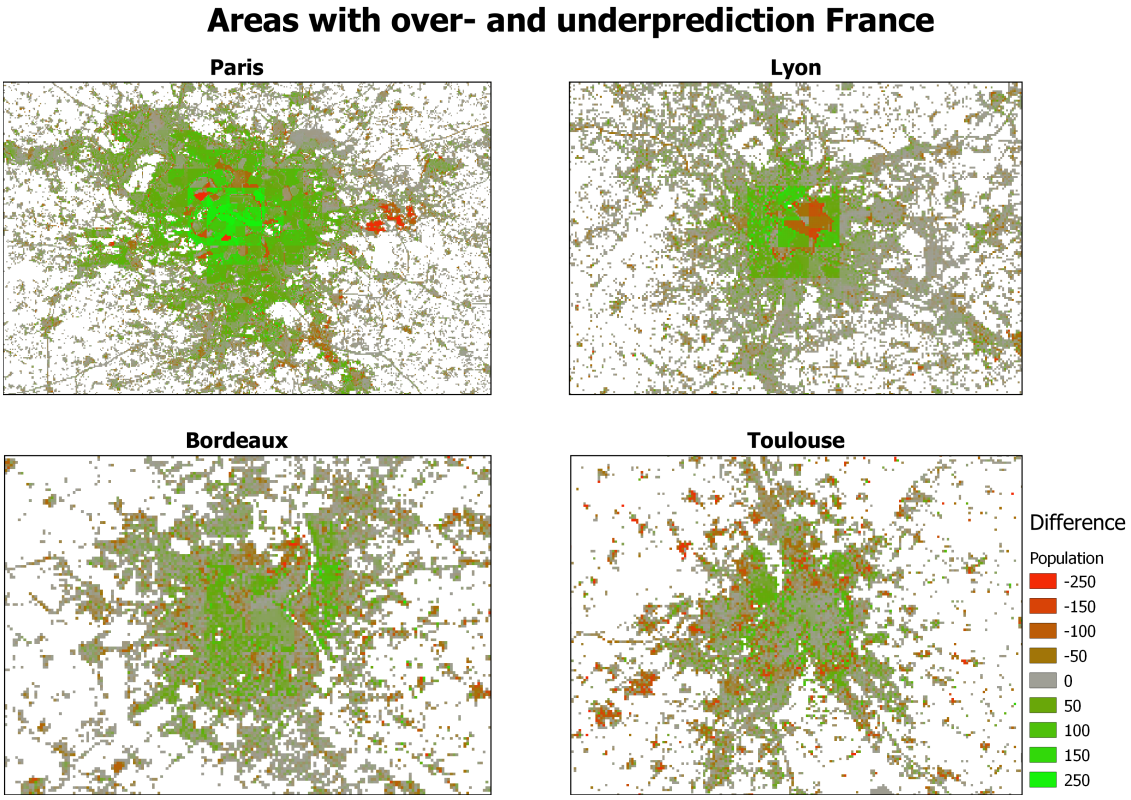

Figure 9.4: Areas in major cities with over- and underprediction

Looking ahead and into the future predictions two scenarios have been run on PopNet. One for France and one for Denmark, on the period from 2020 to 2100 in a ten year interval based on the IIASA SSP2 population scenarios (cf. section 4). These will be evaluated to assess the CNNs ability to predict long term spatial population distribution and uncover insights into strengths and weaknesses, as well as getting a thorough assessment of the way the model works. This could be both on country scale trends, related to interesting local or city wide developments, or other changes that warrant further investigation. The immediate result of the scenarios are nine country wide TIFF images for both Denmark and France, where each iteration is aimed to achieve a population number close to that of the SSP2 scenario for the respective year. These predictions for the future population distribution and their numbers for Denmark and France, can be seen from table 9.5 and table 9.6 respectively.

While they do not hit the precise numbers of the projection made by IIASA for the SSP2 they are all within ±0.3 percent of the value predicted for the year in question. These results were achieved by adjusting the input population value in the architecture, for each year until the output fit. This was done as the population value input is not fully understood by the model as the maximum population value when run, but does have an impact that control the output. As such the input population value in the architecture was not actual SSP2 values for each period but the output achieved was. Getting values closer to the SSP2 scenario could likely have been achieved, but as the results were within ±0.3 percent they were deemed precise enough. This is because the uncertainty cannot be eliminated completely anyway, as the model is based on projected numbers and trained on data that already provide a certain degree of uncertainty (cf. chapter 4).

Year	2020	2030	2040	2050	2060	2070	2080	2090	2100
SSP2 population	5,806,000	6,087,000	6,338,000	6,574,000	6,825,000	7,047,000	7,225,000	7,354,000	7,426,000
PopNet population	5,831,093	6,064,529	6,366,482	6,580,994	6,829,117	7,036,016	7,227,473	7,372,575	7,449,710
Difference	-25,093	22,471	-28,482	-6,994	-4,117	10,984	-2,473	-18,575	-23,710
Percentage difference	-0.43%	0.37%	-0.45%	-0.11%	-0.06%	0.16%	-0.03%	-0.25%	-0.32%
Avg. difference	15,878								

Table 9.5: Denmark - population projection overview

Year	2020	2030	2040	2050	2060	2070	2080	2090	2100
SSP2 population	66,609,124	70,324,320	73,707,429	76,503,876	78,866,247	80,865,224	82,440,492	83,226,046	82,969,660
PopNet population	66,567,282	70,273,243	73,784,940	76,680,008	78,818,427	80,687,394	82,652,028	83,510,848	82,767,475
Difference	-41,842	-51,077	77,511	176,132	-47,820	-177,830	211,536	284,802	-202,185
Percentage difference	-0.06%	-0.07%	0.11%	0.23%	-0.06%	-0.22%	0.26%	0.34%	-0.24%
Avg. difference	141,193								

Table 9.6: France - population projection overview

When looking through the individual years some differences between the two trained models can be observed, this can be seen from table 9.7 and 9.8. The Danish model and prediction is more volatile and varies greater in terms of lowest and highest values within each 10 year period where the prediction for France seems to be more stable. Where the maximum value of France stays around 3100 to 3700 over the 80 year time span the maximum values observed on the results from Denmark vary 4300 over 8700 and ending back at 5700 in the last iteration for the year 2100. The same tendency can be seen in the minimum values where the France predictions are stable around -1 and the predictions for Denmark vary from -30 to -81. The number of cells with a negative value larger than -1 is very low and seem to disappear or change place from iteration to iteration which suggests

that the occurrence of negative cells is likely a by-product of the chunk cost function the model have learned to use as a way to balance the population sum.

Denmark									
Year	2020	2030	2040	2050	2060	2070	2080	2090	2100
Min. population value:	-30	-44	-70	-65	-41	-43	-46	-81	-70
Max. population value:	4352	4767	5367	6568	7575	8687	7360	6455	5707

Table 9.7: Denmark - Minimum and maximum population values

France									
Year	2020	2030	2040	2050	2060	2070	2080	2090	2100
Min. population value:	-1	-1	-1	-1	-1	-1	-2	0	-1
Max. population value:	3124	3333	3487	3578	3644	3674	3698	3674	3580

Table 9.8: France - Minimum and maximum population values

9.1 Country Evaluation

Getting close to the target values that are provided by the SSP2 projections and closely mimicking historical development is a good first indication that the model can make somewhat realistic predictions. However, we need to analyse the predictions more closely to determine how people are distributed and evaluate whether that distribution is reasonable. Examining the development from 2015 to 2100 for Denmark, that is shown in figure 9.5, gives insight on how the distribution of the cells have changed. Cells with zero population are not shown on the figure as they distort the visualization.

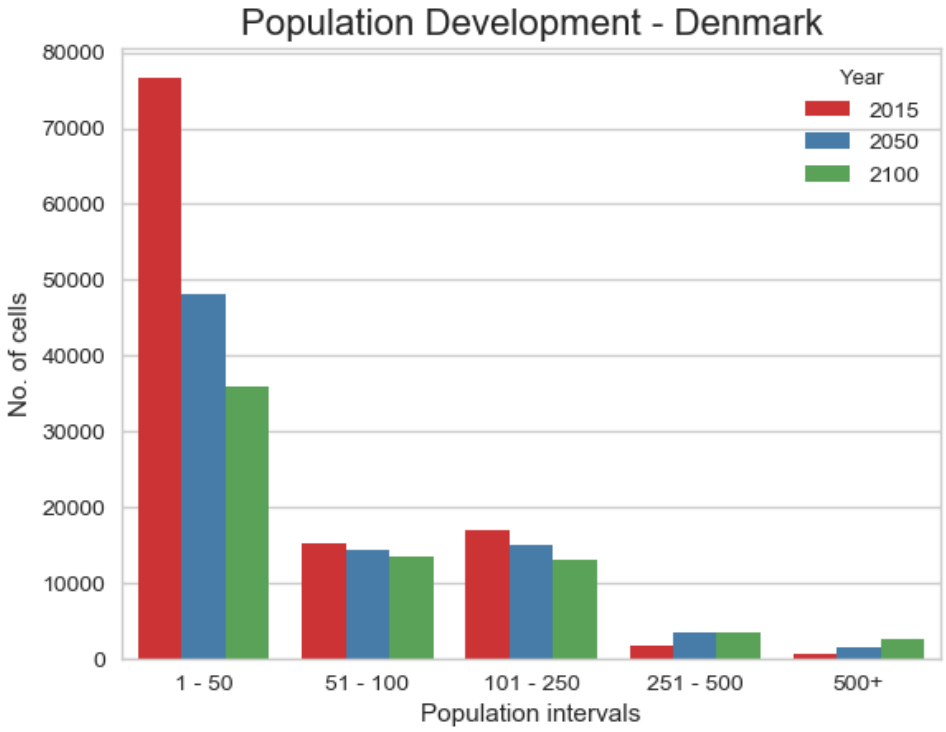

Figure 9.5: Population development in Denmark from 2015 to 2100

From the loss of cells in the 1 - 50 population category it is apparent, that the model has a strong tendency to depopulate rural cells. This trend can also be seen in the categories 51 - 100 and 101 - 250, however not quite as strong. This tells us that the model has picked up on the urbanization pattern in Denmark from the historic data. Especially larger cities become more dense, while also expanding the spatial extents as can be seen from the increase of cells in the two last categories.

France has some of the same tendencies as can be seen from figure 9.6. Nonetheless, the urbanization is not nearly as strong and while Denmark loses cells in the categories 51 - 100 and 101 - 250 France actually gains cells. This is interesting, as it shows us that the neural network does in fact evolve differently from country to country even when build with the same architecture.

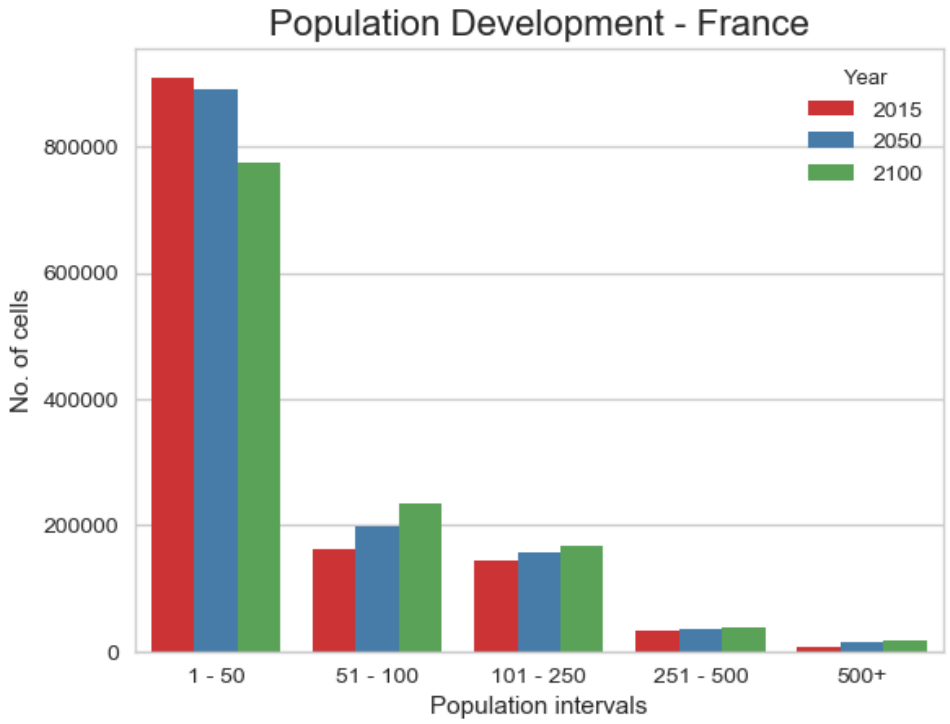

Figure 9.6: Population development in France from 2015 to 2100

Figure 9.7 and 9.8 stresses the fact that it is indeed the large cities that accounts for most of the population increase, as the latitudes representing Copenhagen, Paris and other cities spikes the most from 2015 to 2100. In addition the maximum population value of a given cell increases over the years, as given by table 9.7 and 9.8, which further indicates that the cities are predicted to become more dense.

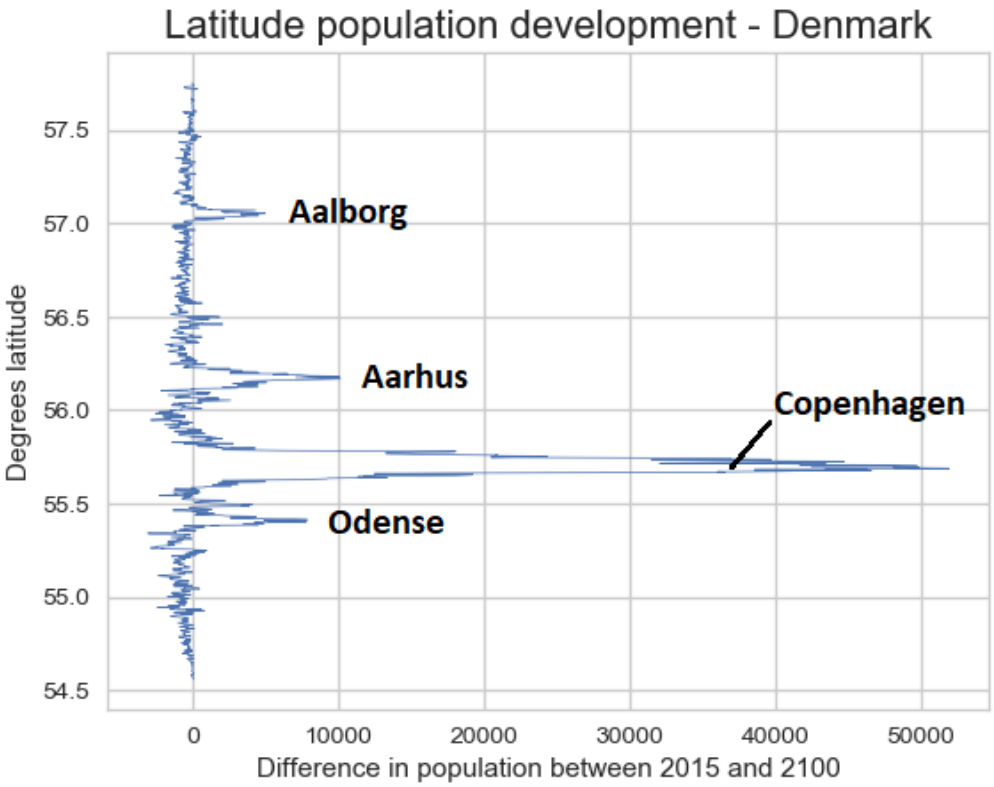

Figure 9.7: Latitude population development in Denmark from 2015 to 2100

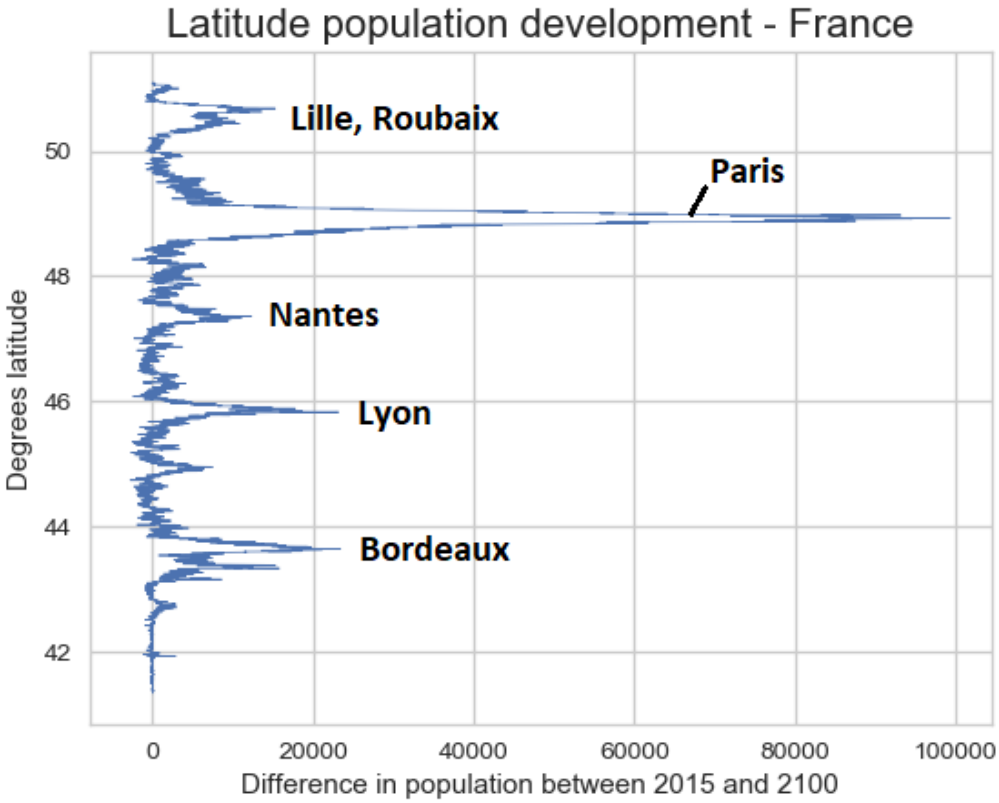

Figure 9.8: Latitude population development in France from 2015 to 2100

In addition, longitude also shows the same trend as shown on figure 9.7. However, this also appears to show a slight shifting in the population towards west from 2015 to 2100. This is especially clear when looking at the zoomed in subplot. This issue will be investigated further in the next section that analyse the local trends.

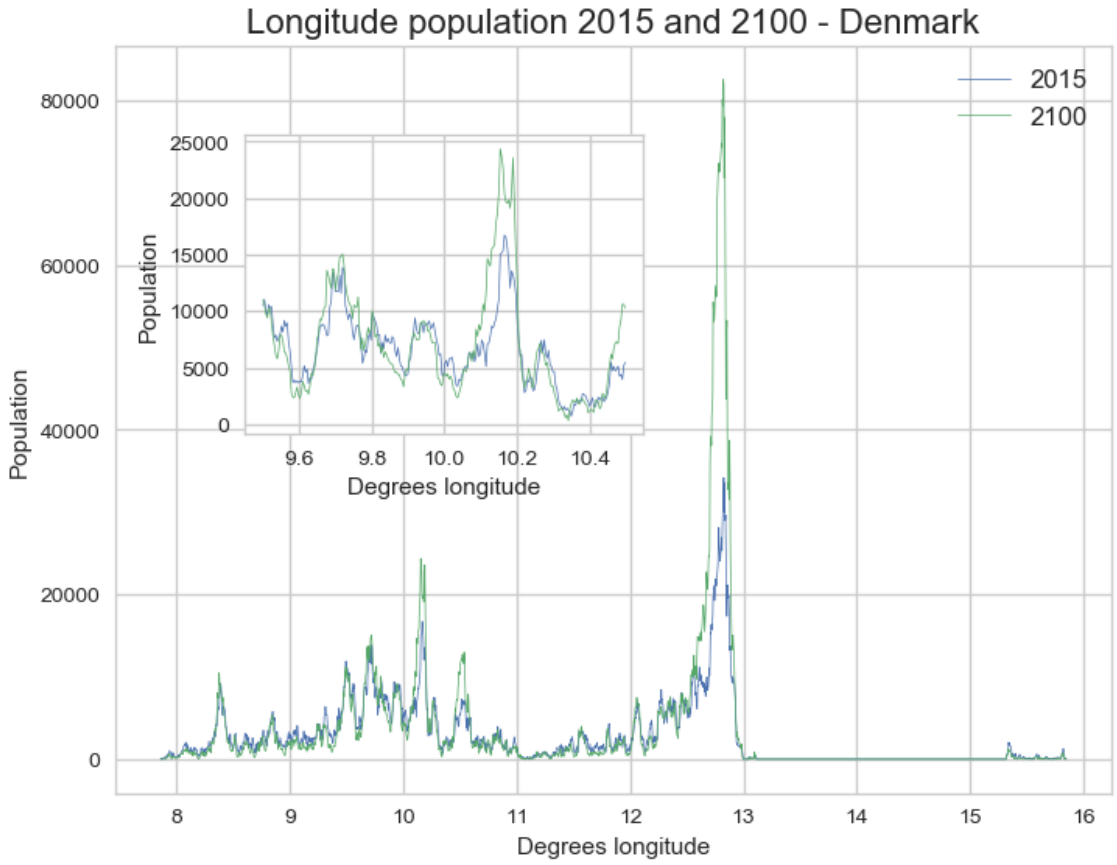

Figure 9.9: Longitude population in Denmark for 2015 and 2100

9.2 Local Trends

Looking closer at the population distribution predictions made by PopNet it is possible to get a more detailed view of the predicted changes over time. The most affected places are the largest cities, which in the countries the model was tested on is Paris and Copenhagen. The changes and interesting insights that can be gained by evaluating on a local scale is however not limited to these two cities and as such other examples will be presented as well.

To help make the model, its logic and choices become more transparent the relevant ancillary data is visualized as well. From this, certain patterns or lack thereof can be seen as the model is trained and run on all of these simultaneously. As such, all of them could have potential impact on the predicted spatial distribution. When looking at the visualization of Paris – France, respectively, the ancillary data in figure 9.10 and the PopNet population distribution prediction in figure 9.11, one can see a consistent growth in the city across each ten-year period of time.

When looking at the change across the 2020 to 2100-timespan, an increasing spread of population can be observed from the city centre and out towards the suburban areas around the inner city. The city suburbs do however also seem to spread from local population centres into their own surrounding areas.

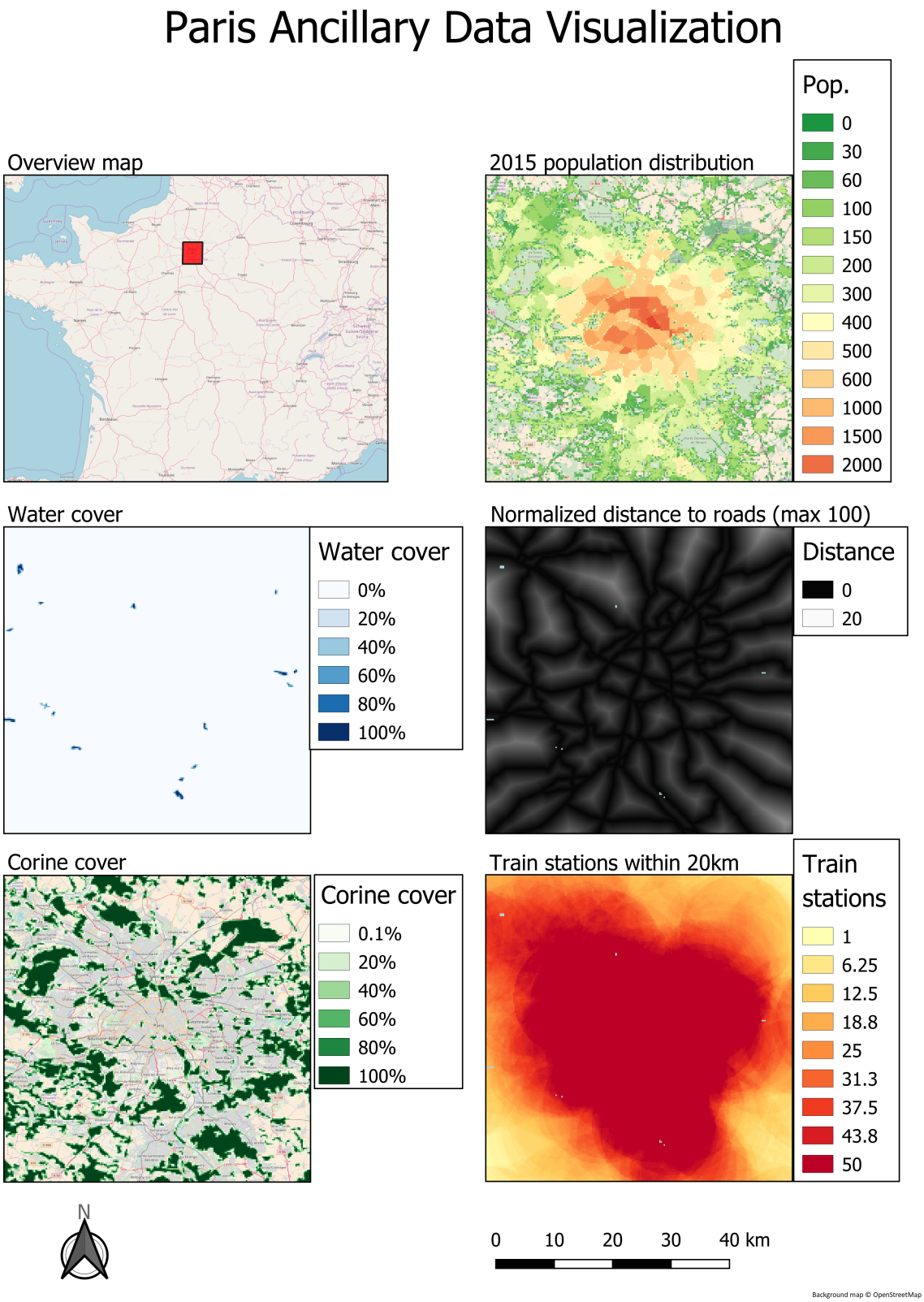

Figure 9.10: Ancillary data in the area around Paris

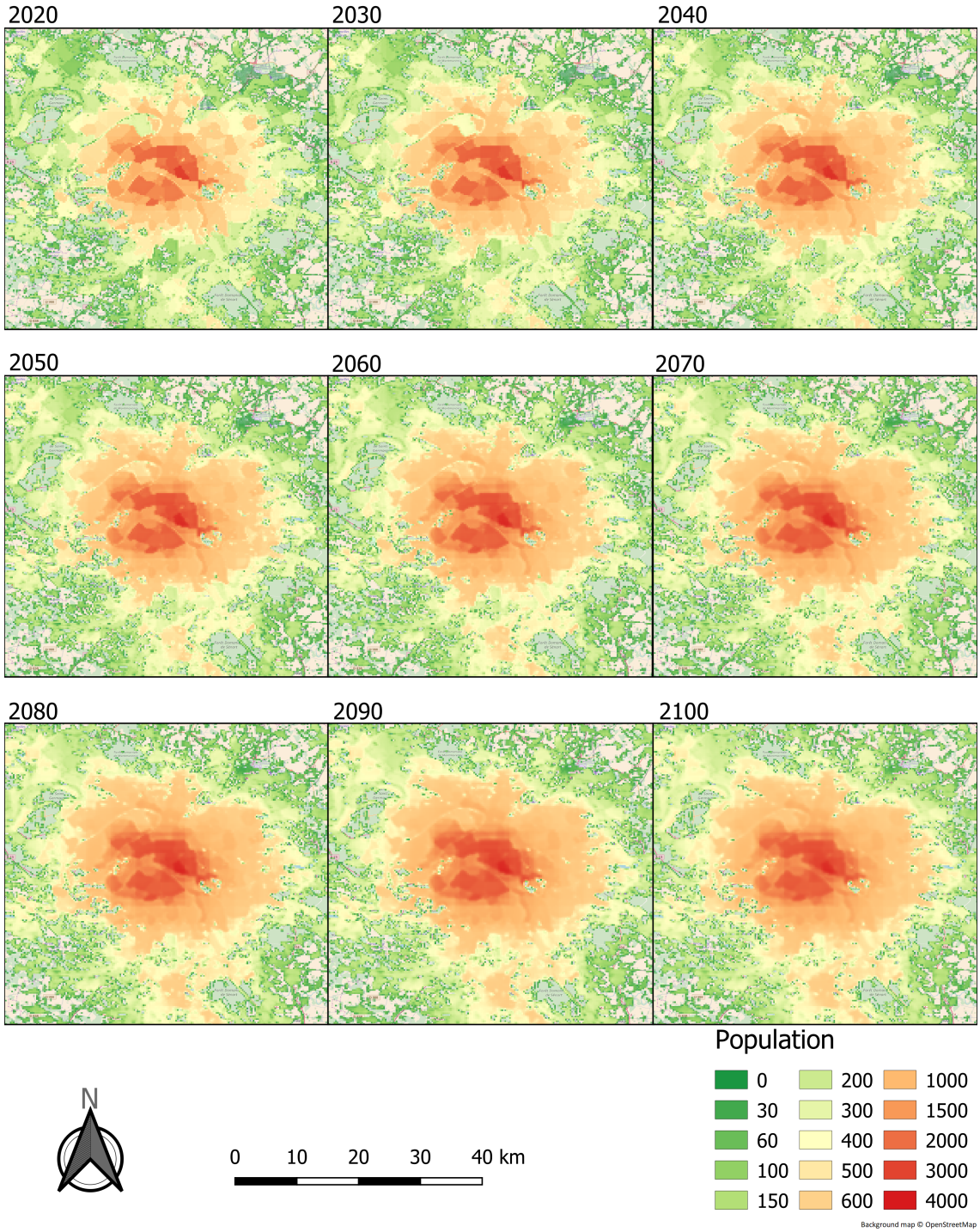

Figure 9.11: PopNet predicted population distribution of Paris

At first sight there is no clear tendencies to the pattern in relation to the ancillary data layers. As can be seen from the ancillary layers visualized in figure 9.10, many of the layers have either a high degree of uniformity or cover. There is only a few spots of water as

rivers such as the Seine is not part of the input data layers as the source for inland water only takes lakes into account (cf. section 5.4). This could however likely have provided a more detailed water feature even though the Seine river can be seen in the population data as a less populated line of pixels snaking through the city centre, but this stem from the GHS data input.

Both the train station layer and the road layer are almost universally present within the area and as such show no real patterns of impact on the local population distribution here.

Assessing the difference between each ten year period, as seen from figure 9.12, it is clear that even though the chunks are shifted between each iteration of the model a pattern can still be observed based on the chunk implementation of the population increase. Another interesting development can be observed in the last predicted maps for 2090 and 2100. Here we see population fall slightly in some of the highly populated areas but rising in less populated areas. The occurrence of this development corresponds with the stagnation and decrease of overall population projected between 2080 and 2100 covered in table 9.6. As such, the model does handle decrease of population but also contentiously disperses the population within a chunk toward a more even distribution.

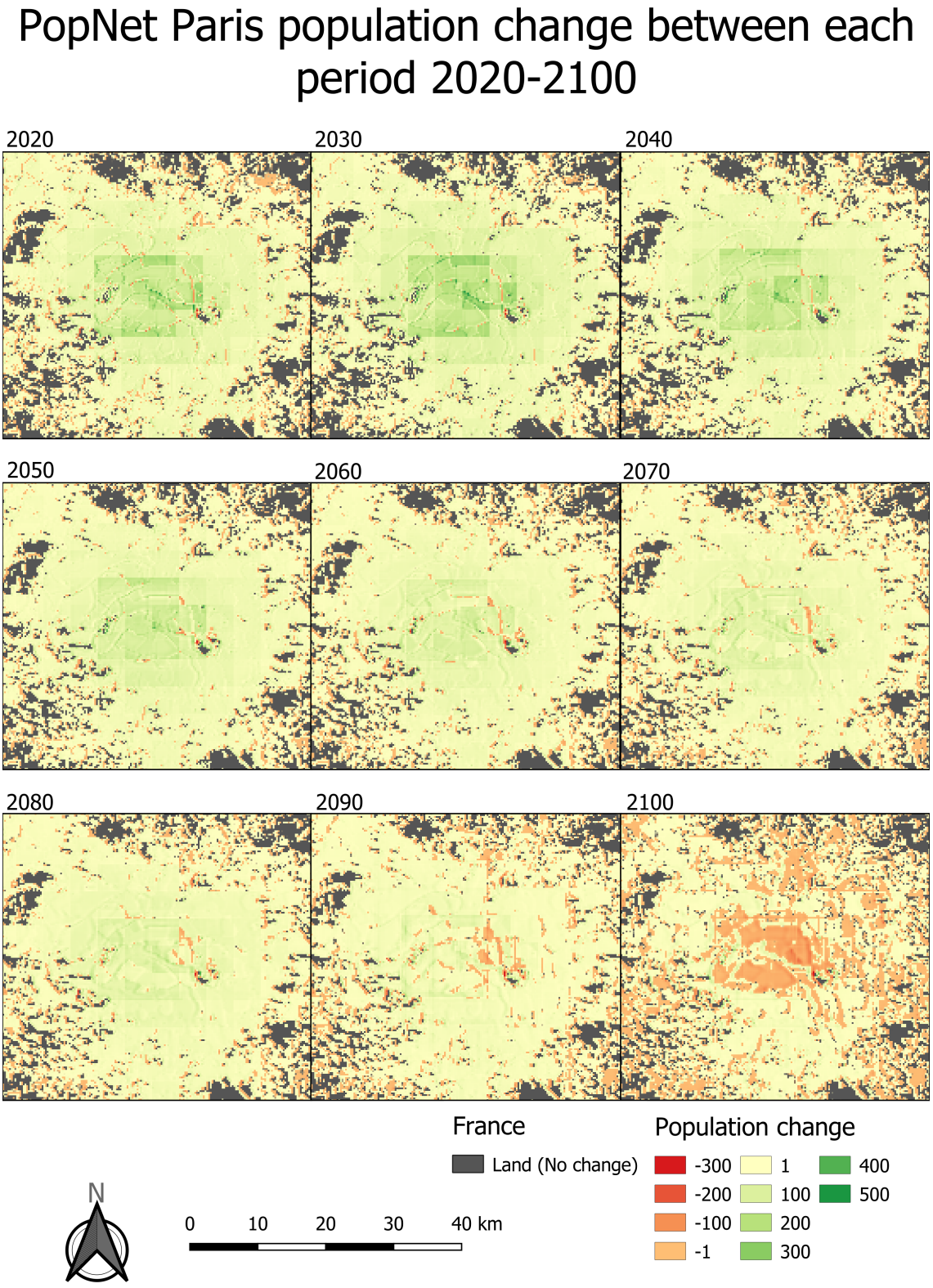

Figure 9.12: Population change of Paris between 2020 and 2100

When looking further at the Paris area some interesting things related to the Corine layer can be seen. If we look closer at the airports we can see that they become increasingly covered by population. The airport areas are part of the Corine ancillary data and should

thereby be distinguishable to the model. This process of population development can be seen on the area surrounding Orly aiport which can be seen in figure 9.13 where the majority of the Corine cover shows the airport extends.

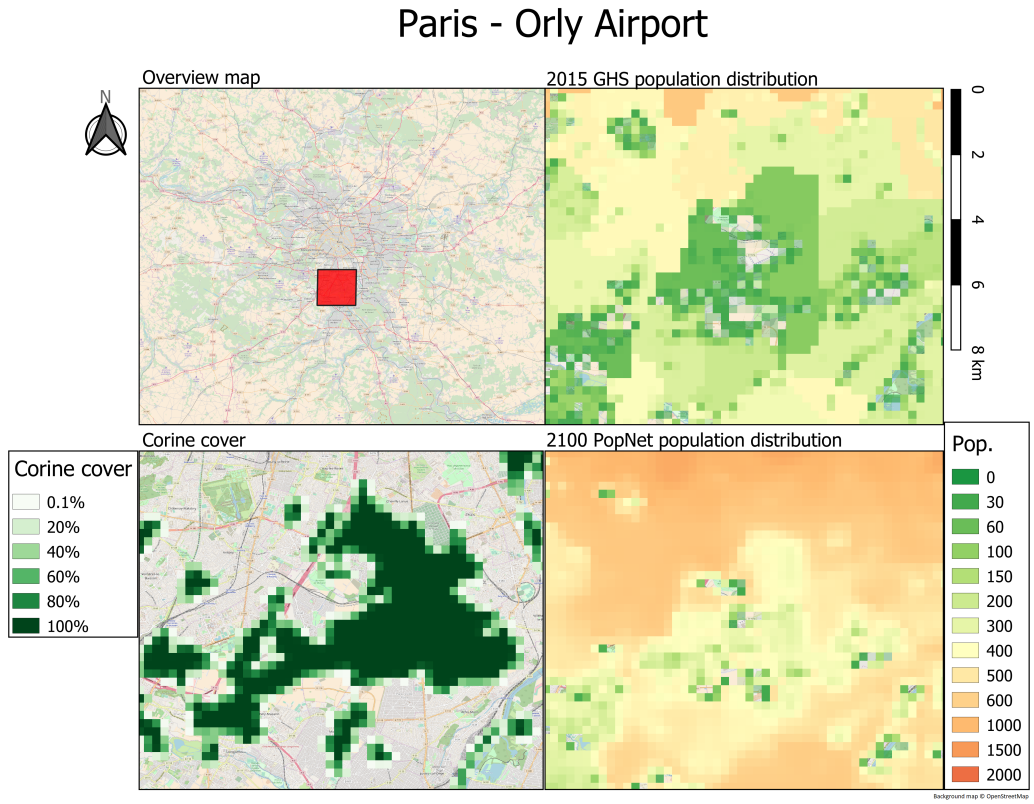

Figure 9.13: Population prediction and input data around Orly Airport

As we can see from the maps of the Corine cover and the 2100 PopNet distribution it does become gradually more covered by population, compared to the 2015 GHS data. This happens even though it does have lower population values compared to some surrounding development. The interesting point here, is that not only is the airport not considered a place where people do not live - but seeing the 2015 GHS population data, we can see that this already have people living there, which is likely contributing to why population is distributed there.

A note to this is that the airport cover is melted together with both industry and forest cover from the Corine layer. This is likely also one reason why the model sees it as susceptible to population development. These layers are likely to have seen population developments during the period from 1975-2015 on which it is trained. Looking towards the real world there is also no saying that an airport cannot be closed down and the area used for different purposes. This has been seen before with for example Tempelhof airport in the centre of Berlin [The Guardian, 2015].

Nonetheless this should not be a general tendency and it highlights that the model inherits uncertainties from the input data, which in this case is the GHS data created from census data and built-up areas identified from satellite imagery (cf. section 5).

Another interesting concept that revolves around the Corine ancillary data layer is the case of forests. Seen in figure 9.14 is the Saint-Germain-en-Laye Forest that lies on the north-eastern outskirts of Paris. Looking at this as an example of the forests covered by the Corine layer in general we can see that the population development does encroach on the forest over time but that in the case of larger forest areas the process is slow.

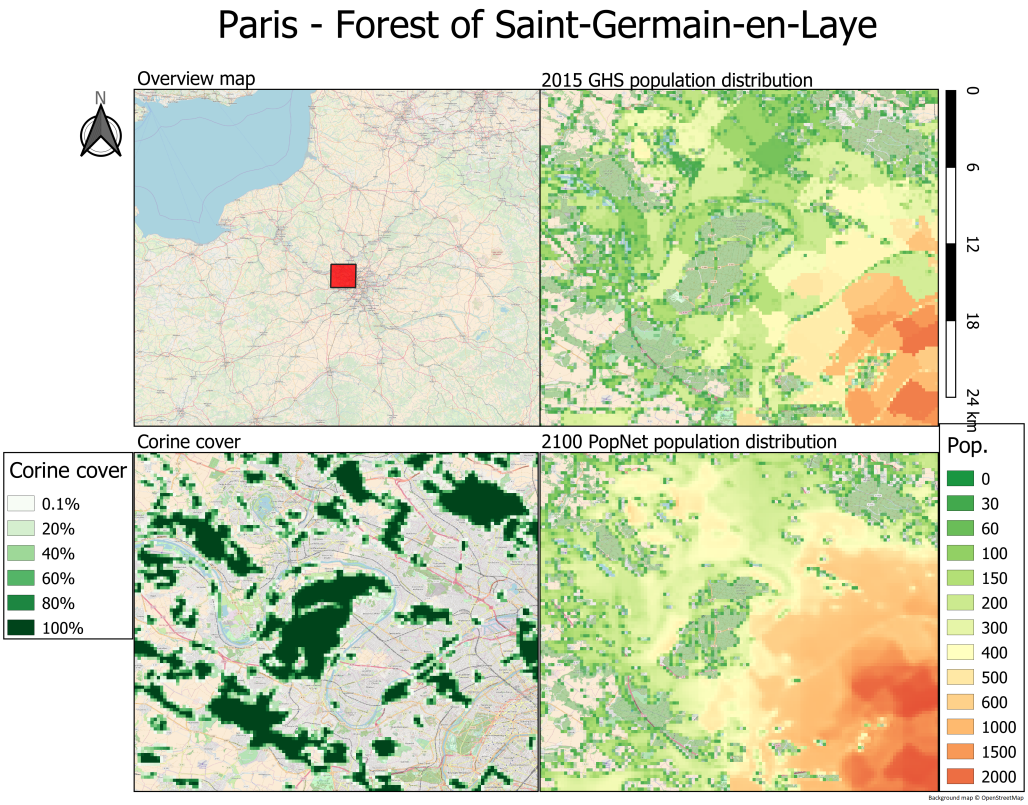

Figure 9.14: Population prediction and input data around the forest of Saint-Germain-en-Laye

Looking at the details it seems that the population development happens from the gaps within the Corine layer that highlights the centre forest but in the 2100 map one can see that the model has also created areas within the Corine layer but even these seem to have grown from already populated cells. As such it is uncertain whether the model is capable of identifying suitable areas separately from already developed areas or is limited to moving population to neighboring cells.

Whether or not this development is likely in the real world, is harder to tell. The model is trained on relatively few data layers and it is possible that some of the forest or nature areas could be protected by laws or regulations that are unlikely to change even over extended periods of time.

When looking closer at areas that lie adjacent to water, different tendencies become clear. The water value is an efficient feature and the model understands that population does not live in tiles containing 100 percent water. That the model should clearly recognize water tiles, is supported by the choices made in the data preparation section, where the distance to roads layer is also set to 100 on water cells, which is the normalized maximum

value for that ancillary data layer (cf. section 7).

What is also interesting is that the population on the local scale seem to shift towards west. This tendency was also previously mentioned in section 9.1. Here it becomes even more clear, as can be seen in figure 9.15, because the shift moves population into the water tiles which does not occur under any other circumstances.

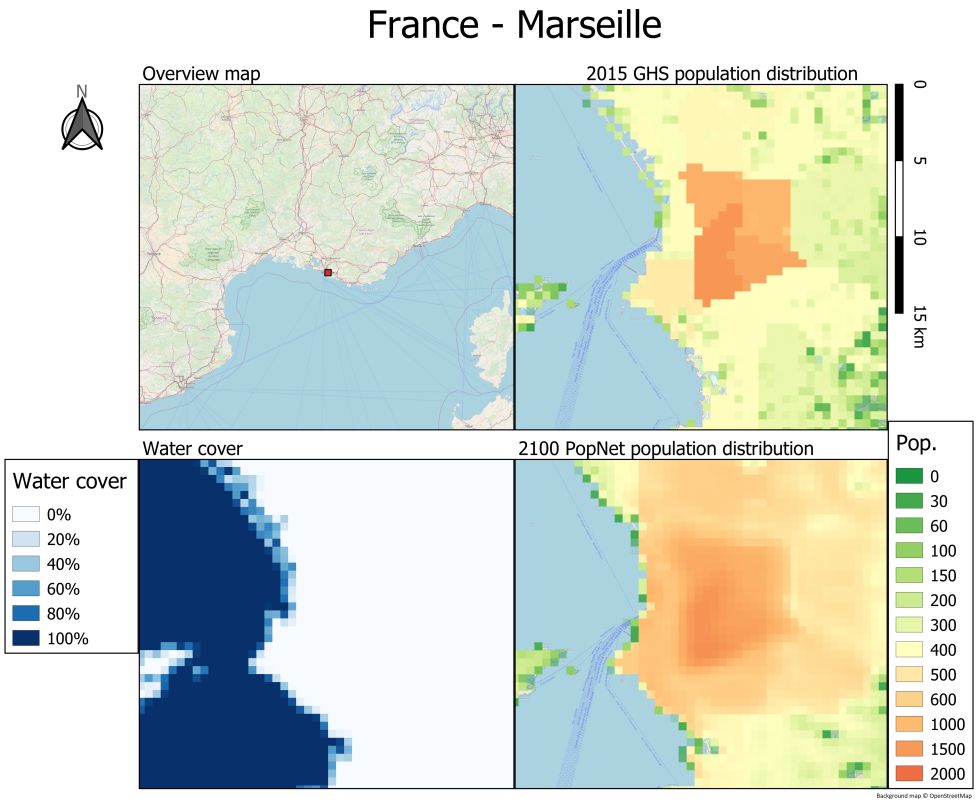

Figure 9.15: Population prediction and input data of Marseille city

The shift is subtle and occurs gradually over the periods or iterations of the model and only moves one or two pixels over the total period. However, one or two pixels still equals a shift of between 250-500 meters which is a lot on a local scale.

This shift also happens when the model is trained on Denmark which points toward it being a general problem that might relate to core functions of the CNN and possibly the chunk cost function, the specific reason is however currently unknown.

Outside of the observed shift, the general predicted population distribution of Marseille seem to follow the same tendencies as Paris with a general increase and spread of population from the city centre.

Looking at the Danish model and outputs, several differences can be seen on a local scale. First of all Copenhagen as illustrated in figure 9.16 and 9.17 below, have a more central development with higher population growth in the centre, and compared to Paris, less urban expansion outwards from the city centre.

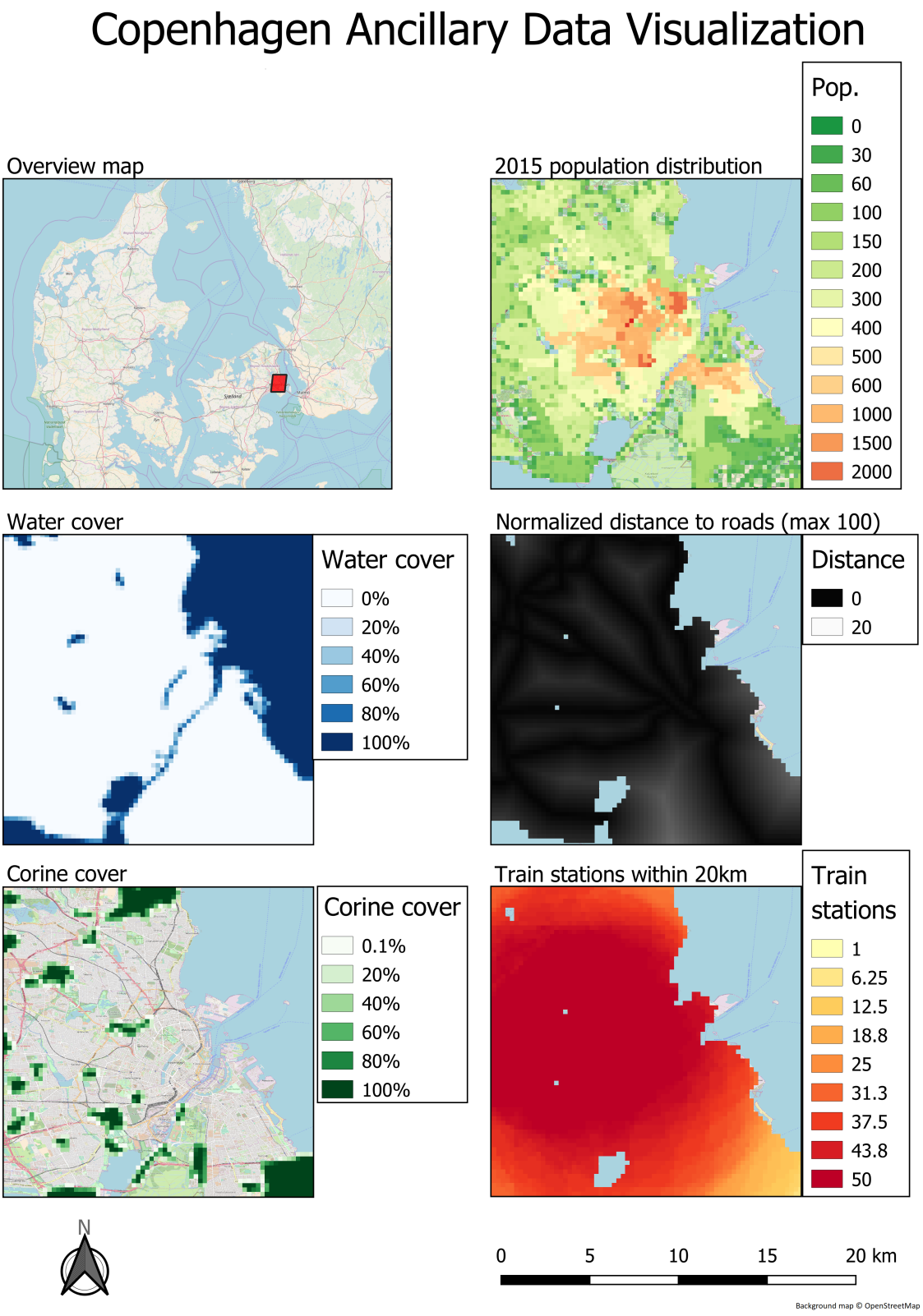

Figure 9.16: Ancillary data in the area around Copenhagen

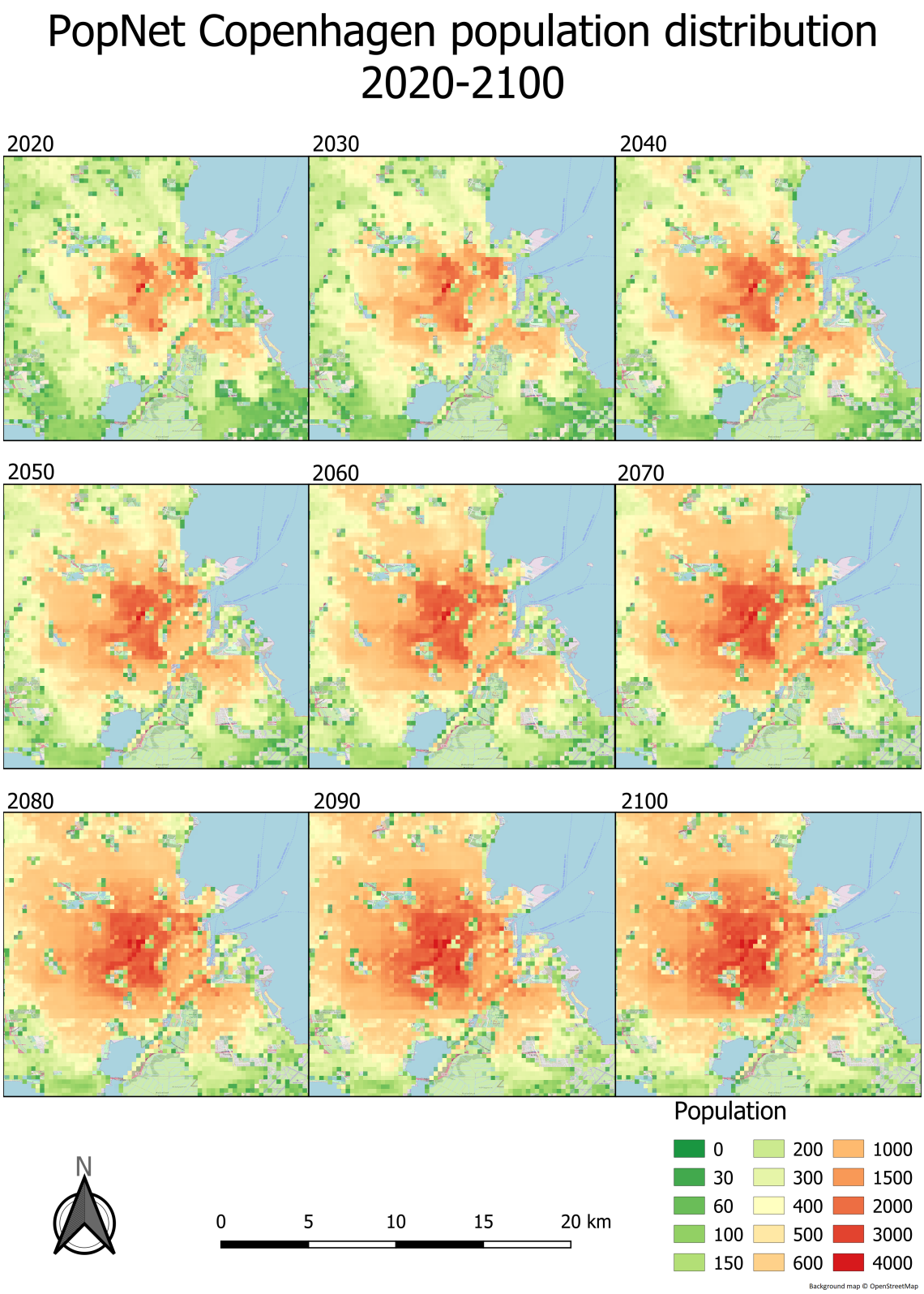

Figure 9.17: PopNet predicted population distribution of Copenhagen

If compared with the visualization of the French capital Paris in figure 9.11, it is also clear that tendencies observed in section 9.1 can be seen here. There is a clear tendency that the model trained on France spread the population over a larger area where the model

trained on Denmark increase inner city cell values of especially Copenhagen. One reason why this could be the case could be due the large amount of water around the centre of Copenhagen that limits the possibilities of the model to expand outwards as effective as for example Paris that does not have this restriction.

Another reason why, could be a difference in the models as they are trained on the different data sets of Denmark and France from which we can see there is a variation as previously covered.

This difference can also be seen in the population change between each period compared to the maps of Paris showing the same thing. As figure 9.18 shows, the variation within the individual chunks vary more between areas. This local variation difference between Copenhagen and Paris could be the result of the input GHS data, being of different spatial quality. While all cells in the GHS layer are 250×250 meters, the population value contained in these are based on census blocks which vary in spatial size. A possible result of this, is that the GHS populaion data for Paris in the input data, is evenly distributed across the individual census blocks. For Copenhagen, this effect can also be seen but on a smaller scale, likely due to smaller census districts that thereby distribute data more precisely and spatially varying in the GHS grid.

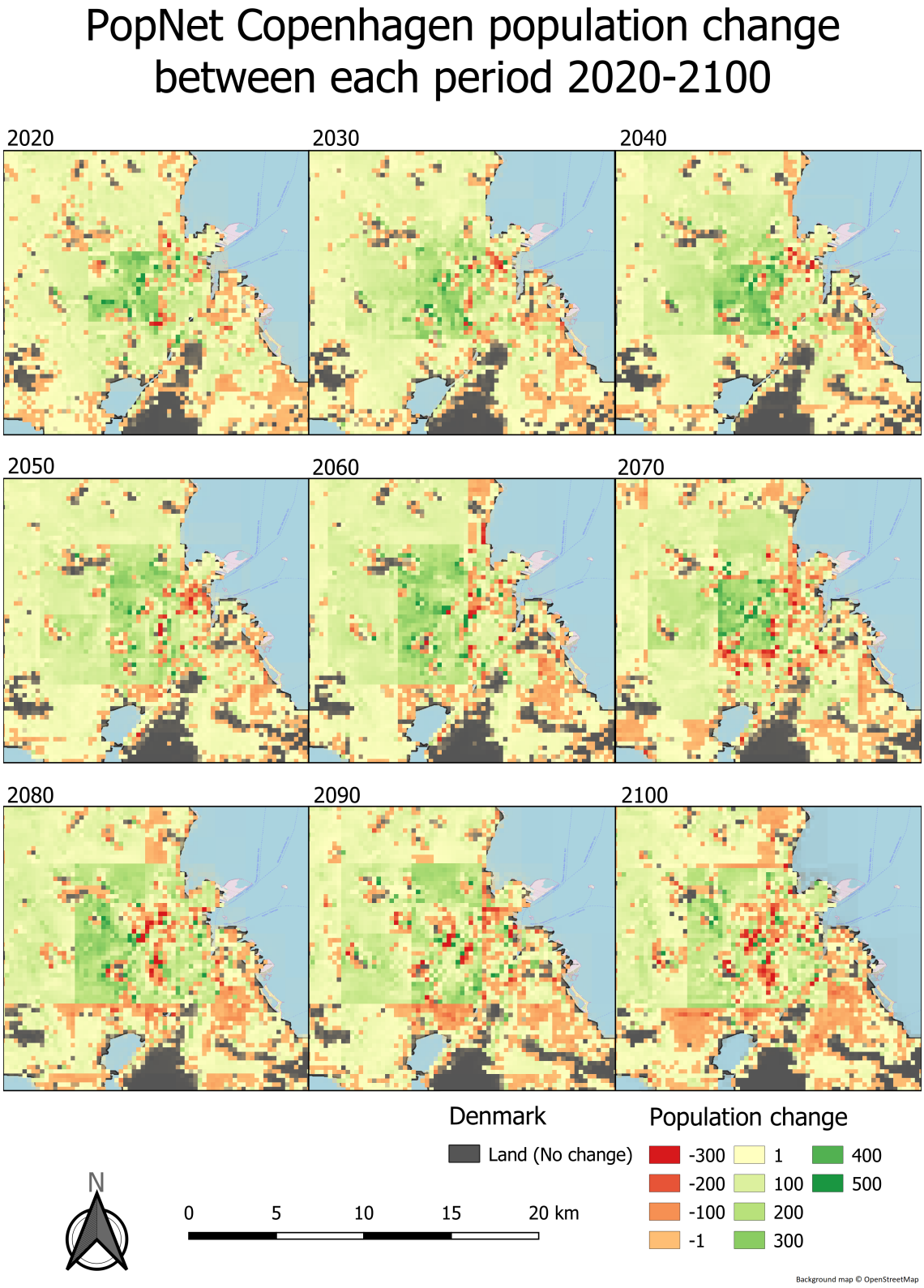

Figure 9.18: Population change of Copenhagen between 2020 and 2100

Another curious thing is how certain areas avoid population increases all through the period even though as just stated certain cells have massive increases. From figure 9.19 an example of this can be seen, the map shows the area of Utterlev-mose which is a park,

lake and marsh area in the north-eastern part of Copenhagen.

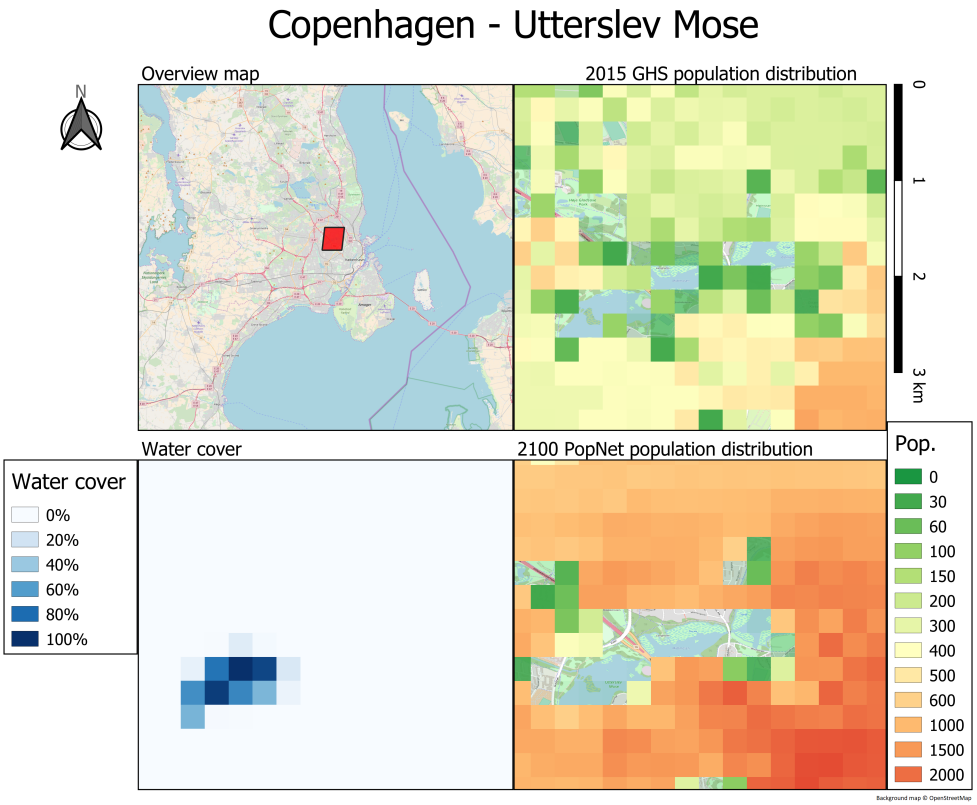

Figure 9.19: Population prediction and input data around Utterslev-mose

Here a development can be seen that highlights the area around the green space even clearer than the 2015 GHS pop layer does initially. While some part of the change can be explained by the occurrence of water in the south-eastern part of the marsh the model itself seemingly recognizes a pattern that makes this identifiable. This could potentially be the sparse population in the GHS layers it is trained on, but no obvious explanation can be seen from the ancillary data. It is as can be seen from figure 9.17 a development that occurs at multiple locations in Copenhagen and tends to be on parks, cemeteries or other likewise places - although none as clear as the example of Utterslev-mose.

When looking at population distribution outside of the large cities we can see a tendency of depopulation in the rural areas. Especially the smallest townships and singular populated pixels with no neighbours seem to have values relocated into the larger towns and cities. Figure 9.20 highlights an example of such a development in Northern Jutland where it can be seen that larger towns and cities grow while the smaller townships grow smaller and less populated over time.

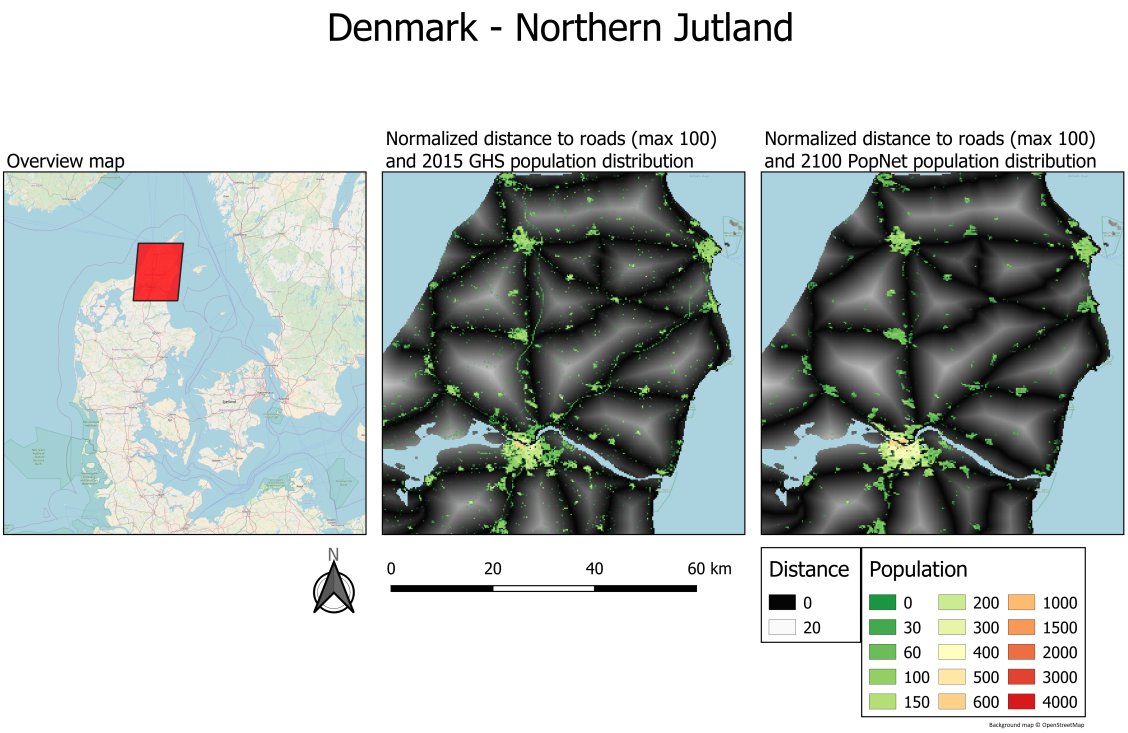

Figure 9.20: Population prediction overlayed distance to roads ancillary data

What figure 9.20 also shows, is a map of the normalized distance to roads. This shows that the distance to roads value seem to have an effect as populated pixels with high distance to road values seem to be almost completely removed over the 85 year time span. This tendency could however also be caused simply by the larger towns and cities which then happen to lie within areas with less distance to roads. Thus making it seem like the development happens due to the distance factor rather than simply due to population gravitating towards larger cities. One thing that seems to point toward this is that the populated cells along the roads which can be seen in the 2015 GHS population distribution in figure 9.20, also become heavily depopulated when comparing to the 2100 PopNet prediction.

When looking into the ancillary data layers of number of train stations within 20 kilometers and the pixel slope percentage we can see that they seem to have an effect. For the slope layer the only place this can be seen is in the mountainous regions of France where population decreases in the pixels with high slope values. It is however hard to ascertain whether this phenomenon happens because of the slope values themselves or due to the urbanization pattern that decreases population outside of towns and cities. When looking at the average population on the pixels for France that have a value above zero categorized by slope value, visualized in figure 9.21, it can be seen that the values all increase over time. The only cells that do not seem to follow a similar pattern are the pixels with a value representing 10 degrees or more slope. These seem to start increasing in population more rapidly from 2040 and onwards but this tendency coincides with the beginning of the shifting cells that occur towards the left. Given the low amount of cells with a slope value above 10 only a few populated shifting to have a large impact on the average. The same tendency could likely be occurring within the other categories as well, but this have

a smaller impact on those as there are more cells as well as higher population. Looking at the average in 2015 however, both the category of 5 to 10 degrees and 10 degree plus are close to each other. This could point toward that the values of 10 degree plus are recognizable by the model as impactful, but that values below a threshold of 10 are to a lesser degree.

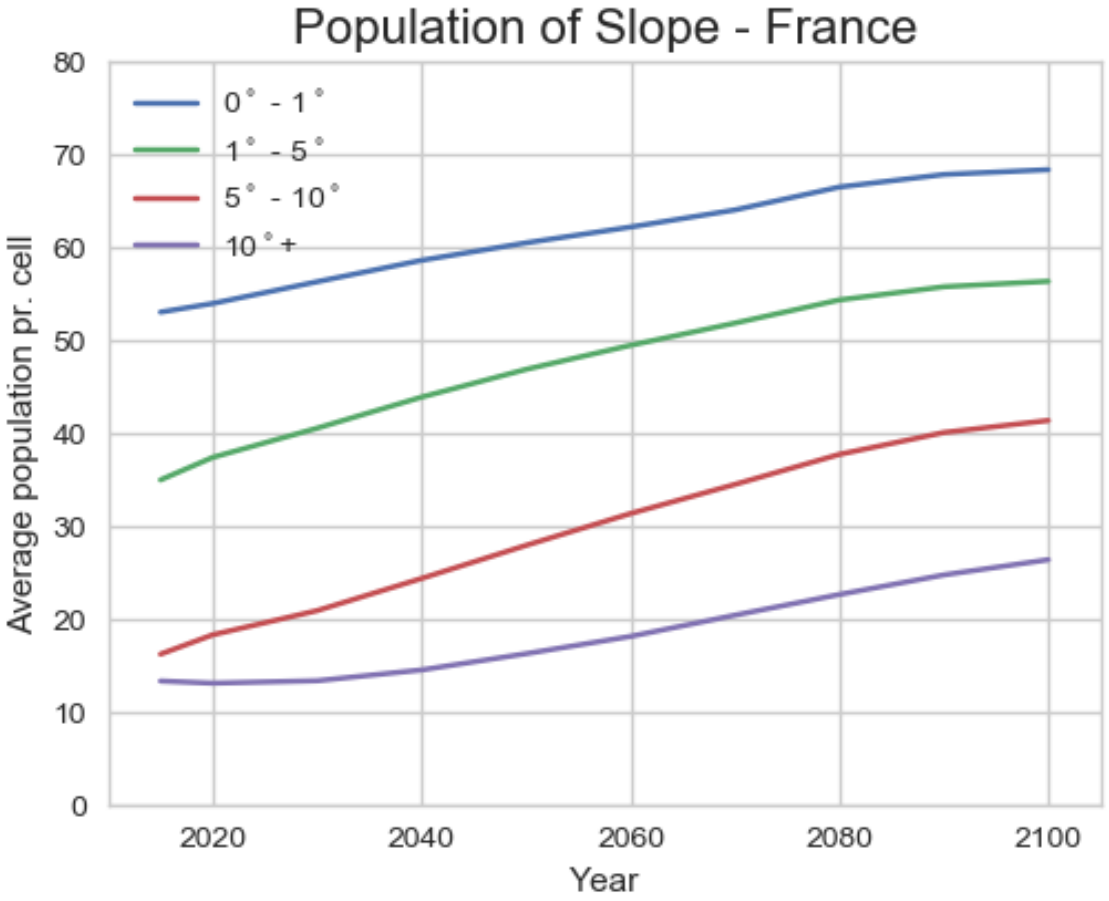

Figure 9.21: Average population per cell with more than zero population of France, categorized by slope

Looking at the train station layer the values seem to align as population increase occur in areas covered by train stations. But whether this happens because of the station pixel value or because stations are logically placed within areas that already have high population is hard to say. As such, we see that there seems to be a correlation between the trains stations and population. We did however already know this, as it was one of the reasons why the train stations was implemented as a support for the model to identify high population areas.

Overall we can thereby see that the model itself can project change that is relatively close to that which have occurred from the year 2000 to 2015 when looking at the numbers. The model is also capable of projecting future population projections within ±0,3 percent of the population numbers projected by the IIASA SSP2 scenario. The models of Denmark and France vary in how much population is moved towards the larger cities but both

models have a tendency of moving rural population to nearby towns. When looking closer at the spatial distribution however the model still have a number of challenges. One of these is an overall tendency that population values seem to shift slightly toward left in the TIFF images. Another is that population does not respect a number of boundaries and that ancillary layers have varying amounts of influence. This is likely caused by various elements from how the CNN architecture works to input data quality as well as input data values. Even with these challenges, the model does clearly identify patterns of population distribution and successfully projects a future scenario based a controllable population input.

This answers *How well does the model predict future population distribution?* and partially *What knowledge can be gained from the results, experiments and tests?*. The findings will be further elaborated in the next chapter where a few experiments are conducted to further investigate the models performance, challenges and limits.

Experiments 10

This chapter tests the models' capabilities and limitations in terms of how they react to altering of data to usability of models on different countries than they have been trained on. The limitations will be discussed in regards to possible improvements that could be implemented. This supports the answering of the sixth sub-research question; *What knowledge can be gained from the results, experiments and tests?*.

10.1 Alteration of Data

There are several interesting aspects of altering the input data, as it allows for simulating different scenarios. One is flooding a city, which we have tried to understand how the model reacts. To simulate the flooding we have placed a lake in the middle of Copenhagen, as shown in figure 10.1.

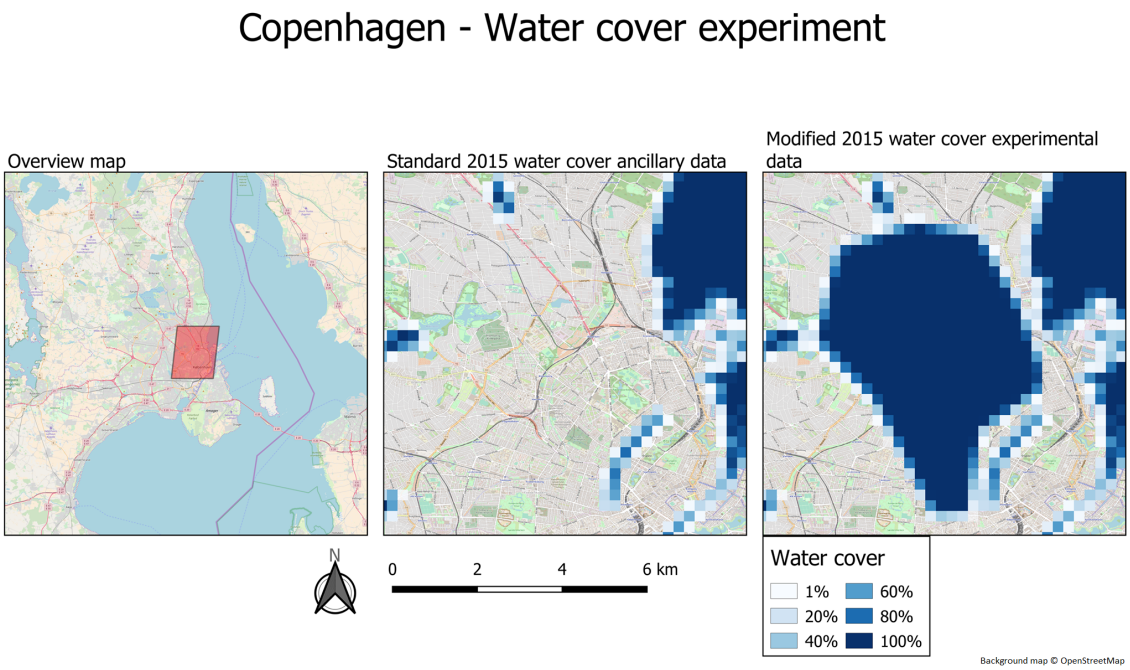

Figure 10.1: Area of lake placed in Copenhagen

Running the model on this data produces a similar result to the original output, which can be seen from figure 10.2 that compares the population distribution within the shown area. This result is unexpected as water should be uninhabitable, and therefore move the population gradually if not at once. The fact that it does not move the population is first of all a sign that population has a bigger weight and impact in the model than that of

water, and thus that water only have little influence. However, the influence it does have is expected as the experiments population distribution has slightly less people than the original. Secondly, the model has not encountered a situation with such vast amount of water and population in the same cells in the training, which probably means that the model goes back to default and relies mostly on the population feature.

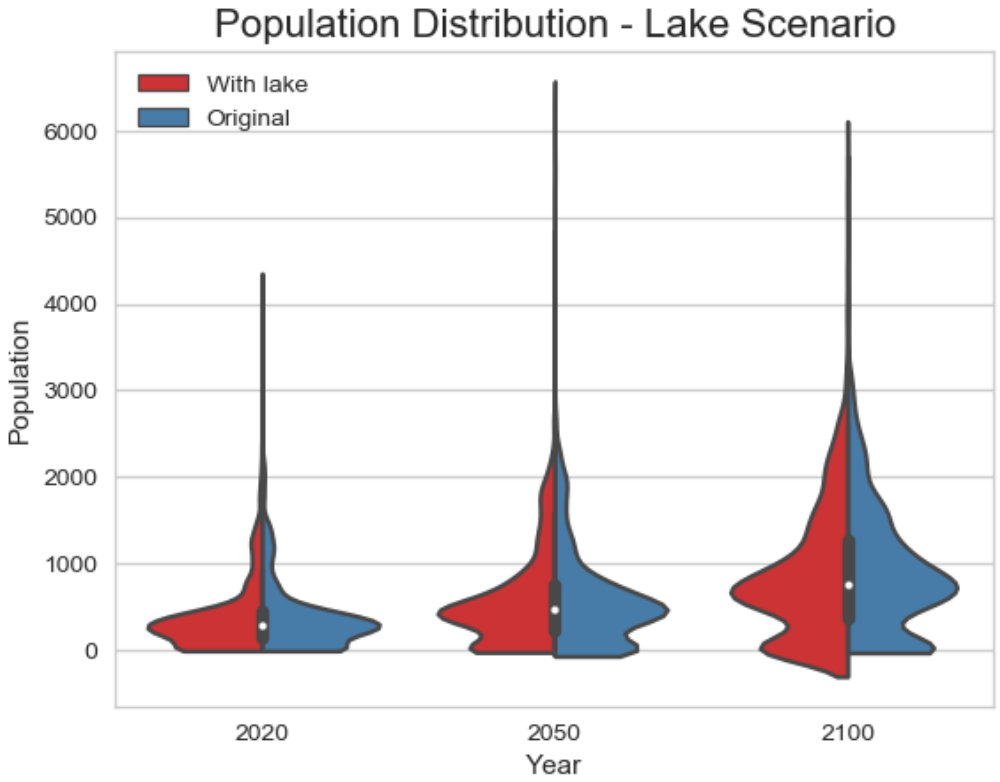

Figure 10.2: Population development distributions around the scenario area for original and lake scenario

Another scenario created is the addition of a road, that is missing in the original data, at Ørestaden on Amager. The area, where the road is added is shown in figure 10.3.

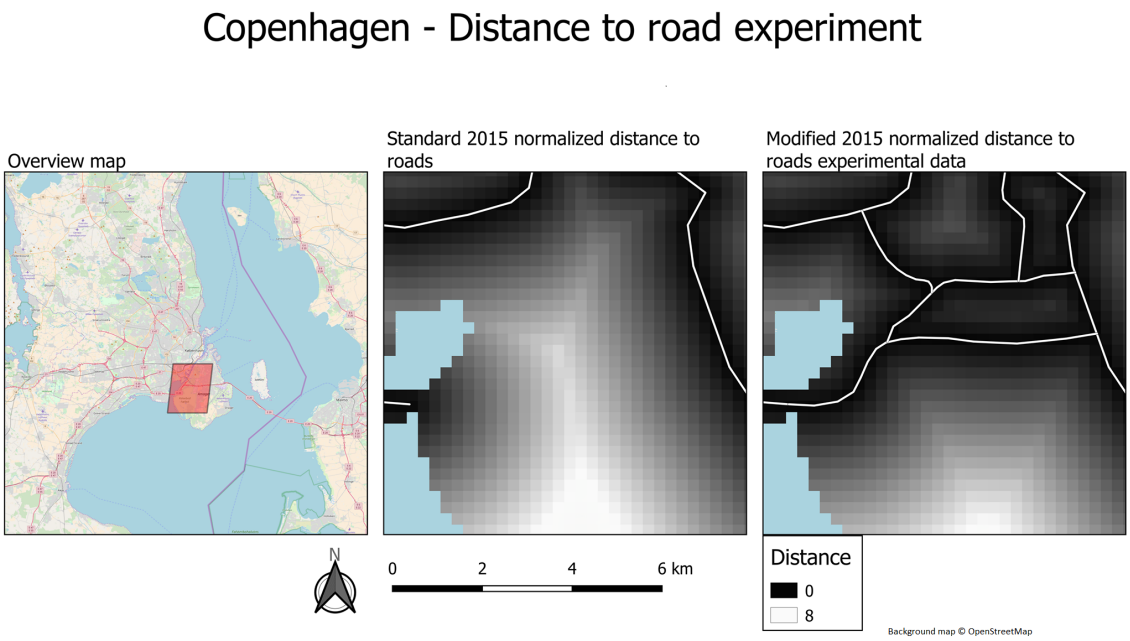

Figure 10.3: Area of road placed Ørestaden Amager

The impact of the road can be seen from the population distribution comparison of the area in figure 10.4, that shows increasing population values. This clearly indicates that roads, and features other than the population do have an impact on the prediction which was uncertain from the evaluation chapter.

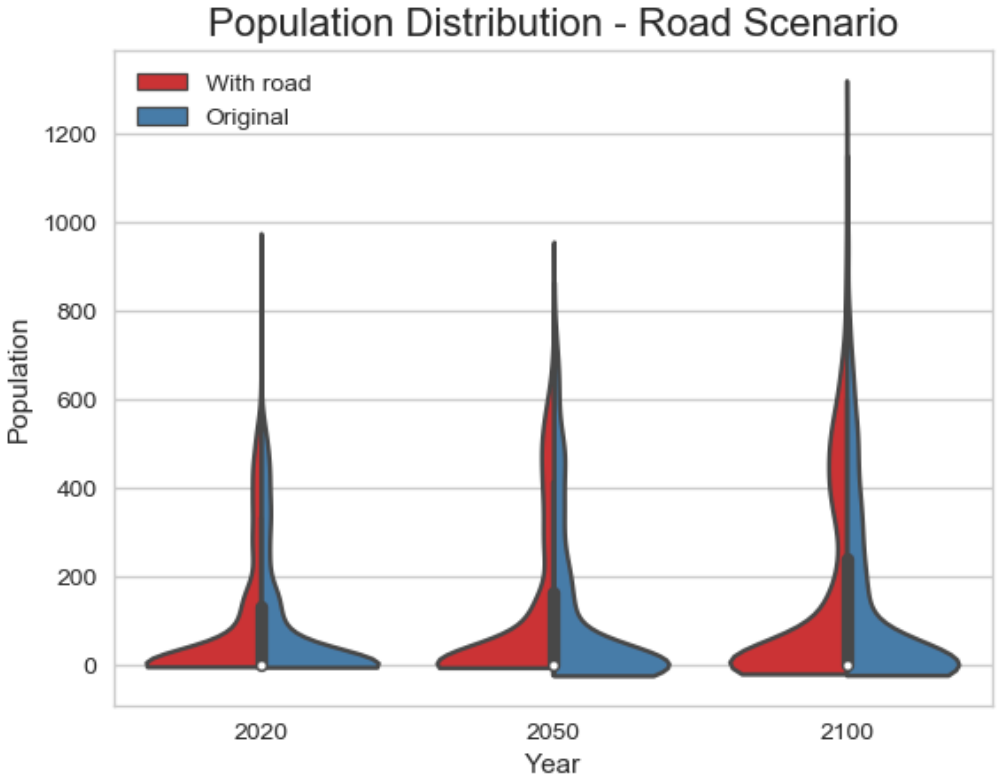

Figure 10.4: Population development distributions around the scenario area for original and road scenario

Looking at figure 10.5, shows the original scenario versus the road scenario. From examining those maps it is evident that a rise in population happens around the new road, which stresses the fact that the road feature does indeed have an impact. In addition, this could be used to incorporate potential and planned future projects to examine effects on population. However, as was shown in the lake scenario, the neural network needs to have encountered a similar situation before it makes sense

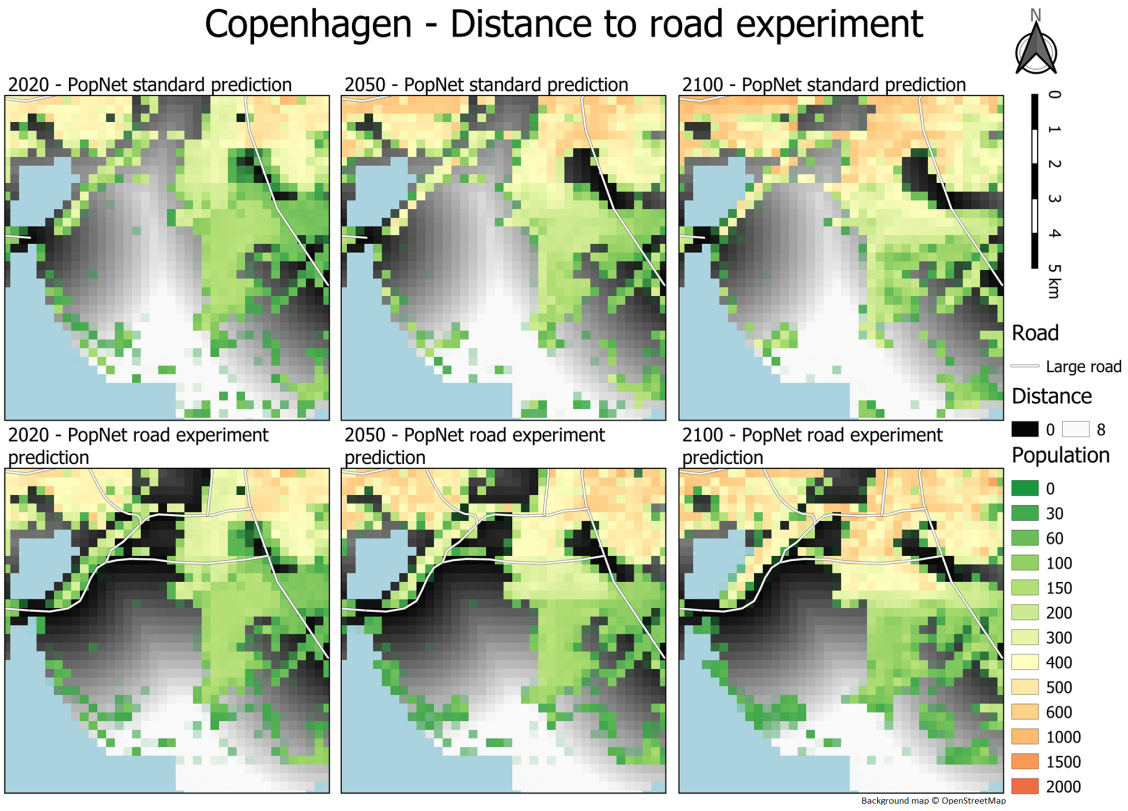

Figure 10.5: Spatial population distribution around the scenario area for original and road scenario

The scenarios illustrate how future political plans manually can be incorporated in the model, and how they impact it. However, the changes that can be made are limited to the input data layers and does not necessarily have the intended effect, as can be seen from the lake scenario. Thereby, these scenarios touches upon a flaw in the model, that is related to the choice and limitation of data. All the layers chosen are spatial features that exist in the physical world, and despite research showing correlation between the layers and population, as documented in chapter 5, they represent only six features to predict such a complex concept. So despite our model showing promising results, as shown from the previous chapter, there are improvements to be made. Some suggestions for improvements are explored below.

As mentioned the model only uses physical spatial features like water, roads etc. but does not capitalise on other data like financial or societal statistics, that has shown to have an impact on health, education and thereby population patterns [Bloom et al., 2008]. One could argue that some of those factors are already implemented in the SSP2 projection number and thus indirectly in the neural network. However, we believe that inputting

these values spatially, could let the neural network find patterns and correlations to the spatial population distribution.

Using ancillary data with the same temporal resolution, as the population data, would allow the model to capture local nuances that was not possible with constant data. This has not been an option in this project, as we simply have not been able to obtain the historical data, but as more data is gathered and stored it will likely be feasible to implement in the future.

Another step to make, is including historical political plans for the training ranging from infrastructure, residence, nature, industry to culture projects and potentially others. This will make the model aware of the changes that are happening to population in relation to such projects and thus be able to add existing plans after training to directly impact the predictions towards political desired or expected goals. While this will be doable for developed countries, especially with projects like INSPIRE, it would probably be hard to implement for developing countries, that do not have much historical data [Directive, 2007].

In regards to planning, it is unlikely that there exist plans up to the year 2100. To deal with this issue, it could be a possibility to give the model flexibility to predict development in the ancillary layers (recreational areas, roads, lakes etc.). This could for example come to fruition in new built up areas, where the model could find patterns that simulate a new park being established. Despite the inherent uncertainty in such predictions, they may show to be more plausible than using static ancillary layers.

10.2 Model Usage

Using a model trained on one country on another will give an understanding of how general the model can be. To research this idea further, we have used the model trained on France to predict the spatial population distribution for Denmark in 2015. The population and MAE is derived from all values and the figures only uses value above or eqal to one, for the same reason as given in the evaluation chapter 9.

Comparing the difference between the predicted and the historical population distribution in 2015, shows that the French model is predicting the total population well and is off by 4537 people compared to the historical population data for Denmark in 2015. The mean absolute error of 1.08, indicates that the prediction in general on average is off by 1.08 per cell. The mean is negative 2.54 and the standard deviation is 29.81. This is exactly the same results that the French model produced previously on France. This indicates that the French model produces the same distribution on Denmark that it did on France, and the same results can be expected to be seen in the evaluation figures.

Denmark - French Model	Predicted Population	Difference in Population	Mean Absolute Error	Mean Population Difference	Standard Deviation of Difference
Prediction	5,664,460	4537	1.08	-2.54	29.81
Historical	5,668,997				

Table 10.1: Denmark - comparison of prediction and actual population data in 2015 using the French model

Examining the distribution in figure 10.6, shows that it is predicting fewer cells in the 0 - 20 population range than the historical data, but is closer to the historical 2015 distribution in the 20 - 500 range than the Danish model used on Denmark. In the 500+ population range it resembles the Danish model used on Denmark, where it overestimates seen by the red color rising above the purple overlap. The French model is again predicting a max population value of around 2900, where the historical data has a max value of 3700.

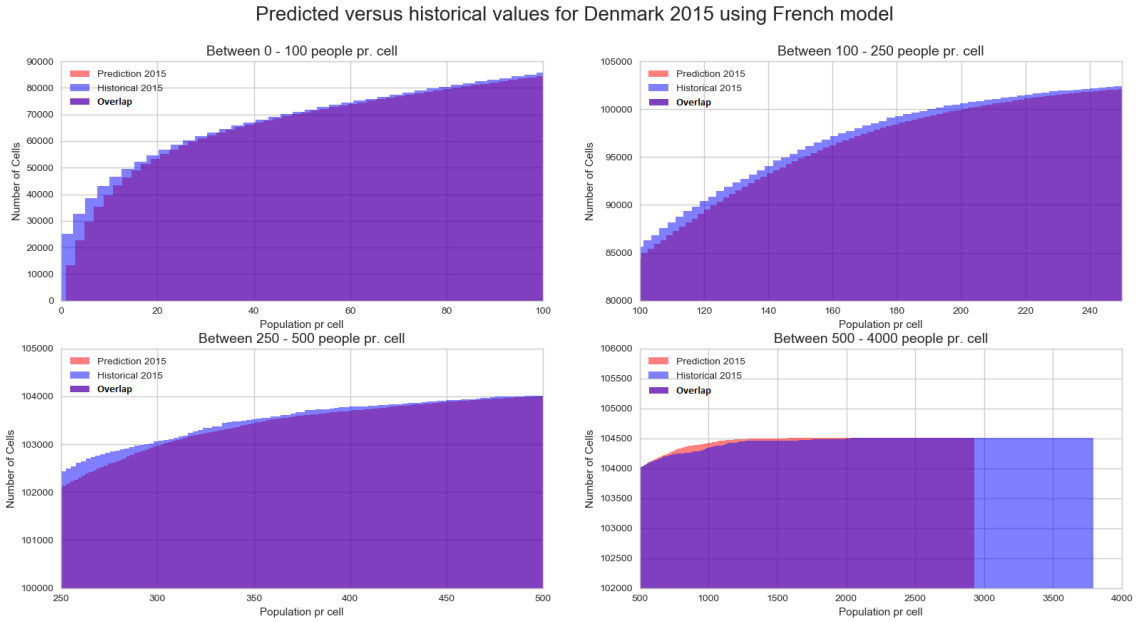

Figure 10.6: Denmark - Cumulative comparison of prediction and historical population distribution in 2015 using the French model

Overall the French model performs well when used on Denmark and is closer to the historical distribution in 2015 than the Danish model, where it predicts closer to the historical data values for 2015 in the 80 - 500 people range per cell.

This resemblance between the experiment and the results of the Danish model presented in chapter 9 can also be seen from the distribution in the four major cities shown in figure 10.7. It over- and underestimates the values in respectively different areas, It does, however, give the impression that the French model is a little bit closer to the historical values overall.

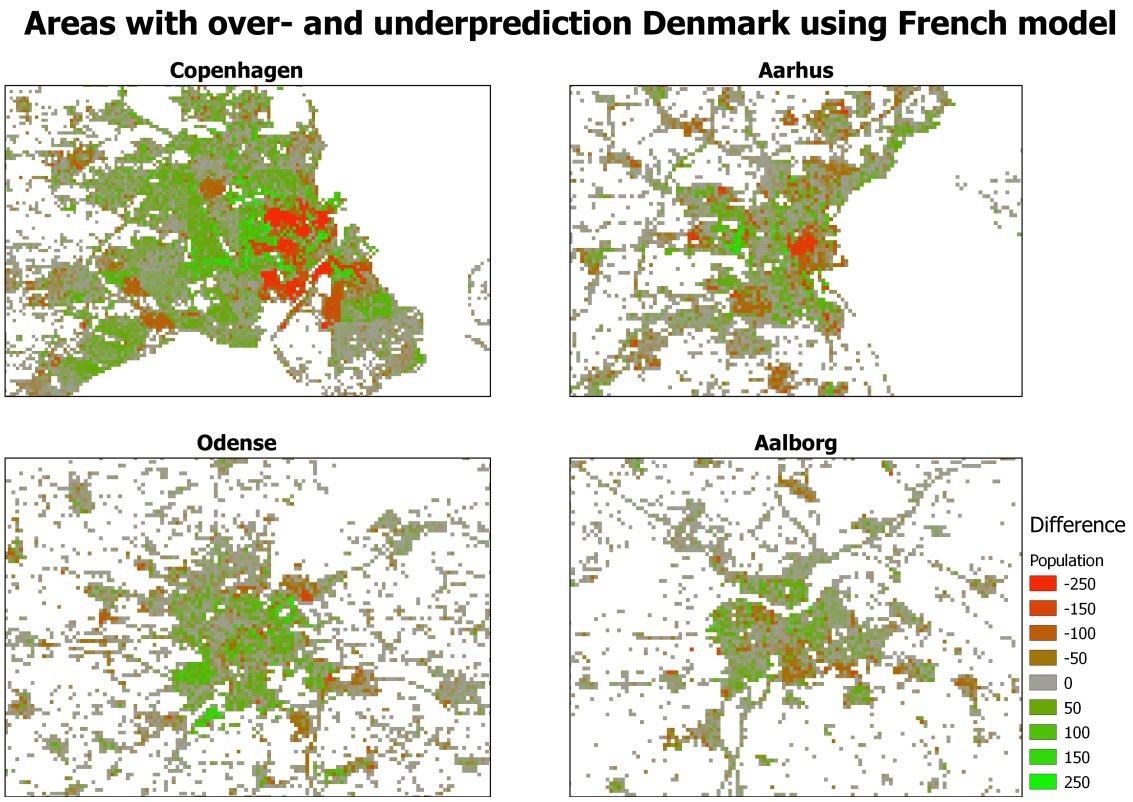

Figure 10.7: Denmark - Areas with over- and underprediction in 2015 using the French model

With insight on how the French model performed on Denmark for the historical data, we will now evaluate how it predicts the future spatial population in Denmark. The model trained on Denmark showed volatility, as the maximum population for a cell rose to above 8.000 from 3.700, as shown in table 9.7. The French trained model had a steady and seemingly more realistic increase in maximum population. Interesting enough, when using this model on Denmark we get a decreasing maximum population up until 2080, where it starts rising again as can be seen in table 10.2.

Denmark - French model									
Year	2020	2030	2040	2050	2060	2070	2080	2090	2100
Min. population value	-1	0	0	0	0	0	0	-1	0
Max. population value	3128	2868	2431	2201	2260	2242	2267	2323	2454

Table 10.2: Denmark - Minimum and maximum population values (French model)

This different behavior in maximum cell-value makes the results of the models quite different, as can be seen from the violin plot of Copenhagen represented in figure 10.8. The maximum values are obvious on this figure, but it is also worth noticing that the French model tends to be more conservative related to raising individual cell population values. This means that it tends to spread the population rather than gathering it, as can also be seen from figure 10.9 of Copenhagen. Here, it is obvious that the model distinguishes itself from the "finger" development plan of Copenhagen, unlike the Danish

model, due to its expansive behaviour [Erhvervsstyrelsen, 2017]. This can be seen as the French model fills the gaps between the "fingers" with population rather than retaining them.

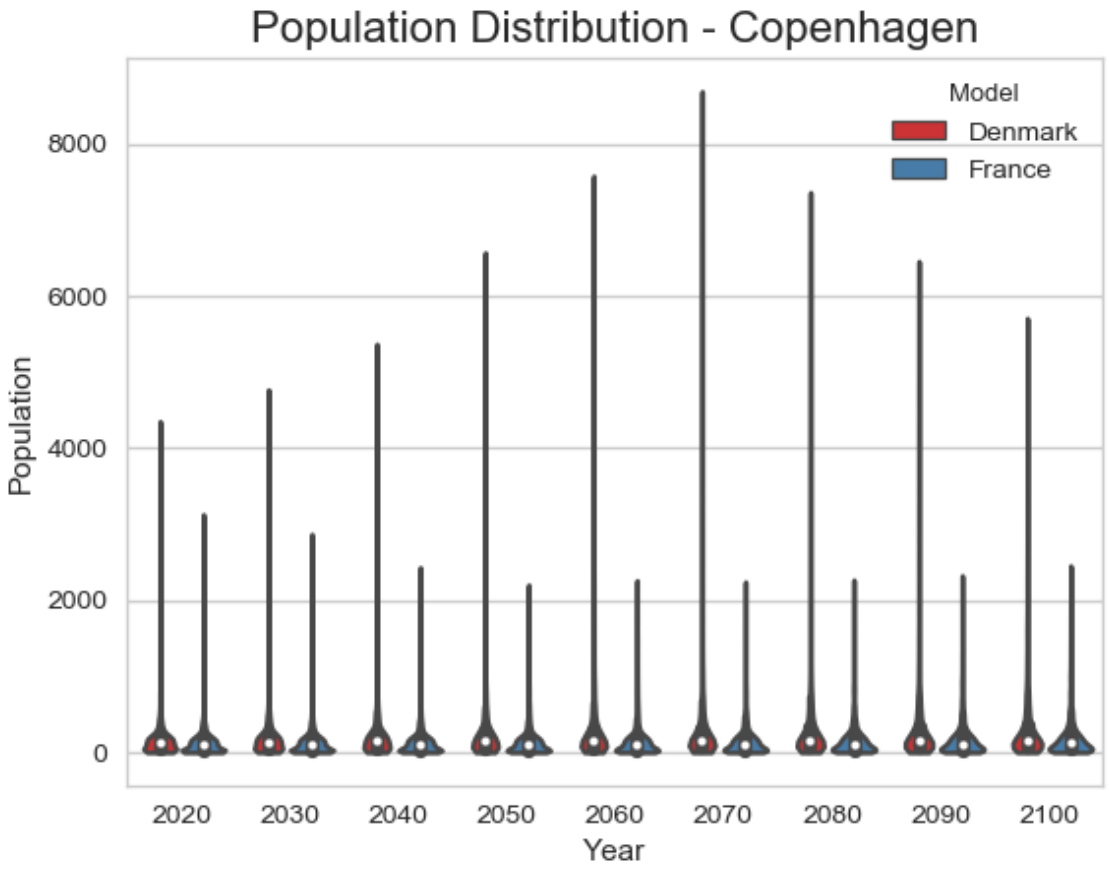

Figure 10.8: Population development in Copenhagen with the Danish and French model. Cells with zero population are omitted.

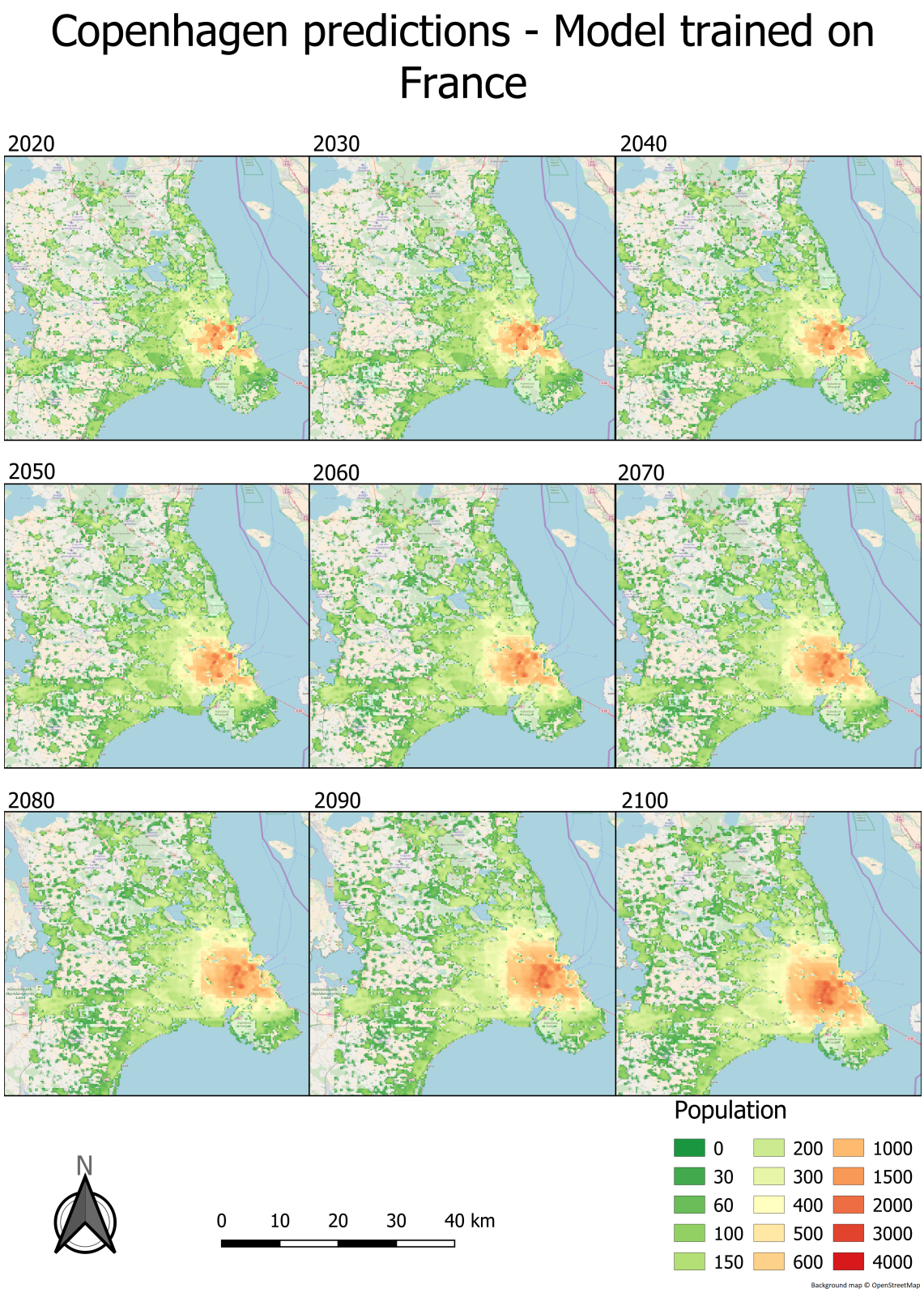

Figure 10.9: Copenhagen spatial population predictions to 2100 with the model trained on France

So while the results are different and have flaws, neither are inherently wrong predictions. The French model is more stable, but does not take the local nuances in Denmark into

consideration, as can be seen from the "fingers" of Copenhagen and their development. However, we believe that training the Danish model, which has been trained considerably less than the French, for longer will make it more stable and thereby possibly better than the French model, when predicting Danish spatial population distribution. Thereby, we believe that training and using a model on the same country will be best practice.

The observations however tells us that using a model trained on one country to predict a different, but similar country, is not necessarily poor. From this observation, we believe that the neural network can benefit from training on multiple countries at once, and potentially use and strengthen patterns based on different countries. This will give more data for the model to train on and thereby more variety, making the neural network less likely to encounter situations it has not seen before, as happened in the lake scenario.

Implementing such a feature will require to input categorical data into the CNN. This could be layers representing the cells' municipality, country and region, so that the neural network can find out which countries and regions are similar. There are several ways of embedding the categorical data to make it feasible for a neural network, we believe hashing or one hot encoding are the most promising, as those have proven useful in a machine learning context [Pentreath, 2017]. Implementation of this would in theory make it possible to train on the whole world and make a very complex, but possibly a very good predictor, for future spatial population distributions.

The next chapter will discuss the findings in the report and finalize the answer to the sixth sub-research question.

Discussion 11

The use of a CNN throughout this project has shown its ability to recognize geographical patterns and use these to predict future spatial population distributions. The projection of future spatial population distributions have already been investigated by Keßler and Marcotullio [2017] and Jones and O'Neill [2016] who have used geosimulation to create realistic spatial population distributions. However, this approach depends on manually created ruleset that are repeated for every year of iteration. This gives the model limitations, as it becomes too simple and uniform to obtain nuances in different local communities. At the same time the approach is inflexible in terms of having multiple data features decide the most optimal placement of the population.

Spatial distributions created with geosimulations are useful at indicating major population trends. Those trends can be used to tackle global, regional and national issues arising with massive population growth combined with for example climate change that makes the world, and especially developing countries, vulnerable to hazards. However, the limitations that were previously mentioned make it inconvenient for decisions and analysis at a local scale, as the quality is not good enough. While our results are not necessarily of better quality, the approach itself is promising, as it eliminates the limitations of geosimulations. The choice for an optimal cell for population is evaluated on multiple features through hidden layers that does indeed represent a complexity that can be representative at a local scale. In addition, it allows for taking in an arbitrary number of features to obtain patterns from historical data without the need to create rules based on the context between them. This means that political plans and economy among others can have direct impact on the spatial population distribution and thereby create realistic scenarios on a local scale.

However, this approach has some of the same limitations that other neural networks have been reported to have. Training a neural network requires a lot of data, and because of the complexity of predicting spatial population distributions means more data the better [Banko and Brill, 2001; Gupta, 2017]. We have been limited by the historic data as only 1975, 1990, 2000 and 2015 have been available. This is arguably too few years to establish a strong foundation for predictions and the temporal resolution even varies, but as more data is created and becomes available the less of an issue this becomes. This paradoxically stresses the fact that developing countries, that arguably need knowledge of future spatial population distributions the most, will have a harder time obtaining a good one with our approach, because they tend to have sparse amount data. Another issue is that the model inherits data flaws, which the data used in this project have plenty of, from sharp population edges between censuses, population distributed onto airports to small towns having absurd amount of people within them. This is a common issue in machine learning and in this case it means that the neural network thinks that above mentioned scenarios

are realistic and thus can distribute population based on those false assumptions [Kellher, 2016; Dietterich and Kong, 1995].

Furthermore, the complexity that neural networks bring comes at a price. Thousands of weights and biases adapted to the label through back-propagation are forming the final model, and there is no reasoning behind those beside minimizing the cost function. This means that it is extremely hard to understand the specific weights and biases essentially making the model a black box. Hereby we do not have control over what it emphasizes other than the inputs and labels used to train on, unlike geosimulations, where every action is controlled by rules. As a consequence we have not found a way to make actual thresholds for the total, minimum and maximum cell population, which makes the result vary a bit from the SSP2 population projection.

Despite the flaws that the CNN has, we believe that it has more potential to produce accurate future spatial population distributions, than that of geosimulations. In this project we have produced realistic results, and there are still improvements to be made. Collecting more and better data would improve the neural network's predictions and finding patterns across borders using categorical data could potentially lead to massive improvements. Complimentary to this the architecture of PopNet is narrow with only three convolutions and two dense layers, and it has been adjusted based on a practical methodology. This means that we have had to test hyperparameters for different architectures with limiting hardware resources. Besides, we do not know whether there is a theoretical approach that could help designing a better architecture. To further improve the CNN it is also possible to feed the chunks to the CNN with surroundings and overlap as suggested in the testing section, which should give the CNN better awareness of the area in which the chunk is placed in (cf. section 8.4).

In relation to the discussion of pros and cons in the geosimulation- and CNN-approach to predict future spatial population distribution, we believe a combination of the two could be worth investigating. Create a ruleset of moves and let a CNN decide which move is the most appropriate in a given situation. This way we can track what moves can and are made and thereby control the model, thus avoiding it becoming a black box. The CNN will still rely on multiple data features to make the decision complex enough to capture local nuances. As a bonus this approach is a categorisation problem rather than regression problem. Categorization problems in relation to CNNs have much more research allocated to it than regression problems, which can be utilized in designing the architecture for a CNN to make decisions.

Doing this and using CNN however, does not remove the underlying challenge of attempting to use recognition software to initially recognize and learn to identify and place population in relation to geographical features. If this is compared to what CNNs are usually used for, which is recognizing the content of an image, for example a horse. This is a thing that are bound by obvious rules as it is a physical object defined by its nature. A horse consists of a body, a head, four legs and there is no doubt that there is a pattern of what a horse looks like. The logic and patterns of where people choose to live are far harder to identify as these consist of complex mechanics, not only related to measurable variables such as economics or geography, but to emotions and human behaviour as well. At the very least this establishes the fact that more types of data can likely improve the model and help

identify where population should be distributed in the future even though this likely would require extensive tests of the influence of each variable to ensure proper weighting. As such it does not mean that observable patterns of population distribution and change, as this report and CNN is based on, do not exist. It does however open a discussion about the concepts of correlation and causality between the data used for projecting future spatial population distribution, as the CNN model is blind to the difference between the two.

We see this in effect in relation to the weights the model gives to each layers as seen from the evaluation of the results data. These all point toward the main attribute in our data is existing population rather than the ancillary data of slope, distance to roads or train stations etc. This is however not necessarily because there is a direct causality between high population and increasing population although one can hardly deny that there might be to some degree. A more likely explanation is that there is a clear correlation between existing population and future population as seen in the data from 1975 to 2015. It does seem unlikely though that this should be the defining explanation to population increase or decrease in a certain area, in complete disregard for other circumstances. The point of this is that a key thing is that the CNN and machine learning itself still needs a large degree of human control as it cannot distinguish between important causal concepts in the data and data that happen to correlate well. This has to be done with a more qualitative and thorough approach. In this report we relied on the research and experience of others to establish a foundation of ancillary data to use as our input data, but this might be even more important when using a CNN as we have less control with the interpretation of the data. Our findings also point toward that this has to be considered heavily in the preparation of the data as can be seen from the experiment with water cover (cf. section 10.1). The models ability to deal with the simulated flood or water rise exemplified by creating a lake in the centre of Copenhagen did not work as expected. While the situation is constructed it should have provoked a sudden shift of population which our model did not do. In this case a causal relationship between water cover and population which is logical to us, should have been clear to the model as correlation, because correlation should exist between the two datasets provided for training the model. This highlights that not only the right data have to be chosen but these also have to be adjusted properly in regard to normalization as this has an impact on how the CNN relates them. In relation to the population value however this was as previously covered not possible to normalize as it would ruin the output prediction. The effects of this skewer should however theoretically lessen if trained over a longer time as the CNN should adjust the weights over time.

As such CNNs open up new possibilities for projecting future spatial population distribution by being able to take large amounts of data and analyse it in order to determine relationships between layers of data, identify unique local variances and applying them to a real world context. It does however also arguably increase the necessary background work needed to assure that the data input into the model is both meaningful, flawless and adjusted correctly as the model will automatically interpret this data and inherit its flaws and errors.

Chapter 9, 10 and this chapter thereby answers the sixth research question of *What knowledge can be gained from the results, experiments and tests?* which leads to the next chapter containing the conclusion.

Conclusion 12

Population growth represents a challenge to humanity, as to how resources are allocated and where people will and should live. The global population growth is expected to continue up until 2070 according to the IIASA medium population projection scenario, SSP2. This is especially true for developing countries with high population growth and likely challenges with both climate and economy, and predicting where people will settle in such countries can help drive a knowledge-based decision-making process to cope with the issues. The best results on predicting spatial population distributions have previously been done with geosimulations, but this project shows that CNNs can be used as an alternative through training on historic data to find population patterns.

Historic data for spatially distributed population has shown to be sparse despite multiple available datasets. Common to the datasets is that they all have several flaws. From low spatial and temporal resolution, inaccurate placement of population to limited spatial and temporal coverage. The dataset from GHSL was deemed best for the purpose of this project, and was accompanied with ancillary data from Copernicus, SEDAC, EEA and OpenDataSoft. The ancillary data covers water and Corine coverage, slope, roads and train stations that all have causality or correlation to population. Using this data in a CNN requires extensive processing from handling of big TIFF-files, spatial operations to creation of chunks from the final input grid to resemble images.

For the architecture of the CNN a sample network called PopNet was built. This consists of three convolutional and two dense layers, a relatively shallow neural network compared to the domain of image recognition. Nonetheless, we find it to be able to identify realistic patterns, at least on a national scale, and project future spatial population distribution scenarios. From the creation and evaluation of PopNet a number of tendencies and challenges could be seen. PopNet was able to predict the historic growth from 2000 to 2015, although with notable deviations, and was capable of producing population predictions within ±0,3 percent of the SSP2 scenario. However, it was unable to precisely hit the projected population numbers both historic and projected. A general shift of population towards the geographical west were observed in the results. In addition, it was found that the input data, both in regard to quality and preparation of values, had a large impact on how the CNN recognized and predicted future spatial population distribution. Arguably even more important compared to previous geosimulation techniques as the process and logic of the CNN is not directly programmable.

The findings- and creation of PopNet thereby addresses the research question; *How can a convolutional neural network be used to project future spatial population distribution and what results can be achieved?*. In addition there are various improvements that could be

made with the training data as well as the neural network architecture itself. Among those are making the model more aware of its surroundings, training on multiple countries to get more data while also establishing links and similarities between them and including other ancillary data like economy and political plans. Those alterations could improve the quality of the model so it will be able to make realistic predictions on a local scale.

Future Work 13

This section introduces future work that could be looked further into, related to the findings and conclusions of this report.

We have throughout the results and evaluation seen a pattern of population data shifting towards the geographical west, over each iteration of applying the model. This could be investigated further as to why it occurs and whether it happens, as we hypothesise, due to the core working of Tensorflow, the CNN or convoluational layers in general, or a combination of these and our own chunk cost function implementation.

Further improvement and testing of the existing nerual network architecture and resulting models could also be done. Changing the hyperparameters and training for longer time could potentially improve the models. This is because a deeper or wider network should improve the ability of the CNN to precisely recognize and identify patterns, but requires necessary hardware as these improvement are, as covered in the architecture and testing chapters, computationally intense. In relation to the architecture, it would also be interesting to investigate whether the model could be made to use an overlap between chunks or in another way, make the model, take the surroundings of a given chunk, into account. The reason for doing so, is as pointed out in the testing chapter 8.4, that the model have problems predicting smooth, natural values along borders between chunks.

Another possible improvement is adjusting the values of implemented data and adding further data. Doing so, adding for example spatial economic information, could in itself improve the CNN's ability to identify population distribution patterns. Another potential that could be further investigated, is the possibility of implementing categorical data. Adding categorical data could potentially teach the model to identify and work with different geographical divisions such as countries, municipalities of local city plans in a practical way. The effect of such an implementation could mean that you could train data on more than one country at a time and thereby gain more data that the model itself can use to find patterns across uniquely identifiable geographic areas. Thereby it should be able to uniquely identify patterns found in a municipality in Denmark as something special to this municipal ID, but also know that this municipality is more related to another Danish municipality, than a French municipality for example, as Danish municipalities could share the same category of country being Denmark.

As shown in the experiment conducted in the experiment section, that tested the effects of implementing a lake in the data, located in the middle of Copenhagen, the model does currently respond as intended to certain geographical changes. This could be further improved and tested in relation to if the model can be trained to understand the effect of environmental changes, for example by implementing constructed scenarios into the

training data. Using the example of a flooded Copenhagen, it should be possible for the model to recognize this pattern by training it on a scenario where a city is depopulated according to an increase of the water cover ancillary data value. If this is possible, the model could possibly be trained to react accordingly to spread of dessert, increasing temperatures or rising water levels in the same way.

The cover of PopNet is currently limited to Europe because of the coherence and cover of the ancillary data. It could however be interesting to look further into the possible differences that countries from different parts of the world have, what patterns the model would recognize and how both results and model would compare. The models trained for Denmark and France tested and evaluated in this report are different from each other which is likely a product of the patterns recognized from the historical data. Further investigation of this will however require more coherent global datasets or a number of identical local datasets.

Bibliography

Ahn et al., 2005. Namkee Ahn, Juha Alho, Herbert Brücker, Harri Cruijsen, Seppo Laakso, Jukka Lassila, Audronė Morkūnienė, Niku Määttänen and Tarmo Valkonen. *The use of demographic trends and long-term population projections in public policy planning at EU, national, regional, and local level.* Report prepared for the European Commission, Brussels: European Commission, 2005.

Banko and Brill, 2001. Michele Banko and Eric Brill. *Scaling to very very large corpora for natural language disambiguation.* pages 26–33, 2001. used: 26-05-2018.

Benenson and Torrens, 2004. Itzhak Benenson and Paul M Torrens. *Geosimulation: Automata-based modeling of urban phenomena.* John Wiley & Sons, 2004.

Bengio, 2012. Yoshua Bengio. *Practical recommendations for gradient-based training of deep architectures.* pages 437–478, 2012.

Bloom et al., 2008. David E Bloom, David Canning et al. *Population health and economic growth.* Health and growth, 53, 2008. used: 22-5-2018.

Bottou, 2010. Léon Bottou. *Large-scale machine learning with stochastic gradient descent.* pages 177–186, 2010. used: 13-03-2018.

Capitaine Train, 2015. Capitaine Train. *European train stations.* `https://public.opendatasoft.com/explore/dataset/european-train-stations/information/`, 2015. used: 14-06-2018.

Carson et al., 2016. Mark Carson, Armin Köhl, Detlef Stammer, ABA Slangen, CA Katsman, RSW Van de Wal, J Church and N White. *Coastal sea level changes, observed and projected during the 20th and 21st century.* Climatic Change, 134(1-2), 269–281, 2016.

Castelluccio et al., 2015. Marco Castelluccio, Giovanni Poggi, Carlo Sansone and Luisa Verdoliva. *Land use classification in remote sensing images by convolutional neural networks.* arXiv preprint arXiv:1508.00092, 2015. used: 06-05-2018.

Churchland and Sejnowski, 2016. Patricia S Churchland and Terrence J Sejnowski. *The computational brain.* MIT press, 2016.

CIESIN, 2016. CIESIN. *Gridded population of the world(gpw).v4.* `http://sedac.ciesin.columbia.edu/data/collection/gpw-v4`, 2016. used: 07-03-2018.

CIESIN, 2017. CIESIN. *Documentation for the Gridded Population of the World, Version 4 (gpwv4), Revision 10 Data Sets.* `http://sedac.ciesin.columbia.edu/downloads/docs/gpw-v4/gpw-v4-documentation-rev10.pdf`, 2017. used: 07-03-2018.

Copernicus, **2018**. Copernicus. *Pan-European.* https://land.copernicus.eu/pan-european, 2018. used: 08-03-2018.

Dietterich and Kong, **1995**. Thomas G Dietterich and Eun Bae Kong. *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms.* 1995. used: 26-05-2018.

Directive, **2007**. INSPIRE Directive. *Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).* Published in the official Journal on the 25th April, 2007. used: 21-5-2018.

Erhvervsstyrelsen, **2017**. Erhvervsstyrelsen. *Fingerplanen.* https://planinfo.erhvervsstyrelsen.dk/fingerplanen, 2017. used: 06-06-2018.

Eskin et al., **2002**. Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy and Sal Stolfo. *A geometric framework for unsupervised anomaly detection.* pages 77–101, 2002.

European Commission, **2015**. European Commission. *Datasets - GHS population grid, derived from GPW4, multitemporal (1975, 1990, 2000, 2015.* http://data.jrc.ec.europa.eu/dataset/jrc-ghsl-ghs_pop_gpw4_globe_r2015a, 2015. used: 07-03-2018.

European Commission, **2016**. European Commission. *Datasets - GHSL.* http://ghsl.jrc.ec.europa.eu/datasets.php#2016public, 2016. used: 07-03-2018.

European Environment Agency, **2012**. European Environment Agency. *European catchments and Rivers network system (Ecrins).* https://www.eea.europa.eu/data-and-maps/data/european-catchments-and-rivers-network#tab-gis-data, 2012. used: 04-05-2018.

Freire et al., **2016**. Sergio Freire, Kytt MacManus, Martino Pesaresi, Erin Doxsey-Whitfield and Jane Mills. *Development of new open and free multi-temporal global population grids at 250 m resolution.* https://agile-online.org/conference_paper/cds/agile_2016/shortpapers/152_Paper_in_PDF.pdf, 2016. used: 07-03-2018.

Ge et al., **2015**. Rong Ge, Furong Huang, Chi Jin and Yang Yuan. *Escaping from saddle points—online stochastic gradient for tensor decomposition.* pages 797–842, 2015. used: 13-03-2018.

Gemy, **2018**. Mahmoud Gemy. *Tensorflow-Project-Template.* https://github.com/MrGemy95/Tensorflow-Project-Template, 2018. used: 08-03-2018.

Goodfellow et al., **2016**. Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

Goodfellow et al., **2013**. Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud and Vinay D. Shet. *Multi-digit Number Recognition from Street View Imagery*

using Deep Convolutional Neural Networks. CoRR, abs/1312.6082, 2013. URL http://arxiv.org/abs/1312.6082.

Google, **2018a**. Google. *Installing Tensorflow.* https://www.tensorflow.org/install/, 2018. used: 09-03-2018.

Google, **2018b**. Google. *Tutorial Tensorflow.* https://www.tensorflow.org/tutorials/layers, 2018. used: 13-03-2018.

Gupta, **2017**. Abhinav Gupta. *Revisiting the Unreasonable Effectiveness of Data.* https://ai.googleblog.com/2017/07/revisiting-unreasonable-effectiveness.html, 2017. used: 29-05-2018.

He et al., **2016**. Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Deep residual learning for image recognition.* pages 770–778, 2016.

Herold et al., **2003**. Martin Herold, Noah C Goldstein and Keith C Clarke. *The spatiotemporal form of urban growth: measurement, analysis and modeling.* Remote sensing of Environment, 86(3), 286–302, 2003.

Hijmans et al., **2015**. Robert Hijmans, Julian Kapoor, John Wieczorek, Nel Garcia, Arnel Rala Aileen Maunahan and Axel Mandel. *Global Administrative Areas.* http://gadm.org/, 2015. used: 07-03-2018.

Hinton and Sejnowski, **1999**. Geoffrey E Hinton and Terrence Joseph Sejnowski. *Unsupervised learning: foundations of neural computation.* MIT press, 1999.

Hubel and Wiesel, **1962**. David H. Hubel and Torsten N. Wiesel. *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* pages 106–154, 1962. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1359523/pdf/jphysiol01247-0121.pdf. used: 07-02-2018.

IIASA, **2014**. International Institute for Applied Systems Analysis IIASA. *World population likely to peak by 2070.* http://www.iiasa.ac.at/web/home/about/news/20141023-population-9billion.html, 2014. used: 02-03-2018.

IIASA, **2018**. International Institute for Applied Systems Analysis IIASA. *SSP Public Database version 1.1.* https://tntcat.iiasa.ac.at/SspDb/dsd?Action=htmlpage&page=about, 2018. used: 05-03-2018.

Jones and O'Neill, **2016**. Bryan Jones and BC O'Neill. *Spatially explicit global population scenarios consistent with the Shared Socioeconomic Pathways.* Environmental Research Letters, 11(8), 084003, 2016.

KC and Lutz, **2017**. Samir KC and Wolfgang Lutz. *The human core of the shared socioeconomic pathways: Population scenarios by age, sex and level of education for all countries to 2100.* Global Environmental Change, 42, 181–192, 2017.

Keßler and Marcotullio, 2017. Carsten Keßler and Peter J. Marcotullio. *A Geosimulation for the Future Spatial Distribution of the Global Population.* (ISBN: 978-3-540-75400-8), 2017. used: 31-01-2018.

Kellher, 2016. Adam Kellher. *Understanding Bias: A Pre-requisite For Trustworthy Results.* `https://medium.com/causal-data-science/understanding-bias-a-pre-requisite-for-trustworthy-results-ee590b75b1be`, 2016. used: 26-05-2018.

Keskar et al., 2016. Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy and Ping Tak Peter Tang. *On large-batch training for deep learning: Generalization gap and sharp minima.* arXiv preprint arXiv:1609.04836, 2016.

Kingma and Ba, 2014. Diederik P Kingma and Jimmy Ba. *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

Krizhevsky et al., 2012. Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. *ImageNet Classification with Deep Convolutional Neural Networks.* pages 1097–1105, 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

Learned-Miller, 2014. Erik G Learned-Miller. *Introduction to Supervised Learning.* I: Department of Computer Science, University of Massachusetts, 2014.

LeCun et al., 1998. Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. *Gradient-based learning applied to document recognition.* pages 2278–2324, 1998. used: 07-02-2018.

Lloyd et al., 2017a. Christopher T. Lloyd, Alessandro Sorichetta and Andrew J. Tatem. *High resolution global gridded data for use in population studies.* (DOI: 10.1038/sdata.2017.1), 2017. used: 31-01-2018.

Lloyd et al., 2017b. Christopher T Lloyd, Alessandro Sorichetta and Andrew J Tatem. *High resolution global gridded data for use in population studies.* Scientific data, 4, 170001, 2017.

Lutz and Samir, 2010. Wolfgang Lutz and KC Samir. *Dimensions of global population projections: what do we know about future population trends and structures?* Philosophical Transactions of the Royal Society B: Biological Sciences, 365(1554), 2779–2791, 2010.

Lutz and Skirbekk, 2017. Wolfgang Lutz and Vegard Skirbekk. *How education drives demography and knowledge informs projections.* World Population & Human Capital in the Twenty-First Century: An Overview, 2017.

Lutz et al., 2014a. Wolfgang Lutz, William P. Butz and KC Samir. *World Population and Human Capital in the Twenty-First Century.* ISBN: 9780198703167. Oxford University Press, 2014.

Lutz et al., 2014b. Wolfgang Lutz, William P. Butz, KC Samir, Warren Sanderson and Sergei Scherbov. *9 billion or 11 billion? The research behind new population*

projections. http://blog.iiasa.ac.at/2014/09/23/9-billion-or-11-billion-the-research-behind-new-population-projections/, 2014. used: 23-03-2018.

Malthus, **1798**. Thomas R. Malthus. *An Essay on the Principle of Population*. 1798. used: 04-02-2018.

Mishkin, **2016**. Dmytro Mishkin. *caffenet-benchmark - Complexity*. https://github.com/ducha-aiki/caffenet-benchmark/blob/master/Complexity.md, 2016.

Mishkin et al., **2016**. Dmytro Mishkin, Nikolay Sergievskiy and Jiri Matas. *Systematic evaluation of CNN advances on the ImageNet*. CoRR, abs/1606.02228, 2016. URL http://arxiv.org/abs/1606.02228.

Nagle et al., **01 2014**. Nicholas Nagle, Barbara Buttenfield, Stefan Leyk and Seth Spielman. *Dasymetric Modeling and Uncertainty*. 104, 80–95, 2014.

Nielsen, **2017**. Michael Nielsen. *Neural Networks and Deep Learning*. http://neuralnetworksanddeeplearning.com/chap1.html, 2017. used: 23-03-2018.

O'Neill et al., **2001**. Brian C O'Neill, Deborah Balk, Melanie Brickman and Markos Ezra. *A guide to global population projections*. Demographic research, 4, 203–288, 2001.

O'Neill et al., **2017**. Brian C O'Neill, Elmar Kriegler, Kristie L Ebi, Eric Kemp-Benedict, Keywan Riahi, Dale S Rothman, Bas J van Ruijven, Detlef P van Vuuren, Joern Birkmann, Kasper Kok et al. *The roads ahead: narratives for shared socioeconomic pathways describing world futures in the 21st century*. Global Environmental Change, 42, 169–180, 2017.

Patz et al., **2005**. Jonathan A Patz, Diarmid Campbell-Lendrum, Tracey Holloway and Jonathan A Foley. *Impact of regional climate change on human health*. Nature, 438 (7066), 310, 2005.

Pentreath, **2017**. Nick Pentreath. *Feature Hashing for Scalable Machine Learning*. 2017. URL https://dzone.com/articles/feature-hashing-for-scalable-machine-learning. used: 23-05-2018.

Pijanowski et al., **2013**. Bryan C. Pijanowski, AminTayyebi, Jarrod Doucette, Burak K.Pekin, David Braun and James Plourde. *A big data urban growth simulation at a national scale: Configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing (HPC) environment*. pages 250–268, 2013. used: 09-02-2018.

Population Reference Bureau, **2001**. Population Reference Bureau. *Understanding and Using Population Projections*. https://www.prb.org/understandingandusingpopulationprojections/, 2001. used: 24-04-2018.

Qi et al., 2016. Charles R Qi, Hao Su, Kaichun Mo and Leonidas J Guibas. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.* arXiv preprint arXiv:1612.00593, 2016. used: 03-12-2017.

Robinson et al., 2017. Caleb Robinson, Fred Hohman and Bistra Dilkina. *A Deep Learning Approach for Population Estimation from Satellite Imagery.* pages 47–54, 2017.

Simonyan and Zisserman, 2014a. Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* CoRR, abs/1409.1556, 2014. URL `http://arxiv.org/abs/1409.1556`.

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

Springenberg et al., 2014. Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox and Martin Riedmiller. *Striving for simplicity: The all convolutional net.* arXiv preprint arXiv:1412.6806, 2014.

Sutton and Barto, 2017. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press Cambridge, 2017.

Szegedy et al., 2015. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich et al. *Going deeper with convolutions.* 2015.

Tang et al., 2006. Zaiyong Tang, Caroline W Leung and Kallol Bagchi. *Improving population estimation with neural network models.* pages 1181–1186, 2006.

Tatem, 2017. Andrew J Tatem. *WorldPop, open data for spatial demography.* Scientific data, 4, 170004–170004, 2017.

The Guardian, 2015. The Guardian. *How Berliners refused to give Tempelhof airport over to developers.* The Guardian, 2015. URL `https://www.theguardian.com/cities/2015/mar/05/how-berliners-refused-to-give-tempelhof-airport-over-to-developers`. used: 21-5-2018.

Tobler, 1970. W. R. Tobler. *A Computer Movie Simulating Urban Growth in the Detroit Region.* Economic Geography, 46(sup1), 234–240, 1970. doi: 10.2307/143141. URL `http://www.tandfonline.com/doi/abs/10.2307/143141`.

Torrens and Benenson, 2005. Paul M Torrens and Itzhak Benenson. *Geographic automata systems.* International Journal of Geographical Information Science, 19(4), 385–412, 2005.

United Nations, 2017a. Department of Economic & Social Affairs United Nations. *World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100.* `https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html`, 2017.

United Nations, **2018**. Department of Economic & Social Affairs United Nations. *FAQ*. https://esa.un.org/unpd/wpp/General/FAQs.aspx, 2018. used: 10-11-2017.

United Nations, **2017b**. Department of Economic & Social Affairs Population Division United Nations. *World Population Prospects: The 2017 Revision*. https://esa.un.org/unpd/wpp/DataQuery, 2017. used: 02-03-2018.

United Nations, **2017c**. Department of Economic & Social Affairs Population Division United Nations. *World Population Prospects: The 2017 Revision, Methodology of the United Nations Population Estimates and Projections*. Working Paper No. ESA/P/WP.250, 2017.

van Vuuren et al., **2017**. Detlef P van Vuuren, Keywan Riahi, Katherine Calvin, Rob Dellink, Johannes Emmerling, Shinichiro Fujimori, Samir KC, Elmar Kriegler and Brian O'Neill. *The Shared Socio-economic Pathways: Trajectories for human development and global environmental change*. Global Environmental Change, 42, 148–152, 2017.

Walia, **2017**. Anish Singh Walia. *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent*. https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradien 2017. used: 27-03-2018.

Willmott and Matsuura, **2005**. Cort J Willmott and Kenji Matsuura. *Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance*. Climate research, 30(1), 79–82, 2005.

WorldPop, **2016a**. WorldPop. *What is WorldPop?* http://www.worldpop.org.uk/, 2016. used: 07-03-2018.

WorldPop, **2016b**. WorldPop. *Data - FAQ*. http://www.worldpop.org.uk/data/faq/, 2016. used: 07-03-2018.

WorldPop, **2016c**. WorldPop. *WorldPop Methods - Mapping Populations*. http://www.worldpop.org.uk/data/methods/, 2016. used: 07-03-2018.

Zeiler, **2015**. Matthew D. Zeiler. *Visualizing and Understanding Deep Neural Networks*. https://www.youtube.com/watch?v=ghEmQSxT6tw, 2015. used: 18-5-2018.

Zeiler and Fergus, **2013**. Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. CoRR, abs/1311.2901, 2013.

Zhang et al., **2018**. Han Zhang, Ian Goodfellow, Dimitris Metaxas and Augustus Odena. *Self-Attention Generative Adversarial Networks*. arXiv preprint arXiv:1805.08318, 2018.