Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis BSc2 Econometrics/Economics

Micro-Spatial Economic Analysis in the Netherlands Using Convolutional Neural Networks

Martijn Quist (457631)

Supervisor: dr. Naghi, A.A.
Second assessor: Koobs, S.J.
Date final version: 2nd July 2023

Abstract

This paper explores the use of convolutional neural network (CNN) models for predicting income and population levels in fine spatial scale urban areas in the Netherlands. Both models that are pre-trained on the US (transfer learning) and re-trained models are used for obtaining predictions with Landsat 7 satellite imagery as input. I also research whether CNN models can capture the positive agglomeration effect of population density on income per capita. Transfer learning predictions have poor accuracy with highly negative R^2 values between -1.906 for large image income predictions and -42.11 for large image population predictions. Newly trained models perform better with R^2 values lower in magnitude between -0.539 and -4.1784. Furthermore, the MSEs of re-trained models are comparable to similar research papers. Thus, re-training provides more accurate predictions than transfer learning for this application. Pre-trained CNN model predictions of small areas seem to best approximate true agglomeration effects with an income per capita increase of 0.15% per 1% increase in population compared to a true value of 0.10%. However, none of the predictions come close to the true effect. Therefore, CNN model predictions are not suitable for analysis of agglomeration effects in fine spatial scale urban areas in the Netherlands.

Keywords— Convolutional neural network, satellite imagery, spatial economics, agglomeration effects, the Netherlands

Contents

1 Introduction

In the past decades machine learning has seen an enormous increase in power for a wide variety of applications, such as image recognition, artificial intelligence, and self-driving cars. One area in which machine learning has been increasingly used is economics. While machine learning initially was mostly used for financial forecasting purposes in economics, its use has also been found for other economic applications (Gogas and Papadimitriou, 2021). One such application is in spatial economics, which analyses economic phenomena at different spatial scales (e.g., national, regional, city/neighbourhood level). Examples of spatial economics analysis include the location decision of firms, how different factors affect regional inequalities, and the benefits and disadvantages of agglomeration.

A machine learning method that has benefited from the increase in available data and increased computer processing power is the convolutional neural network (CNN). Research in spatial economics leverages this benefit by using CNNs with satellite imagery as input to predict economic variables. Doll et al. (2006), for example, find that night-time lights are highly correlated with gross regional product in the US and Western Europe. More recent research by Khachiyan et al. (2022b) shows that daytime satellite imagery performs better than night-time lights imagery. They predict income and population level and change over time at small geographic scale in urban areas in the United States, obtaining high R^2 values.

Accurate prediction of income and population using satellite imagery can be of great benefit for economic analysis. Countries that infrequently perform a population census, for example, may benefit from such predictions. Furthermore, research into agglomeration effects in urban areas may be of interest for urban planning. Income and population predictions could be used for such research, where population could serve as a proxy for the degree of agglomeration. While the applications are interesting, not much research has been performed on this topic, apart from Khachiyan et al. (2022b). Specifically, it is not known how this method performs in other countries than the US. Therefore, in this paper I analyse the use of a method similar to Khachiyan et al. (2022b) to research the use of satellite imagery with CNN models to predict income and population levels in the Netherlands.

The Netherlands is particularly interesting for this application as it has urban areas that can be described as polycentric due to cities lying relatively close together. Polycentricity refers to a connection of smaller urban areas that form a larger urban whole. This connection is based both on geographical distance and shared economic activity. The *Randstad* area in the Netherlands is a prime example of this, consisting of the cities Amsterdam, Rotterdam, the Hague, and Utrecht (Nadin and Zonneveld, 2020). Furthermore, the Netherlands is a rather densely populated country with 522 inhabitants per square kilometre compared to 37 in the US in 2022 (Roser and Ortiz-Ospina, 2022). Therefore, urban planning methods may differ between the two countries. This may result in different spatial features, such as shapes and edges that would be captured by CNNs when predicting income and population levels.

In this paper I investigate how accurately satellite imagery in combination with CNNs can predict income and population levels in urban areas in the Netherlands. CNNs are computationally demanding to train, sometimes taking up to months depending on the amount of training data. A method called transfer learning, which uses pre-trained models on new data can be used to significantly reduce computational demand. Therefore, I also investigate how transfer learning compares to re-training CNN models with respect to the accuracy of income and population predictions. As mentioned previously, the analysis of agglomeration effects could be interesting for policy makers in urban planning. CNN model predictions of income and population may facilitate this analysis where income and population data is unavailable. Thus, I finally investigate if CNN model predictions of income and population are able to capture the agglomeration effects that are typically present in urban areas.

Using both CNN models that are pre-trained by Khachiyan et al. (2022b) and newly trained CNN models, I make predictions of income and population levels for urban areas in the Netherlands. This is done using income and population data for small (1.2km) and large (2.4km) square areas. Multi-spectral satellite imagery captured by the Landsat 7 satellite is used as input for the models. This is relatively high resolution imagery for this purpose with pixels of 30 by 30 metres. R^2 and mean squared error (MSE) values are calculated to evaluate the predictions. Predictions are also evaluated by making plots of true versus predicted values to detect possible skewness. Furthermore, agglomeration effects are analysed by regressing log income per capita on log population level using linear regression. Besides the aforementioned methods, the methodology of Khachiyan et al. (2022b) relevant to the present research is replicated. Their CNN model predictions are recalculated and the R^2 values are computed for income and population and for small and large images. The replicated results are compared to their reported results.

The replicated results are almost similar to the results reported by Khachiyan et al. (2022b) with the R^2 differing at most 0.007. Furthermore, transfer learning applied to the Netherlands results in very negative R^2 values and high MSEs. The least accurate predictions are those of large image population with an R^2 of -42.1120. The most accurate predictions are those of large image income with an R^2 of -1.9061. The re-trained CNN models also provide negative R^2 values, although much lower in magnitude. The lowest R^2 in magnitude is -0.5391 for large image population predictions and the highest in magnitude -4.1784 for small image income predictions. Both transfer learning and re-trained model predictions are consistently too low. Agglomeration effects in terms of the effect of population on income per capita are closest to the true value for small image transfer learning predictions with a 0.148% increase per 1% increase in population. The true effect is 0.101% for small imagery. The effects of other predictions are either not significant or significantly negative.

The paper is structured as follows. First, a review of the relevant literature is provided in section 2. Then, a description of the necessary data is given in 3. This includes data sources, collection methods, and descriptive statistics of satellite imagery and economic variables. Next, section 4 provides an explanation of the methodology. This includes a brief explanation of CNNs. After the results are given in section 5, a conclusion and discussion are provided in section 6.

2 Literature Review

The use of satellite imagery in combination with CNNs for spatial economic research has seen significant growth in recent years. This method has opened up new opportunities for analysing and understanding various economic phenomena. Despite being a relatively novel method, the research conducted thus far has yielded promising results. This literature review aims to provide a comprehensive overview of the existing research in this domain while identifying potential areas for further investigation.

Satellite imagery has some key benefits, such as wide geospatial coverage, easy access, and high spatial resolution. Therefore, it is increasingly being used in spatial economic research. Donaldson and Storeygard (2016) provide an overview of the use of satellite imagery in the field of economics. They highlight the increasing feasibility of using this imagery due to higher processing power and lower costs of deploying smaller satellites. One type of satellite imagery that has been used extensively in economic research is night-time light imagery. Doll et al. (2006) find that night-time lights are highly correlated with gross regional product in the US and western Europe. However, night-time light imagery poses the problem of over-saturation of pixels at finer geospatial scales (He et al., 2014). Therefore, applications with night-time light imagery are not suitable for small-scale spatial economic analysis.

One of the methods for spatial economic analysis using satellite imagery is the convolutional neural network (CNN). A CNN is able to extract features from high-dimensional input data for a multitude of purposes. Jean et al. (2016) use high-resolution satellite imagery with CNNs to predict consumption expenditure and assets in five African countries. Their model is able to explain up to 75% of variation in local-level economic outcomes. Lehnert et al. (2023) show that CNNs using daytime satellite imagery have potential for proxying economic activity. Their model is able to predict economic activity more accurately at small regional levels than models using night-time lights. Khachiyan et al. (2022b) use daytime satellite imagery to predict level and change of income and population in the US. Their results outperform similar models, with R^2 levels of 0.85 and 0.91 in level and 0.32 and 0.46 in decadal changes for income and population, respectively.

Khachiyan et al. (2022b) use satellite imagery as an input in CNNs in order to predict, among others, income and population levels and in the continental United States. They do this for square areas of 1.2km and 2.4km per side using satellite imagery from the Landsat 7 satellite. This imagery consists of pixels with sizes of 30m by 30m. The pixels corresponding to urbanized US census blocks (600 to 3,000 residents) are collected. The labels (population and income) are constructed from the US census in 2000, 2010, and 2020. For level predictions, data from 2000 and 2010 is pooled and used for training and testing. Their income predictions achieve R^2 values of 0.8374 and 0.7494 for large and small imagery, respectively. For population, they achieve R^2 values of 0.8684 and 0.7492 for large and small imagery, respectively. No similar research has been performed for a different country than the US. Therefore, it is worth exploring their method in a different country, such as the Netherlands, using their trained models in a process called transfer learning.

Transfer learning is a method in machine learning where a pre-trained neural network model is used for a new task, instead of training a new model. This approach has gained popularity due to its ability to leverage the knowledge obtained from training a model and apply it to a task with limited label data (Tajbakhsh et al., 2016). Transfer learning has shown to be effective in achieving similar or even better performance then re-training CNNs from scratch. For example, in the context of tumor detection Alla and Athota (2022) achieve high prediction accuracy when using transfer learning with CNNs to classify brain tumors based on MRI images. Albahli and

Albattah (2021) find that transfer learning is specifically useful when dealing with limited data. They use transfer learning with CNN models to diagnose COVID-19 in patients using X-ray images. Overall, transfer learning limits the need for large data sets and can improve accuracy and reduce training time. However, studies on the use of transfer learning methods for economic applications are scarce.

Besides simply predicting economic variables using CNNs, predictions can also be used to see whether CNNs can capture certain characteristics of economic theories. A relevant economic theory in the context of urban areas, income, and population is that of agglomeration effects. Agglomeration effects refer to the positive externalities that arise when economic activities concentrate in a particular geographic area. Noted effects include knowledge spillovers, economies of scale, and access to a skilled workforce. These effects are caused by agglomeration of people and firms, but also attract people firms to establish themselves in these areas (Feldman, 1999). In their paper, "Consumer City", Glaeser et al. (2001) highlight different types of premia that present themselves in cities: an urban productivity premium, an urban rent premium, and an urban amenity premium. The urban productivity premium, measured in wages, refers to the higher productivity of firms in urban areas. This is the result of access to better ideas and technology and lower transportation costs (Glaeser et al., 2001). Thus, it is suggested that urban areas have higher income levels per capita than non-urban areas. CNN predictions of income and population could be used to calculate the effects of population on income per capita. This may be useful for assessing agglomeration effects in urban areas using satellite imagery. A thorough search of relevant literature has not yielded any research into whether CNN predictions capture this agglomeration effect through predictions of income and population levels.

In conclusion, the use of satellite imagery together with CNNs has emerged as a promising approach for spatial economic research. Both night-time light and daytime satellite imagery have been used for this purpose, with the latter yielding higher prediction accuracy. Different economic outcomes, such as economic activity and income have been explored. Furthermore, in different fields, such as diagnostic imaging, transfer learning has shown high predictive accuracy at significantly lower computational cost. As these methods are relatively novel, there remain many potential avenues for further research.

3 Data

This section describes the data that is used to replicate part of the research by Khachiyan et al. (2022b), perform transfer learning, and re-train CNNs for the Netherlands. This is done using satellite imagery, data on income and population, and code and previously trained model weights of Khachiyan et al. (2022b).

3.1 Satellite Imagery

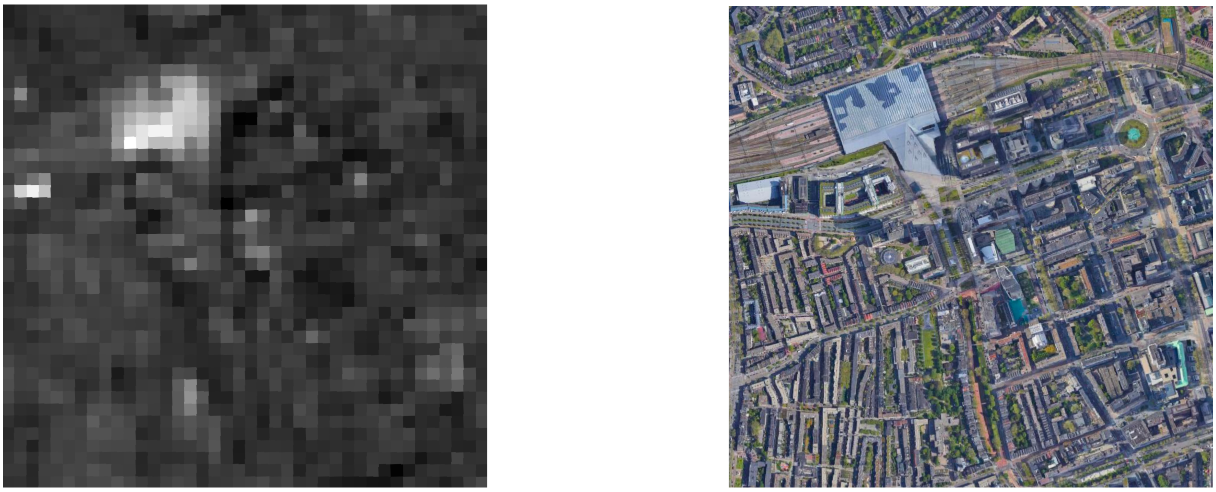
For the replication of Khachiyan et al. (2022b), satellite imagery of the urban areas of the continental United States is collected from their online repository (Khachiyan et al., 2022a). This data consists of the seven spectral bands collected by the Landsat 7 satellite with a spatial resolution of 30 metres; 1 blue, 1 green, 1 red, 1 near-infrared, 1 short-wave infrared, 1 thermal, and 1 mid-infrared. This imagery was collected using the Google Earth Engine (GEE) (Gorelick et al., 2017) API in Python for the years 2000 and 2010. Seeing as the presence of clouds may inhibit feature extraction and decrease prediction accuracy, annual composites were created using cloud-free images in the period May-August.

For the re-training of CNN models for the Netherlands Landsat 7 satellite imagery was also collected for urban areas using GEE (Gorelick et al., 2017). Specifically, similar to Khachiyan et al. (2022b), Landsat 7 Collection 1 Tier 1 8-day TOA Reflectance Composite imagery is collected. Collection 1 Tier 1 refers to the data quality, where Tier 1 is the best quality. 8-day TOA Reflectance Composite is data that is collected by making a composite of Earth surface reflectance using all data collected over each 8 days (TOA stands for top of atmosphere). Urban areas are described using a shapefile obtained from Statistics Netherlands (Statistics Netherlands, 2023), which contains squares (500 x 500 metres) (from now on: 500m squares) with multiple attributes. The attribute amount of addresses ("aantal_adressen") is used to determine urban areas. This is done using the geographic information system application QGIS (QGIS Development Team, 2023). Statistics Netherlands considers an area urban if it contains more than 1500 private addresses per square kilometre, translating to 375 addresses per 500m square. Urban 500m squares are filtered and merged, after which edges are smoothed to remove jaggedness. Areas consisting of 4 contiguous squares or less are removed, as both the small (1.2km x 1.2km) and large (2.4km x 2.4km) square satellite images that will be collected will exceed their borders. The resulting urban areas of interest can be seen in figure 1 below.

Figure 1: Urban areas in the Netherlands consisting of merged squares (500m) with more than 375 addresses per square and more than 4 squares per contiguous area.

Satellite imagery is collected for these urban areas. This is done for the year 2020 only, as 2020 is the only year with convenient reports of income and population at fine spatial scale as will be described in the following subsection. The collected imagery is separated into small (1.2km) and large (2.4km) square images. Seeing as each pixel is 30m x 30m, small images are 40 x 40 pixels and large images are 80 x 80 pixels. An example of a small Landsat 7 image containing the blue band in grayscale of part of the city centre of Rotterdam can be seen in figure 2 below. This figure also contains high definition satellite imagery of the same area for comparison. This area contains the central train station of Rotterdam in the upper left corner.

Clearly, the train station emits more light than the surrounding areas, which is evident from the whiter pixels. The data of each satellite image used as input in the CNN models consist of the detected intensity of each pixel for the given spectral band.



(a) Landsat 7

(b) Satellite

Figure 2: Landsat 7 (a) blue band (visualised in greyscale) and satellite imagery (b) of a small image (1.2km x 1.2km) of the city centre of Rotterdam, the Netherlands.

Note: Landsat 7 Collection 1 Tier 1 8-day TOA Reflectance Composite imagery is used.

3.2 Economic Variables

The shapefile with 500m squares downloaded from Statistics Netherlands (Statistics Netherlands, 2023) also contains the variables population level (attribute "aantal_inwoners"), yearly average income per household in thousands of euros (attribute "gemiddeld_inkomen_huishouden") , and number of private households (attribute "aantal_part_huishoudens") as recorded on January 1, 2020 for each 500m square. The models trained by Khachiyan et al. (2022b)predict total population and total yearly personal income per small and large image. Therefore, total yearly income is calculated for each 500m square in the Netherlands by multiplying average income per household by the number of households. All income data used by Khachiyan et al. (2022b) is in 2012 US dollars, such that their CNN models predict income in 2012 US dollars. Therefore, the 2020 euros are converted to 2012 US dollars using the US inflation rates between 2012 and 2020 (MacroTrends, 2023)and the euro to US dollar exchange rate on January 1, 2020 (European Central Bank, 2023). Some descriptive statistics of income and population for the filtered urban 500m squares can be found in table 1.

Table 1: Descriptive statistics of income and population data per 500m urban square in the Netherlands in 2020.

Variable	**Mean**	**St. Dev**	**Min**	**Max**
Population	1.50×10^3	6.92×10^3	4.30×10^2	7.21×10^2
Yearly income per household (€)	3.08×10^4	6.20×10^3	5.50×10^3	9.59×10^4
Number of households	7.25×10^2	4.11×10^2	3.80×10^2	4.81×10^3

Notes: A 500m square is considered urban if it contains more than 375 addresses. Data is collected from Statistics Netherlands (Statistics Netherlands, 2023).

3.3 Weights and Code

In order to replicate their methodology and apply the CNN models trained by Khachiyan et al. (2022b) to the Netherlands, their model weights are downloaded from their online repository (Khachiyan et al., 2022a). Replication is performed using part of the Python code provided in their online repository. Their README file is used as guideline for replication. This is the code that makes predictions for the relevant image areas in the US. For transfer learning and re-training, their code from the online repository is slightly adapted for the new dataset. This includes both Python and Stata code. A combination of GEE, Google Drive, and ArcGIS Python APIs are used. ArcGIS (ESRI, 2023a) is a geographical information system application similar to QGIS.

4 Methodology

In this section, I first provide an explanation of the machine learning method CNN. Next, I delve into the method used for replication of Khachiyan et al. (2022b). Subsequently, I explore the utilisation of their weights for transfer learning predictions in the Netherlands. Moreover, an explanation of the re-training of CNN models for predictions in the Netherlands is given. Lastly, I investigate the agglomeration effects in urban areas using the predicted and true income and population data.

4.1 Convolutional Neural Network

A neural network is a machine learning method that is based on the functioning of the neurons in the human brain. A CNN is a specific type of neural network that can be used to process visual data, such as images and videos. A CNN does this by extracting meaningful features from this data using connections of neurons between different layers. A CNN distinguishes itself from a regular artificial neural network (ANN) through convolutional layers. A convolutional layer applies a set of small filters or kernels, sliding them over the pixels of the input image. For each position on the pixels, the filters perform a mathematical operation between the pixel values and the filter's weights. This results in feature maps, which are fed to the following layers. Any number of feature layers can be output to the following layer. Another type of layer present in CNNs is the pooling layer. Pooling layers reduce the dimensionality of the data and extract the most relevant data. The most common pooling operation is max pooling, which outputs the maximum value of a subset of the input. Pooling helps reduce computational complexity, while retaining important information. An example of the convolutional part of a CNN can be seen in figure 3. The two-dimensional input layer is convoluted to eight feature maps in the following layer. Then, following a pooling operation, the dimensionality is reduced from 128 x 128 neurons to 64 x 64 neurons. After a number of convolutional and pooling layers, the data is vectorized and passed to the dense layers.

The fully connected, or dense, layers of a CNN resemble the typical ANN. In these layers each neuron is connected to the neurons in the following layer. Each connection has an associated weight. The input of a neuron consists of the output of each previous neuron multiplied by their respective weight plus a bias term. These operations are continued until the output layer

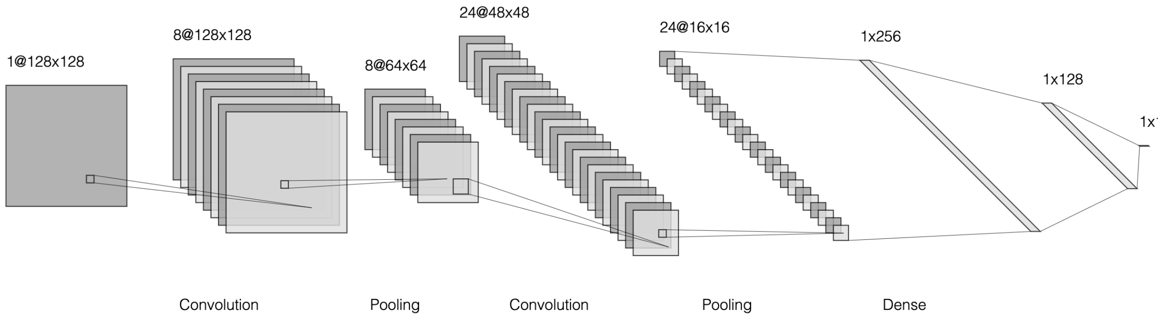


Figure 3: Example of a convolutional neural network with (from left to right) two convolutional and two pooling layers and two hidden layers with one output.

Notes: The numbers above each set of layers in the convolutional and pooling layers denote the dimension; e.g., 24@48x48 denotes 24 layers with 48 by 48 pixels each. The numbers above the fully connected layers denotes the number of neurons; e.g., 1x128 denotes a layer with 128 neurons.

is reached, which can output any number of values based on the application at hand. The input of each neuron, the weighted sum of outputs from the previous layer, is typically transformed using an activation function. A widely used activation function is the rectified linear unit (ReLU) function: $f(x) = \max(0, x)$, where x denotes the input. ANNs are universal function approximators (Sonoda and Murata, 2017), meaning they have the ability to represent any function, given appropriate weights and architecture. The dense layers of the CNN in figure 3 also represent an ANN with one output. Figure 4, a different example of an ANN, more clearly shows the connections between neurons in dense layers.

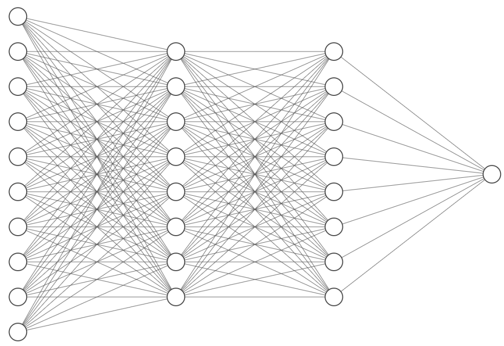


Figure 4: Example of an artificial neural network with (from left to right) 10 inputs, 2 dense layers and 1 output neuron.

Training of CNNs is performed by optimising the weights and biases. Using labelled training data, weights and biases are updated by minimizing a loss function. This is typically the MSE between the outputs and the true values. This method, called backpropagation, iteratively updates the weights by propagating the error backwards through the network. Different hyperparameters may be used to control this process: The learning rate determines the strength of updating of weights and biases; the dropout rate determines if and how many neurons are randomly removed to reduce dimensionality; regularisation, such as L2 regularisation, may be used to regularise the weights; the number of epochs determines how often the entire training set is used to optimise the weights. Testing different hyperparameters through validation is an important step in training CNNs. After training, a test set is used to determine the accuracy of the CNN for the relevant application.

4.2 Replication

The part of the research by Khachiyan et al. (2022b) that is relevant to answering the research question of the present research is replicated. This includes level predictions of income and population for small and large images. Khachiyan et al. (2022b) train CNNs using a specialized University of California computing cluster and estimate re-training would take more than a month if run on a modern super-computer cluster. Therefore, rather than re-training CNNs for the US, pre-trained model weights downloaded from their online repository (Khachiyan et al., 2022a) for prediction purposes. Furthermore, the imagery used in the test subset of images is downloaded from this repository, rather than re-extracting it. This imagery includes both small and large images. Together, the model weights and test images are used to obtain predictions of income and population level. This is done using the code from their online repository. Predictions are exported as csv files and keyed by image id. The true values, downloaded from their repository are used to calculate R^2 values using the formula

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}, \tag{1}$$

where y_i denotes the true value, $\hat{y}_i$ denotes the predicted value, and $\bar{y}$ denotes the mean of all true values. The R^2 values are presented in a table alongside the values reported by Khachiyan et al. (2022b) for comparison. In principle the replicated and reported values should be equal. MSE values are also calculated for comparison to the re-trained model predictions. Khachiyan et al. (2022b) do not calculate MSE values, so no reported values can be provided.

4.3 Transfer Learning and Re-Training

This section provides an explanation of the method of transfer learning and training of CNNs to create predictions for the Netherlands. Firstly, the preparation of necessary data for the Netherlands is explained. Next, the details of transfer learning are described. Finally, an outline of the process of training new CNN models is given.

4.3.1 Geo-Spatial Data

Firstly, data from Statistics Netherlands, consisting of 500m squares in the Netherlands with various attributes is downloaded in geopackage format (Statistics Netherlands, 2023) and loaded into QGIS. These are filtered, such that only squares with more than 375 addresses per square remain using the filter function. The remaining squares are merged using the dissolve function. Any empty space in remaining areas is filled with the delete holes function and jagged edges are removed with the smooth function. Areas consisting of four or less squares are removed by filtering out areas less than one square kilometre. The remaining areas constitute the urban areas and are exported as a shapefile for further imagery extraction and analysis.

Using the GEE API (Gorelick et al., 2017) in Pycharm, Landsat 7 imagery is extracted over the urban areas shapefile for the year 2020. This is done by creating a cloud-free composite of the available images from May 1, 2020 until August 30, 2020. The year 2020 is chosen for analysis, as income per household, a crucial variable in the present research, is not available in years before and after. Furthermore, a cloud mask is used so that images are not disturbed by light reflectance from clouds. This would inhibit the ability of a CNN in making accurate predictions. The collected imagery per urban area is exported to a google drive file in TFRecord format, an efficient format, which eases use of Python packages in further steps. This is done both for small and large imagery, which are keyed by image id and latitude and longitude. Next, the imagery is downloaded from the Google Drive folder using Google Drive API (Gwak and Nabel, 2016). The data is stored in HDF5 files for small and large imagery separately. HDF5 files are useful for storing large amounts of images in a single file. After downloading imagery in the correct formats, population and income labels must be assigned to the images.

For the assignment of income and population labels to the images, the shapefile with 500m squares is used again. This file contains population, number of private households, and average income per household. Total income per square is calculated by multiplying number of private households and average income per household. First the overlap between the imagery and the 500m squares is calculated. This results in a file containing an area and a percentage overlap between the two. This is done using the ArcGIS Pro package arcpy (ESRI, 2023b). Then, based on the geographic overlap, population and income are interpolated for each image. An example of how this is done for population is as follows. If there is an small image overlapping a number of 500m squares as in figure 5, the population and income are calculated based on the percentage overlap. If each square in figure 5 would have 1000 inhabitants, then the population would be

$$\text{population} = 1 \times 1000 + 4 \times 0.7 \times 1000 + 4 \times 0.49 \times 1000 = 5760.$$

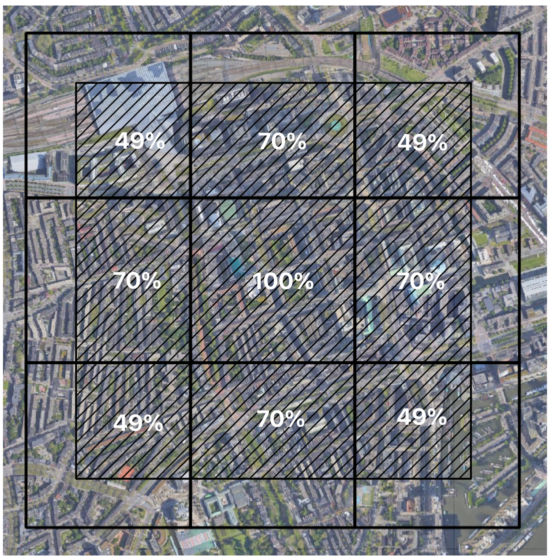


Figure 5: Small image (1.2 x 1.2km) of the Rotterdam city centre (dashed) overlapping 500m squares collected from Statistics Netherlands with corresponding percentage overlap

Notes: The percentages in each square denote the percentage overlap between the 500m squares and the 1.2km square image in the middle (dashed).

4.3.2 Transfer Learning

To extend upon the paper by Khachiyan et al. (2022b) a similar method is applied to a new dataset, the Netherlands. Urban areas, as described in the data section, are used to extract Landsat 7 satellite imagery. In a process called transfer learning, the pre-trained model weights are used to make predictions for the Netherlands. Therefore, I follow the steps laid out by Khachiyan et al. (2022b) in their research paper.

After assigning population and income values to each image, predictions of income and population level are obtained using the model weights trained by Khachiyan et al. (2022b). This is done for both income and population level and small and large imagery. Thus, four sets of predictions are made. For this purpose, slightly adapted code of Khachiyan et al. (2022b) is used. This relies on the tensorflow package (Abadi et al., 2015), as images and weights are in TFRecord format. The CNN models of Khachiyan et al. (2022b) are trained to predict log population and log income in 2012 US dollars. Therefore, the income values for the Netherlands are converted to 2012 US dollars.

The conversion from 2020 euros 2012 US dollars is performed using the US inflation rates from 2012 until 2019[1] and the US dollar/Euro exchange rate in at the beginning of 2020[2]. The inflation rate in 2020 is excluded, because data from the Netherlands is collected on January 1, 2020. The first available US dollar/euro exchange rate in 2020 on the website of the European Central Bank is on January 2 and was 1.1193 US Dollar/euro. The 2020 euro income values are multiplied by this number to obtain 2020 US dollars. Then, the inflation values from 2012 to 2019 in the US are compounded through multiplication[3]. Finally, the 2020 US dollar values are divided by this compounded number to obtain 2012 US dollars. Finally, the natural logarithm of population and income are calculated to obtain the true values in the correct format for comparison with transfer learning predictions.

R^2 values and mean squared error (MSE) are calculated for the predictions. These values are displayed in a table together with the results of newly trained CNN models, as explained in the following subsection. Besides R^2 and MSE values, plots containing predicted against true values are made. These plots contain a 45-degree line, which helps recognize any skewness in the prediction errors.

4.3.3 Re-Training CNNs

The training of the new CNN models is done in a similar fashion to Khachiyan et al. (2022b). The geo-spatial data of the Netherlands is split into a training (50%), validation (20%), and test set (30%) for both the small and large image data. These sets are created based on urban area instead of per image to analyse any differences in prediction accuracy between areas. The training set is used to train the model weights, the validations set is used to tune the CNN hyperparameters, and the test set is used to calculate prediction accuracy measures. For each outcome variable, different combinations of hyperparameters are used to find the optimal com-

[1] `https://www.macrotrends.net/countries/USA/united-states/inflation-rate-cpi`

[2] `https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/eurofxref-graph-usd.en.html`

[3] The inflation from 2012 until and not including 2020 is: 2.07%, 1.46%, 1.62%, 0.12%, 1.26%, 2.13%, 2.44%, and 1.81%. Compounding yields a value of 1.1364 (13.64% inflation from 2012 until 2020)

bination. This is done by minimizing the mean squared error between the predictions and true values for the validation set. The set of hyperparameters consists of learning rate (0.0001), L2 regularisation parameter (1e-08, 1e-07, or 1e-06), batch size (16), decay steps (50, 100, or 200), number of convolutional filters (32), and dropout rate (0.5). There are 9 possible combinations of hyperparameters that are used for validation. Training stops after 200 epochs have been conducted, or the R^2 of the validation set does not increase for 50 epochs in a row. The tensorflow package (Abadi et al., 2015) in Python is used for training, validation, and obtaining predictions. R^2 and MSE values are calculated using the test set and presented in a table together with the values calculated using transfer learning. Furthermore, plots of predicted vs. true values are included for the different outcomes.

4.4 Agglomeration Effects

Agglomeration of people and firms in urban areas can have different economic implications. Using the income and population predictions and true values, the effects of agglomeration on income per capita will be analysed. In theory, agglomeration may lead to benefits to companies and people, which may increase income per capita. In order to research this, log income per capita will be regressed on the log of total population for small and large images. This will be done for the true values as well as the predicted values to uncover if the CNNs are able to capture any present agglomeration effects. Population per image instead of the typical population density is used as a measure of agglomeration, as images have uniform sizes. The regression formulas are as follows:

$$\ln \frac{Inc_i}{Pop_i} = \beta_{true} \times \ln Pop_i + \varepsilon_i \tag{2}$$

$$\ln \frac{\widehat{Inc}_{i,tl}}{\widehat{Pop}_{i,tl}} = \beta_{tl} \times \ln \widehat{Pop}_{i,tl} + \omega_i \tag{3}$$

$$\ln \frac{\widehat{Inc}_{i,cnn}}{\widehat{Pop}_{i,cnn}} = \beta_{cnn} \times \ln \widehat{Pop}_{i,cnn} + \nu_i \tag{4}$$

Inc_i and Pop_i denote income and population for image i. Where i is either in the dataset of large or small images. $\widehat{Inc}_i$ and $\widehat{Pop}_i$ denote predicted income and population for image i. The subscripts *tl* and *cnn* refer to the transfer learning and re-trained CNN model predictions, respectively. The results consist of the estimated β coefficients of these equations. Furthermore, the R^2 values are also included.

5 Results

5.1 Replication

The results obtained by replicating part of the paper by Khachiyan et al. (2022b) can be seen in table 2 below. The replicated and reported R^2 values are very close together and differ at most 0.0007. The discrepancies are small and may be due to differences in rounding. The MSE is the highest for income in small imagery, and lowest for population in large imagery. For both income and population, the MSEs are lowest for large imagery, indicating that large imagery provides more accurate predictions. This is also highlighted by the higher R^2 values for large imagery.

Table 2: Replicated and reported R^2 and MSE for predictions of income and population using the data and methodology of Khachiyan et al. (2022)

		Replicated R^2	Reported R^2	MSE
Income	**Small**	0.7497	0.7494	1.5249
	Large	0.8367	0.8374	0.4003
Population	**Small**	0.7488	0.7492	0.7095
	Large	0.8681	0.8684	0.2991

Notes: Khachiyan et al. (2022b) train convolutional neural networks to predict income and population level using square satellite imagery of 1.2km (small) and 2.4km (large) in the continental United States for the years 2000 and 2010. For replication, the pre-trained model weights are collected and the predictions are made using the data in the test set, which comprises 20% for both the small and large imagery. The R^2 and MSE values are calculated using the true values. The reported values are the values obtained by Khachiyan et al. (2022b). All necessary data is collected from their online repository (Khachiyan et al., 2022a).

5.2 Transfer Learning and Re-Training

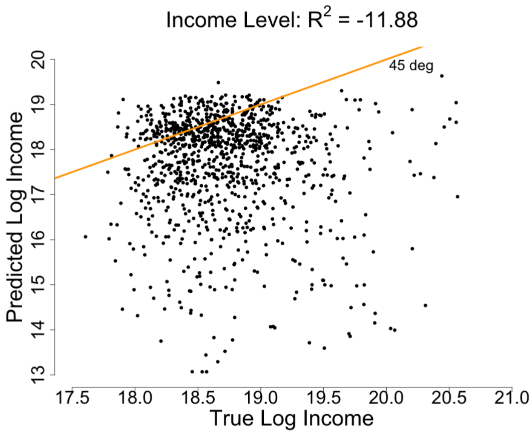
The table containing the results of hyperparameter tuning in re-training of the CNN models can be found in appendix A. The R^2 and MSE values of the transfer learning and re-trained CNN models are included in table 3. The R^2 values are highly negative for the transfer learning method, indicating very low accuracy. Only the R^2 value for large image income predictions, although negative, is somewhat low in magnitude at -1.9061. This is also reflected in the MSE, which is the lowest at 0.7228. The R^2 values of the re-trained CNNs are also all negative, but lower in magnitude. The most negative R^2 value is that of small image income (-4.1784) and the highest R^2 is that of large image population (-0.5391). The MSEs are also lower than for transfer learning with large image population being the lowest (0.2315) and small image income the highest (1.5017). While the R^2 values are much lower than those obtained by Khachiyan et al. (2022b) for predictions in the US, the MSEs show more promising results when comparing table 2 and 3. The MSEs of small image income and small and large image population predictions are lower for the re-trained CNNs.

Table 3: R^2 and MSE values of income and population predictions for large and small imagery using transfer learning methods and re-trained CNN models.

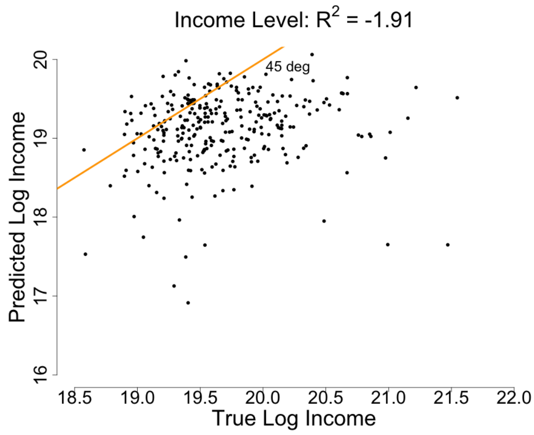
		Transfer Learning		Re-trained CNN	
		R^2	MSE	R^2	MSE
Income	**Small**	-11.8832	2.8417	-4.1784	1.5017
	Large	-1.9061	0.7228	-1.7721	0.5395
Population	**Small**	-25.3281	4.3180	-0.5642	0.3341
	Large	-42.1120	8.2520	-0.5391	0.2315

Notes: Transfer learning refers to the use of weights trained by Khachiyan et al. (2022b) for income and population predictions for small (1.2km) and large (2.4km) satellite imagery in the Netherlands. These weights are obtained by training CNN models using urban areas in the United States. Re-trained CNN refers to CNNs trained for the present research using urban areas in the Netherlands as dataset. All R^2 and MSE values are calculated using predictions of income and population for the Netherlands.

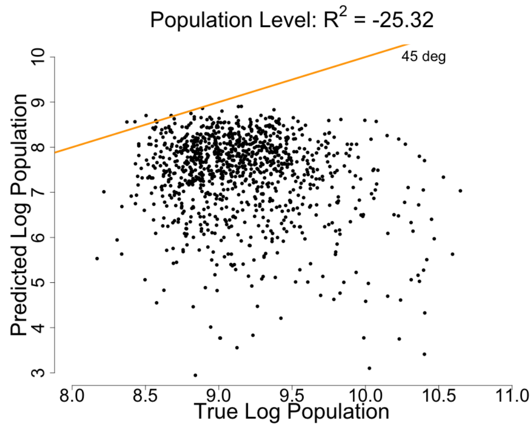
The difference in R^2 and MSE may be the result of the fact that they measure prediction accuracy differently. R^2 measures the proportion of variance in the outcome variable explained by the model, while MSE does not. Therefore, a model may have high accuracy in terms of MSE, while failing to capture the underlying trends in the data. This phenomenon is best visualised in plots with true values plotted against predicted values. These plots for the four outcome variables using transfer learning can be seen in figure 6. The income predictions for



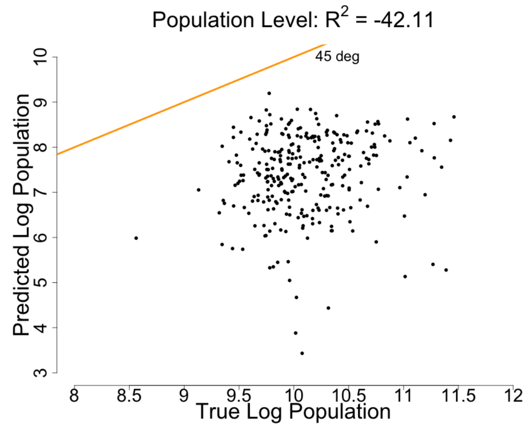
(a) Income Small



(b) Income Large



(c) Population Small

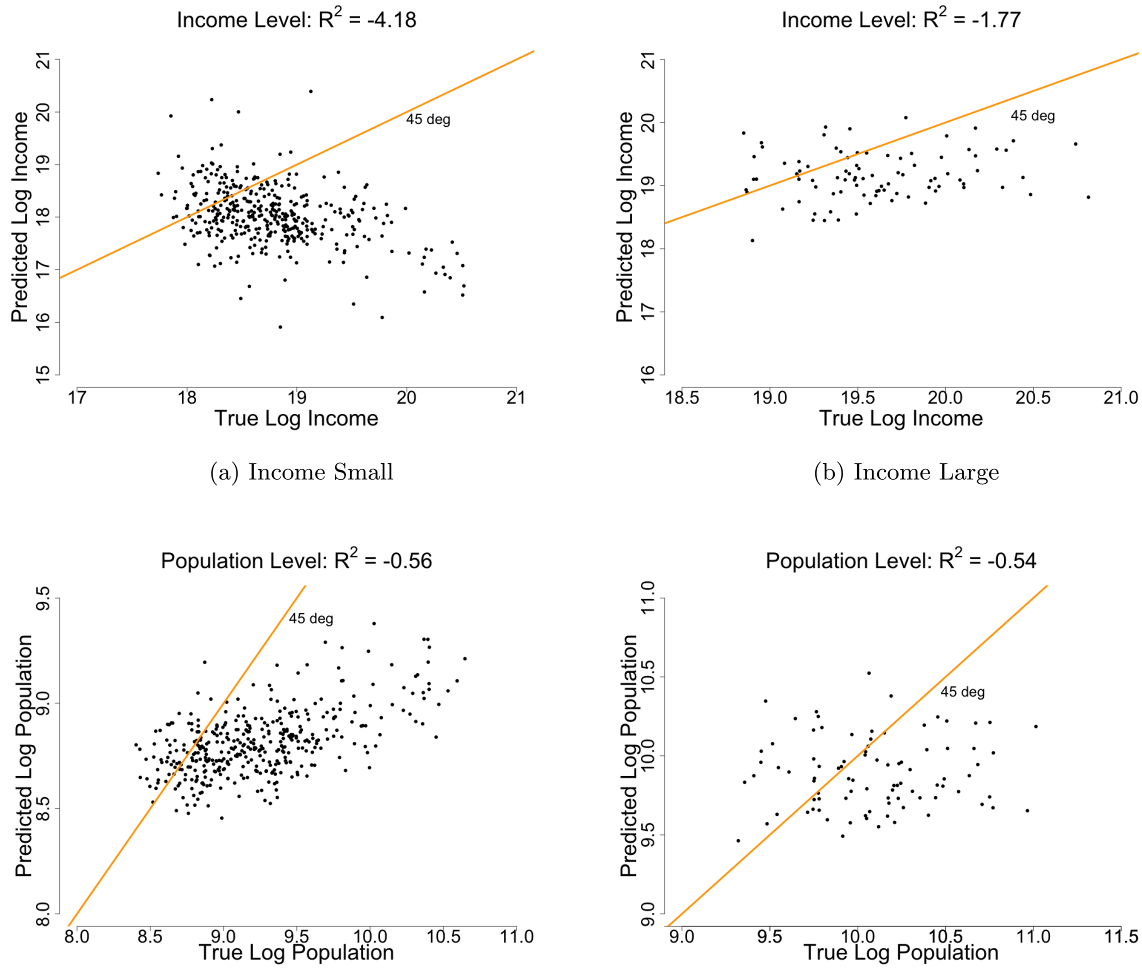


(d) Population Large

Figure 6: Predicted versus true values of income and population level for small and large imagery using transfer learning with convolutional neural networks

Note: These predictions are obtained using Landsat 7 satellite imagery in the Netherlands of small (1.2km) and large (2.4km) square urban areas. The weights used in the convolutional neural networks are obtained from Khachiyan et al. (2022b), who train CNNs for a similar purpose in the United States. The R^2 values of the predictions are included above each figure, and the 45-degree line has been added, labelled "45 deg".

small and large imagery (subfigures 6a and 6b, respectively) show that income is often predicted too low as it lies under the 45-degree line. Income predictions for small imagery have much more downward outliers, with log income values of around 13 and 14. This results in the very low R^2 value of -11.88. This is less the case for large imagery with a few outliers around 17, also visible in the R^2 value of -1.91. The population predictions are almost all below the 45-degree line for small imagery (see subfigure 6c) and all below the 45-degree line for large imagery (see 6d). Part of the reason of the highly negative R^2 values for these population predictions is the lack of variance in the true values, seeing as this causes a small denominator in equation 1. All in all, transfer learning using the models trained by Khachiyan et al. (2022b) on the dataset of the Netherlands results in predictions with poor accuracy. The same plots for the predictions using



(a) Income Small

(b) Income Large

(c) Population Small

(d) Population Large

Figure 7: Predicted versus true values of income and population level for small and large imagery using convolutional neural networks

Note: These predictions are obtained using Landsat 7 satellite imagery in the Netherlands of small (1.2km) and large (2.4km) square urban areas. The R^2 values of the predictions are included above each figure, and the 45-degree line has been added, labelled "45 deg".

re-trained CNNs can be seen in figure 7. Seeing as the R^2 values for these predictions are lower in magnitude, it is no surprise that the predictions seem to follow a more accurate trend. The income predictions for large images, for example, lie around the 45-degree line (see subfigure 7b). This is also somewhat true for the large image population predictions (see subfigure 7d). The cloud of points for the small image population predictions seems to follow a trend that is less steep than the 45-degree line (see subfigure 7c). However, most points lie around the 45-degree line. The small image income predictions (see subfigure 7a) seem to follow a downward trend, hence the lowest R^2 value at -4.18. Compared to the transfer learning plots in figure 6 these predictions generally seem to follow a more accurate trend. Therefore, retraining CNN models seems more suitable than transfer learning for obtaining accurate predictions in the Netherlands.

Finally, maps containing the average prediction error per urban area for each outcome variable can be found in appendix B. This prediction error is calculated as prediction minus the true value. It is interesting to note that in larger urban areas population and income are underestimated more often than in smaller urban areas. This may be due to a difference in urban planning in larger urban areas containing different features not recognized by the CNN models. Furthermore, these underestimations seem to limit themselves mostly to the Randstad, an area in the west of the Netherlands containing the four largest cities. This is also the case for smaller urban areas. This may be the result of these smaller areas belonging to the larger cities in this region.

5.3 Agglomeration Effects

The results of the regressions in equation 2, 3, and 4 can be seen in table 4. Both the regression coefficients are significant at the 0.1% significance level for the large and small imagery when using true income per capita and population level. Thus, there seem to be significant positive agglomeration effects when comparing small and large imagery. However, one would need to control for other variables influencing income per capita to truly make this conclusion. Seeing as log variables are used, the coefficients can be interpreted as the percentage change, when population increases by 1%. Thus, for example, using large imagery the income per capita increases by 0.086% when population level increases by 1%. The R^2 values are roughly the same at 0.077 and 0.079 for large and small imagery, respectively. The estimated coefficient for

Table 4: Agglomeration effects measured by regression of income per capita on population level based on true and predicted values

		Large	**Small**
True values	β_1	0.086*** (0.018)	0.101*** (0.011)
	R^2	0.077	0.079
Transfer Learning	β_2	-0.033 (0.108)	0.148* (0.063)
	R^2	0.000	0.818
Re-trained models	β_3	0.090 (0.058)	-0.606*** (0.065)
	R^2	0.016	0.180

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$

Notes: Regressions are performed for small (1.2km) and large (2.4km) urban squares within the Netherlands. log incomer per capita is regressed on log population level. β_1 denotes the estimated regression coefficient for true values provided by Statistics Netherlands and interpolated to the geographic space of the images. β_2 denotes the estimated regression coefficient for values income and population values predicted by the convolutional neural networks in the present research. Standard errors are displayed between parentheses.

the large imagery using the predicted income and population levels is not significantly different from 0 at -0.033. Therefore, there is no evidence that agglomeration effects are present when comparing different large urban areas with predicted income and population values. For the predictions to be useful in the context of using them in agglomeration effects research, the coefficient results would have to be similar to the true value regression. This is somewhat the case for the small image regression with transfer learning predicted values. The regression coefficient is significant at the 5% significance level at 0.148. It is still quite different from the value of 0.101 for small imagery and true values, but it indicates some evidence of positive agglomeration effects as in the regression with true values. Thus, contrary to large imagery, perhaps small imagery predictions could be used to research agglomeration effects in different settings.

6 Conclusions

In this paper, CNN models using transfer learning and re-training are used to predict income and population levels at fine spatial scale. This is done using satellite imagery for small (1.2 x 1.2km) and large (2.4 x 2.4km) urban areas in the Netherlands. Transfer learning is applied using models pre-trained on similar images in the continental United States by Khachiyan et al. (2022b). Furthermore, it is analysed whether CNN models can capture any present agglomeration effects in these urban areas. This is done by comparing the effect of population on income per capita using true values, transfer learning predictions, and re-trained model predictions. For this, a regression of income per capita on population is performed for both small and large imagery.

Transfer learning provides poor predictions with highly negative R^2 values and high MSE values. Only the income predictions for large imagery are somewhat accurate. The R^2 values of the re-trained CNN models are more accurate with negative R^2 values that are much closer to zero. The MSE values also indicate higher accuracy, especially for the population predictions. Overall, the population predictions of re-trained models are very accurate and even lower than Khachiyan et al. (2022b). Based on these results, it can be concluded that re-training CNN models provides significantly better predictions of income and population than using transfer learning with models trained on the US.

Furthermore, agglomeration effects seem to be present in urban areas in the Netherlands, as log population has a significantly positive effect on log income per capita for both small and large areas when using true values. When population increases by 1%, income per capita increases by 0.086% and 0.101% for large and small imagery, respectively. However, only the re-trained CNN model for small imagery is able to capture this significantly positive effect. This effect is a 0.148% increase in income per capita per 1% percentage increase in population. This effect is however roughly 50% larger than for the true data, indicating that the correct magnitude is not captured. Thus, it does not seem that CNN model predictions are suitable for analysing this agglomeration effect.

The present research poses some limitations with respect to what could be researched with the available computational power. The transfer learning method in the present research used full pre-trained models rather than re-training part of the models. Typically, the pre-trained convolutional layers are used to capture features and the dense layers are retrained to optimise how these features are used in the new dataset. It may be interesting in future research to see how this changes results. Furthermore, sensitivity analysis could have been performed based on the threshold used to define urban areas in the Netherlands. The definition of Statistics Netherlands was used (more than 1500 private addresses per square kilometre), but perhaps this does not align with the dataset the pre-trained models were trained on in the US. Also, different agglomeration effects could be analysed to find more relevant economic applications of CNN model predictions of income and population. Finally, defining CNN model architectures is a process of trial and error. Thus, different architecture types could be researched. Application in other countries with different urban features could also be explored.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. `https://www.tensorflow.org/`

Albahli, S., & Albattah, W. (2021). Deep transfer learning for covid-19 prediction: Case study for limited data problems. *CMIR*, *17*, 973–980.

Alla, S. S. M., & Athota, K. (2022). Brain tumor detection using transfer learning in deep learning. *IJST*, *15*, 2093–2102. `https://doi.org/10.17485/ijst/v15i40.1307`

Doll, C. N., Muller, J.-P., & Morley, J. G. (2006). Mapping regional economic activity from night-time light satellite imagery. *Ecological Economics*, *57*(1), 75–92.

Donaldson, D., & Storeygard, A. (2016). The view from above: Applications of satellite data in economics. *Journal of Economic Perspectives*, *30*(4), 171–198.

ESRI. (2023a). *Arcgis*. Version 10.8.2. `https://www.arcgis.com`

ESRI. (2023b). ArcPy (2.8.1). `https://developers.arcgis.com/documentation/arcgis-add-ins-and-automation/arcpy/`

European Central Bank. (2023). ECB euro reference exchange rate: US Dollar (USD). `https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/eurofxref-graph-usd.en.html`

Feldman, M. P. (1999). The new economics of innovation, spillovers and agglomeration: A review of empirical studies. *Economics of innovation and new technology*, *8*(1-2), 5–25.

Glaeser, E. L., Kolko, J., & Saiz, A. (2001). Consumer city. *Journal of economic geography*, *1*(1), 27–50.

Gogas, P., & Papadimitriou, T. (2021). Machine learning in economics and finance. *Computational Economics*, *57*, 1–4.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*. `https://doi.org/10.1016/j.rse.2017.06.031`

Gwak, B. S., JunYoung, & Nabel, R. (2016). PyDrive (1.2.1). `https://pythonhosted.org/PyDrive/`

He, C., Ma, Q., Liu, Z., & Zhang, Q. (2014). Modeling the spatiotemporal dynamics of electric power consumption in mainland china using saturation-corrected dmsp/ols nighttime stable light data. *International Journal of Digital Earth*, *7*(12), 993–1014.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *353*(6301), 790–794.

Khachiyan, A., Thomas, A., Zhou, H., Hanson, G., Cloninger, A., Rosing, T., & Khandelwal, A. (2022a). Data and code for: Using neural networks to predict micro-spatial economic growth: Weights [Data set]. `https://doi.org/10.3886/E158002V1-138640`

Khachiyan, A., Thomas, A., Zhou, H., Hanson, G., Cloninger, A., Rosing, T., & Khandelwal, A. K. (2022b). Using neural networks to predict microspatial economic growth. *American Economic Review: Insights*, *4*(4), 491–506.

Lehnert, P., Niederberger, M., Backes-Gellner, U., & Bettinger, E. (2023). Proxying economic activity with daytime satellite imagery: Filling data gaps across time and space. *PNAS nexus*, *2*(4), 1–10.

MacroTrends. (2023). U.S. Inflation Rate 1960-2023. https://www.macrotrends.net/countries/USA/united-states/inflation-rate-cpi

Nadin, V., & Zonneveld, W. Introducing the randstad: A polycentric metropolis. In: In *The randstad*. Routledge, 2020, pp. 3–25.

QGIS Development Team. (2023). QGIS Geographic Information System. *QGIS Association*. https://www.qgis.org

Roser, M., & Ortiz-Ospina, E. (2022). Population density, 2022. https://ourworldindata.org/grapher/population-density

Sonoda, S., & Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, *43*(2), 233–268.

Statistics Netherlands. (2023). Kaart van 500 meter bij 500 meter met statistieken. https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische-data/kaart-van-500-meter-bij-500-meter-met-statistieken

Tajbakhsh, N., Shin, J. I., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imaging*, *35*, 1299–1312. https://doi.org/10.1109/tmi.2016.2535302

Appendix

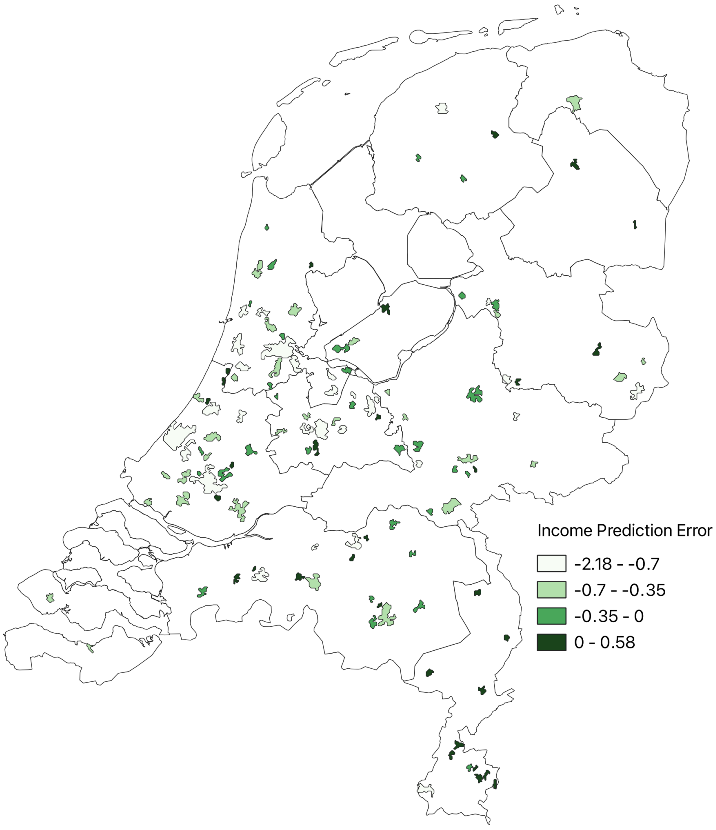
Appendix A Hyperparameter Tuning

Table 5: Results of running the CNN training for each set of hyperparameters using minimization based on the MSE of the validation set

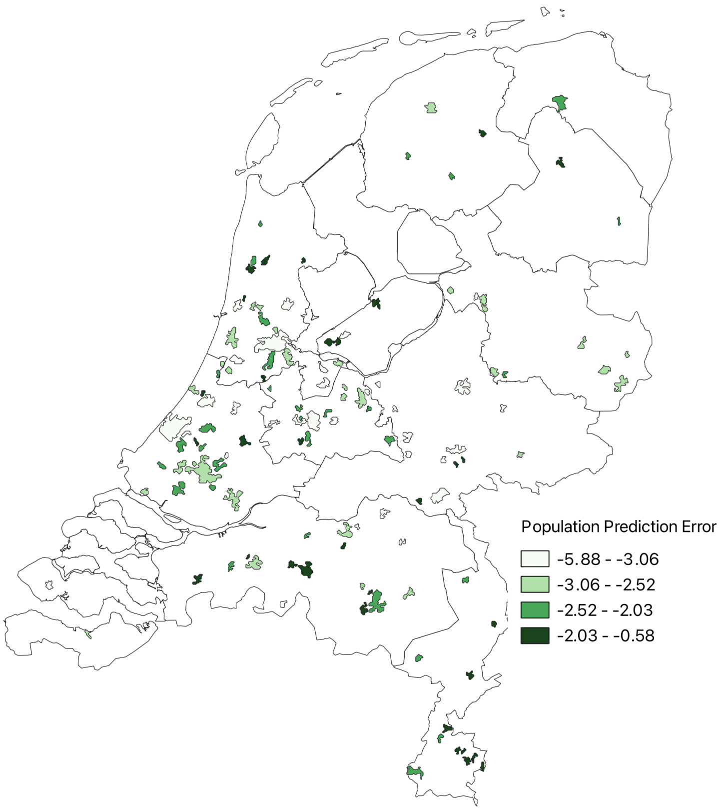
Small Imagery						Income			Population		
lr	l2	bs	ds	nf	dr	MSE	R^2	epochs	MSE	R^2	epochs
0.001	1e-08	16	50	32	0.5	1.7875	-11.7216	...	0.6445	-5.1575	54
0.001	1e-08	16	100	32	0.5	1.0717	-6.6271		0.5901	-4.6381	96
0.001	1e-08	16	200	32	0.5	1.2890	-8.1741		0.2114	-1.0193	96
0.001	1e-07	16	50	32	0.5	**0.7079**	-4.0368		0.6586	-5.2900	52
0.001	1e-07	16	100	32	0.5	2.1613	-14.3807		**0.1986**	-0.8963	108
0.001	1e-07	16	200	32	0.5	1.7462	-11.4264		0.5987	-4.7180	57
0.001	1e-06	16	50	32	0.5	2.1226	-14.0906		0.3033	-1.8776	61
0.001	1e-06	16	100	32	0.5	2.1826	-14.5207		0.3395	-2.2251	76
0.001	1e-06	16	200	32	0.5	0.4399	-2.1170		0.4922	-3.6838	67
Large Imagery						Income			Population		
lr	l2	bs	ds	nf	dr	MSE	R^2	epochs	MSE	R^2	epochs
0.001	1e-08	16	50	32	0.5	2.2309	-5.0461	54	0.9700	-3.0179	70
0.001	1e-08	16	100	32	0.5	1.4349	-2.8888	92	1.0091	-3.1800	56
0.001	1e-08	16	200	32	0.5	**0.9713**	-1.6323	163	0.5528	-1.2898	67
0.001	1e-07	16	50	32	0.5	1.6765	-3.5429	98	**0.3884**	-0.6080	96
0.001	1e-07	16	100	32	0.5	2.6549	-6.1945	124	0.4302	-0.7809	88
0.001	1e-07	16	200	32	0.5	2.2920	-5.2111	66	0.4427	-0.8329	124
0.001	1e-06	16	50	32	0.5	2.4072	-5.5185	124	0.9438	-2.8995	56
0.001	1e-06	16	100	32	0.5	2.0996	-4.6838	56	0.8596	-2.5506	54
0.001	1e-06	16	200	32	0.5	1.2834	-2.4720	79	0.9787	-3.0443	63

Notes: This table shows the tuning of the hyperparameters used to train the convolutional neural networks for income and population level predictions in the Netherlands. For a full explanation of the training of convolutional neural networks, please consult the methodology in the main paper. The abbreviations of the hyperparameters are as follows:
lr: learning rate (of the gradient descent algorithm in CNN weight optimisation)
l2: L2 regularisation parameter
bs: Batch size (used to update weights before a new batch is used)
ds: Number of steps in training after which the learning rate is lowered
nf: Number of convolutional filters used in the CNN
dr: Dropout rate (The fraction of nodes randomly removed between layers)
The mean squared error (MSE) and R^2 refer to that of the validation set. The set of hyperparameters with the lowest MSE is used for training (The lowest MSEs for each application are made bold). Epochs refers to the number of epochs used for training the CNNs.

Appendix B Maps



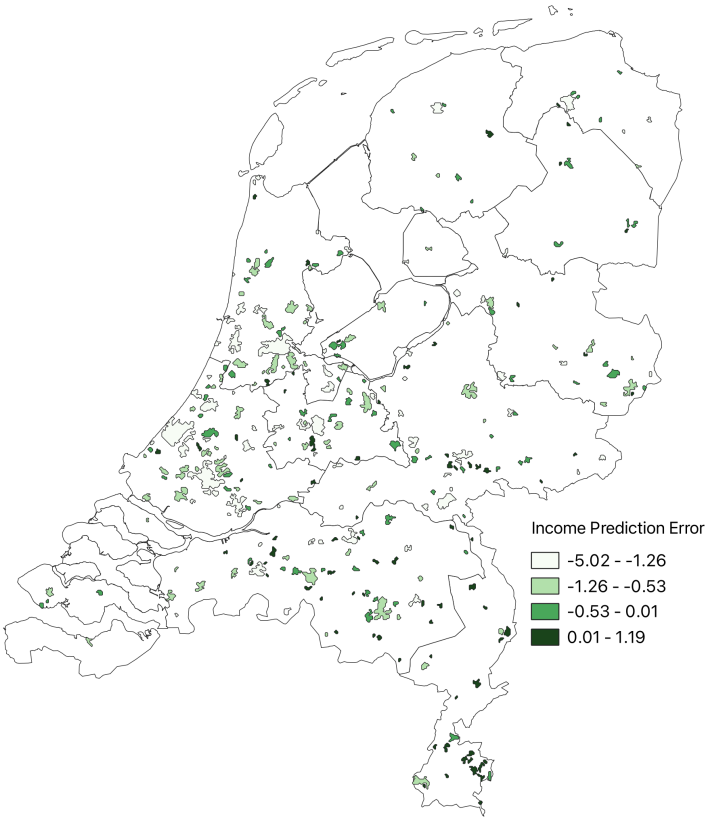
(a) Income prediction errors for large imagery



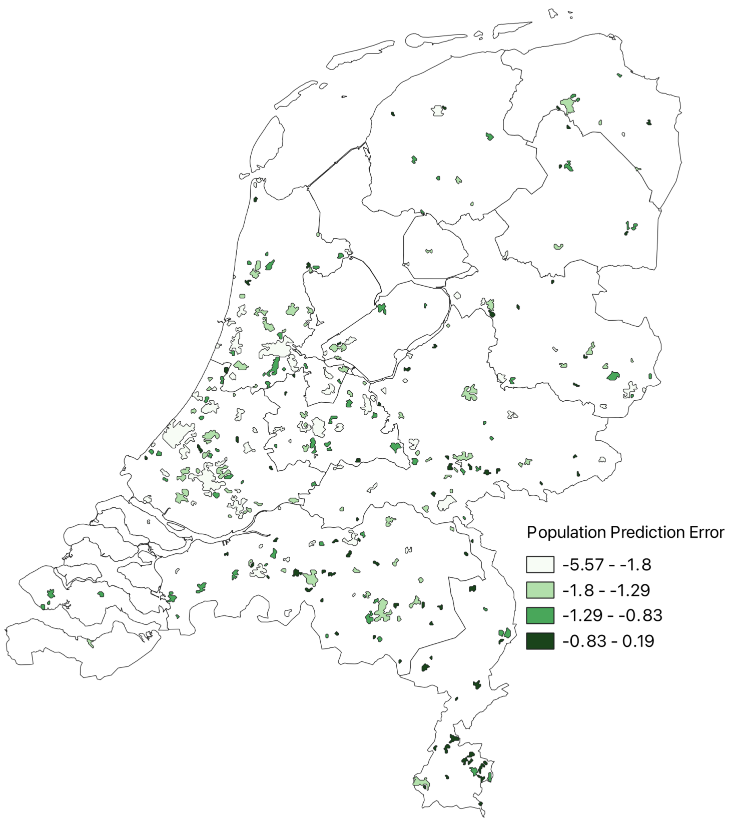
(b) Population prediction errors for large imagery

Figure 8: Maps containing the income (a) and population (b) transfer learning prediction errors of large images for urban areas in the Netherlands in the test set.

Notes: The prediction error is calculated as the average prediction error per urban area, where the error is the prediction minus the true value.



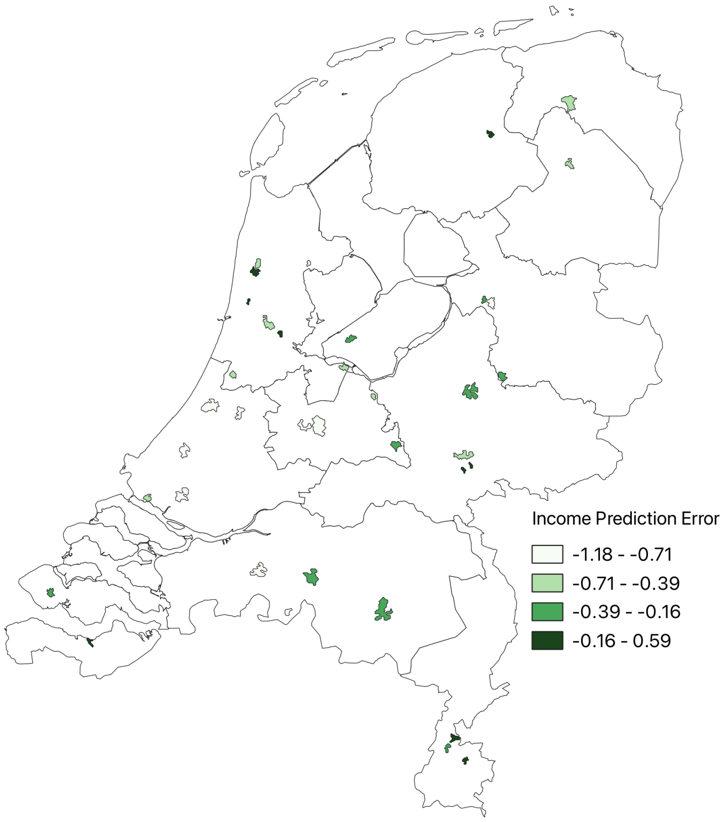
(a) Income prediction errors for small imagery



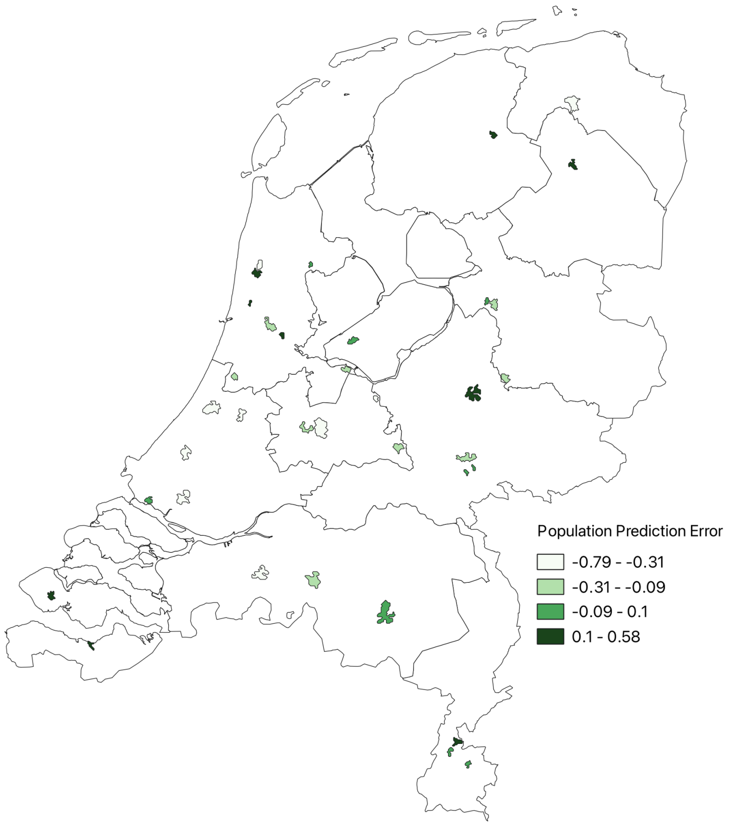
(b) Population prediction errors for small imagery

Figure 9: Maps containing the income (a) and population (b) transfer learning prediction errors of small images for urban areas in the Netherlands in the test set.

Notes: The prediction error is calculated as the average prediction error per urban area, where the error is the prediction minus the true value.



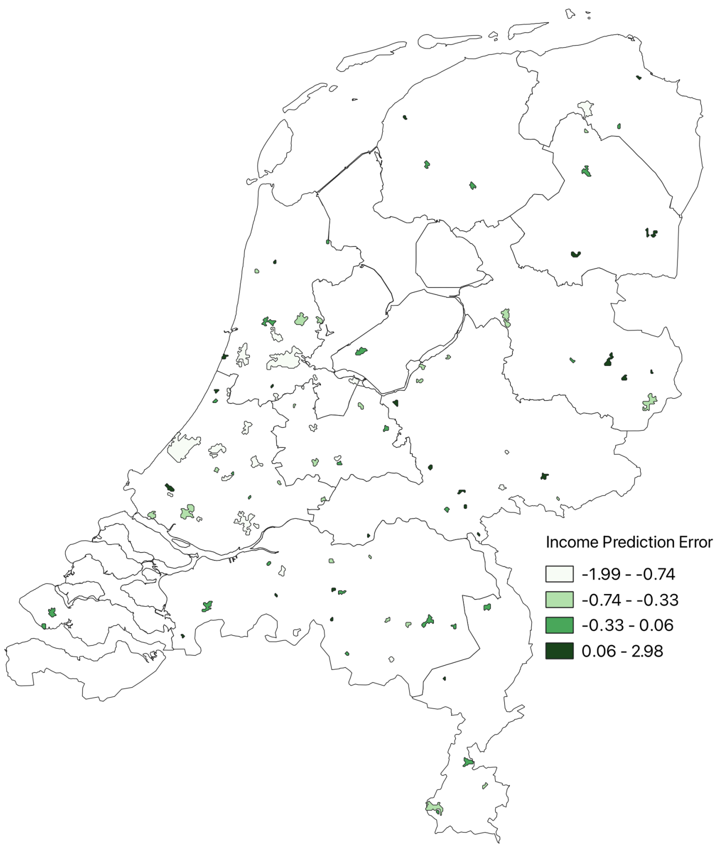
(a) Income prediction errors for large imagery



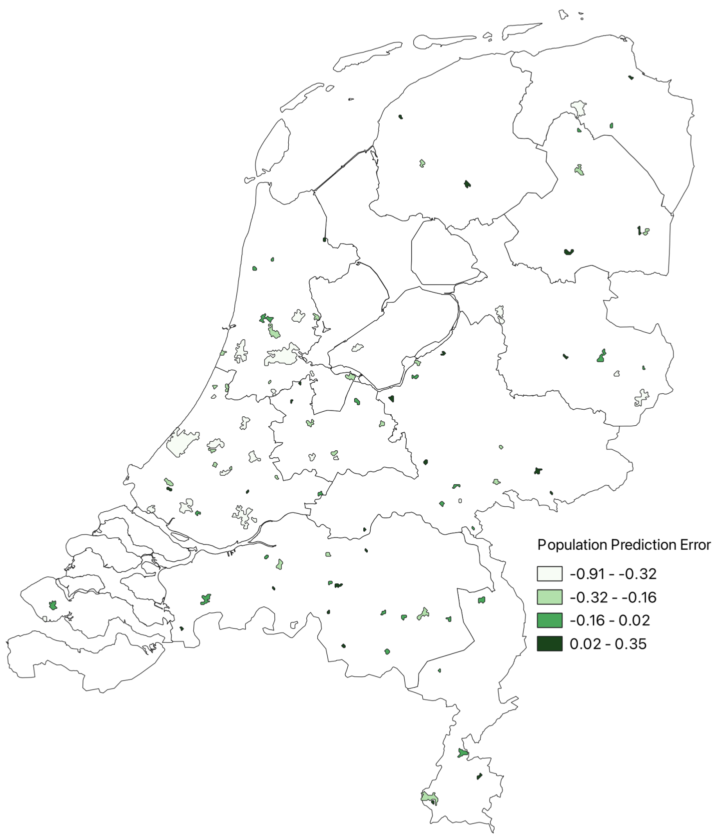
(b) Population prediction errors for large imagery

Figure 10: Maps containing the income (a) and population (b) re-trained CNN prediction errors of large images for urban areas in the Netherlands in the test set.

Notes: The prediction error is calculated as the average prediction error per urban area, where the error is the prediction minus the true value.



(a) Income prediction errors for small imagery



(b) Population prediction errors for small imagery

Figure 11: Maps containing the income (a) and population (b) re-trained CNN prediction errors of small images for urban areas in the Netherlands in the test set.

Notes: The prediction error is calculated as the average prediction error per urban area, where the error is the prediction minus the true value.

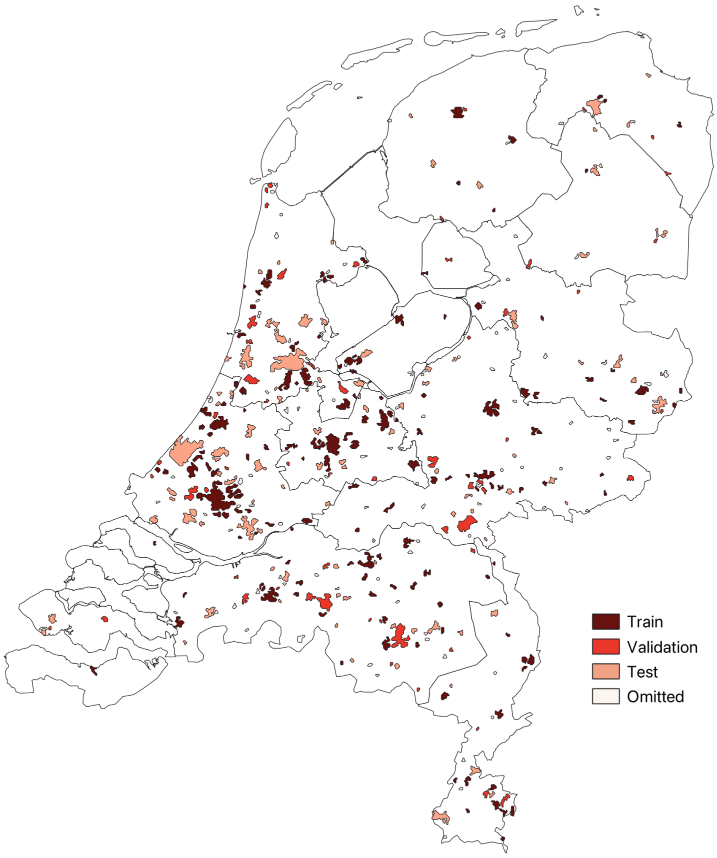


Figure 12: Map of the Netherlands showing which urban areas were randomly placed in which subset for training, validation and testing CNN models for the small imagery