

Evaluation of Land Surface Temperature based on Local Climate Zones Scheme in Greater Copenhagen Area

Authors:

Bogdan Mocanu

Student nr.:

20181384

Supervisor:

Jamal Jokar Arsanjani

August 2020

Title:

Evaluation of Land Surface Temperature based on Local Climate Zones Scheme in Greater Copenhagen Area

Project:

Speciale Thesis, Geoinformatics

Aalborg University - Copenhagen

Project Period:

February 2020 - August 2020

Student:

Bogdan Mocanu

Supervisor:

Prof.Dr.Jamal Jokar Arsanjani

Number of pages: 71

Number of standard pages:

Approx. 51 (350 words per page)

Number of appendices: 2

Number of ZIP files: 1

Finished: August 2020

Abstract:

This project tries to investigate if there is a connection between extensive build-up areas and climate change in Copenhagen area from 2015-2019. For this aspect, Local Climate Zones scheme was used to map Copenhagen area using Landsat 8 images with a focus for Nordhavn area. This research has been produced in Google Earth Engine with Random Forest Classifier with an Overall Accuracy of roughly 0.75, where on one hand LCZ maps were produced for each year analysing the dynamic of the classes. On the other hand LST was calculated for each year based on Landsat8 image collection. Though an increase trend for LST was identified in a range in-between min: 20.57 and max:29.33. Based on this project it can only show a connection between LST growing trend and newly built-up areas, but further, for the sources of error, adjustments are required. The project can be improved on one hand for deeper analysis, but on the other hand it can be extended for larger areas where at least for the vegetation classes the LCZ scheme can be considered, while for the built-up classes the digital surface model is mandatory for extraction, all together with land surface temperature calibration and temperature records validation

Preface

This is the Speciale Thesis of Bogdan Mocanu at Geoinformatics at Aalborg University Copenhagen.

This project would not have been possible without the constant moral support and guidance of Prof.Dr.Jamal Jokar Arsanjani.

Reader's guide

The report is made for easy electronic reading with hyperlinks and internal references, but the text is also structured to easily be read when printed.

Terms of Reference

Here is a glossary of the abbreviations used in the project, for a quick and easy lookup:

- WUDAPT = World Urban Database and Access Portal Tools
- DSM = Digital Surface Model
- LCZ = Local Climate Zones
- LST = Land Surface Temperature
- UHI = Urban Heat Island
- JS = JavaScript
- NDVI = Normalised Difference Vegetation Index
- UEB = Urban Energy Ballance
- SUEWS = urface Urban Energy and WaterBalance
- LULC = Land Use Land Classification
- OBIA = Object Based Image Analysis
- API = Application Programming Interface
- IDE = Interactive Development Environment
- GEE = Google EArth Engine
- HTML = Hypertext Markup Language
- CSS = Cascading Style Sheets
- NIR = Near Infrared
- NGRDI = Normalized Green Red Difference Index
- DVI = Difference Vegetation Index
- NDWI = Normalized Difference Water Index
- SWIR = Short Wave Infrared
- RVI = Ratio Vegetation Index
- SR = Simple Ratio
- EVI = Enhached Vegetation Index
- SAVI = Soil Adjust Vegetation Index
- IBI = Index-based Built-up Index
- ASTER = Advanced Spaceborne Thermal Emission and Reflection Radiometer

Source referencing

Citations to literature that has been used in this project is done by the Harvard Method. The in-text references tells the author or organization and the year of issue. For the full reference, the reader should head to the Bibliography at the end of this project report. Here, a list of all references can be seen in alphabetic order, with information about name of author or organization, title of reference, publisher/URL, year, and lastly the time of access if the source is a link to a web page.

In the same way as text, figures and tables will be cited by author's name and year of issue. If a figure or table does not have any reference, it is of own production.

Contents

Preface	iii
Reader's guide	iii
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Local Climate Zone state of the art	4
1.2 Previous study findings	11
1.3 Problem statement	12
1.4 Study Area	12
2 Methods	15
2.1 Process for the project	15
2.2 Software	16
2.3 Google Earth Engine	18
3 Theory	21
3.1 Landsat Programe	21
3.2 JavaScript(JS)	23
3.3 Google Earth Engine (GEE)	24
3.4 Indexes	29
3.5 Land Surface Temperature (LST)	32
4 Implementation	35
4.1 Data	35
4.2 Google Earth Engine	36
4.3 Land Surface Temperature	44
5 Results	47
5.1 Maps	47
5.2 Accuracy2019 and 2015	52
5.3 Graphs	53
6 Discussion	55
6.1 Land Use Land Change and sources of error	55
6.2 Temperature and sources of error	55
6.3 Nordhavn urbanisation impact	56
7 Conclusion	57
8 Reflections	59

8.1 Reproducibility	59
Bibliography	61
A Appendix guide	71

List of Figures

1.1	Local Climate Zones. <i>Source: Stewart and Oke [2012]</i>	5
2.1	Land Surface Temperature Flowchart	15
2.2	Local Climate Zones Flowchart	16
2.3	NDVI-visualisation. <i>Source: GisGeography.com [n.d.]</i>	18
3.1	The coverage period of each landsat satellite. <i>Source: Landsat Science [n.d.a]</i> .	21
3.2	Wave length. <i>Source: Wim H. Bakker [2009]</i>	22
3.3	Sensor principle <i>Source: Wim H. Bakker [2009]</i>	22
3.4	Google Earth Engine	25
4.1	Training samples	37
5.1	Map of Local Climate Zones 2015	47
5.2	Map of Local Climate Zones 2019	48
5.3	Nordhavn 2015	48
5.4	Nordhavn 2019	49
5.5	Emissivity map	49
5.6	Emissivity Nordhavn Area	50
5.7	Land Surface Temperature	51
5.8	LST Nordhavn Area	51
5.9	Results 2019	52
5.10	Results 2015	52
5.11	Results pixels 2019	53
5.12	Results pixels 2015	53
5.13	Copenhagen graph kelvin	54
5.14	Copenhagen line graph kelvin	54
5.15	Copenhagen graph kelvin	54
5.16	Nordhavn graph kelvin	54

List of Tables

4.1	The bands of Landsat 8. <i>source: eos [n.d.]</i>	36
-----	---	----

Urban areas represent a combination of green areas and built spaces that generate the flow of matter and energy that are vital features for the functioning of the ecosystem. In order to have a good ecosystem condition, the green areas within the urban landscape should not be limited. Consequently, those urban landscapes characterized by high buildings, a higher density of built structures and limited green areas are considered as being part of bad ecosystem condition. A healthy urban environment may be determined by open arrangements of low buildings combined with large surfaces covered with green space. In other words, a healthy urban environment is equivalent with a good ecosystem condition [Nedkov et al., 2017].

In the area of urban design, planning and development research of urban morphology through spatial form and function represents an important step [Verma and Jana, 2019]. Urban morphology is described as the result of specific thematic land cover classification created based on various scientific approaches. The resulted thematic land cover classification provide relevant and specific spatial information for decision-makers in urban planning, management and development [Nedkov et al., 2017]. Climate-related studies have to use urban morphology and vegetation structure as important factors in this type of studies [Zhao, 2018].

As part of the urbanization process built-up areas and impervious surfaces (such as asphalt, concrete, roof tops or building walls) are chosen to replace the natural landscapes. This process brings several changes within the urban landscape. The most important changes include lower albedo, higher heat capacity, changes in energy absorption and energy storage, wind intensification due to buildings arrangement, higher humidity as a consequence of vegetation cover reducing that generate lower evapotranspiration, but also higher anthropogenic energy release [Zhao, 2018]. [Nedkov et al., 2017] consider that the spatial arrangement of parks and vegetation belts within the urban areas represents an important aspect to consider while assessing the ecosystem condition in urban areas.

[Alexander et al., 2015]) mentioning a report of the United Nation from 2012 underline that the urban population is expected to grow to 2.6 billion. There are various effects of this urban population growth. Among these effects the urban climate change is one of the most important. Urban local climate in built-up areas is generated as a consequence of buildings, human activities and air pollution related to these activities and emission heat [Levlovics et al., 2013].

Natural vegetation and agricultural surfaces in urban areas are replaced with specific urban land covers such as buildings, asphalt, pavements etc. The building infrastructure associated with transportation infrastructure specifically for urban areas contribute to

the generation of excessive heat in the cities. This excessive heat is produced into a continuously manner and it is also transmitted into the atmosphere generating a higher heat in the urban area and a lower comfort for the population [Buyantuyev and Wu, 2009].

Urban population is vulnerable to urban climate and climate changes at the urban level. In order to analyze the urban climate and climate changes in urban areas effects, quantitative models for urban landscapes are required as inputs in urban climate analysis [Bechtel et al., 2015] [Leconte et al., 2015] quoting Christensen et al (2007) highlight that most European cities are expected to face heat waves that will be more intense and more frequent in the next decades. In the context of global warming is already proved that temperatures within urban areas are higher. Thermal conditions at urban level influence human activities and living conditions [Wang et al., 2017]. [Wong et al., 2019] highlight that local climate zone and land cover of the global scale have a highly contribution in the urban meteorological modeling.

Urban areas are characterized by large surfaces covered with buildings and impervious surfaces. The high density of buildings together with low surfaces covered with vegetation contributes to the formation of the urban heat island (UHI) phenomenon [Zhao, 2018]. One of the topics intensely researched in the last decades is represented by the urban heat island effect. The UHI effect is highly influenced by the characteristics of the urban surface [Grimmond et al., 2010]. Increased temperature in urban areas manifested as the urban heat island may be reduced substantially through a sustainable urban planning and well-design measures [Geletič et al., 2019].

[Vandamme et al., 2019] affirm that modifications in thermal climate in the urban areas are inevitable as the result of urbanization, land-cover changes and higher density of man-made features. Higher surface temperature in urban areas is mostly generated by the high urbanization degree that increase the built-up areas and consequently the anthropogenic heat is also increased [Cai et al., 2017]; [Bechtel et al., 2019].

There is a gap in the current information regarding the cities at the global level. This gap is considered by the World Urban Database and Access Portal Tools (WUDAPT) as an impediment in the urban climate research. The missing data refers especially when it comes about informing and preparing climate strategies in the urban areas aiming to adapt and to mitigate climate changes challenges [New, 2018]. In order to obtain relevant information regarding thermal data in the urban areas various approaches are developed, such as urban energy balance model (UEB) or the Surface Urban Energy and Water Balance model (SUEWS) [Alexander et al., 2015]. [Wang et al., 2017] highlight that at local scale various strategies having as goals the mitigation and the adaptation to urban warming are not sufficient in order to overcome excessive urban heat. Consequently, the LCZ approach is utilized in order to analyze the complexity and diversity of urban climate.

The problems that generate the need of a standardized method for urban site classification regarding climatic conditions include aspects such as the poor descriptions of sites in past studies, the difficulty to define and categorize developed and undeveloped human landscape and also the necessity to have a generalized method to compare the UHI values in various landscapes and regions [Coseo and Larsen, 2014].

Urban heat island In urban climatology the urban heat island (UHI) phenomenon is

studied intensively in recent years in various cities and mega cities all over the world [Buyantuyev and Wu, 2009]; [Mirzaei and Haghighat, 2010]; [X., 2015]; [Mirzaei, 2015]; [Kaloustian and Bechtel, 2016]. UHI is one of the major problems that urban planners and decision-makers in urban areas are aiming to mitigate [Coseo and Larsen, 2014]. Climatic response of urban surface can provide reliable source for comparison and guiding in the urban planning strategies [Ren et al., 2016]. UHI at metropolitan scale is highly impacted by the land cover conditions and characteristics [Stone et al., 2013].

Urban heat island is a phenomenon that is influenced by the different thermal properties of urban land cover and land use, urban morphology, building materials and human activities [Collins and Dronova, 2019]. The urban heat island (UHI) phenomenon represents the approach that describes those urban areas that are hotter as temperature than the non-urbanized areas in the surroundings of the cities [Kaloustian and Bechtel, 2016].

K[Kaloustian and Bechtel, 2016] citing a study developed by the French National Institute of Health in 2015 mention that the urban heat island phenomenon influences various aspects within the urban area: microclimate of the city, public health, energy use and even quality of life. Due to this, in the recent years the causes, effects and mitigation options of the UHI phenomenon are more studied and discussed.

UHI is an unwanted phenomenon in any city since it is considered as a factor that increases the heat in the urban environment, reduces the comfort for the urban population, increases the pollution that can generate higher mortality rates and also increases the energy demand of buildings [Mirzaei, 2015]. The UHI phenomenon can generate a wide range of adverse effects of air pollution, climate warming, health risks for population living in urban areas and energy consumption [Zhao, 2018]. Moreover, the UHI phenomenon has small scale heat release caused by anthropogenic activities and also meso-scale interactions at atmospheric level [Mirzaei and Haghighat, 2010]. In order to integrate the aspects specific for both the atmosphere and terrestrial surface layers of the UHI phenomenon in various measurements methodologies, this phenomenon has been defined for multiple layers [Zhao, 2018].

In the field of UHI research a standardized method to classify the results is required. Such a standardized classification framework may help urban planners, urban climatologists and local authorities to create appropriate strategies in order to prevent and to overcome hard urban climates [Coseo and Larsen, 2014].

[Estacio et al., 2019] affirm that one of the major challenges in UHI studies is delimitating the urban and rural areas. Since the administrative boundaries are not always the limit of the urban elements, this approach in delineating urban from rural may come with inaccuracies. Moreover, very often rural areas situated in the neighborhood of urban landscapes present highly urbanized characteristics. Furthermore, as a result of distinct urban functions and forms variations in urban atmosphere can be noticed within a city [Estacio et al., 2019].

Urban design as structure, building forms, street orientation, buildings materials are highly influenced by local climatic conditions such as local air temperature, precipitation, sunlight, wind and humidity. The urbanization process has a significant influence toward all the components involved in the city design. At the same time urbanization process is

influenced by human socio-economic activities. Due to this, in time the UHI phenomenon parameters are supposed to change. Moreover, the speed of the urbanization process and the spatiotemporal pattern of urban expansion are influenced by the socio-economic activities and their relationship with all the components in the urban areas [Zhao, 2018].

Traditional UHI measurements and comparison may show uncertainty due to the limits of discrete classification of rural and urban areas. Furthermore, local factors define each city and due to this the study of the UHI phenomenon is even more complex. Local characteristics and discrete definitions of urban and rural areas lead to a difficult comparison of the UHI phenomenon [X., 2015]. In order to overcome this challenge, the LCZ classification scheme has been proposed as the method to complete the weather analysis in urban areas using thermal distinct properties of urban surface covers [Zhao, 2018]. The LCZ classification scheme is meant to overcome these challenges by using the localized climatic conditions in order to differentiate rural areas from urban areas [Stewart and Oke, 2012]. Local Climate Zones (LCZs) framework is an approach that is also discussed and presented in the media as an important topic in the nowadays urban climate changes [New, 2015], [New, 2018], [New, 2019].

Hotter areas within the UHI phenomenon are associated with various urban physical characteristics that combine air temperature and urban surface cover. There is a lack of standardization in comparing various urban areas regarding these hotter areas and the LCZs classification is proposed as the method to describe and identify the physical characteristics that contribute to the UHI phenomenon in urban areas [Coseo and Larsen, 2014].

1.1 Local Climate Zone state of the art

Local Climate Zone (LCZ) is a classification scheme that uses a standardized scientific manner describing the land cover and the thermal performance regarding the urban heat island. The aim of LCZ classification is to have a homogeneous air temperature. Moreover, LCZ is characterized by qualitative and quantitative properties [?].

Local Climate Zones (LCZs) are those areas within urban landscapes that are constructed to be generic and have uniform surface cover structure, materials and human activities [Stewart and Oke, 2012]. LCZ approach is used in urban climate study in order to classify both built-up areas and natural features within the urban boundaries. Moreover, the LCZ classification scheme brings an important support to identify the urban arrangement and the urban morphology [Verma and Jana, 2019]. In other words the LCZ classification system is the LULC in the urban environment [Stewart and Oke, 2012]. Currently, the LCZ framework is considered as the core method in order to study the urban climate and to create a worldwide database regarding the urban morphology and urban metabolism [Bechtel et al., 2015].

Local Climate Zones represent the first urban landscape classification approach regarding the internal urban structure. The LCZs scheme can be described as complete, universal and broad regarding the internal structures within the urban areas. Moreover, the LCZs


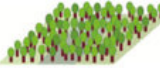

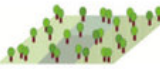

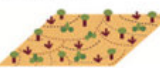






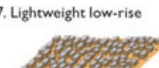




Built types	Definition	Land cover types	Definition
1. Compact high-rise 	Dense mix of tall buildings to tens of stories. Few or no trees. Land cover mostly paved. Concrete, steel, stone, and glass construction materials.	A. Dense trees 	Heavily wooded landscape of deciduous and/or evergreen trees. Land cover mostly pervious (low plants). Zone function is natural forest, tree cultivation, or urban park.
2. Compact midrise 	Dense mix of midrise buildings (3–9 stories). Few or no trees. Land cover mostly paved. Stone, brick, tile, and concrete construction materials.	B. Scattered trees 	Lightly wooded landscape of deciduous and/or evergreen trees. Land cover mostly pervious (low plants). Zone function is natural forest, tree cultivation, or urban park.
3. Compact low-rise 	Dense mix of low-rise buildings (1–3 stories). Few or no trees. Land cover mostly paved. Stone, brick, tile, and concrete construction materials.	C. Bush, scrub 	Open arrangement of bushes, shrubs, and short, woody trees. Land cover mostly pervious (bare soil or sand). Zone function is natural scrubland or agriculture.
4. Open high-rise 	Open arrangement of tall buildings to tens of stories. Abundance of pervious land cover (low plants, scattered trees). Concrete, steel, stone, and glass construction materials.	D. Low plants 	Featureless landscape of grass or herbaceous plants/crops. Few or no trees. Zone function is natural grassland, agriculture, or urban park.
5. Open midrise 	Open arrangement of midrise buildings (3–9 stories). Abundance of pervious land cover (low plants, scattered trees). Concrete, steel, stone, and glass construction materials.	E. Bare rock or paved 	Featureless landscape of rock or paved cover. Few or no trees or plants. Zone function is natural desert (rock) or urban transportation.
6. Open low-rise 	Open arrangement of low-rise buildings (1–3 stories). Abundance of pervious land cover (low plants, scattered trees). Wood, brick, stone, tile, and concrete construction materials.	F. Bare soil or sand 	Featureless landscape of soil or sand cover. Few or no trees or plants. Zone function is natural desert or agriculture.
7. Lightweight low-rise 	Dense mix of single-story buildings. Few or no trees. Land cover mostly hard-packed. Lightweight construction materials (e.g., wood, thatch, corrugated metal).	G. Water 	Large, open water bodies such as seas and lakes, or small bodies such as rivers, reservoirs, and lagoons.
8. Large low-rise 	Open arrangement of large low-rise buildings (1–3 stories). Few or no trees. Land cover mostly paved. Steel, concrete, metal, and stone construction materials.	VARIABLE LAND COVER PROPERTIES	
9. Sparsely built 	Sparse arrangement of small or medium-sized buildings in a natural setting. Abundance of pervious land cover (low plants, scattered trees).	b. bare trees	Leafless deciduous trees (e.g., winter). Increased sky view factor. Reduced albedo.
10. Heavy industry 	Low-rise and midrise industrial structures (towers, tanks, stacks). Few or no trees. Land cover mostly paved or hard-packed. Metal, steel, and concrete construction materials.	s. snow cover	Snow cover > 10 cm in depth. Low admittance. High albedo.
		d. dry ground	Parched soil. Low admittance. Large Bowen ratio. Increased albedo.
		w. wet ground	Waterlogged soil. High admittance. Small Bowen ratio. Reduced albedo.

Figure 1.1: Local Climate Zones. *Source: Stewart and Oke [2012]*

scheme is globally standardized and can be implemented regardless of time, place, and culture according to [Bechtel et al., 2015]. The LCZs classification scheme is suitable for urban studies from climate perspectives since this method uses more than just build-ups characteristics, namely climate-relevant surface properties and urban functions [Stewart and Oke, 2012].

LCZ classification is considered as a rapid mapping method and also as a reliable method while comparing across similar urban landscapes and characteristics. However, even the LCZ framework has some warnings, such as the impossibility to capture the urban growth in all the studied cities and due to this is necessary to provide local customization [Perera and Emmanuel, 2018].

Even if urban climate knowledge would help in the making decisions process at the urban level, there is a lack of data regarding such urban climate knowledge. The lack of

data includes the land use and land cover, but also the effects of the land use/land cover to the local climatic conditions. LCZ classification scheme could be the approach to overcome the difficulties caused by the lacuna of land use/land cover data and the effects of the land use/land cover on the urban climatic conditions [Perera and Emmanuel, 2018].

In order to categorize the classes while implementing the LCZ framework, it is necessary to classify several aspects such as the roughness characteristics of built-up areas, the urban geometry and the openness and vegetation as permeability features within the urban landscape [Bechtel et al., 2015]. Due to specific urban forms, geographical location of cities and urban development history on the scale of LCZs vary at global level. Moreover, heterogeneity and spatial auto-correlation are characteristics for both natural and urban landscapes. Consequently, there is very difficult to describe a region as absolutely homogeneous and it is very difficult to define clearly a boundary between distinct LCZs types [Zhao, 2018]. [Verma and Jana, 2019] present the LCZ classification scheme as an approach that clearly delineates the intra-urban classes based on a large documentation. This documentation defines the classification particularities in order to ensure a universal applicability and reproducibility of this mapping classification.

Urban planners have to integrate in the future urban planning this climate variability and the heat island phenomenon is among the most used approach in this direction. According to [Cai et al., 2016] Local Climate Zone (LCZ) classification can solve the gap between the urban development and the consequences of this development towards the local climatic conditions. LCZ classification mapping is based on the method proposed by the World Urban Database and Portal Tool (WUDAPT), which is considered as the conventional method in the LCZ scheme [Bechtel et al., 2015]. Based on the LCZ classification scheme local experts all over the world are capable of generate LCZ maps for cities around the globe by only following the WUDAPT protocol [Verma and Jana, 2019].

[Brousse et al., 2016] highlight that one of the gaps regarding urban landscape is represented by the lack of climatic relevant data in order to obtain urban climatic models for the progress in the urban landscape analysis. In order to overcome to this data gap, the World Urban Database and Access Portal Tools (WUDAPT) project has proposed a protocol aimed to create a global database and the first step of this protocol was to define the Local Climate Zones classification system. By using this protocol the urban areas are partitioned into various neighborhood types according to different parameters analyzed by using this protocol [Brousse et al., 2016]. The World Urban Database and Access Portal Tools (WUDAPT) project is focused to acquire, to store and to disseminate data regarding the cities all over the world in order to develop various tools in the field of climate research [Ching et al., 2018]. The data should focus on different aspects regarding cities' landscapes, such as the physical form of urban areas and the functions of the city supporting human activities. The physical form includes urban land cover, urban buildings geometry and construction materials types, while the functions supporting human activities include functions such as energy usage, transportation or waste production [Demuzere et al., 2019].

In the context of specific urban planning and climate sensitive in urban landscapes the LCZ classification system has meaningful advantages due to fact that this approach incorporates the urban morphology as surface cover and geometric shape, thermal properties

and anthropogenic characteristics [Perera and Emmanuel, 2018]. Another major advantage of LCZs is that this classification scheme can be used to compare into a coherent way the results obtained in various cities around the world. This comparison can be done into an easy and methodical manner. Moreover, architects and urban planners can benefit from the LCZs framework while projecting future developments of the urban landscape and even beyond the scale of the city taking into account the urban climate. [Alexander et al., 2015] mention as other LCZ framework advantages the fact that the LCZ classification is a universal approach regardless the landscape and climate conditions, it is simple to identify different categories for analyzed urban neighborhoods based on both fieldwork and digital data sources, and each LCZ type can be linked to a certain range of parameter that is specific for urban surface cover, building heights or building materials. Other advantages proposed by the LCZ approach are represented by the high degree of flexibility regarding the stages in further urban development, both in design and monitoring of these stages. This is the result of enough heterogeneity of the LCZ mapping results obtained based on the value ranges within LCZ types that will lead to a higher flexibility in the urban fabric design. Moreover, aspects such as social and economic can be also incorporated in the LCZ approach by using land use covering information [Perera and Emmanuel, 2018]. The findings of [Collins and Dronova, 2019] have shown a strong relation between the urbanization process and the LCZ changes, while the most notable LCZ transformations were observed in non-developed areas or in sparsely built-up areas. On the other hand, developed areas have shown a lower transformation regarding the LCZs classes. Moreover LCZ classifications are reliable for both past periods' analysis and change assessments on long term periods.

In order to determine the relevant Local Climate Zone for each urban block the observed data within the study area is correlated with selected data according to the guideline developed by [Stewart and Oke, 2012] giving importance to aspects such as the building types, certain building characteristics, land cover properties and land cover and land use types [Perera and Emmanuel, 2018]. Urban surface properties influence into a high degree the near-surface air temperature on the local scale. The LCZ concept is based on this local-scale while categorizing the neighborhoods, but this framework neglects the heterogeneity of the urban structure at sub-scale level and the effects of this heterogeneity towards the near-surface air temperature [Qua, 2018].

[Kaloustian and Bechtel, 2016] affirm that LCZ approach is helpful in order to develop a basic assessment for local weather, but on the other hand the accuracy is limited. In order to tackle the accuracy limitations, high resolution data both as spatial and temporal resolution should be used in the LCZ classification mapping process [Wang et al., 2017]. Since land cover information is not easy to be obtained, LiDAR derived-products can be very helpful in order to obtain more accurate results in LCZs mapping [New, 2015];[Zhao, 2018]. LiDAR products can be used in order to standardize and to identify the urban morphology for comparative studies within the metropolitan urban areas [New, 2015].

However, LCZ classification is feasible in large metropolitan areas [Kaloustian and Bechtel, 2016]. Furthermore, [Cai et al., 2016] consider that LCZ scheme can be a helpful tool in the sustainable urban development regarding the urban climatic conditions. Moreover, LCZ classification may be used as extensive input data for various urban climate simulation and

models at regional and local scales. Another limitation of the LCZ approach is considered the application of this framework in the urban planning process due to the difficulty in the process of scale selection and boundaries settings. When the study areas are very large the urban fabric mix is highly extensive and the issue of expressing a uniform characteristic for climatic conditions for that area can appear [Perera and Emmanuel, 2018].

According to [Stewart and Oke, 2012], the LCZs mapping is a three step processes. The first step is to collect the metadata site, the second step is to define the thermal source area and the third step is represented by selecting the local climate zone.

LCZ classifications can be done applying three different methods, namely the manual sampling method, remote sensing method and the GIS method [Bechtel et al., 2016b]; [Zheng et al., 2017]. All these three methods are different due to the data sources and the methodology involved in the analysis. Moreover, the GIS method used in LCZ mapping has other two approaches: the one based on raster datasets and the method based on vector datasets [Zheng et al., 2017]. Estacio et al. (2019) consider that remote sensing method used in the LCZs classification requires local knowledge. Moreover, [Estacio et al., 2019] affirms that remote sensing method may bring errors in the process due to the need of creating training areas, while GIS-based method is considered as a solution to solve the problems of remote sensing method.

The GIS-based methodologies used in the LCZ classification are based on surface, derived geometric and radiative properties in order to differentiate the types of LCZs classes. In order to define distinct layers of LCZ classes fuzzy logic and reclassifying techniques are utilized. However, even these methods have a problem, namely the noisy pixels that are produced requiring filtering processes [Estacio et al., 2019]. In order to solve these issues related to the GIS-based method in the LCZ classification process, [Estacio et al., 2019] have developed a GIS toolbox called “Local Climate Zones”. This toolbox incorporates specifications of fuzzy logic and cellular automata specifications and it may be used for any city only by applying the property layers of the studied area.

Various methodologies have been used in order to develop LCZ classification. Remote sensing method is used in the LCZ classification scheme by [Brousse et al., 2016], while [Zheng et al., 2017] uses GIS raster-based method. The GIS vector-based method has been used by [Levlovics et al., 2013]. [Zhang et al., 2019] use the remote sensing method combined with OpenStreetMap (OSM) data, while [Vandamme et al., 2019] has used a methodology combining manual training areas sampling and remote sensing method. [Collins and Dronova, 2019] developed a research related to LCZ scheme using OBIA as remote sensing method, while [Wong et al., 2019] has used Sentinel-2 satellite image in order to derive LCZ classification scheme in the urban areas in the Pearl River Delta, China. [Brousse et al., 2016] propose an approach where the LCZ classification scheme is compared with the CORINE land cover data. Furthermore, by implementing OBIA methodology in the LCZs classification scheme, [Collins and Dronova, 2019] have identified that there is a need to standardize and refine the results of LCZs framework in the validation step due to the various strategies implemented in accuracy assessments in this field of research. The large variety of accurate assessments regarding LCZs framework is a great challenge to design a generalized approach regarding a certain method.

Local Climate Zone (LCZ) scheme has been used in various studies all over the world in order to analyze the cities climate change at local level [Leconte et al., 2015]. Among the cities where the LCZ scheme has been developed could be mentioned Szeged, Hungary [Levlovics et al., 2013]; Chicago, USA [Coseo and Larsen, 2014]; Singapore, Singapore [X., 2015]; Nancy, France [Leconte et al., 2015]; Dublin, Ireland [Alexander et al., 2015]; Olomouc, Czech Republic [Lehnert et al., 2015]; Beirut, Liban [Kaloustian and Bechtel, 2016]; Guangzhou, China [Cai et al., 2016]; Madrid, Spain [Brousse et al., 2016]; Dallas-Fort Worth, Austin, and San Antonio metropolitan areas, Texas, U.S.A. [Zhao, 2018]; Hong Kong [Zheng et al., 2017]; Colombo, Sri Lanka [Perera and Emmanuel, 2018]; Berlin, Germany [Qua, 2018]; Kunming, China [Vandamme et al., 2019]; Salt Lake Metro Region, Utah, USA [Collins and Dronova, 2019]; Brno, Czech Republic [Geletič et al., 2019]; Mumbai, India [Verma and Jana, 2019]; Quezon City, Philippines [Estacio et al., 2019]. On the other hand, there are various studies using LCZ classification scheme to analyze climatic conditions in urban environments by comparing several cities from various continents such as Amsterdam, Chicago, Madrid and Xi'an [Zhang et al., 2019], while [Demuzere et al., 2019] have developed a research using the LCZ classification framework for the entire European continent and then focusing on several cities in Europe.

The LCZ framework is widely used nowadays in order to analyze the urban thermal conditions in urban areas. During the classification process of the LCZ mapping both optimal scale analysis and local knowledge are required. Local knowledge during the LCZ approach refers to detailed information regarding the topographic features, land surface characteristics and the socio-cultural background. Moreover, by obtaining an image regarding the urban climate landscapes an overall climate description of the urban landscape can be generated [Wang et al., 2017]. By using the LCZ classification system the heterogeneity of urban landscapes can be revealed and the spatial arrangements of built-up areas and green spaces may be presented into an appropriate manner in order to provide strong arguments for the assessment of these spatial arrangements conditions [Nedkov et al., 2017].

By using the LCZ scheme the urban heat island phenomenon can be redefined in magnitude and this is an important advantage of LCZ scheme [Leconte et al., 2015] since the “urban” and “rural” classes are replaced with temperature differentiation using LCZ classes [Cai et al., 2016]; [Bechtel et al., 2016a]. There are 17 standard (basic) types of classes in the LCZ scheme having 10 built classes and 7 land cover classes [Bechtel et al., 2016a], [Cai et al., 2016].

Using these 17 standard types of the LCZ scheme, the intra-urban temperature is analyzed into a comparative approach regarding the different urban classes analyzing the effects of heterogeneous urban morphology on the climatic conditions at urban level [Cai et al., 2016]. Each one of these zone types is derived using different combinations that can be read regarding the urban cover, urban fabric, urban functions, urban forms and urban metabolism. All these features can generate a certain impact on the near surface climate and this impact can be measured [Bechtel et al., 2016a].

The level of information while using the LCZ classification provides more information on each zone type in the urban land cover. Local Climate Zones address to different uniform regions within the urban environment such as urban morphology, urban cover, materials and human activity [Bechtel et al., 2016a]. Characteristics of the LCZs regions has to be

uniform and based on these characteristics the sizes of these regions vary from hundreds of meters up to thousands of meters. These characteristics of the LCZs regions are highly influenced by the local surface characteristics [Zheng et al., 2017]. For each urban category the LCZ classification scheme has certain characteristics and variables that describe as value ranges the material properties and the urban morphology [Brousse et al., 2016]. Urban types in the LCZs classification scheme are associated with numerical values based on the key variables of each type. These numerical values are used in order to create a model revealing the atmospheric response to the urbanization processes [Stewart and Oke, 2012];[Ching et al., 2018].

Among the difficulties during LCZ studies could be mentioned the fact that numerous structures are fitting at classes edges during the classification process, the difficulty to estimate the building height and the high diversity of the background image regarding the land use and land cover [Kaloustian and Bechtel, 2016]. Making a comparison of two different LCZ classification schemes is difficult due to the fact that classes of each scheme cannot be harmonized direct manner [Bechtel et al., 2016a].

The accuracies for LCZ classification are varying from study to study. If [Cai et al., 2016] have accuracies from 80percent to 84percent, [] Ren et al. (2019) report overall accuracies ranging from 60percent to 89percent. According to Ren et al. (2019) the classification accuracies have large variations especially in built-up classes due to the poor performance in compact and open buildings classification. Other factors that contribute to these variations in accuracies are represented by the number and the quality of training samples, but also classes' similarities.

Based on the study developed by [Vandamme et al., 2019] the LCZ scheme is an approach that can be used with confidence in order to analyze the spatiotemporal dynamics in the land-use and land-cover in urban areas, to create assessments regarding the urban planning policies and strategies and also to offer a simple approach to monitor the urban landscape changes to allow urban planners to develop rational strategies for sustainable development of urban landscapes. Moreover, [Vandamme et al., 2019] propose combining LCZ classifications with thermal environmental studies in order to provide for urban planners a feasible tool to design cities that in the future would be more inhabitable.

Spatiotemporal dynamics of the LCZ results are caused mostly by the urban planning policies that were implemented in order to design an urban environment that is more suitable for living [Vandamme et al., 2019]. Moreover, [Collins and Dronova, 2019]) considers that LCZs framework can represent a reliable source of information regarding the urban transformation as a consequence of urbanization. Furthermore [Collins and Dronova, 2019] recommended LCZs classification as a tool to characterize the relevant aspects to microclimate level in the urban landscapes.

[Perera and Emmanuel, 2018] state that certain requirements are necessary to be met by data and process before the LCZ approach is carried on in the urban planning. These requirements of the LCZ classification approach are distinctive for each stage of the framework process, whether it is about the planning and policy phase, the design and approval phase or the monitoring and compliance phase. For the primary stage the free maps available are extremely important as a source for building footprints, roads, natural

features and land use. All these features can help in the understanding of the urban fabric as key elements in the climate-influencing parameters. Another requirement in the first stage of the process is to be familiar with the planning process and the desired urban landscape in the future. In the second phase the requirements include transparency and certainty in order to ensure the utilization of the resources into an appropriate manner and timely decision making. The last stage requires a continuous update and maintenance of the database with the urban characteristics and features. Moreover, within this phase of the LCZ approach the socio-economic feedback is an important aspect to have in mind (Perera Emmanuel, 2018).

The building level regulations are not negated within the LCZ classification approach since these regulations are the spine of the built-up areas within the urban landscape. The LCZ framework discusses about several characteristics of the building level controls such as plot surface, street width, surface fraction, height of the buildings and thermal conditions of the buildings [Perera and Emmanuel, 2018].

However, [Coseo and Larsen, 2014] consider that one of the problems with the LCZ classification system is represented by the reducing of the urban variables that contribute to higher temperatures within urban sites. Aiming to provide weather sensitive development aspects such as building regulations and zone specific density of buildings are highly important to city scale and neighborhood level. Among the building regulations that are sensitive regarding urban climate may be included aspects like the mitigation of UHI, indoor energy use and outdoor thermal comfort [Perera and Emmanuel, 2018].

[Demuzere et al., 2019] point out that a LCZ mapping at large scale is possible while using computational techniques and enough training areas for the studied areas. Furthermore, states that a complete and consistent LCZ map on continental scale can be obtained based on sufficient training areas and cloud computing resources in order to ensure support in the global climate research.

1.2 Previous study findings

By implementing the LCZ framework [Geletič et al., 2019] highlight that the highest number of hot days in the summer is situated within the compact development and/or midrise zones within the city of Brno, Czech Republic. On the other hand, the lowest number of such hot days is expected to be in the zone describing the forested areas. Moreover, forested areas and open areas with low plants show the lowest values attributed to tropical nights and warm nights, while the highest values for these nights are expected to be seen in compact midrise development, compact low-rise development and industrial areas [Geletič et al., 2019].

Based on the findings of their research in Chicago, [Coseo and Larsen, 2014] state that the LCZ system should provide more than just standardized characterization of physical features at local level in urban areas and it should provide the image of urban landscapes into a realistic manner.

[X., 2015] has identified using the LCZ classification mapping approach in Singapore

that green space has lower temperature compare with Central Business District (CBD). The findings show that in order have a city that is more resilient to climate change impacts and more liveable, such green space is very important.

Based on the LCZ classification scheme developed in Szeged, Hungary, [Levlovics et al., 2013] have found that higher temperatures are characteristic to those areas having compact built-up space and midrise areas, while surfaces having lower temperatures are open space and lowrise areas.

The findings of [Qua, 2018] applying the LCZ classification scheme in Berlin, Germany shows that even if the LCZ types seem to be very homogeneous, the air temperature pattern has unique characteristics as a consequence of the large variety of surface cover properties in the urban areas.

1.3 Problem statement

According to the literature research, the problem statement is:

Copenhagen Municipality plans to extend inhabited areas in the future [TheLokal, n.d.] with soil reclamation projects, while there is a global climate change trend,at a micro-level this can produce a climate change by changing the land surface temperature of the areas.

Based on this problem statement the following research questions are automatically triggered :

1. How Local Climate Zones evolved during the last 5 years in Nordhavn?
2. How can LCZ methodology helps spotting the micro-climate differences in Copenhagen's newly urbanised areas?

1.4 Study Area

Nordhavn is a big new district in the city of Copenhagen, Denmark. This new district is expected to be home for more than 40,000 people and it should generate almost 40,000 new job opportunities according to the City and Port Development report from 2012 [Nissen, 2014]. Copenhagen is among the most appreciated cities in the world considering the living conditions and the urban planning aspects. Copenhagen has been awarded in recent years descriptions such as one of the most liveable cities (according [Monocle, n.d.]) or as one of the top ten healthiest cities in the world (according to CNN) [Nissen, 2014].

Nordhavn is an idea that starts from several objectives that are aimed to be achieved while planning and developing the new urban districts. Accordingly, Nordhavn is a project that intends to be eco-friendly, a vibrant community, a dynamic city that supports sustainable mobility and a city for everyone [Nissen, 2014]. The new urban district in Copenhagen is aimed to offer a green infrastructure and to support sustainable development of urban space from both environmental and social perspectives[Nissen, 2014]. The

project of Nordhavn proposes a low temperature heating-district based on smart energy projects, smart houses having intelligent systems and seawater and groundwater used for cooling systems. Moreover, public lighting will be based on solar and wind energy. By implementing these approaches this new district will provide an optimal energy storage, use and management, but also sustainable transport [Nissen, 2014]. Nissen (2014) quoting Nielsen (2008) mentions that nowadays climate and sustainability aspects are intensely discussed and considered among the biggest challenges of modern times. These aspects are also discussed in the context of urban planning and development. Consequently, environmental and sustainability aspects are addressed in the urban planning context in Nordhavn district also.

Methods 2

This chapter reveal the methods implemented in this project. The reader will be familiarised with the generalities of the methods to prepare him with an overview of the direction chosen. Further details and are revealed in chapter 4 to present how the methods are used and combined for achieving the desired result.

2.1 Process for the project

2.1.1 Land Surface Temperature Diagram

This part shows the diagrams of the flowchart for this project.

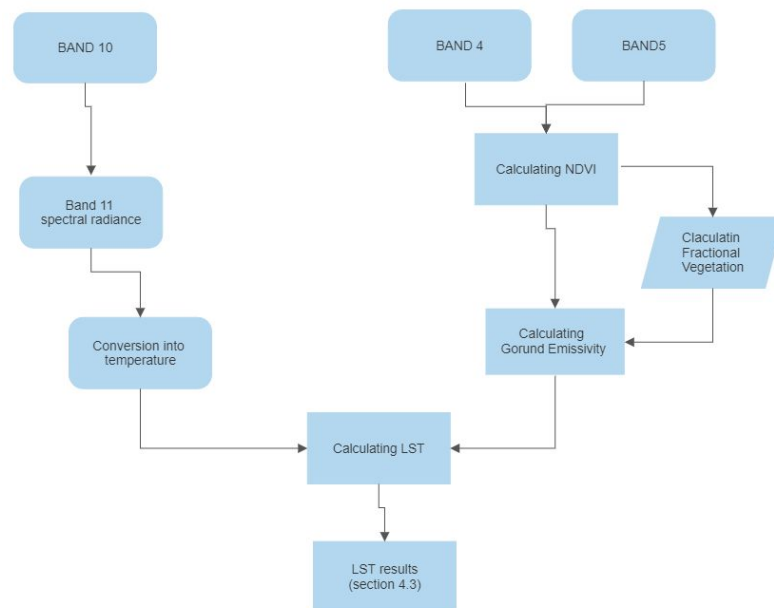


Figure 2.1: Land Surface Temperature Flowchart

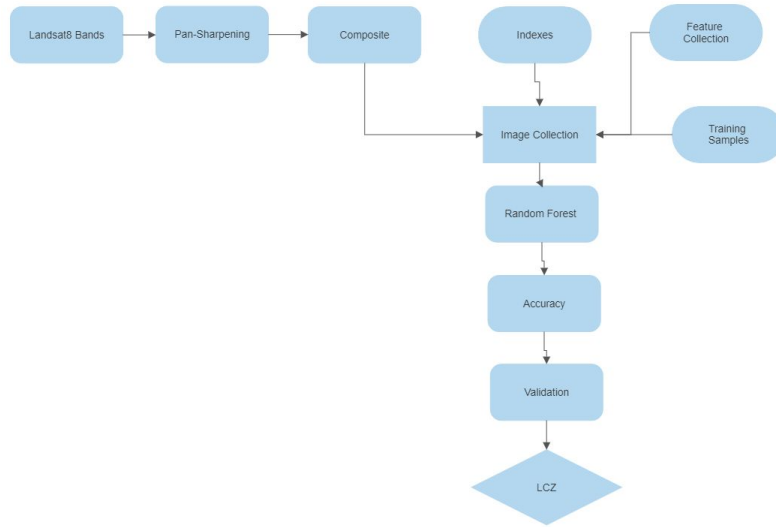


Figure 2.2: Local Climate Zones Flowchart

2.2 Software

In this section it is presented the software used for remote sensing analysis which is Google Earth Engine. The description of the software is at a general point of view.

2.2.1 Google Earth Engine

A cloud-based platform released by Google with a great capability of geo-spatial analysis released as a servant for monitoring several global issues like : climate monitoring, deforestation, drought, environmental protection. It's uniqueness integrated platform come in hand for the remote-sensing scientists which rely on big computational power. It is a analysis ready data catalog accessed throughout an application programming interface (API) corelated with an interactive development environment(IDE). [Gorelick et al., 2017]. On a freely user basis it can be accessed throughout the homepage <https://earthengine.google.com>. In this project Google Earth Engine is used from the beginning to the end for downloading the satellite image collections - analyzing - visualisation. The following chapters will describe how it works and why it was chosen for the process.

2.2.2 Tools

In this section the tools used for this project in Google Earth Engine will be described. The tools described in this section are used to derive data out of satellite images and run analysis on it.

2.2.3 ee.ImageCollection

Is a collection of images collected by satellite and indexed by ID; the image can afterwards be displayed introducing the image ID (timestamp/location) into the ImageCollection constructor.[Google Earth Engine, n.d.b]

This tool is helpful for reducing the time and hardware size used for displaying satellite images directly in the console; used for downloading projcet's ground start data.

2.2.4 ee.Filter

Coming in handy for the user, Google Earth Engine provides ee.Filter as argument applicable for dates and bounds like imageCollection.filterDate and imageCollection.filterBounds.[Google Earth Engine, n.d.f]

This is used in the project for the easiness of displaying satellite images according to the time-stamps and areas requested.

2.2.5 ImageCollection.mosaic

This function allows compiling spatial data into a continuous image based on the differences sorted out by time and location resulting a maximum value composite. By mosaicking it is created an overlap of the images arranged by the date registered into the collection where the top place will be always the newest record.In this way the image collection reduces.[Google Earth Engine, n.d.h]

2.2.6 Compositing

While Landsat 8 pictures the same location every 16th day which results a collection of at least 12 images, in a 6 months period, Google Earth Engine reduces the collection to only one image known as temporal reduction.In this case using a **reducer** triggers the way Earth Engine will do a pixel overlap, so every pixel displayed in the Console is derived. Moreover Earth Engine will primarily place on top the latest pixel which in this case a median pixel value can be triggered using **median()**. [Google Earth Engine, n.d.c]

2.2.7 Masking

Masking helps improve the analysis data by excluding desired pixel turning them into transparent. To mention here that on each band a pixel will always have a mask.Moreover the pixels presenting a value above 0 will be turned over in this way considering 2 images masked will turnout an image where the pixels with value 0 will changed with the pixels from the second picture if there wil be a value above 0 for each pixel of the previous image.[Google Earth Engine, n.d.g]

2.2.8 Normalized Difference VegetationIndex-NDVI

NDVI is an index which measures the vegetation based on remote-sensing. In this case the difference from near-infrared(reflected by vegetation) and red light(absorbed by the vegetation).

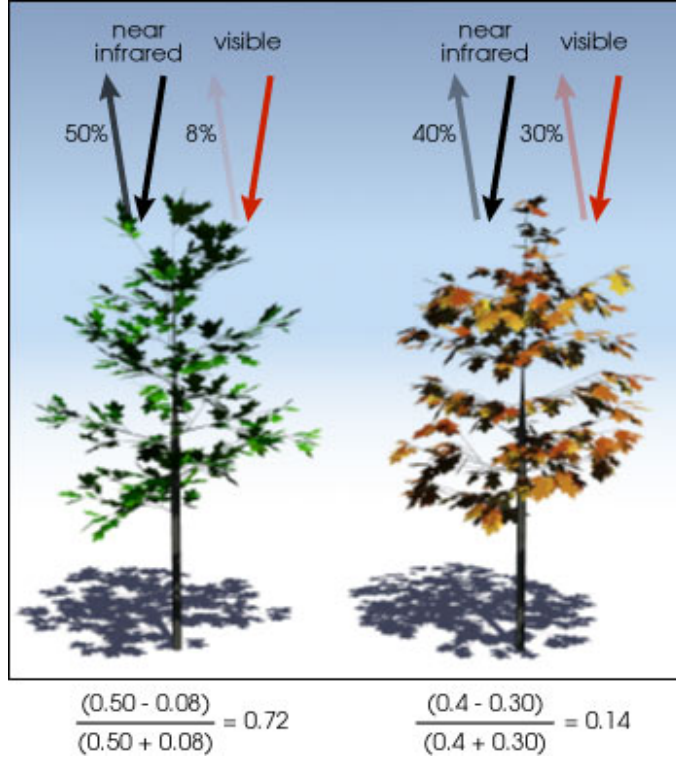


Figure 2.3: NDVI-visualisation. *Source: GisGeography.com [n.d.]*

Important to mention is that the result will always be in between -1 and +1.

$$NDVI = (NIR - RED) \div (NIR + RED) \quad (2.1)$$

[Bakker et al., 2001]

In this project NDVI is essential by its capability of differentiating the different vegetation types whereas for creating the correct classes of the training sample which relies on the whole mapping process.

2.3 Google Earth Engine

Google Earth Engine (GEE) is a platform, which can be used for geospatial analysis. This can be done in Google's infrastructure, meaning that a lot of the images and data are the same for Google Earth and Google Earth Engine, though only some of them are available for GEE. Images are being collected daily and are available for everyone. The engine stores satellite imagery in a public archive, and it has images from more than 40 years ago [Google Earth Engine, n.d.d] and [Google Earth Engine, n.d.e].

The platform can be interacted with in several ways: either by the code editor, by the explorer or by the client libraries. The code editor is a web-based IDE for scripting and is designed to make geospatial analysis easy to work with. The Explorer for Google Earth Engine is a simple to use web app, which can be used to visualize data and do simple analyses [Google Earth Engine, n.d.d].

In this project Google Earth Engine is used to calculate the monthly average NDVI of Copenhagen. While there are other remote sensing options for studying the spatio-temporal pattern of green spaces, NDVI seemed suitable for an area such as Copenhagen.

Theory 3

Throughout this chapter, different theories which are relevant in order to answer the research questions, will be described. The theories that are included are about Landsat8, Land Surface Temperature, Google Earth Engine, the impact of green areas how calculations of this can be done. In chapter 4 it will be explained how these theories will be implemented in the project.

3.1 Landsat Programme

The Landsat Program is one of the longest data archive which started with the first launch in 1972 and continued until present with Landsat 8 launched in 2013 and with an expectation for 2021 for launching Landsat9[Landsat Science, n.d.b]. The satellites of the Landsat program "have created the longest continuously-acquired space-based, moderate-resolution data archive." [USGS, n.d.] due to the longevity on the orbit.

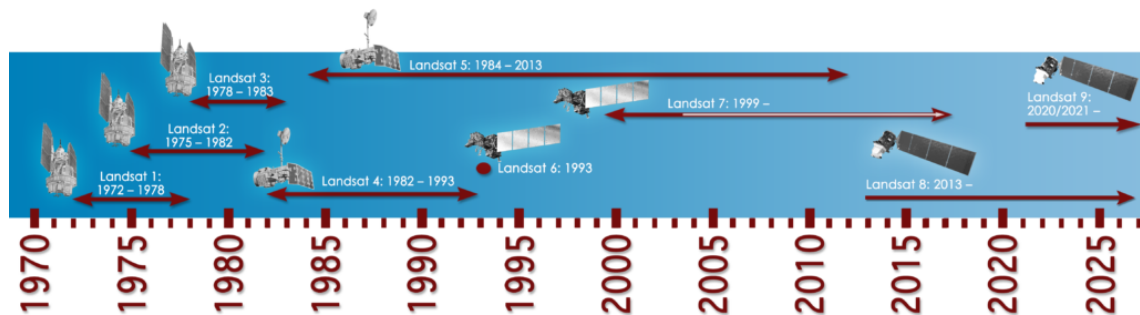


Figure 3.1: The coverage period of each landsat satellite. Source: Landsat Science [n.d.a]

3.1.1 Satellite scanning

Human eye perceives only the Red Green Blue segments of the spectrum and important is to mention that each object has a spectral signature according to the radiation generated by the light impact or any other electromagnetic wave.

The electromagnetic spectrum is the total range of the wavelengths. The figure 3.2, shows the classes that is divided into. Human eye can see only the Red Green Blue classes.

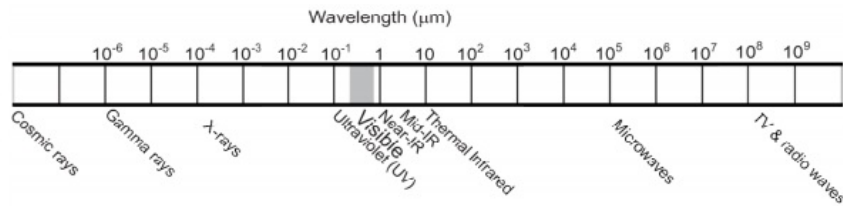


Figure 3.2: Wave length. Source: Wim H. Bakker [2009]

As an example, the wave generated into the red spectrum allows to visualise different types of vegetation. Google Earth Engine comes up with possibility of studying vegetation based on different vegetation indexes that will lead to specific types of healthy, non-healthy, stressed vegetation or above ground biomass estimation.

In the figure 3.3, it is shown how sensors are using the solar energy combined with the reflected energy.

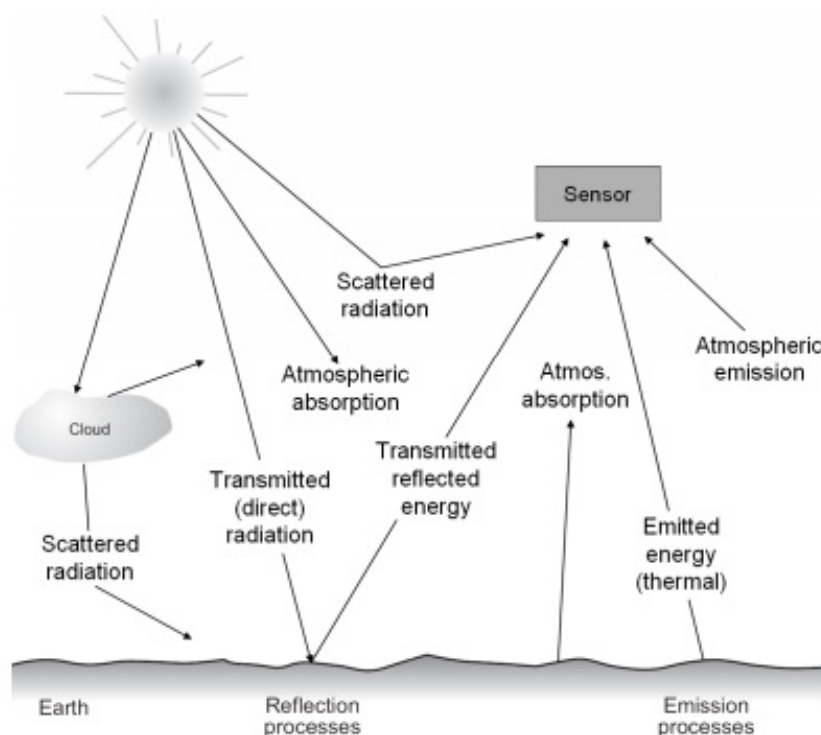


Figure 3.3: Sensor principle Source: Wim H. Bakker [2009]

Multi-temporal satellite images are the best choice for studying trends and most precise way to evaluate a prediction model. This image collections are the best contributors for remote-sensing analysis.

Landsat started in 1972 with Landsat and it is still running and providing global images. It started with Landsat 1 placed on orbit and continuing modifying the sensors up to

Landsat 8, which was released in 2013. This program offers a collection that helps analyzing earth [Wim H. Bakker, 2009].

3.2 JavaScript(JS)

JavaScript (JS) represents a programming or scripting language that helps users to achieve various complex elements or items on website pages and it is seen as the best known programming language for web pages, even if it is also used in non-web environments. It is extremely important to know that JavaScript is not the same thing with Java programming language. Even if both Java Script and Java programming language are developed by Oracle, they have significant differences in syntax, semantics and uses.[JavaScript, n.d.a]

Moreover, JavaScript is considered to be a programming language described as lightweight interpreted due to the fact that the web browser runs the script from the original text form received from the JavaScript code. In order to use JavaScript into a web page, user needs only the script element for the HTML page. Very often JS is reckoned also as just-in-time compiled programming language, meaning that scripts and codes are compiled during the execution of the script instead of being compiled before the execution. JS is also described as object-oriented programming language. However, JS can function also as a procedural programming language.[JavaScript, n.d.a]

This script language is available on the client side of the web. It is helpful to establish how a web page would behave when a certain event occurs. Since JS is an easy to learn programming language, it is commonly used in order to control the behavior of the web pages. JS is easy to learn and easy to use because it brings numerous standard built-in objects that may be used by developers to implement on web pages. All these built-in objects have detailed methods and properties in order to help developers to use them.[JavaScript, n.d.a]

JS is considered as the third layer of the standard web technologies, while the first two of these layers are considered HTML and CSS. Even if at first sight web pages seems only to display static features and information for users, it should be known that the web pages have to display constant updates, animated graphics that are 2D or 3D, interactive maps or even scrolling videos. It is very probable that JavaScript is implicated in all these features and elements displaying on the web pages.[JavaScript, n.d.a]

The first layer of standard web technologies, HTML, is a language used in order to define the content uploaded on the web pages. HTML helps users to define paragraphs, tables, headings or to embed images, graphs or videos in the web page.[JavaScript, n.d.b]

The second layer, CSS, is used in order to provide style and design to the HTML content previously created. Accordingly, the CSS language may help users in order to establish certain background colors for web pages, certain fonts or to display content in columns. After HTML and CSS technologies, JavaScript comes to help users to update into a dynamic mode most of the features present on the web pages (text, animated images, multimedia etc.).[JavaScript, n.d.b] JS programming language work using various types of data: strings, numbers, lists, objects, dictionaries and functions. The dynamic capabilities

of JavaScript programming language include several elements: runtime object construction, function variables, variable parameter lists, object introspection and source code recovery. Through function developer can store part of a programming code that generates only one task in a coding block. Moreover, a function allows users to call the code any time only by using a short command.[JavaScript, n.d.c]

There are some common programming features describing JavaScript. This programming language can store important values inside different variables, can support operations on text or it can run different programming codes in order to provide a response to various events that may happen on the web page.

Furthermore, JavaScript has also a top functionality for the client-side language, namely Application Programming Interfaces (APIs). This functionality provides superpower to the users that need to implement JavaScript languages. APIs represents “prefabricated” batches of different code building blocks that may help a developer to create complex programs. Without using APIs would be very complicated and sometimes even impossible for the developer to implement such complex programs. Two categories of APIs are ready for web developers. The first category is represented by browser APIs. These APIs are implemented in the web browser pages and they can display data from computers located in the surroundings or to run effective and complex things. The second category of APIs is known as third party APIs. These APIs are not available by default into the web browser. Accordingly, developers must search for code or information in order to comply with a code from other web pages.[Google Earth Engine, n.d.a]

JS runs on many web pages such as Google Maps, PayPal or Facebook. Accordingly, JS is part also of the Google Earth Engine (GEE). The JS APIs implemented in GEE may help users to run a wide range of tasks. For example there are JS APIs to visualize images and band images, to add a layer to the map, to customize the layer visualization, to compute various terrain indices using images etc.[Google Earth Engine, n.d.a]

3.3 Google Earth Engine (GEE)

Google Earth Engine (GEE) is recognized as the largest and most advanced geospatial processing platform in the world that is cloud-based. The aims of the GEE is to offer an interactive platform that allows the development of geospatial algorithm at scale, to ensure a high impact in the geospatial environment, to support the science based on data and to enable the progress regarding global challenges that require large geospatial datasets. GEE is made of several components as follows: geospatial datasets, computational infrastructure, APIs and apps. Geospatial datasets are represented by all the data obtained using remote sensing methods and other spatial data acquisition methods. These datasets are growing very fast constantly and the datasets used in GEE are publicly available. Computational infrastructure refers to Google’s tools allowing parallel processing of geospatial datasets. APIs used in GEE include JavaScript, Python and REST, while the apps include all those applications that are built using APIs in GEE and that are dealing with geospatial datasets. Google Earth Engine as a cloud-based platform for spatial analysis at global scale can support various social and environmental issues such as deforestation, natural

hazards, climate change and monitoring, food security, environmental protection and water management. This is due to Google’s computational infrastructure implemented in the GEE [Gorelick et al., 2017]. Google Earth Engine has been established at the end of 2010. Besides the satellite and geospatial data, GEE provides access to algorithms for processing large datasets and to cloud computing [Kumar and Mutanga, 2018].

GEE may be easily accessed and also controlled by using an application programming interface (API) that is accessible via Internet. This platform is also associated with an interactive development environment (IDE) that provides a fast visualization and prototyping of the results [Gorelick et al., 2017]. In order to have access to Earth Engine user interface API it should be used the [ui] package. This package allows users to create graphical interfaces for all the scripts that users need to use in GEE. Graphical interfaces for GEE scripts can be simple input widgets, more complex widgets, panels to handle the user interface layout and also event handlers to highlight the interactions between the widgets that are part of the user interface. Simple input widgets are represented by buttons and checkboxes, while complex widgets include maps and charts. GEE hosts a large spatial data catalog that is publicly available. The datasets are a results from a large variety of aerial and satellite imaging systems, climate and weather forecasts, environmental variables, land use and land cover data, socio-economic datasets and also topographic data. It is important to highlight that the entire data catalog available on GEE provides preprocessed data that is ready-to-use [Gorelick et al., 2017].

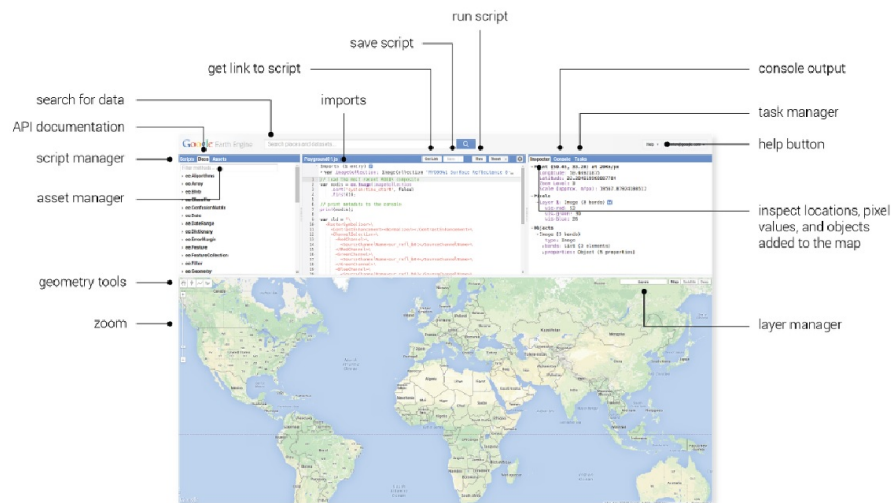


Figure 3.4: Google Earth Engine
[GoogleEarthEngine, n.d.]

The datasets available through GEE include Landsat, MODIS, Sentinel and High-Resolution imagery. The available datasets cover a wide range of scientific domains such as weather and climate, surface temperature and geophysics. The archive used in GEE covers more than 40 years of historical geospatial datasets, but also up-to-date spatial data. According to [Gorelick et al., 2017] the data catalog of GEE is updated continuously with about 6000 imagery scenes each day, these scenes being obtained from active missions and they are usually uploaded in about 24 hours from the acquisition time. Due to this, GEE platform allows measurements and monitoring of the dynamics at a global scale in

the Earth's environment. Measurements are based on a large catalog providing Earth observation datasets. Moreover, GEE ensure computational access to an impressive number of computers in Google's data centers in order to run intrinsically-parallel processing of Earth datasets [Moore and Hansen, 2011].

[Johansen et al., 2015] state that GEE is a free and globally accessible source of online geospatial datasets and also of various processing tools in land cover dynamics researches and studies. GEE aims to ensure access for scientists all over the world and for developing world countries to large spatial datasets in order to enhance and to strengthen the understanding of natural resources in public institutions and among the civil society [Moore and Hansen, 2011]. Google Earth Engine (GEE) is a reliable, powerful and rapid tool in land cover mapping research and analysis ([Huang et al., 2017]; [Campos-Taberner et al., 2018]). This cloud-based platform is also very flexible for Earth observation research [Kumar and Mutanga, 2018]. [Ravanelli et al., 2018] highlight that using GEE a wide spatiotemporal and long term approach can be implemented in the research of the urban heat island and land cover change by processing a large database of Landsat images. GEE is a feasible tool in order to analyze and to detect land cover dynamics by comparing recent land cover types with archived Landsat images [Huang et al., 2017]. Various studies have used Google Earth Engine in mapping land cover changes and dynamics. One study is combining Landsat images and GEE in order to detect major land cover dynamics using NDVI has been developed by [Huang et al., 2017] in Beijing, China obtaining an overall accuracy of 86.61percent for the classification map. Another research using GEE and Landsat images has been developed in Queensland, Australia by [Johansen et al., 2015] in order to analyze the vegetation change map. [Dong et al., 2016] have combined GEE, Landsat 8 imagery and phenology-based algorithm in mapping paddy rice planting areas in NE Asia, obtaining a producer accuracy of 73percent and a user accuracy of 92percent. [Xie et al., 2019] state that an automated land cover mapping approach can bring accurate results using the GEE cloud-based platform based on their study developed in the north of China.

Besides land cover mapping, GEE platform is also used in other studies all over the world. Biophysical variables are mapped and analyzed at global level using MODIS datasets in the GEE cloud-based platform [Campos-Taberner et al., 2018], while urban heat island phenomenon is monitored together with land cover dynamics using GEE by [Ravanelli et al., 2018]. Among the major advantages of using GEE as a combination of science, massive geospatial datasets and computational techniques may be mentioned the high computational speed, transparency in using methods and datasets, the ease-of-use of this tool, but also the enhancement in collaboration [Moore and Hansen, 2011]. Since satellite and aerial imagery have very often surfaces covered by clouds, [Mateo-García et al., 2018] have proposed a methodology in order to detect clouds in the GEE geospatial processing platform. [Kumar and Mutanga, 2018] consider that another important advantage of the GEE is that users don't need any more to store, to process and to analyze large geospatial datasets on their office computer or on their personal computer. Moreover, users are not dependent on remote sensing software to analyze satellite data since GEE allows a large variety of remote sensing methods, except from object-based image analysis [Kumar and Mutanga, 2018]. Users can access both public catalog and private spatial data. The public catalog is provided through GEE user interface, while private data can be accessed and analyzed based on the library of operators that are available in the Earth Engine

API. All these operators are part of a massive parallel processing system. This system is automatically subdivided and it assigns the computational operations offering high spatial analysis returns [Gorelick et al., 2017].

In order to access GEE users should utilize the web browser and the Earth Engine Code Editor. There are 4 modules existing in the Code Editor of the GEE, namely Manager, Code Editor, Console and Map. Manager is the elementary module of the file system in the GEE and it contains additional 3 sub-modules: Scripts, Docs and Assets. Scripts module is that part that stores all the scripts created by the user in addition to the scripts from Google. Docs hosts the entire documentation for the GEE, while Assets module is that one that allows users to upload their local files to GEE. Code Editor is that module that allows users to write and to execute all the scripts created in JavaScript. Here scripts are formatted and any problem in users' scripts is underlined and autocompleted where it is necessary. Moreover, this module provides hints to use GEE functions. Above the Code Editor, users can find buttons that allow to run, to reset or to save scripts. Console is another module in GEE that has 3 sub-modules as follows: Inspector, Console and Tasks. Using Inspector module users can query into an interactively manner the map by clicking it. Console sub-module helps users to obtain metadata from those files used in scripts, while Tasks sub-module permits users to view those tasks that need more time in order to be completed after users have submitted them to the Earth Engine platform. The 4th module is Map and it allows users to have the visual output of their scripts. Users will have their scripts' results overlying a 15-meter base map image of Earth surface. Google Earth Engine is a tool that allows to process substantial amounts of data into useful information regarding a wide range of environmental issues. GEE is designed to work with large datasets and this can be a great support for researchers using remote sensing data. Moreover, GEE provides the opportunity to share scripts and codes among users. Consequently, users do not have to gain knowledge of coding in Python or JavaScript. Furthermore, even researchers from developing countries without access to spatial datasets or computer hardware and software have now the opportunity to run spatial analysis using GEE [Kumar and Mutanga, 2018].

ee.FeatureCollection

FeatureCollection permits to combine various groups of features that relate. By combining these groups of features additional operations are available, such as sorting, filtering or rendering. In order to generate a FeatureCollection, users should provide a list of features to the constructor. It is not necessary that properties and geometry are the same for the features. In order to upload one of the numerous table datasets hosted in EarthEngine, users should use the table ID during the construction of the FeatureCollection. Moreover, a FeatureCollection can also include individual geometry of only one Feature.

ee.Date

Earth Engine uses date objects in order to represent time. In order to construct an *ee.Date* users can choose various options: a string, JavaScript Date or the static methods included in the *ee.Date* class. Date objects are utilized to filter the collections.

GetData.ImportData is used to get data from Excel files, such as .xlsx or.csv in order to be used in Earth Engine or to import vector and raster data. *.median()* This function is used

in the reduction of image collection (`ee.Reducer`). When using this function the pixel value is calculated as the median value for the pixels selected in the script. *Map.CenterObject* This tool allows users to center the map on the object used in the analysis and then display that image at a zoomed level. Higher numbers indicate a more zoomed in display meaning a larger scale.

Map.addLayer

Using `Map.addLayer` users can add directly to the map feature collections. Moreover, `Map.addLayer` may be useful to obtain the desired visualization effects when a layer is added to the map since the Code Editor assigns RGB values. By using `Map.addLayer` desired parameters for visualization may be provided.

.bandNames

Using this function will provide information about satellite images bands by choosing each band by its name. Band names can be printed as a list or band names can be discovered by displaying the image on the console and inspecting it.

.select(bands).sampleRegions() Is used in order to create a training dataset by extracting pixels value from satellite images. Using it the training points are overlaid on the satellite image in order to get the training dataset.

.randomColumn() is used to create random sampling on the training points used to extract pixels values. By utilizing *randomColumn()* users can create manually multiple classifiers and may perform manually bootstrap. Moreover, *randomColumn()* may help in order to take only one sample to train a classifier. While using *randomColumn()* to *ee.FeatureCollection*, a column of deterministic pseudorandom numbers is added to the collection.

filter() This constitutes one of the methods provided by GEE as filtering methods for the image collections. The complete filtering functionality of Earth Engine can be accessed using this feature with an *ee.Filter* as argument.

RandomForest One of the most used classification methods is *RandomForest*. It is included in the GEE Classifier package and it can be used as an algorithm in the GEE classifications. This method is used for classification, regression and different tasks in order to provide various decisions trees during the training time. The output is the class representing the mode of the classes for the classification process and the class representing the mean prediction for the regression process.

ee.Classifier.smileRandomforest().train() *ee.Classifier* is the package that includes algorithms used for the supervised classifications in Earth Engine. Among the classifiers are included Random Forest, CART, SVM and NaiveBayes. Using *ee.Classifier.smileRandomforest* an empty Random Forest classifier is created, while adding *train()* means that the classifier is trained based on certain training features. *ConfusionMatrix()* This is used in order to calculate a cross-tabulation of both observed and predicted classes. Moreover, both these classes will have associated statistics. In GEE *ConfusionMatrix()* is used to evaluate the accuracy of a classifier. *accuracy()* is used in order to create an assessment regarding the accuracy of a classifier using *ConfusionMatrix()*. Accuracy can be obtained as training

accuracy and validation accuracy, but also as overall accuracy of a confusion matrix as a ratio between correct and total. *.kappa()* This is a statistical coefficient utilized to measure the reliability between qualitative items. It uses for example to compute the statistics for the confusion matrix. *.order()* With this function items can be rearranged as ascending or descending. *.producersAccuracy()* Obtain the probability of a certain value is a given class to be correctly classified. *.reduceRegion()* This function is helpful in order to reduce the pixel in the region or regions represented as geometry up to a statistic or a different compact representation of the pixel data in that region. The region can contain numerous pixels if the region is a polygon or it can contain only one pixel for the point geometry. *.ee.Reducer.count()* Represents an unweight reducer and it include the pixels having the centroids in the region. *ui.Chart.image.byClass()* This function represents an image band by plotting the image data aggregated by classes.

ee.Image().addBands() Additional band or bands are added to the transition image object *.setSeriesNames()* This is utilized in order to define a name for series belonging to different variables (chart, histogram etc.). *ui.Label()* represents areas displaying the text. *ui.Panel()* This is a user interface container situated at an upper-level utilized in order to arrange widgets. In order to control the arrangement of the widgets on the screen each ui.Panel has a ui.Panel.Layout. ui.Panel is not printed on the console and due to this a special panel should be created. This special panel is aimed to hold the entire list of other panels in user interface Earth Engine. *Map.add()* This button is used in order to add different features to the map, as for example title, panels of charts etc.

Export.image.toDrive() Images from Earth Engine can be exported to users' Drive account by using this *Export.image.toDrive()*. This button is used in order to add different features to the map, as for example title, panels of charts etc.

3.4 Indexes

Normalized Difference Vegetation Index (NDVI) This index represents a numerical indicator used in remote sensing measurements in order to identify surface vegetation cover on satellite imagery. In order to obtain the NDVI values the red and near infrared spectral bands obtained from Advanced Very High Resolution Radiometer (AVHRR) are used based on the surface reflectance calculations. Red and near infrared bands are the most sensitive spectral bands to vegetation. This index is useful for studies focused to analyze and monitor the health vegetation across the world. NDVI is an index that is commonly used in studies related to vegetation cover. In order to determine the density of green researchers have to identify different colors based on the wavelengths on the satellite images. These colors are based on the sunlight reflected by the plants in the visible (red) and near infrared spectral bands. Healthy vegetation will reflect more in the near infrared, while unhealthy vegetation will reflect more visible light. NDVI is calculated based on the following formula: Results

$$NDVI = (NIR - RED) \div (NIR + RED) \quad (3.1)$$

[www.usgs.gov, n.d.b]

of NDVI calculation always give values ranging from -1 to 1. While the highest values

close to 1 represent the highest density of green vegetation, values close to zero means surfaces having no vegetation.[www.usgs.gov, n.d.b]

Normalized Difference Green/Red Difference Index (NGRDI) As it name says, this index represents a ratio between the subtraction of the values of green band of the values of the red band, and the sum of those two spectral bands. In other words NGRDI use the reflectance of the green and red bands of the electromagnetic spectrum.[Liu et al., 2019] This index is usually used in order to analyze vegetation fraction and it is very sensitive to bare land [Liu et al., 2019]. NGRDI is used to analyze aerial biomass and the nutrient situation regarding the weed management in a certain crop area.

$$NDVI = (GREEN - RED) \div (GREEN + RED) \quad (3.2)$$

[www.usgs.gov, n.d.b]

Difference Vegetation Index (DVI) This is probably the most simple vegetation index and it is derived as the difference between near infrared and red bands. Accordingly, the algorithm for this vegetation index looks like this: This index is highly sensitive to

$$DVI = (NIR - RED) \quad (3.3)$$

the amount of vegetation and it is used to make a clear distinction between soil and vegetation. It is important to emphasize that this vegetation index is not influenced by the difference given by radiance and reflectance generated by the atmosphere. DVI is mostly used to analyze the vegetation ecological environment. Due to this DVI is also known as Environmental Vegetation Index (EVI) ([Xue and Su, 2017].

Normalized Difference Water Index (NDWI) There are two indices called Normalized Difference Water Index (NDWI). One of them monitors vegetation changes based on (a) water content in vegetation structure, while the other one monitors the (b) water content in water bodies using satellite images. (a) The NDWI monitoring vegetation water content is derived from satellite images using near-infrared and short wave infrared spectral bands. This is an indicator is sensitive to the water content of vegetation leaves. The near infrared spectral band reflectance is influenced by the leaf structure but it is not influenced by the water content. On the other hand, short wave infrared band reflectance is influenced by thy changes in the vegetation water content and in the structure of vegetation canopies. This vegetation index is very sensitive to the water content in vegetation structure. In order to derive NDWI the next formula is used: The obtained results that use NDWI vary

$$NDWI = (NIR - SWIR) \div (NIR + SWIR) \quad (3.4)$$

between -1 and +1, where highest values indicate high water content in the vegetation and also a high vegetation fraction cover. On the other hand, low NDWI values indicate reduce water content in vegetation and also a low vegetation fraction cover. This vegetation index is less sensitive to atmospheric conditions such as other vegetation indices, as for instance NDVI [cai Gao, 1996]. The NDWI rate will decrease after periods without precipitation. Moreover, this vegetation index may be very useful to monitor the potential fire risk for vegetation areas based on the presence of moisture in the vegetation cover.

(b) The NDWI monitoring the water content in water bodies using satellite images uses green and near infrared spectral bands. The formula calculating this index is as follows: $NDWI = \frac{GREEN - NIR}{GREEN + NIR}$. In order to underline the presence of water bodies based on satellite images, this index eliminates the soil cover and vegetation cover on Earth surface. Moreover, the NDWI index may be a helpful tool to estimate turbidity of the water bodies using satellite images [McFEETERS, 1996]. Values for the NDWI index analyzing water bodies based on remotely sensed data are categorized based on one threshold that is equal to 0.3. Values below 0.3 are attributed to non-water surfaces, while values higher or at least equal to 0.3 are considered as water bodies [McFEETERS, 1996]

Ratio Vegetation Index (RVI)

Ratio Vegetation Index (RVI) is also known as Simple Ratio (SR) due to the simplest ratio-based formula used to derive this index, namely the ratio between near infrared and red spectral bands. Accordingly, the algorithm used to derive RVI is:

$$RVI(SR) = (NIR) \div (RED) \quad (3.5)$$

RVI is founded on the fact that leaves absorb more red light and less infrared light. Accordingly, this vegetation index is mostly used in research focused on monitoring and estimation of green biomass, especially in areas with a very high density of vegetation since this index is sensitive to vegetation [Xue and Su, 2017]. The results obtained by applying this vegetation index from satellite images show high values for the vegetation, while low values are attributed to surfaces covered by soil, water, ice etc. Consequently, this index is used in order to indicate the amount of vegetation in a certain study area. By using RVI to analyze the amount of vegetation, the effects of atmosphere and topography are reduced. Topography effect is reduced because the index eliminates from the equation the irradiance, while atmospheric effects are reduced from the RVI consequences because it largely eliminates transmittance. [Xue and Su, 2017]

Enhanced Vegetation Index (EVI) EVI is an index that is very similar to NDVI and it is used with success in studies aimed to quantify vegetation greenness. This vegetation index is highly sensitive in regions having a high density of vegetation. Moreover, it corrects some of the atmospheric effects and also some canopy noise that may be in the background. Consequently, while calculating EVI the formula has also an “L” parameter to correct canopy background and “C” values representing the coefficients used for the atmospheric resistance. The spectral bands used in EVI calculation are near infrared, red and blue. The formula used to derive this index is: , where G is the gain factor. If NDVI is

$$EVI = Gx(NIR - RED) \div (NIR + C1xRED - C2xBLUE + L) \quad (3.6)$$

[www.usgs.gov, n.d.a]

sensitive to leaves chlorophyll, EVI is an index that is more sensitive to canopy structural variations. [www.usgs.gov, n.d.a]

Soil Adjust Vegetation Index (SAVI) Soil Adjust Vegetation Index (SAVI) is another surface reflectance index derived based on satellite images. This index is used in order to monitor the influence of soil brightness in those areas having a low vegetative cover. It is

$$SAVI = ((NIR - RED) \div (NIR + RED + L))x(1 + L) \quad (3.7)$$

[Huete, 1988]

calculated as the ratio between the values from red and near infrared spectral bands using also an “L” factor correction for soil brightness established at 0.5 in order to fit for most of the land cover categories. By using this factor correction the variations induced by the soil in vegetation indices are almost eliminated [Huete, 1988].

Index-Based Built-Up Index (IBI) This index aims to extract the built-up features on satellite images very fast. This is a different index since it uses thematic index-derived bands in order to generate an index instead of using original spectral bands from satellite images[Xu, 2008]. There are three thematic index-derived bands used to calculate IBI: SAVI (Soil Adjusted Vegetation Index), MNDWI (Modified Normalized Difference Water Index) and NDBI (Normalized Difference Built-Up Index). Each one of these three represents the major components of the urban environment: vegetation, water and built-up areas[Xu, 2008]. The mathematical formula to calculate IBI is the following one:

$$IBI = (NDBI - (SAVI + MNDWI) \div 2 \div (NDBI + (SAVI + MNDWI) \div 2) \quad (3.8)$$

[Xu, 2008]

Using this index is very easy to extract built-up land in urban space since the subtraction of SAVI and MNDWI index-derived bands from the NDBI band would highlight the presence of positive values only for those pixels representing built-up land. Values obtained from IBI range from -1 to +1 [Xu, 2008].

3.5 Land Surface Temperature (LST)

[Jeevalakshmi et al., 2017] define land surface temperature as the temperature measured when the ground surface is in direct contact with the measurement instrument. Land Surface Temperature (LST) is part of Earth’s energy budget. It closely relates to the distribution between latent and sensible heat flux. Various studies use LST derived from satellite images in different approaches analyzing land surface conditions and mapping various aspects regarding urban areas (extent, micro-climate, evapotranspiration, vegetation stress etc.)[Ermida et al., 2020]. LST is used in urban heat island (UHI) studies in order to quantify this phenomenon in urban areas[Estoque and Murayama, 2017]; [Peng et al., 2018]. According to [Meng et al., 2019], remote sensing is the unique approach to obtain LST data both at regional and global scales.

[Ermida et al., 2020] state that Landsat satellite images are appropriate for LST studies based on the high-resolution offered by these satellite data. Accordingly, since GEE stores a large amount of Landsat data, GEE online platform may help users to derive LST from Landsat satellite imagery. Such a code to derive LST in GEE online platform has been uploaded in GEE by [Ermida et al., 2020] in order to help users to obtain LST for numerous

study areas or to edit this code in order to meet users' requirements.

[Jeevalakshmi et al., 2017] mention that Landsat 8 provides many possibilities to the researcher to monitor and to analyze the land process with the help of remote sensing tools. [Salih et al., 2018] state that Landsat 8 is very suitable satellite data to derive LST due to the technical specifications provided by the Operational Land Imager (OLI) and the Thermal Infrared Sensor (TIRS) instruments offering resolution ranging from 15 meters to 100 meters. Accordingly, various studies have used Landsat 8 satellite imagery in order to derive LST (Rajeshwari Mani, 2014; Avdan Jovanovska, 2016; [Jeevalakshmi et al., 2017]; [Salih et al., 2018]; [Meng et al., 2019]). Based on Landsat 8 satellite images, [Avdan and Jovanovska, 2016] and [Jeevalakshmi et al., 2017] have used thermal infrared sensor (TIR) band 10 for the brightness temperature assessment and bands 4 and 5 were used to derive NDVI. The resolution for the thermal band of Landsat 8 is 100 m, while for bands 4 and 5 the resolution is 30 m. This is an important aspect in the LST validation process. If LST is compared with air temperature, then it should be noted that there is a difference between those two features [Avdan and Jovanovska, 2016] ; [Jeevalakshmi et al., 2017].

[Salih et al., 2018] have used both bands 10 and 11 of the Thermal Infrared Sensor as tools to generate an estimation of brightness temperature, while bands 4 and 5 from Operational Land Imager were utilized in order to derive NDVI parameter. In order to estimate LST parameter, from NDVI was derived land surface emissivity (LSE). LSE is the parameter expressing the emissivity of different land cover classes [Salih et al., 2018].

In urban heat studies Landsat 8 is very reliable due to thermal infrared sensor (TIRS) with those two thermal infrared channels [Meng et al., 2019]. In order to retrieve LST using Landsat 8 data many algorithms were developed: the algorithm based on one single channel ([Wang et al., 2017]; [Cristóbal et al., 2018]), the algorithm based on split-window technique (Rozenstein et al., 2014; Du et al., 2015), and the algorithm based on the separation of temperature and emissivity (Wang et al., 2015). Meng et al. (2019) has proposed an alternative approach in order to derive LST from Landsat 8 by using the NOAA JPSS Enterprise algorithm.

Implementation 4

This chapter describes the project starting with the data used. The process describes how was it was achieved by decisions taken in the development.

4.1 Data

This chapter presents how the data was used based om the 3rd chapter.

4.1.1 Landsat 8

Landsat 8 was used as data provider for this project for the time series needed for analysis. Throughout the chapter will be shown the motivation behind this selection.

Sensor	Band number	Band name	Wavelength (μm)	Resolution (m)	Band Applications
OLI	1	Coastal	0.43 - 0.45	30	Coastal and aerosol studies
OLI	2	Blue	0.45 - 0.51	30	Bathymetric mapping, distinguishing soil from vegetation, and deciduous from coniferous vegetation
OLI	3	Green	0.53 - 0.59	30	Emphasizes peak vegetation, which is useful for assessing plant vigor
OLI	4	Red	0.63 - 0.67	30	Discriminates vegetation slopes
OLI	5	NIR	0.85 - 0.88	30	Emphasizes biomass content and shorelines
OLI	6	SWIR 1	1.57 - 1.65	120	Discriminates moisture content of soil and vegetation; penetrates thin clouds
OLI	7	SWIR 2	2.11 - 2.29	30	Improved moisture content of soil and vegetation and thin cloud penetration
OLI	8	Pan	0.50 - 0.68	15	15 meter resolution, sharper image definition
OLI	9	Cirrus	1.36 - 1.38	30	Improved detection of cirrus cloud contamination
TIRS	10	TIRS 1	10.60 - 11.19	30(100)	100 meter resolution, thermal mapping and estimated soil moisture
TIRS	11	TIRS 2	11.50 - 12.51	30(100)	100 meter resolution, thermal mapping and estimated soil moisture

Table 4.1: The bands of Landsat 8. *source: eos [n.d.]*.

4.2 Google Earth Engine

This section will show how the project was implemented explaining step-by-step how the code works. The indexes calculated and algorithms used are explained in the previous chapter. Further more, the Landsat 8 collection was choosen to be a good way to determine the spatio-temporal analysis due to the fact that a result is achieved with a time series

covered by the Landsat 8 program.

4.2.1 The Google Earth Engine scripts

Google Earth Engine scripts are based [Developer'sGuide, n.d.] and the Landsat 8 image collection are enough to elaborate the project as it follows:

STUDY AREA

```
1 //Selecting the area using GEE Feature Collection
2 var studyArea = ee.FeatureCollection(studyArea);
```

The Study Area is a feature collection of coordinates for the desired area which are interconnected into a geometry. In this way, the amount of data is significantly reduced so as the processing time and resources.

FEATURE COLLECTION - TRAINING SAMPLES In order to help the machine

```

> var openHighRise: FeatureCollection (26 elements)
> var compactMidRise: FeatureCollection (19 elements)
> var openMidRise: FeatureCollection (22 elements)
> var openLowRise: FeatureCollection (45 elements)
> var largeLowRise: FeatureCollection (47 elements)
> var sparselyBuilt: FeatureCollection (16 elements)
> var heavyIndustry: FeatureCollection (28 elements)
> var denseTrees: FeatureCollection (53 elements)
> var scatteredTrees: FeatureCollection (43 elements)
> var bushScrubs: FeatureCollection (36 elements)
> var lowPlants: FeatureCollection (52 elements)
> var bareRocksPaves: FeatureCollection (26 elements)
> var soilSand: FeatureCollection (87 elements)
> var water: FeatureCollection (32 elements)

```

Figure 4.1: Training samples

learning algorithm a series of training samples are feeded. This selection, as described in the Introduction chapter, requires the knowledge of the area or Open Street Map for determine the "golden" pixels that describes most accurate the value of the class. In this way a selection in range of 20 to 70 pixels was performed for each individual class.

Get Data

```
1 var GetData = require("users/bogdanmocanu99/TestFinal:importData");
2 var getIndexes = require("users/bogdanmocanu99/TestFinal:Indexes");
```

For this part of the script, the composite and indexes scripts were loaded from a different file of the same repository called "ImportData" and "Indexes". This approach was desired for a cleaner script so that will be easier to visualize and understand by the reader.

Google Earth Engine assets

```

1 var trainData = openHighRise.merge(compactMidRise).merge(openMidRise).merge(openLowRise).
  merge(largeLowRise).merge(sparselyBuilt).merge(heavyIndustry).merge(denseTrees).merge(
    scatteredTrees).merge(bushScrubs).merge(lowPlants).merge(bareRocksPaves).merge(
    soilSand).merge(water);
2 var train = ee.FeatureCollection(trainData);

```

In this section it is shown the training data for each class which is a feature collection and will be loaded into the script whenever it will be called.

Importdata

```

1 exports.importData = function(studyArea,startDate,endDate, cloudFilter){
2
3
4 //Set up bands and corresponding band names
5 var inBands = ee.List([1,2,3,4,5,7,6,8])
6 var outBands = ee.List(['blue','green','red','nir','swirl','temp', 'swir2','panchromatic
  '])
7
8 var multImageDict = {
9   'SR': ee.Image
10     ([0.0001,0.0001,0.0001,0.0001,0.1,0.0001,0.0001,0.0001,0.0001,0.0001])
11 };
12 // Pan-Sharpening
13 var panSharpenL8 = function(image) {
14   var rgb = image.select('red', 'blue', 'green');
15   var pan = image.select('panchromatic');
16   // Convert to HSV, swap in the pan band, and convert back to RGB.
17   var huesat = rgb.rgbToHsv().select('hue', 'saturation');
18   var upres = ee.Image.cat(huesat, pan).hsvToRgb();
19   return image.addBands(upres);
20 };
21
22 // Get Landsat data
23
24 var l8s = ee.ImageCollection("LANDSAT/LC08/C01/T1_TOA")
25   .filterDate(startDate,endDate)
26   .filterBounds(studyArea)
27   .select(inBands,outBands)
28   .map(panSharpenL8)
29   .filter(ee.Filter.lt("CLOUD_COVER",10))
30
31
32 l8s = l8s.map(function(img){
33   return img.multiply(multImageDict["SR"]).copyProperties(img,['system:time_start','
    system:footprint']).copyProperties(img);}
34   return l8s;
35
36 };

```

In the ImportData file of the same repository, the separate script is in charge for creating the composite based on the bands selection as it is shown in the table [eos, n.d.] not all the 11 bands were relevant for use and getting an 30m resolution for the composite. An important part of the script is to the Pan-Sharpening which allows to make a visual

correction achieving a better resolution respectively a 15m based on the Panchromatic band 8 which has a 15meter resolution. Furthermore the Cloud Cover Filter it is set to 10 which will result a collection with a low coverage of clouds. In this way the canvas will load a good image collection ready for further analysis.

Date

```

1  /// Date
2  var startyear = 2019;
3  var endyear = 2019;
4
5  // Date object
6  var startDate = ee.Date.fromYMD(startyear,4,1);
7  var endDate = ee.Date.fromYMD(endyear,9,30);

```

This part of the script chooses the time of the year where the collection is loaded on the canvas. To be mentioned that Landsat8 will provide a picture for every 16 day interval, but in this way the composite with the best results will be loaded on canvas.

Get images

```

1  print("getting images");
2  var study = GetData.importData(studyArea, startDate, endDate, 10);
3  print("found ",study.size(),"images");
4  var study = study.median().clip(studyArea);
5  print(study);
6
7  //Add True color Image mosaic
8  var image = ee.ImageCollection("LANDSAT/LC08/C01/T1_SR")
9    .filterDate('2019-07-01', '2019-09-30')
10   .filterBounds(studyArea)
11   .filter(ee.Filter.lt("CLOUD_COVER",10))
12   .mosaic()
13  Map.addLayer(image.clip(studyArea), {bands: ['B4', 'B3', 'B2'],min:0, max: 3000}, 'True
    colour image');

```

This part of the script releases creates a median for each pixel from the entire collection in this way a better result based on the filters applied is loaded then the raw data collection.

Indexes Used

```

1  /exports.getIndexes = function(image) {
2
3    // Normalized difference vegetation index (NDVI)
4    var ndvi = image.normalizedDifference(['nir','red']).rename("ndvi");
5    image = image.addBands(ndvi);
6
7    // Red Edge Normalized difference vegetation index (NDVI)
8    var rendvi = image.normalizedDifference(['nir','red']).rename("rendvi");
9    image = image.addBands(rendvi);
10
11   // // Normalized Difference Green/Red difference index (NGRDI)
12   var ngrdi = image.normalizedDifference(['green','red']).rename('ngrdi');
13   image = image.addBands(ngrdi);
14
15   // Difference Vegetation Index(DVI)
16   var dvi = image.select('nir').subtract(image.select('red')).rename('dvi');

```

```

17 image = image.addBands(dvi);
18
19 // Normalized Difference Water Index (NDWI)
20 var ndwi = image.normalizedDifference(['nir', 'swir1']).rename("ndwi");
21 image = image.addBands(ndwi);
22
23 // Ratio Vegetation Index (RERI)
24 var rvi = image.select('nir').divide(image.select('red')).rename('rvi');
25 image = image.addBands(rvi);
26
27 // Red Edge Ratio Vegetation Index (RERI)
28 var rervi = image.select('nir').divide(image.select('red')).rename('rervi');
29 image = image.addBands(rervi);
30
31 // add Enhanced Vegetation Indexes
32 var evi = image.expression('2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {
33   'NIR' : image.select('nir'),
34   'RED' : image.select('red'),
35   'BLUE': image.select('blue') }).float();
36 image = image.addBands(evi.rename('evi'));
37
38 // Add Soil Adjust Vegetation Index (SAVI)
39 var savi = image.expression('(NIR - RED) * (1 + 0.5) / (NIR + RED + 0.5)', {
40   'NIR': image.select('nir'),
41   'RED': image.select('red')}).float();
42 image = image.addBands(savi.rename('savi'));
43
44 // Add Index-Based Built-Up Index (IBI)
45 var ibiA = image.expression('2 * SWIR1 / (SWIR1 + NIR)', {
46   'SWIR1': image.select('swir1'),
47   'NIR' : image.select('nir')}).rename(['IBI_A']);
48
49 var ibiB = image.expression('(NIR / (NIR + RED)) + (GREEN / (GREEN + SWIR1))', {
50   'NIR' : image.select('nir'),
51   'RED' : image.select('red'),
52   'GREEN': image.select('green'),
53   'SWIR1': image.select('swir1')}).rename(['IBI_B']);
54
55 var ibiAB = ibiA.addBands(ibiB);
56 var ibi = ibiAB.normalizedDifference(['IBI_A', 'IBI_B']);
57 image = image.addBands(ibi.rename('ibi'));
58 return(image);
59
60 }
61
62 exports.getEVI = function(image) {
63   print(image)
64   // add Enhanced Vegetation Indexes
65   var evi = image.expression('2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {
66     'NIR' : image.select('nir'),
67     'RED' : image.select('red'),
68     'BLUE': image.select('blue') }).float();
69   image = image.addBands(evi.rename('evi'));
70
71   return(image);
72 }

```

The indexes used for this collection make sense due to the considerably great number of classes which involves different types of vegetation together with classes that correlates the built-up areas mixed with vegetation, as well as shallow waters with surface vegetation that needed to be differentiated. For each index and it's best use chapter 4 describes them on a wider point of view.

Get indexes

```
1 // get Indexes
2 print("getting indexes");
3 study = getIndexes.getIndexes(study);
4 print(study);
```

This part of the script applies the indexes from the separate file "Indexes" into the image collection on the canvas.

Get bands

```
1 // get bands
2 var bands = study.bandNames()
3 print(bands)
```

Training samples

```
1 // integers starting from zero in the training data.
2 var label = 'lcode'
3 // Overlay the points on the imagery to get training.
4 var trainings = study.select(bands).sampleRegions({
5   collection: train,
6   properties: [label],
7   scale: 30
8 });
9 var sample = trainings.randomColumn();
10 var split = 0.7; // Roughly 70% training, 30% testing.
11 var training = sample.filter(ee.Filter.lt('random', split));
12 print(training.size());
```

This part of the code is the supervised classification is performed based on the training samples gathered in the previous step of the script. Using it the classifier will be trained likewise the data will be split into random column with 70percent for training and 30percent for validation.

Random Forest

```
1 // Random forest
2 var classifier = (ee.Classifier.smileRandomForest(15)
3   .train({
4     features: training,
5     classProperty: label,
6     inputProperties: bands
7   }));
8 var classified = study.classify(classifier);
9 print(classified);
```

This part of the code will classify the pixels according to the classifier trained on the previous step.

Confusion Matrix

```

1 // Get a confusion matrix representing resubstitution accuracy.
2 var trainAccuracy = classifier.confusionMatrix();
3 print('Resubstitution error matrix: ', trainAccuracy);
4 print('Training overall accuracy: ', trainAccuracy.accuracy());
5
6 var validation = sample.filter(ee.Filter.gte('random', split));
7
8 // Classify the validation data.
9 var validated = validation.classify(classifier);
10
11 // Get a confusion matrix representing expected accuracy.
12 var testAccuracy = validated.errorMatrix(label, 'classification');
13
14 var OA = testAccuracy.accuracy();
15 var CA = testAccuracy.consumersAccuracy();
16 var Kappa = testAccuracy.kappa();
17 var Order = testAccuracy.order();
18 var PA = testAccuracy.producersAccuracy();
19
20 print(testAccuracy, 'Confusion Matrix');
21 print(OA, 'Overall Accuracy');
22 print(CA, 'Consumers Accuracy');
23 print(Kappa, 'Kappa');
24 print(Order, 'Order');
25 print(PA, 'Producers Accuracy');

```

This step of the code will generate a matrix that will afterwards show the performance of the classifier model built up in the previous steps

Pixel Classes

```

1 var stats = classified.reduceRegion({
2   reducer: ee.Reducer.count(),
3   geometry: studyArea,
4   scale: 90
5 });
6 print(stats);
7
8 var options = {
9   lineWidth: 1,
10  pointSize: 2,
11  hAxis: {title: 'Classes'},
12  vAxis: {title: 'Num of pixels'},
13  title: 'Number of pixels in each class.',
14  colors: ['f79605', 'e68a02', 'f73302', 'fad102', 'bfa522', 'bdc22f', 'edede4', '047d14', '06c71f', '9ef7ce', '12a182', '7e8f8b', 'f2ffba', '0278cc'],
15 };
16 var pixelChart = ui.Chart.image.byClass({
17   image: ee.Image(1).addBands(classified),
18   classBand: 'classification',
19   region: studyArea,
20   scale: 90,
21   reducer: ee.Reducer.count()
22 }).setSeriesNames(['openHighRise', 'compactMidRise', 'openMidRise', 'openLowRise', 'largeLowRise', 'sparselyBuilt', 'heavyIndustry', 'denseTrees', 'scatteredTrees', 'bushScrubs', 'lowPlants', 'bareRocksPaves', 'soilSand', 'water']).setOptions(options);

```



```

23
24 print(pixelChart)
25
26 Map.addLayer(classified,{ 'min': 1, 'max': 14, palette: ['f79605','e68a02','f73302','fad102',
    'bfa522','bdc22f','edede4','047d14','06c71f','9ef7ce','12a182','7e8f8b','f2ffba','
    0278cc','0278cc']},
27     'classification');

```

For this part of the script, the pixels classified are distributed on every class. This step will help a better visualisation of the statistical data results, which will be available in the next chapter.

Map add legend

```

1  var names = ['openHighRise','compactMidRise','openMidRise','openLowRise','largeLowRise','
    sparselyBuilt','heavyIndustry','denseTrees','scatteredTrees','bushScrubs','lowPlants','
    bareRocksPaves','soilSand','water' ];
2
3  var values = [ '1', '2', '3',];
4
5  var legendsPalette = ['#f79605','#e68a02','#f73302','#fad102','#bfa522','#bdc22f','#edede4',
    '#047d14','#06c71f','#9ef7ce','#12a182','#7e8f8b','#f2ffba','#0278cc'];
6
7  // set position of panel
8  var legend = ui.Panel({style: { position: 'bottom-right', padding: '8px 15px'}});
9
10 // Create legend title
11 var legendTitle = ui.Label({value: 'Land Cover Legends ',style: {
12     fontWeight: 'bold', fontSize: '18px', margin: '0 0 4px 0', padding: '0' }});
13
14 // Add the title to the panel
15 legend.add(legendTitle);
16
17 var makeRow = function(color, name) {
18     // Create the label that is actually the colored box.
19     var colorBox = ui.Label({
20         style: {
21             backgroundColor: '#' + color, padding: '8px',margin: '0 0 4px 0' });
22
23     // Create the label filled with the description text.
24     var description = ui.Label({
25         value: name, style: {margin: '0 0 4px 6px'}});
26
27     // return the panel
28     return ui.Panel({
29         widgets: [colorBox, description],layout: ui.Panel.Layout.Flow('horizontal')}});
30
31 // Add color and and names
32 for (var i = 0; i < 14; i++) {
33     legend.add(makeRow(legendsPalette[i], names[i]));
34 }
35
36 // Add the legend to the map.
37 Map.add(legend);

```

Adding a legend for the whole canvas will give the better spatio-temporal understanding

of the data, in this way a ready made map can be downloaded to hard-drive

4.3 Land Surface Temperature

Land Surface Temperature is calculated based on NDVI values on the study area

```

1 //cloud mask
2 function maskL8sr(image) {
3   // Bits 3 and 5 are cloud shadow and cloud, respectively.
4   var cloudShadowBitMask = (1 << 3);
5   var cloudsBitMask = (1 << 5);
6   // Get the pixel QA band.
7   var qa = image.select('pixel_qa');
8   // Both flags should be set to zero, indicating clear conditions.
9   var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
10              .and(qa.bitwiseAnd(cloudsBitMask).eq(0));
11   return image.updateMask(mask);
12 }
13
14 //vis params
15 var vizParams = {
16   bands: ['B5', 'B6', 'B4'],
17   min: 0,
18   max: 4000,
19   gamma: [1, 0.9, 1.1]
20 };
21
22 var vizParams2 = {
23   bands: ['B4', 'B3', 'B2'],
24   min: 0,
25   max: 3000,
26   gamma: 1.4,
27 };

```

Choosing image collection for the graphical exposure of the data

```

1 var image = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
2   .filterBounds(geometry)
3   .filterDate('2014-06-01', '2019-06-30')
4   .sort('CLOUD_COVER')
5   .first();
6 print(image)
7 var ndvi = image.normalizedDifference(['B5',
8   'B4']).rename('NDVI');
9 var ndviParams = {min: -1, max: 1, palette: ['blue', 'white',
10   'green']};
11 print(ndvi, 'ndvi');
12 Map.addLayer(ndvi, ndviParams, 'ndvi');

```

Selecting thermal band nr.10

```

1 //select thermal band 10(with brightness temperature), no calculation
2 var thermal= image.select('B10').multiply(0.1);
3 var b10Params = {min: 291.918, max: 302.382, palette: ['blue',
4   'white', 'green']};
5 Map.addLayer(thermal, b10Params, 'thermal');

```

Find min and max NDVI

```

1 // find the min and max of NDVI
2
3 var min = ee.Number(ndvi.reduceRegion({
4   reducer: ee.Reducer.min(),
5   geometry: geometry,
6   scale: 30,
7   maxPixels: 1e9
8 }).values().get(0));
9 print(min, 'min');
10 var max = ee.Number(ndvi.reduceRegion({
11   reducer: ee.Reducer.max(),
12   geometry: geometry,
13   scale: 30,
14   maxPixels: 1e9
15 }).values().get(0));
16 print(max, 'max')

```

Fractional vegetation

```

1 //fractional vegetation
2 {
3   var fv =(ndvi.subtract(min).divide(max.subtract(min))).pow(ee.Number(2)).rename('FV');
4   print(fv, 'fv');
5   Map.addLayer(fv);
6 }

```

Emissivity

```

1 //Emissivity
2 var a= ee.Number(0.004);
3 var b= ee.Number(0.986);
4 var EM=fv.multiply(a).add(b).rename('EMM');
5 var imageVisParam3 = {min: 0.9865619146722164, max:0.989699971371314};
6 Map.addLayer(EM, imageVisParam3,'EMM');

```

LST

```

1 //LST in Celsius Degree bring -273.15
2 //NB: In Kelvin don't bring -273.15
3 var LST = thermal.expression(
4   '(Tb/(1 + (0.00115* (Tb / 1.438))*log(Ep)))-273.15', {
5     'Tb': thermal.select('B10'),
6     'Ep': EM.select('EMM')
7   }).rename('LST');
8 Map.addLayer(LST, {min: 20.569706944223423, max:29.328077233404645, palette: [
9   '040274', '040281', '0502a3', '0502b8', '0502ce', '0502e6',
10  '0602ff', '235cb1', '307ef3', '269db1', '30c8e2', '32d3ef',
11  '3be285', '3ff38f', '86e26f', '3ae237', 'b5e22e', 'd6e21f',
12  'fff705', 'ffd611', 'ffb613', 'ff8b13', 'ff6e08', 'ff500d',
13  'ff0000', 'de0101', 'c21301', 'a71001', '911003'
14  ]}, 'LST');

```

Graph

```

1 //Graph
2 var regions = ee.FeatureCollection([

```

```
3 ee.Geometry(geometry), {label: 'Copenhagen'})),  
4  
5 // Load Landsat 8 brightness temperature data for 1 year.  
6 var temps2013 = ee.ImageCollection('LANDSAT/LC08/C01/T1_32DAY_TOA')  
7   .filterDate('2014-12-25', '2019-12-25')  
8   .select('B11');
```

Create time-series Chart

```
1 // Create a time series chart.  
2 var tempTimeSeries = ui.Chart.image.seriesByRegion(  
3   temps2013, regions, ee.Reducer.mean(), 'B11', 200, 'system:time_start', 'label')  
4   .setChartType('ScatterChart')  
5  
6   .setOptions({  
7     title: 'Temperature over time in Copenhagen',  
8     vAxis: {title: 'Temperature (Kelvin)'},  
9  
10    lineWidth: 1,  
11    pointSize: 4,  
12  
13  });  
14  
15 // Display.  
16 print(tempTimeSeries);
```

Results 5

This chapter will present the results achieved through the implementation, but further discussion on the accuracy and whether the results answer the initial problem of the project, will not be discussed here but in chapter 6.

5.1 Maps

This section will present the maps achieved during each step of the analysis.

5.1.1 Local Climate Zones

The following map shows the result achieved by pixel based image classification for the year 2015 which is the beginning of the analysis. One thing to be stated is that red, yellow, orange represents the built-up areas which includes different types of vegetation at different levels of development.

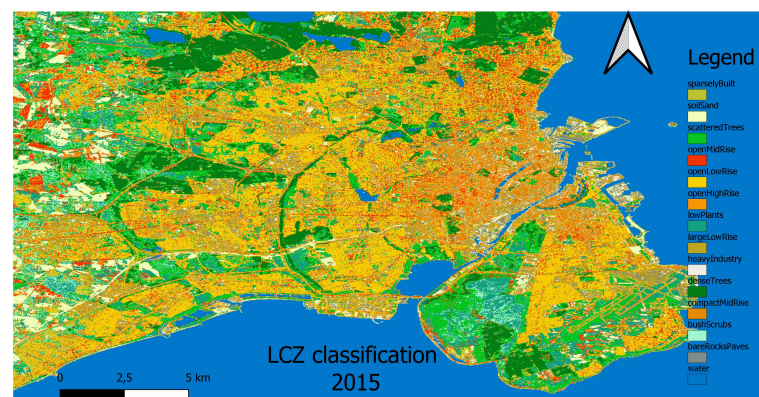


Figure 5.1: Map of Local Climate Zones 2015

As a comparison, here is the map resulted for the last time series analysis -2019-. Just by visual inspection is not enough to detect the land changes during the period. For this two greater zones it is though slightly difficult to see a change in the land use. For better understanding of this changes, charts were generated. The purpose of the great area

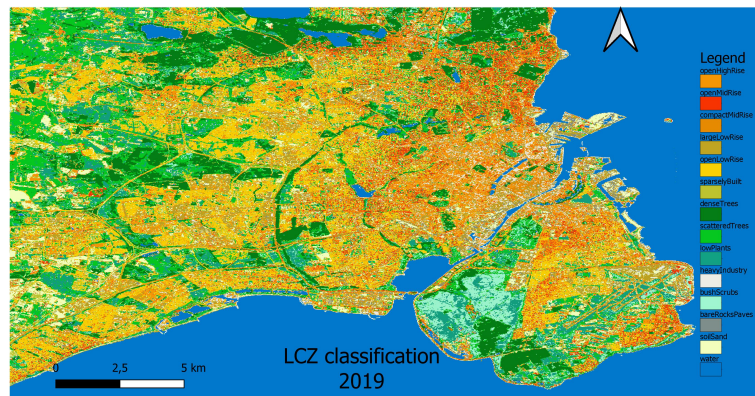


Figure 5.2: Map of Local Climate Zones 2019

taken into consideration at the beginning was to show the capability of the project for producing result for a larger area, which proves the manoeuvrability of the algorithm. At a closer look, the newly built-up areas by land reclamation Nordhavn is in focus were even visual inspection reveals the differences from year 2015 to year 2019. The considerable greater region is visible.

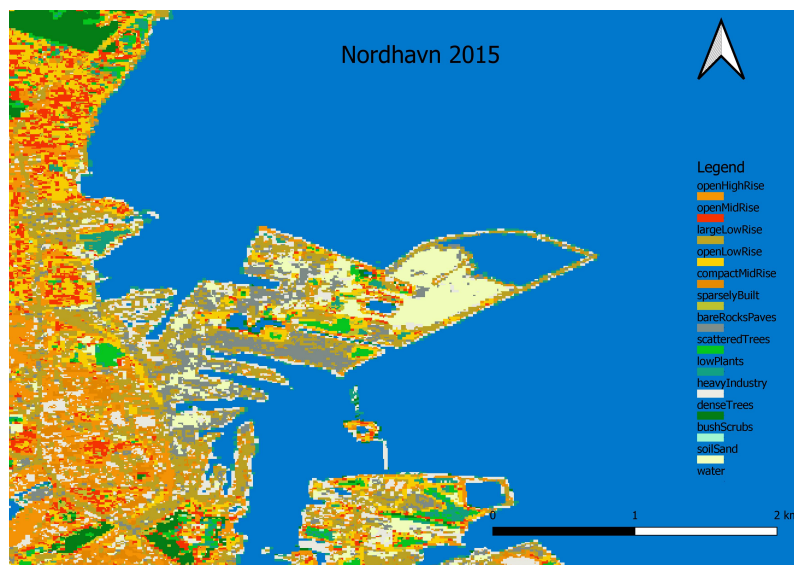


Figure 5.3: Nordhavn 2015

By comparing the 2 time stamps there is an obvious land use change therefor it can be stated that a change has been detected for the water pixels. Further with the analysis will show, using the same manner, how this change affect the temperature.

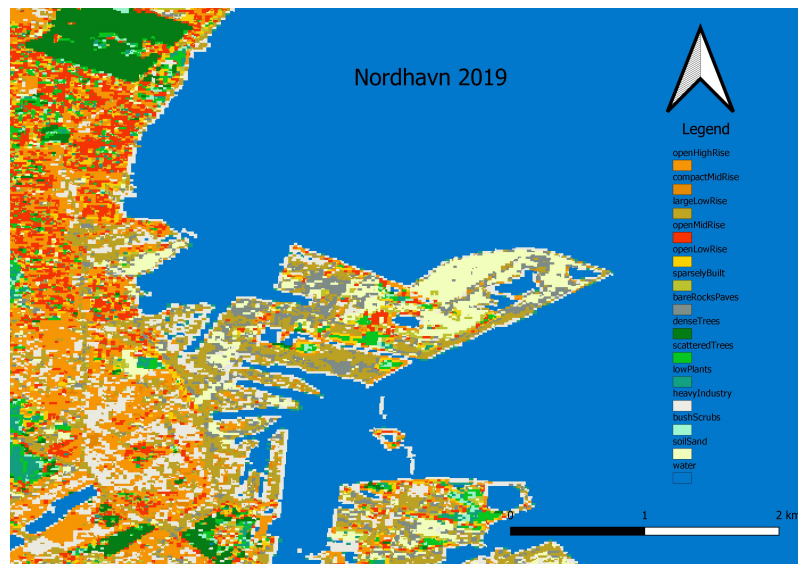


Figure 5.4: Nordhavn 2019

5.1.2 Emissivity

Further in the analysis process the capability of a surface to emit thermal radiation was considered. In this case maps have been generated, trying to prove that extensive built-up areas come up with a change for the local climate. In the same manner with the previous maps, the larger area was taken into consideration with the purpose of showing that the same procedure can be followed for extensive research. Nordhavn area has a considerable

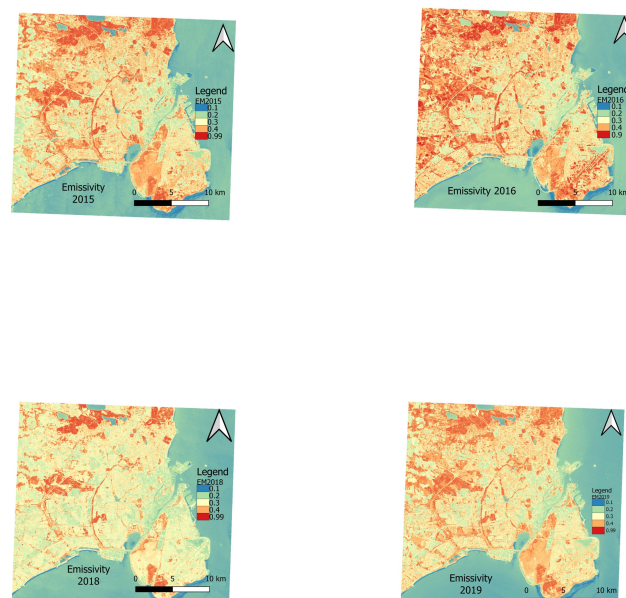


Figure 5.5: Emissivity map

land use change detected for the timestamps selected for the analysis. Further with the analysis, the emissivity coefficient has been transformed into temperature.

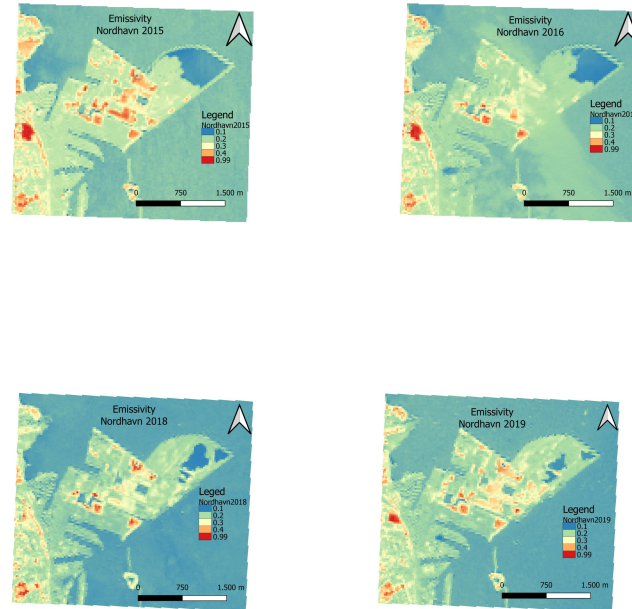


Figure 5.6: Emissivity Nordhavn Area

5.1.3 Land Surface Temperature

For continuing with the analysis the Land Surface Temperature was derived based on surface emissivity. In the same way maps were represented for the greater area for different time stamps, but also for deeper analysis the focus maps were generated for Nordhavn area. The Celsius values calculated are min: 20.569706944223423, max: 29.328077233404645.



Figure 5.7: Land Surface Temperature

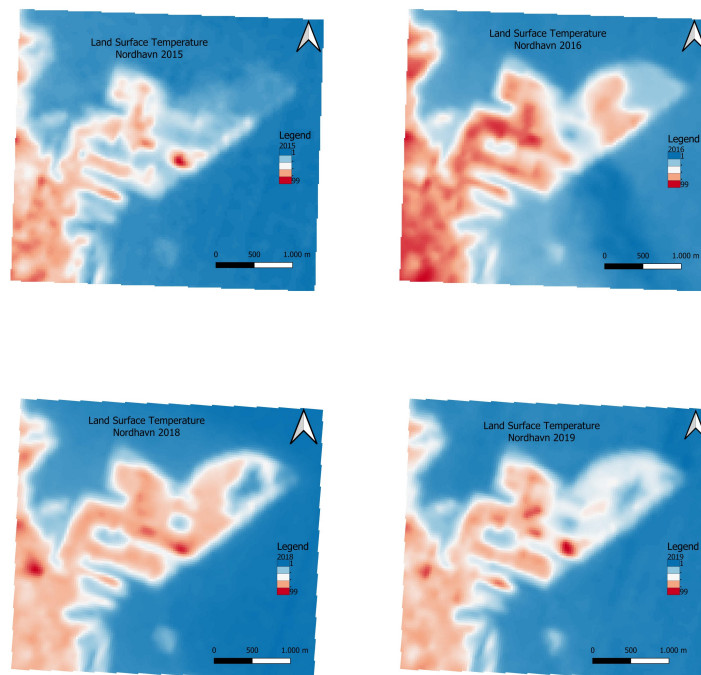


Figure 5.8: LST Nordhavn Area

5.2 Accuracy2019 and 2015

Inspector	Console	Tasks
	Resubstitution error matrix:	JSON
	► List (14 elements)	JSON
	Training overall accuracy:	JSON
	0.978319783197832	
	► List (14 elements)	JSON
	Confusion Matrix	JSON
	0.7484662576687117	
	Overall Accuracy	JSON
	► List (1 element)	JSON
	Consumers Accuracy	JSON
	0.7218661561511571	
	Kappa	JSON
	► [0,1,2,3,4,5,6,7,8,9,10,11,12,13]	JSON
	Order	JSON
	► List (14 elements)	JSON
	Producers Accuracy	JSON

Figure 5.9: Results 2019

At a robust level the Kappa coefficient achieved for 2019 is 0.7218661561511571 Kappa which according to [McHugh, 2012] it is considered to be *substantial*. Training overall accuracy got the value of 0.978319783197832. An overall accuracy has been reached at 0.7484662576687117 .

Inspector	Console	Tasks
	► Classifier.train	JSON
	Resubstitution error matrix:	JSON
	► List (14 elements)	JSON
	Training overall accuracy:	JSON
	0.9763157894736842	
	► List (14 elements)	JSON
	Confusion Matrix	JSON
	0.75	
	Overall Accuracy	JSON
	► List (1 element)	JSON
	Consumers Accuracy	JSON
	0.7248082328838915	
	Kappa	JSON
	► [0,1,2,3,4,5,6,7,8,9,10,11,12,13]	JSON
	Order	JSON
	► List (14 elements)	JSON
	Producers Accuracy	JSON
	► Object (1 property)	JSON

Figure 5.10: Results 2015

While for 2015 Kappa Coefficient is 0.7248082328838915 which it is considered to be *substantial*. Training overall accuracy got a value of 0.9763157894736842 and Overall Accuracy is 0.75.

5.3 Graphs

Related to the graphs results, there is a slightly dynamic of the built-up areas (open midrise , open lowrise). By analysing this charts, a direct correlation can't be distinguished between these classes moreover the class openLowRise shows a decrease but for all other built-up areas there is a growth which show that the built-ul area suffers a change and the built-up area becomes more compact while the denseTrees class doesn't record any change. Considering this situation, the dynamic of the green areas suffers with the decrease of Open Low Rise class and the increase of Open High Rise class.

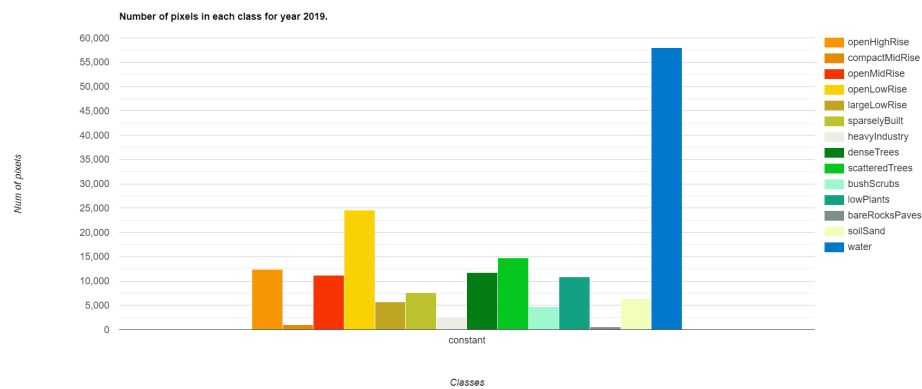


Figure 5.11: Results pixels 2019

On the other hand, checking a specific built-up area like Nordhavn points out and increase LST for the time-stamps analysed. From this focus, the research question can be answered stating that an

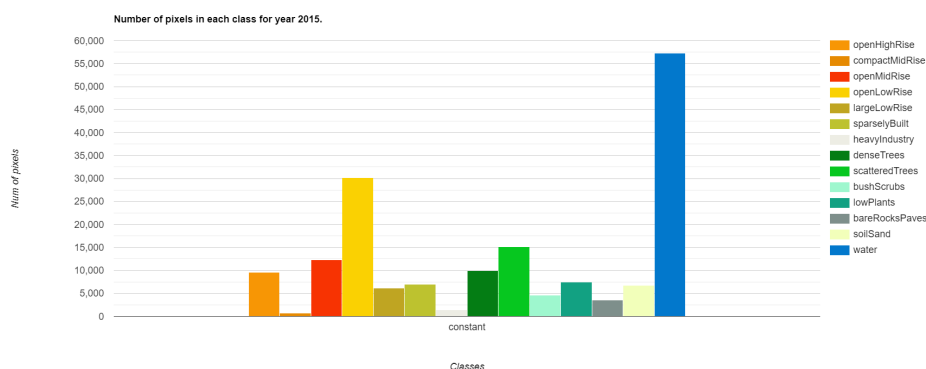


Figure 5.12: Results pixels 2015

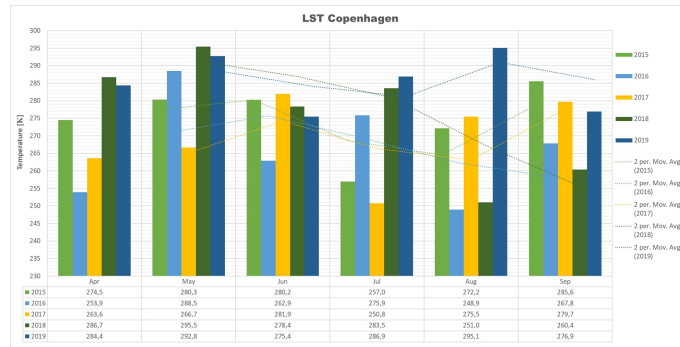


Figure 5.13: Copenhagen graph kelvin

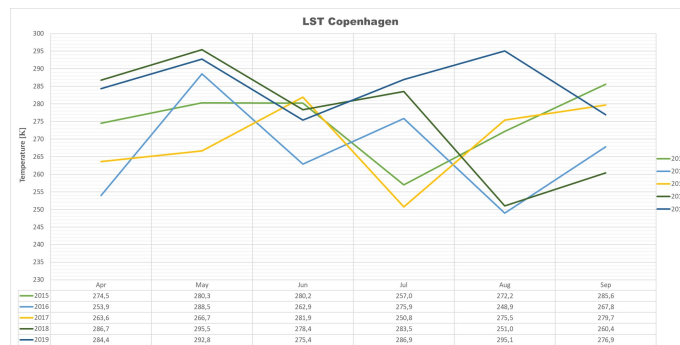


Figure 5.14: Copenhagen line graph kelvin

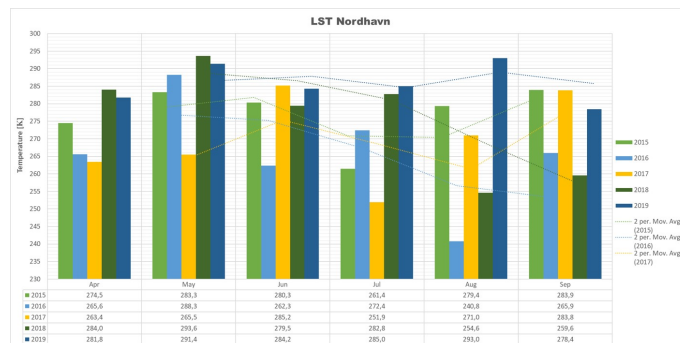


Figure 5.15: Copenhagen graph kelvin

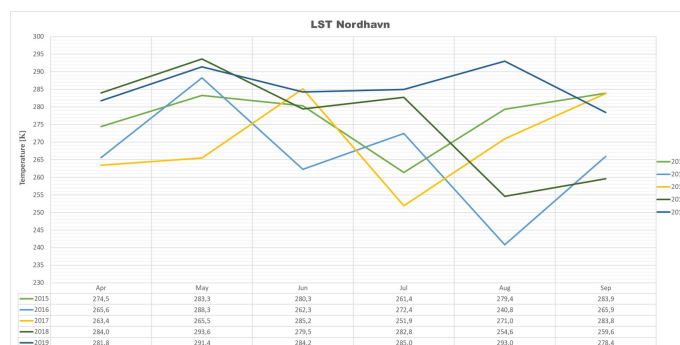


Figure 5.16: Nordhavn graph kelvin

Discussion 6

In this chapter will be presented the answers of the research questions from the Introduction Chapter.

6.1 Land Use Land Change and sources of error

This project based on LCZ methodology, didn't study in a full scale as for the class of OPEN HIGH RISE it was taken into consideration the "on-site" documentation like [Estacio et al., 2019] stated. In this case the image analysis was fully pixel based, as zonal knowledge was take into consideration. This approach was chosen due to the fact that Heat Island Effect was not analysed on a full scale and just as identification part with the effects studied from the literature. In this case and issue occurred due to the fact that based on the training samples, only the color of the pixels were take into consideration. It could have been a solution for identifying specific classes only if it was the same materials used in the process of developing the built-up area corresponding tot the class assigned. Moreover for trying to implementing such a solution it could have been possible at a smaller scale with a better resolution than 15m after pan-sharpening algorithm. As a conclusion, pixel based classification would have been more appropriate for crop classification and not for built-up areas classification, where Object Based Image Analysis method would have been the best choice. The project can be improved by considering this part of the analysis in the beginning where afterwards overlaid should be a segmentation for each class done on a Digital Surface Model and analyse if the classes correspond to the hot-spots mapped by the emissivity analysis.

The whole project can be extended for any region of interest just by adding or changing the desired geometry and setting the center-point of the map. The main issue for the results achieved with an accuracy of roughly 0.69 is that Digital Surface Model wasn't introduced in the project. The whole project tried to analyse the capability of creating a Local Climate Zone map on a pixel-based method which in this case led to a result that cannot be entirely used as a base for further analysis.

6.2 Temperature and sources of error

Land surface temperature was estimated based on the surface emissivity and charts were generated as shown in sub-chapter 5.3 and it is pointing out an increase in top temperature from year 2015 to year 2019. The calculation indicated an increase in the

temperature. Though this increase is for now irrelevant due to the fact that corrections haven't been performed. According to [Jiménez-Muñoz and Sobrino, 2006] the first source of error in calculating LST is **Land Surface Emissivity** where errors up to 10percent may occur. Moreover **effects of aerosols and other gaseous absorbers** as stated in [Jiménez-Muñoz and Sobrino, 2006] there are great differences between the transmissivity spectrum and a default visibility. For instance the main absorbers are the water vapors and the second main absorbers are carbon dioxide and methane **Angular effects** are some other types of errors in LST calculation due to the fact that nadir view angle isn't very precise with field of views over 50degree, while there is a difference registered between nadir and different angles view. Other errors that should have been taken into consideration in order to calibrate the LST for more precise acquisition **Wavelength uncertainty** and **Bandpass and Full-Width Half-Maximum(FWHM) effect**, while **Atmospheric correction** haven't been performed and neither **Noise equivalent delta error** was taken into consideration.

Considering the surface emissivity of built-up areas the connection is that water surface which has a low emissivity has been reclaimed and transformed into built-up area, which in this case increases the emissivity of the area, though a detailed research of local weather stations for the temperature records like [Mijani et al., 2020] it is required for further analysis.

6.3 Nordhavn urbanisation impact

As it was expected and deducted from the literature, changing the land use land cover with built up areas triggers an increase of LST and a possibility of evolving new heat islands. Nordhavn is tended to be a region where no heat islands were identified 5 years ago but can be considered a growing factor for the climate change. On one hand in this situation for the latest satellite imagery a change in LST pattern is distinguished but on the other hand a deeper analysis is necessary to reach out for the results of the impact on human and climate factors.

Overall a pixel based image analysis can be used as the starting point of the project, but for better results a segmentation of the classes based on a digital elevation model is required together with a calibrated LST based on the sources of error, corroborated with local temperatures registered by local stations.

Conclusion 7

This chapter will conclude the 2 research questions, and through their completion the problem statement will be answered. The questions will be presented, as they are concluded on

The first research question to be concluded on is: **"How Local Climate Zones evolved during the last five years in Nordhavn?"**

First of all it was found that based on LCZ scheme that water class area was diminished and soil class has increased, thus a more specific classification can be achieved with golden training sets and Random Forest Classifier, though a pixel-based approach is not suitable for studying LCZ and especially the built-up classes that involves the height of the buildings. This approach was proven to be inappropriate for the classes *compact mid-rise* and *open high-rise* classes that were most likely mixt up, creating a *salt and pepper* classification result. Nonetheless object based classification is far more suitable for this type of analysis, while a pixel-based approach should be maintained only for a starting point. In this case the project needs for a deeper analysis a Digital Surface Model analysis for extraction of the classes where height is considered a factor.

The second research question

How can LCZ methodology help spotting the micro-climate differences in Copenhagen's newly urbanised areas? For this question, the fact that *soil* class has been extended by reclaiming water areas, one obvious factor has been easily identified, but the inaccuracy is in the classes that have height as a factor. For this issue, of not using the digital elevation model, the Land Surface Temperature was mapped and calculated, pointing out the heat-zones. From the literature research of the heat-islands effect, it was pointed out that different LCZ classes have a distinctive emissivity. The main issue in this study is that the LST wasn't calibrated according to the sources of error, where from the inaccurate results in the temperature chart. Moreover the whole temperature part of the study needs to include local temperature records for validation.

In conclusion can be improved on one hand for deeper analysis, but on the other hand it can be extended for larger areas, where at least for the vegetation classes the LCZ scheme can be considered, while for the built-up classes the digital surface model is mandatory for extraction, all together with land surface temperature calibration and temperature records validation.

Reflections 8

8.1 Reproducibility

This project is reproducible due to the methods, algorithms and theories used, are at a general level and not specifically to Greater Copenhagen Area or just Nordhavn. Moreover the Google satellite image collections are freely available and easy to access to update or extend the time-series.

The data used is for this project is Open Source Data so it can be available for anyone that has the repository link that is listed in Appendix, to develop or integrate this project for further analysis.

Bibliography

- , **Mar 15 2019**. *Urban Research - Urban Systems; Researchers from Texas State University Describe Findings in Urban Systems (Application of Airborne Remote Sensing Data On Mapping Local Climate Zones: Cases of Three Metropolitan Areas of Texas, U.s)*, 2019. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/2190633701?accountid=8144>. Copyright - Copyright 2019, NewsRx LLC; Last updated - 2019-03-13.
- , **Aug 31 2015**. *Climate Research; New Climate Research Findings Has Been Reported by Investigators at University of Toronto (Mapping Local Climate Zones for a Worldwide Database of the Form and Function of Cities)*, 2015. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/1707615725?accountid=8144>. Name - Climate Research; University of Toronto; Copyright - Copyright 2015, NewsRx LLC; Last updated - 2015-08-29.
- , **Nov 30 2018**. *Information Technology; Studies from University of North Carolina Further Understanding of Information Technology (WUDAPT An Urban Weather, Climate, and Environmental Modeling Infrastructure for the Anthropocene)*, 2018. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/2136834489?accountid=8144>. Name - University of North Carolina; American Meteorological Society; Copyright - Copyright 2018, NewsRx LLC; Last updated - 2020-03-25.
- , **Jan 2018**. *Micro-Scale Variability of Air Temperature within a Local Climate Zone in Berlin, Germany, during Summer*. *Climate*, 6(1), 5. ISSN 2225-1154. doi: 10.3390/cli6010005. [ONLINE] Available at: <http://dx.doi.org/10.3390/cli6010005>.
- Alexander et al., 2015**. Paul John Alexander, Gerald Mills and Rowan Fealy. *Using LCZ data to run an urban energy balance model*. *Urban Climate*, 13, 14 – 37. ISSN 2212-0955. doi: <https://doi.org/10.1016/j.uclim.2015.05.001>. [ONLINE] Available at: <http://www.sciencedirect.com/science/article/pii/S2212095515000152>.
- Avdan and Jovanovska, 2016**. Ugur Avdan and Gordana Jovanovska. *Algorithm for Automated Mapping of Land Surface Temperature Using LANDSAT 8 Satellite Data*. *Journal of Sensors*, 2016, 1–8. doi: 10.1155/2016/1480307. [ONLINE] Available at: <https://doi.org/10.1155/2016/1480307>.
- Bakker et al., 2001**. Wim H. Bakker, Lucas L. F. Janssen, Colin V. REeves, Ben G. H. Gorte, Christine Pohl, Michael J. C. Weir, John A. Horn, Anupma Prakash and Tsehaie Woldai. *Principles of Remote Sensing*. <http://www.gdmc.nl/oosterom/PoRSHyperlinked.pdf>. Accessed: 28-05-2019.

- Bechtel et al., Jun 2016a.** B. Bechtel, M. Pesaresi, L. See, G. Mills, J. Ching, P. J. Alexander, J. J. Feddema, A. J. Florczyk and I. Stewart. *TOWARDS CONSISTENT MAPPING OF URBAN STRUCTURES – GLOBAL HUMAN SETTLEMENT LAYER AND LOCAL CLIMATE ZONES*. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLI-B8, 1371–1378. doi: 10.5194/isprsarchives-xli-b8-1371-2016. [ONLINE] Available at: <http://dx.doi.org/10.5194/isprsarchives-XLI-B8-1371-2016>.
- Bechtel et al., 2016b.** B. Bechtel, M. Pesaresi, L. See, G. Mills, J. Ching, P. J. Alexander, J. J. Feddema, A. J. Florczyk and I. Stewart. *TOWARDS CONSISTENT MAPPING OF URBAN STRUCTURES – GLOBAL HUMAN SETTLEMENT LAYER AND LOCAL CLIMATE ZONES*. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XLI-B8, 1371–1378. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/1991265935?accountid=8144>. Copyright - Copyright Copernicus GmbH 2016; Last updated - 2018-01-26.
- Bechtel et al., Feb 2015.** Benjamin Bechtel, Paul Alexander, Jürgen Böhner, Jason Ching, Olaf Conrad, Johannes Feddema, Gerald Mills, Linda See and Iain Stewart. *Mapping Local Climate Zones for a Worldwide Database of the Form and Function of Cities*. ISPRS International Journal of Geo-Information, 4(1), 199–219. ISSN 2220-9964. doi: 10.3390/ijgi4010199. [ONLINE] Available at: <http://dx.doi.org/10.3390/ijgi4010199>.
- Bechtel et al., 2019.** Benjamin Bechtel, Paul J. Alexander, Christoph Beck, Jürgen Böhner, Oscar Brousse, Jason Ching, Matthias Demuzere, Cidália Fonte, Tamás Gál, Julia Hidalgo, Peter Hoffmann, Ariane Middel, Gerald Mills, Chao Ren, Linda See, Panagiotis Sismanidis, Marie-Leen Verdonck, Guang Xu and Yong Xu. *Generating WUDAPT Level 0 data – Current status of production and evaluation*. Urban Climate, 27, 24 – 45. ISSN 2212-0955. doi: <https://doi.org/10.1016/j.uclim.2018.10.001>. [ONLINE] Available at: <http://www.sciencedirect.com/science/article/pii/S221209551830302X>.
- Brousse et al., 07 2016.** Oscar Brousse, Alberto Martilli, Mícheál Foley, Gerald Mills and Benjamin Bechtel. *WUDAPT, an efficient land use producing data tool for mesoscale models? Integration of urban LCZ in WRF over Madrid*. Urban Climate, 17, pp 116–134. doi: 10.1016/j.uclim.2016.04.001.
- Buyantuyev and Wu, 2009.** Alexander Buyantuyev and Jianguo Wu. *Urban heat islands and landscape heterogeneity: linking spatiotemporal variations in surface temperatures to land-cover and socioeconomic patterns*. <http://leml.la.asu.edu/jingle/Wu-Publications-PDFs/2010/Buyantuyev+Wu-2010-UHI.pdf>. Accessed: 17-06-2020.
- Cai et al., 2016.** Meng Cai, Chao Ren, Yong Xu, Wei Dai and Xue Mei Wang. *Local Climate Zone Study for Sustainable Megacities Development by Using Improved WUDAPT Methodology – A Case Study in Guangzhou*. Procedia Environmental Sciences, 36, 82–89. doi: 10.1016/j.proenv.2016.09.017. [ONLINE] Available at: <http://dx.doi.org/10.1016/j.proenv.2016.09.017>.

- Cai et al., 06 2017.** Meng Cai, Chao Ren, Yong Xu, Kevin Lau and Ran Wang. *Investigating the relationship between local climate zone and land surface temperature using an improved WUDAPT methodology – A case study of Yangtze River Delta, China.* Urban Climate. doi: 10.1016/j.uclim.2017.05.010.
- cai Gao, 1996.** Bo cai Gao. *NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space.* Remote Sensing of Environment, 58(3), 257 – 266. ISSN 0034-4257. doi: [https://doi.org/10.1016/S0034-4257\(96\)00067-3](https://doi.org/10.1016/S0034-4257(96)00067-3). [ONLINE] Available at: <http://www.sciencedirect.com/science/article/pii/S0034425796000673>.
- Campos-Taberner et al., Jul 2018.** Manuel Campos-Taberner, Álvaro Moreno-Martínez, Francisco García-Haro, Gustau Camps-Valls, Nathaniel Robinson, Jens Kattge and Steven Running. *Global Estimation of Biophysical Variables from Google Earth Engine Platform.* Remote Sensing, 10(8), 1167. ISSN 2072-4292. doi: 10.3390/rs10081167. [ONLINE] Available at: <http://dx.doi.org/10.3390/rs10081167>.
- Ching et al., 10 2018.** J. Ching, G. Mills, B. Bechtel, L. See, J. Feddema, X. Wang, C. Ren, O. Brousse, A. Martilli, M. Neophytou, P. Mouzourides, I. Stewart, A. Hanna, E. Ng, M. Foley, P. Alexander, D. Aliaga, D. Niyogi, A. Shreevastava, P. Bhalachandran, V. Masson, J. Hidalgo, J. Fung, M. Andrade, A. Baklanov, W. Dai, G. Milcinski, M. Demuzere, N. Brunsell, M. Pesaresi, S. Miao, Q. Mu, F. Chen and N. Theeuwes. *WUDAPT: An Urban Weather, Climate, and Environmental Modeling Infrastructure for the Anthropocene.* Bulletin of the American Meteorological Society, 99 (9), 1907–1924. ISSN 0003-0007. doi: 10.1175/BAMS-D-16-0236.1. [ONLINE] Available at: <https://doi.org/10.1175/BAMS-D-16-0236.1>.
- Collins and Dronova, 07 2019.** Jed Collins and Iryna Dronova. *Urban Landscape Change Analysis Using Local Climate Zones and Object-Based Classification in the Salt Lake Metro Region, Utah, USA.* Remote Sensing, 11, 1615. doi: 10.3390/rs11131615.
- Paul Coseo and Larissa Larsen, 01 2014.** Paul Coseo and Larissa Larsen. *How Local Climate Zones Predict Urban Heat Islands in Eight Chicago Neighborhoods.* 01 2014. doi: 10.13140/RG.2.1.2164.1689.
- Cristóbal et al., March 2018.** Jordi Cristóbal, Juan Jiménez-Muñoz, Anupma Prakash, Cristian Mattar, Dražen Skoković and José Sobrino. *An Improved Single-Channel Method to Retrieve Land Surface Temperature from the Landsat-8 Thermal Band.* Remote Sensing, 10(3), 431. doi: 10.3390/rs10030431. [ONLINE] Available at: <https://doi.org/10.3390/rs10030431>.
- Demuzere et al., 2019.** Matthias Demuzere, Benjamin Bechtel, Ariane Middel and Gerald Mills. *Mapping Europe into local climate zones.* PLoS one, 14(4), 1. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/2215024103?accountid=8144>. Date completed - 2019-12-27; Date created - 2019-04-25; Date revised - 2020-03-10; SuppNotes - Conflict of Interest: The authors have declared that no competing interests exist. Cited By: PLoS One. 2016 Mar 28;11(3):e0147121 27018852] Sensors (Basel). 2015 Jun 11;15(6):13763-77 26110405] Nature. 2004 Jan

22;427(6972):332-6 14716318] C R Biol. 2008 Feb;331(2):171-8 18241810] Science. 2008 Feb 8;319(5864):756-60 18258902] Rev Geophys. 2015 Jun;53(2):323-361 27478878; Last updated - 2020-03-10.

Developer'sGuide, n.d. Developer'sGuide. *Google EArth Engine*.

<https://developers.google.com/earth-engine/playground>. Accessed: 29-06-2020.

Dong et al., November 2016. Jinwei Dong, Xiangming Xiao, Michael A. Menarguez, Geli Zhang, Yuanwei Qin, David Thau, Chandrashekhar Biradar and Berrien Moore. *Mapping paddy rice planting area in northeastern Asia with Landsat 8 images, phenology-based algorithm and Google Earth Engine*. Remote Sensing of Environment, 185, 142–154. doi: 10.1016/j.rse.2016.02.016. [ONLINE] Available at: <https://doi.org/10.1016/j.rse.2016.02.016>.

eos, n.d. eos. *LANDSAT 8*. <https://eos.com/landsat-8/>. Accessed: 02-05-2019.

Ermida et al., May 2020. Sofia L. Ermida, Patrícia Soares, Vasco Mantas, Frank-M. Götsche and Isabel F. Trigo. *Google Earth Engine Open-Source Code for Land Surface Temperature Estimation from the Landsat Series*. Remote Sensing, 12(9), 1471. ISSN 2072-4292. doi: 10.3390/rs12091471. [ONLINE] Available at: <http://dx.doi.org/10.3390/rs12091471>.

Estacio et al., 12 2019. Ian Estacio, Jennieveive Babaan, Nereo Pecson, Ariel Blanco, J. Escoto and C. Alcantara. *GIS-BASED MAPPING OF LOCAL CLIMATE ZONES USING FUZZY LOGIC AND CELLULAR AUTOMATA*. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W19, 199–206. doi: 10.5194/isprs-archives-XLII-4-W19-199-2019.

Estoque and Murayama, 11 2017. Ronald Estoque and Yuji Murayama. *Monitoring surface urban heat island formation in a tropical mountain city using Landsat data (1987–2015)*. ISPRS Journal of Photogrammetry and Remote Sensing, 133, 18–29. doi: 10.1016/j.isprsjprs.2017.09.008.

Geletič et al., Mar 2019. Jan Geletič, Michal Lehnert, Petr Dobrovolný and Maja Žuvela-Aloise. *Spatial modelling of summer climate indices based on local climate zones: expected changes in the future climate of Brno, Czech Republic*. Climatic Change, 152 (3), 487–502. ISSN 1573-1480. doi: 10.1007/s10584-018-2353-5. [ONLINE] Available at: <https://doi.org/10.1007/s10584-018-2353-5>.

GisGeography.com, n.d. GisGeography.com. *ImageCollection Overview*.

<https://gisgeography.com/ndvi-normalized-difference-vegetation-index/>. Accessed: 21-06-2020.

Google Earth Engine, n.d.a. Google Earth Engine. *JavaScript*.

https://developers.google.com/earth-engine/tutorial_js_01. Accessed: 21-06-2020.

Google Earth Engine, n.d.b. Google Earth Engine. *ImageCollection Overview*.

https://developers.google.com/earth-engine/ic_creating. Accessed: 18-06-2020.

- Google Earth Engine, n.d.c.** Google Earth Engine. *ImageCollection Overview*.
https://developers.google.com/earth-engine/tutorial_api_05. Accessed:
22-06-2020.
- Google Earth Engine, n.d.d.** Google Earth Engine. *Platform*.
<https://earthengine.google.com/platform/>. Accessed: 18-04-2019.
- Google Earth Engine, n.d.e.** Google Earth Engine. *FAQ*.
<https://earthengine.google.com/faq/>. Accessed: 18-04-2019.
- Google Earth Engine, n.d.f.** Google Earth Engine. *ImageCollection Overview*.
https://https://developers.google.com/earth-engine/ic_filtering. Accessed:
18-06-2020.
- Google Earth Engine, n.d.g.** Google Earth Engine. *ImageCollection Overview*.
https://developers.google.com/earth-engine/tutorial_api_05. Accessed:
21-06-2020.
- Google Earth Engine, n.d.h.** Google Earth Engine. *ImageCollection Overview*.
https://developers.google.com/earth-engine/ic_composite_mosaic. Accessed:
18-06-2020.
- GoogleEarthEngine, n.d.** GoogleEarthEngine. *Google Earth Engine Platform*.
<https://earthengine.google.com/platform/>. Accessed: 29-06-2020.
- Gorelick et al., 2017.** Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau and Rebecca Moore. *Google Earth Engine: Planetary-scale geospatial analysis for everyone*. Remote Sensing of Environment, 202, 18 – 27. ISSN 0034-4257. doi: <https://doi.org/10.1016/j.rse.2017.06.031>. [ONLINE] Available at:
<http://www.sciencedirect.com/science/article/pii/S0034425717302900>. Big Remotely Sensed Data: tools, applications and experiences.
- Grimmond et al., 2010.** Christine Grimmond, Matthias Roth, T. Oke, Y.C. Au, M. Best, Richard Betts, Gregory Carmichael, Helen Cleugh, Walter Dabberdt, R. Emmanuel and Edmilson Freitas. *Climate and More Sustainable Cities: Climate Information for Improved Planning and Management of Cities (Producers/Capabilities Perspective)*. Procedia Environmental Sciences, 1, 247–274. doi: 10.1016/j.proenv.2010.09.016.
- Huang et al., 2017.** Huabing Huang, Yanlei Chen, Nicholas Clinton, Jie Wang, Xiaoyi Wang, Caixia Liu, Peng Gong, Jun Yang, Yuqi Bai, Yaomin Zheng and Zhiliang Zhu. *Mapping major land cover dynamics in Beijing using all Landsat images in Google Earth Engine*. Remote Sensing of Environment, 202. doi: 10.1016/j.rse.2017.02.021.
- Huete, 1988.** A.R Huete. *A soil-adjusted vegetation index (SAVI)*. Remote Sensing of Environment, 25(3), 295 – 309. ISSN 0034-4257. doi: [https://doi.org/10.1016/0034-4257\(88\)90106-X](https://doi.org/10.1016/0034-4257(88)90106-X). [ONLINE] Available at:
<http://www.sciencedirect.com/science/article/pii/003442578890106X>.

- JavaScript, n.d.a.** JavaScript. *JavaScript*.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Accessed:
18-06-2020.
- JavaScript, n.d.b.** JavaScript. *JavaScript*. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Accessed: 18-06-2020.
- JavaScript, n.d.c.** JavaScript. *JavaScript*. https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript. Accessed: 18-06-2020.
- Jeevalakshmi et al., 01 2017.** D. Jeevalakshmi, S. Reddy and Balakrishnan Manikiam. *Land surface temperature retrieval from LANDSAT data using emissivity estimation*. International Journal of Applied Engineering Research, 12, 9679–9687.
- Jiménez-Muñoz and Sobrino, March 2006.** J. C. Jiménez-Muñoz and J. A. Sobrino. *Error sources on the land surface temperature retrieved from thermal infrared single channel remote sensing data*. International Journal of Remote Sensing, 27(5), 999–1014. doi: 10.1080/01431160500075907. [ONLINE] Available at: <https://doi.org/10.1080/01431160500075907>.
- Johansen et al., 07 2015.** Kasper Johansen, Stuart Phinn and Martin Taylor. *Remote Sensing Applications: Society and Environment*. Remote Sensing Applications: Society and Environment, 1, 36–49. doi: 10.1016/j.rsase.2015.06.002.
- Kaloustian and Bechtel, 12 2016.** Noushig Kaloustian and Benjamin Bechtel. *Local Climatic Zoning and Urban Heat Island in Beirut*. Procedia Engineering, 169, 216–223. doi: 10.1016/j.proeng.2016.10.026.
- Kumar and Mutanga, Sep 2018.** Lalit Kumar and Onesimo Mutanga. *Google Earth Engine Applications Since Inception: Usage, Trends, and Potential*. Remote Sensing, 10 (10), 1509. ISSN 2072-4292. doi: 10.3390/rs10101509. [ONLINE] Available at: <http://dx.doi.org/10.3390/rs10101509>.
- Landsat Science, n.d.a.** Landsat Science. *Landsat 5*.
<https://landsat.gsfc.nasa.gov/a-landsat-timeline/>. Accessed: 21-05-2019.
- Landsat Science, n.d.b.** Landsat Science. *Landsat 5*.
<https://landsat.gsfc.nasa.gov/landsat-5/>. Accessed: 25-04-2019.
- Leconte et al., 2015.** Francois Leconte, Julien Bouyer, Rémy Claverie and Mathieu Pétrissans. *Using Local Climate Zone scheme for UHI assessment: Evaluation of the method using mobile measurements*. Building and Environment, 83, 39 – 49. ISSN 0360-1323. doi: <https://doi.org/10.1016/j.buildenv.2014.05.005>. [ONLINE] Available at: <http://www.sciencedirect.com/science/article/pii/S0360132314001413>. Special Issue: Climate adaptation in cities.
- Lehnert et al., 11 2015.** Michal Lehnert, Jan GeletiAe, Jan Husak and Miroslav Vysoudil. *Urban field classification by "local climate zones" in a medium-sized Central European city: the case of Olomouc (Czech Republic)*. Theoretical and Applied Climatology, 122(3-4), 531–541. ISBN 0177-798X, 0177-798X. [ONLINE] Available at:

- <https://search-proquest-com.zorac.aub.aau.dk/docview/1727694142?accountid=8144>. Date revised - 2015-10-01; Last updated - 2016-07-07; SubjectsTermNotLitGenreText - Temperature effects; Standardization; Classification; Climate; Climatology; Temperature differences; Climatic zones; Air temperature; Methodology; Applied climatology; Local climates; Documentation; Air Temperature; Climatic Zones; Climates; Networks; Temperature; Zones; Czech Rep.
- Levlovics et al., 03 2013.** E. Levlovics, Tamás Gál and J. Unger. *Mapping local climate zones with a vector-based GIS method*. Aerul și Apa: Componente ale Mediului, 2013, 230–423.
- Liu et al., May 2019.** Cheng-Chien Liu, Yi-Hsin Chen, Mei-Heng Margaret Wu, Chiang Wei and Ming-Hsun Ko. *Assessment of forest restoration with multitemporal remote sensing imagery*. Scientific Reports, 9(1). doi: 10.1038/s41598-019-43544-5. [ONLINE] Available at: <https://doi.org/10.1038/s41598-019-43544-5>.
- Mateo-García et al., Jul 2018.** Gonzalo Mateo-García, Luis Gómez-Chova, Julia Amorós-López, Jordi Muñoz-Marí and Gustau Camps-Valls. *Multitemporal Cloud Masking in the Google Earth Engine*. Remote Sensing, 10(7), 1079. ISSN 2072-4292. doi: 10.3390/rs10071079. [ONLINE] Available at: <http://dx.doi.org/10.3390/rs10071079>.
- McFEETERS, May 1996.** S. K. McFEETERS. *The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features*. International Journal of Remote Sensing, 17(7), 1425–1432. doi: 10.1080/01431169608948714. [ONLINE] Available at: <https://doi.org/10.1080/01431169608948714>.
- McHugh, 2012.** Mary L. McHugh. *Interrater reliability: the kappa statistic*. Biochemia medica, 22(3), 276–282. ISSN 1330-0962. [ONLINE] Available at: <https://pubmed.ncbi.nlm.nih.gov/23092060>. 23092060[pmid].
- Meng et al., January 2019.** Xiangchen Meng, Jie Cheng, Shaohua Zhao, Sihan Liu and Yunjun Yao. *Estimating Land Surface Temperature from Landsat-8 Data using the NOAA JPSS Enterprise Algorithm*. Remote Sensing, 11(2), 155. doi: 10.3390/rs11020155. [ONLINE] Available at: <https://doi.org/10.3390/rs11020155>.
- Mijani et al., October 2020.** Naeim Mijani, Seyed Kazem Alavipanah, Mohammad Karimi Firozjaei, Jamal Jokar Arsanjani, Saeid Hamzeh and Qihao Weng. *Modeling outdoor thermal comfort using satellite imagery: A principle component analysis-based approach*. Ecological Indicators, 117, 106555. doi: 10.1016/j.ecolind.2020.106555. [ONLINE] Available at: <https://doi.org/10.1016/j.ecolind.2020.106555>.
- Mirzaei, 05 2015.** Parham Mirzaei. *Recent Challenges in Modelling of Urban Heat Island*. Sustainable Cities and Society, 76. doi: 10.1016/j.scs.2015.04.001.
- Mirzaei and Haghighat, 10 2010.** Parham Mirzaei and Fariborz Haghighat. *Approaches to study Urban Heat Island – Abilities and limitations*. Building and Environment, 45, 2192–2201. doi: 10.1016/j.buildenv.2010.04.001.

- Monocle, n.d.** Monocle. *Most liveable city: Copenhagen*.
<https://monocle.com/film/affairs/most-liveable-city-copenhagen/>. Accessed: 19-06-2020.
- Moore and Hansen, December 2011.** R. T. Moore and M. C. Hansen. *Google Earth Engine: a new cloud-computing platform for global-scale earth observation data and analysis*.
- Nedkov et al., 2017.** Stoyan Nedkov, Miglena Zhiyanski, Stelian Dimitrov, Bilyana Borisova, Anton Popov, Ivo Ihtimanski, Rositsa Yaneva, Petar Nikolov and Svetla Bratanova-Doncheva. *Mapping and assessment of urban ecosystem condition and services using integrated index of spatial structure*. *One Ecosystem*, 2, e14499. doi: 10.3897/oneeco.2.e14499. [ONLINE] Available at:
<https://doi.org/10.3897/oneeco.2.e14499>.
- Nissen, 2014.** Nikolai Nissen. *Nordhavn: A Governmentality analysis of urban planning*. [ONLINE] Available at:
https://research.cbs.dk/files/58445242/nikolai_nissen.pdf.
- Peng et al., September 2018.** Jian Peng, Jinglei Jia, Yanxu Liu, Huilei Li and Jiansheng Wu. *Seasonal contrast of the dominant factors for spatial distribution of land surface temperature in urban areas*. *Remote Sensing of Environment*, 215, 255–267. doi: 10.1016/j.rse.2018.06.010. [ONLINE] Available at:
<https://doi.org/10.1016/j.rse.2018.06.010>.
- Perera and Emmanuel, 03 2018.** Narein Perera and R. Emmanuel. *A “Local Climate Zone” based approach to urban planning in Colombo, Sri Lanka*. *Urban Climate*, 23, 188–203. doi: 10.1016/j.uclim.2016.11.006.
- Ravanelli et al., 09 2018.** Roberta Ravanelli, Andrea Nascetti, Raffaella Cirigliano, Clarissa Rico, Giovanni Leuzzi, Paolo Monti and Mattia Crespi. *Monitoring the Impact of Land Cover Change on Surface Urban Heat Island through Google Earth Engine: Proposal of a Global Methodology, First Applications and Problems*. *Remote Sensing*, 10, 1488. doi: 10.3390/rs10091488.
- Chao Ren, Meng Cai, Ran Wang, Yong Xu and Edward Ng, 05 2016.* Chao Ren, Meng Cai, Ran Wang, Yong Xu and Edward Ng. *Local Climate Zone (LCZ) Classification Using the World Urban Database and Access Portal Tools (WUDAPT) Method: A Case Study in Wuhan and Hangzhou*. 05 2016.
- Salih et al., November 2018.** Muhammad Mejbel Salih, Oday Zakariya Jasim, Khalid I. Hassoon and Aysar Jameel Abdalkadhum. *Land Surface Temperature Retrieval from LANDSAT-8 Thermal Infrared Sensor Data and Validation with Infrared Thermometer Camera*. *International Journal of Engineering & Technology*, 7(4.20), 608. doi: 10.14419/ijet.v7i4.20.27402. [ONLINE] Available at:
<https://doi.org/10.14419/ijet.v7i4.20.27402>.
- Stewart and Oke, 12 2012.** I.D. Stewart and T. Oke. *Local Climate Zones for Urban Temperature Studies*. *Bulletin of the American Meteorological Society*, 93, 1879–1900. doi: 10.1175/BAMS-D-11-00019.1.

- Stone et al., Jul 2013.** Brian Stone, Jason Vargo, Peng Liu, Yongtao Hu and Armistead Russell. *Climate Change Adaptation Through Urban Heat Management in Atlanta, Georgia*. Environmental Science & Technology, 47(14), 7780–7786. ISSN 0013-936X. doi: 10.1021/es304352e. [ONLINE] Available at: <https://doi.org/10.1021/es304352e>.
- TheLokal, n.d.** TheLokal. *Denmark plans man-made islands to draw business*. <https://www.thelokal.dk/20190107/denmark-plans-man-made-islands-to-draw-business>. Accessed: 29-06-2020.
- USGS, n.d.** USGS. *Landsat*. <https://www.usgs.gov/land-resources/nli/landsat>. Accessed: 25-04-2019.
- Vandamme et al., 07 2019.** St  phanie Vandamme, Matthias Demuzere, Marie-Leen Verdonck, Zhiming Zhang and Frieke Van Coillie. *Revealing Kunming's (China) Historical Urban Planning Policies Through Local Climate Zones*. Remote Sensing, 11, 1731. doi: 10.3390/rs11141731.
- Verma and Jana, 09 2019.** Deepank Verma and Arnab Jana. *LULC classification methodology based on simple Convolutional Neural Network to map complex urban forms at finer scale: Evidence from Mumbai*.
- Wang et al., 09 2017.** Wang, Q. Zhan and Wanlu Ouyang. *Impact of Urban Climate Landscape Patterns on Land Surface Temperature in Wuhan, China*. Sustainability, 9, 1700. doi: 10.3390/su9101700.
- Wim H. Bakker, 2009.** Wim Feringa Gabriel N. Parodi Ambro S. M. Gieske Christine Pohl Ben G. H. Gorte Colin V. Reeves Karl A. Grabmaier Frank J. van Ruitenbeek Chris A. Hecker Ernst M. Schetselaar John A. Horn Klaus Tempfli Gerrit C. Huurneman Michael J. C. Weir Lucas L. F. Janssen Eduard Westinga Norman Kerle Tsehaie Woldai Wim H. Bakker, Freek D. van der Meer. *Principles of remote sensing*. <https://webapps.itc.utwente.nl>. Accessed: 22-05-2019.
- Wong et al., Mar 2019.** Michael Mau Fung Wong, Jimmy Chi Hung Fung and Peter Pak Shing Yeung. *High-resolution calculation of the urban vegetation fraction in the Pearl River Delta from the Sentinel-2 NDVI for urban climate model parameterization*. Geoscience Letters, 6(1), 2. ISSN 2196-4092. doi: 10.1186/s40562-019-0132-4. [ONLINE] Available at: <https://doi.org/10.1186/s40562-019-0132-4>.
- www.usgs.gov, n.d.a.** www.usgs.gov. *EVI*. https://www.usgs.gov/land-resources/nli/landsat/landsat-enhanced-vegetation-index?qt-science_support_page_related_con=0#qt-science_support_page_related_con.
- www.usgs.gov, n.d.b.** www.usgs.gov. *NDVI*. https://www.usgs.gov/land-resources/nli/landsat/landsat-normalized-difference-vegetation-index?qt-science_support_page_related_con=0#qt-science_support_page_related_con.
- X., 2015.** Ng Y. X. *view of a study of urban heat island using local climate zones the case of singapore 2015*. British Journal of Environment Climate Change. [ONLINE] Available at: <http://www.journalijecc.com/index.php/IJECC/article/view/27298/51239>.

- Xie et al., Dec 2019.** Shuai Xie, Liangyun Liu, Xiao Zhang, Jiangning Yang, Xidong Chen and Yuan Gao. *Automatic Land-Cover Mapping using Landsat Time-Series Data based on Google Earth Engine*. Remote Sensing, 11(24), 3023. ISSN 2072-4292. doi: 10.3390/rs11243023. [ONLINE] Available at: <http://dx.doi.org/10.3390/rs11243023>.
- Xu, June 2008.** H. Xu. *A new index for delineating built-up land features in satellite imagery*. International Journal of Remote Sensing, 29(14), 4269–4276. doi: 10.1080/01431160802039957. [ONLINE] Available at: <https://doi.org/10.1080/01431160802039957>.
- Xue and Su, 2017.** Jinru Xue and Baofeng Su. *Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications*. Journal of Sensors, 2017, 1–17. doi: 10.1155/2017/1353691. [ONLINE] Available at: <https://doi.org/10.1155/2017/1353691>.
- Zhang et al., May 29 2019.** Guichen Zhang, Pedram Ghamisi and Xiang Z. Xiao. *Fusion of Heterogeneous Earth Observation Data for the Classification of Local Climate Zones*, 2019. [ONLINE] Available at: <https://search-proquest-com.zorac.aub.aau.dk/docview/2232267056?accountid=8144>. Copyright - © 2019. This work is published under <http://arxiv.org/licenses/nonexclusive-distrib/1.0/> (the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2019-10-24.
- Zhao, 2018.** Chunhong Zhao. *STUDYING SURFACE URBAN HEAT ISLAND PHENOMENON USING REMOTE SENSING IN THREE METROPOLITAN AREAS OF TEXAS, USA*. <https://digital.library.txstate.edu/bitstream/handle/10877/7411/ZHAO-DISSERTATION-2018.pdf?isAllowed=y&sequence=1>. Accessed: 17-06-2020.
- Zheng et al., 06 2017.** Yingsheng Zheng, Chao Ren, Yong Xu, Ran Wang, Justin Ho, Kevin Lau and Edward Ng. *GIS-based mapping of Local Climate Zone in the high-density city of Hong Kong*. Urban Climate. doi: 10.1016/j.uclim.2017.05.008.

Appendix guide A

Appendix guide

Here is the Appendix guide:

Appendix contains 1 ZIP file with two separate files one with the Local Climate Zones and Land Surface Temperature scripts available for copy paste.

Direct link to the Google Earth Engine code of the Local Climate Zones Code where the user should run the *Index* file, where any user that has this link can access it run it , copy it as follows : Users who have access to the project will be able to access it in the Code Editor via this link:

`code.earthengine.google.com/LCZ`

Direct link to the Google Earth Engine code of the Land Surface Temperature where the user should run the *Graph* file, where any user that has this link can access it run it , copy it as follows :

`code.earthengine.google.com/LST`

