AALBORG UNIVERSITY
AAU CPH
GEOINFORMATICS

Automatic Assessment of District Cooling Potential Based on Aerial Images

Authors:	*Student nr.:*	*Supervisor:*
Ionut Vieru	20191183	Prof. Carsten Keßler

June 2021

Title:

Automatic Assessment of
District Cooling Potential
Based on Aerial Images

Project:

Master Thesis, Geoinformatics

Aalborg University - Copenhagen

Project Period:

February 2021 - June 2021

Student:

Vieru Ionut

Supervisor:

Prof. Carsten Keßler

Number of ZIP files: 1
Finished: June 2021

Abstract:

The current cooling market is dominated by conventional air-conditioning systems powered by electricity having a significant impact on the environment. In the 21st Conference of Parties, held in Paris, 175 countries agreed to take measurements to tackle climate change and keep the average global temperature below 1.5°C above the pre-industrial levels. Sustainable sources of energy can help meeting this target and significantly reduce the greenhouse gas emissions. From a societal perspective, switching to sustainable alternatives to conventional cooling systems brings massive benefits to the community. District cooling paves the road to a low-carbon society contributing to a sustainable way of living. To support decision making in the process of developing the district cooling market, automated workflows for estimating the cooling demand can help the utility companies to identify new areas of expansion. This master thesis explores this potential and aims at developing a fully automated process for assessing the cooling potential using aerial imagery.

Preface

In recent years, urbanization and globalization have led to an increase in energy consumption, leading to higher greenhouse gas emissions. There is a high motivation in the energy sector in using renewable sources of energy for its environmental benefits and the cost savings associated with it. District energy has a great potential in shifting to more sustainable energy sources, providing cheap and clean energy for the entire community.

District cooling is a relatively new field in district energy, but it registered a continuous expansion in recent years, especially in the Nordic countries where district energy is widely used.

Energy efficiency is one of the great benefits of district cooling, but also the higher utilization of equipment makes it cheaper than conventional systems like central or single room air conditioning systems.

The consumer has various reasons to choose district cooling: better economy, no noise from chiller, reduced risk of sudden repair costs, no hazardous refrigerants in the building, easier to certify the building with a green certificate etc.

Not only the end-consumer benefits from it but the whole community in general. The society has the opportunity to become carbon neutral in a sustainable and cost-effective way. The electricity demand would be reduced, relieving the pressure on the electricity grid, and making room for more sustainable energy. The utility company benefits through implementing alternative cooling methods that are sustainable and inexpensive. One example would be to use sea water to chill down the water from the district cooling system. Another example would be reducing production costs by obtaining synergies between heating and cooling grids.

In the summertime office buildings require cooling while apartment buildings next door are in need for hot water. Solar waste heat can be collected with district cooling network, refined, and used for producing hot water.

These synergies between systems coupled with renewable sources of energy contribute to the overall efficiency of the system leading to less energy losses and considerable reducing the energy costs.

Contents

List of Figures

List of Tables

Introduction 1

1.1 General context

The topic of climate change drives a lot of attention in scientific communities these days, the consequences of global warming has severe implications in our daily life and further actions is needed to be taken to limit its effects on the environment.

One of the main factors responsible for climate change is the greenhouse gas emissions, which traps the solar radiation in the atmosphere causing an increase in the temperature of the earth's crust and oceans.[EESI]

Another factor accounting for climate change is deforestation, one quarter of the earth's surface is covered by forests which are responsible for absorbing three quarters of the CO2 from the atmosphere while releasing oxygen.[Bennett, 2017]

Urbanization has also a profound effect on the climate as commercialization and industrialization encourages the use of fossil fuels leading to global warming and climate changes. Changes in the land use and surface materials associated with the road infrastructure and buildings also contribute to an increase of the temperature in the urban settlements, known as the heat island effect. Urban temperatures are 1-3 degrees higher compared to the rural areas and it can go as far as 10 degrees under certain conditions.[GRIMMOND, 2007]

According to United Nations projections, a 25% increase in urban population was registered between 1950 and 2018, from 30% in 1950 to 55% in 2018, with an annual average rate of 0.92% per year. Towards 2030 the share of population living in urban settlements is expected to reach 60%. [UnitedNations, 2018]

Although urban areas cover only 2% of the earth's surface, 75% of the resources are channeled in these areas leading to an increase in transport distances. Urban migration also implies an increase in the economic activity in the city, creating potential for economic development and industrialization. This translates to a higher energy demand in the urban areas, mostly based on fossil fuels like oil, gas and coal which are less expensive and pollutant. A higher concentration of population in the urban areas creates a demand for transportation services as well, resulting in a higher number of vehicles also powered by fossil fuels.[Madlener and Sunak, 2011]

Residential and commercial buildings require electricity to power household appliances and industrial equipment. According to Eurostat the energy mix in European Union for 2018 had five different sources: Petroleum products (36%), natural gas (21%), solid fossil fuels (15%), renewable energy (15%) and nuclear energy (13%).

Except nuclear energy and renewable energy, the rest of the sources are highly pollutant and are also responsible for environmental changes [Eurostat].

Another factor accountable for carbon emissions is thermal energy. From the total energy consumption in European Union, 30% is used for heating and cooling the buildings being responsible for 36% of the total EU carbon emissions. Switching to modern district energy systems has the potential to significantly reduce the greenhouse gas emissions. Combining different renewable energy solutions with modern district heating sources like geothermal, waste incineration or solar thermal are sustainable alternatives to conventional heating systems. [Danfoss]

These cutting-edge solutions are extensively implemented in Denmark, about 62% of the households were connected to district heating in 2012. Denmark also had 541 CPH units in 2012 generating 13% of the total installed electric capacity. [Sorknæs et al., 2015]

Combining heat and power is an efficient technology that captures the heat used to generate electricity and reuses it for district heating, household hot water and industrial processes, achieving over 80% efficiencies, compared with 50% achieved in conventional systems. [EPA]

1.2 District energy

District energy is a centralized energy solution that provides thermal energy to public, residential and commercial buildings on a neighborhood or even city level. It has been used for many years on a wide range of cities across the globe for its potential to lower energy costs, include renewable sources of energy, improve air quality, make cities more sustainable and resilient.[UNEP]

Countries like France, New Zealand, France, Germany, Denmark and European Union in general are leaders in energy efficiency and adopt district energy solutions.[UNEP]

Energy sources like electricity generated by nuclear powerplants or fossil fuels to provide thermal energy are considered to be inefficient. These sources of energy can be used more efficiently in mechanical processes that don't have the posibility of using an alternative source of energy. A good example to highlight this premise is Dubai where 70% of the total electricity is spent on air-conditioning.[UNEP]

District energy also creates synergies between different industries, and it has the potential to lower the energy costs by utilizing heat sources generated by industrial processes like metal smelting plants, CHP plants or data centers. Another free and sustainable source of energy are the rivers, lakes and oceans which can be exploited by chilling the water used in district cooling [UNEP].

Another benefit of district energy systems is the use of thermal storage. Excess energy can be stored and utilized later, when the thermal demand is high, using thermal storage facilities. CPH plats can store the excess heat, being able to operate only when is beneficial for the electricity market. This way creates room for energy generated from sustainable and renewable sources.[UNEP]

Interconnecting the district heating networks enables the exchange of excess energy with the neighboring districts making the entire system more efficient and less volatile. Modern district energy systems have a high degree of flexibility in terms of the energy sources used and ability to expand over time.[UNEP]

CPH powerplants have numerous benefits, but there are some situations where some balancing is needed. For example during very windy and cold days, CPH plants would have to run in order to fulfill the heat demand, even though it would be unprofitable since the wind can generate high amounts of electricity, lowering the market price. On the other hand, during the hot summer days, the electricity demand is high, but the heat demand is low. In these conditions the CPH plant cannot run because it doesn't generate any heat revenues, otherwise it would be very profitable due to high electricity prices.[UNEP]

In both situations, electric boilers and thermal storage can help. The CPH plant can release the stored heat in the first situation, and in the second situation the excess heat can be stored in seasonal thermal storage facilities and used for district heating at a later point in time [UNEP].

District energy has a variety of benefits not only for building owners but the society in general:

- **Societal benefits:**
  - Creates local workforce demand since district energy systems need to be operated and maintained locally
  - Creates local workforce demand since district energy systems need to be operated and maintained locally
  - Increased comfort and reliability for building owners
- **Economic benefits:**
  - Reduced fuel consumption and costs
  - Is using local energy resources contributing to local economy
  - Reduced heating and operational costs
  - Reduced water consumption compared to conventional systems
- **Environmental benefits:**
  - Reduction in greenhouse emissions
  - Minimizes the heat island effect by releasing less heat in the atmosphere
  - It successfully integrates in environmental programs favoring energy efficient and sustainable sources of fuels
  - Optimizes waste management process [Rezaie and Rosen, 2012]

1.3 District heating

District heating is a centralized solution for providing thermal energy in the form of heat to commercial and residential buildings. The hot water is distributed from a central heat source through an underground network of pipes to the end consumer [CEWEP, 2019].

The heat is transfered from the pipe network to the end consumer using a transfer substation placed in the connected building. Usually buildings have two distribution systems, one for the heat supply and another one for supplying domestic hot water. Sometimes a third system is involved to provide heat to the ventilation system. In the substation a heat exchanger is installed to transfer the heat from the distribution pipes to the internal system.[Werner, 2004]

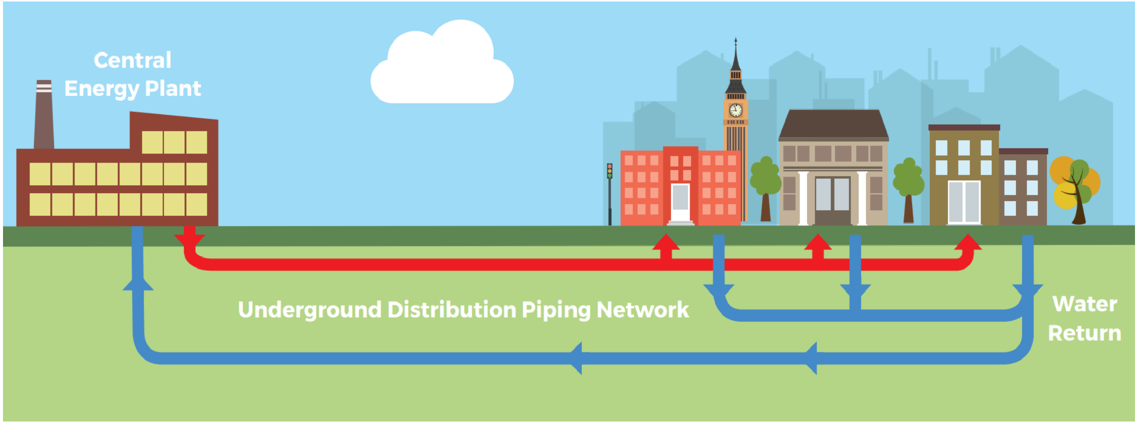

Figure 1.1: District Heating System *Source: calderdale.gov.uk*

A district heating system has three main components: the central heating plant, the consumer system, and the distribution pipe network. [Werner, 2004]

The central plant is the source of the thermic agent and is generated using different technologies like boilers, CPH, solar thermal, heat pumps etc.[relatedproject.eu]

In boiler stations the heat is produced through combustion using fossil fuels, solid waste(biomass) and steam or water is used as a carrier for the thermic agent. The stations powered by fossil fuels are used for smaller utility networks or as a back-up for covering the peak demands. Compared to the traditional household boilers, are more efficient and filter out the pollutants.[relatedproject.eu]

Boilers fueled by biomass are promoted since it is a renewable energy source. In general biomass products are wood chips, residual waste from processed wood products or wood pellets.[relatedproject.eu]

Another sustainable fuel source is waste, burned in a furnace to generate heat, is used in waste-to-energy district heating facilities. [relatedproject.eu]

Solar thermal plants are highly efficient in the summer season, when the heating demand is low, but using thermal storages the heat can be conserved and used at a later point in time. They reach peak efficiencies when combined with CPH systems or gas boilers since are easy to start and stop. In Silkeborg, Denmark was installed the biggest solar system in the world, with a double capacity than a typical solar thermal plant.[relatedproject.eu]

The heat pump systems allow the heat extraction from the ground, as a primary energy source. It also uses electricity, but it converts it into heat very efficiently. These systems are suitable for district heating systems that have intermittent supply of electricity from sustainable sources.[relatedproject.eu]

The heat pumps have the advantage of using low temperature energy sources, complementing renewable and surplus energy sources.[relatedproject.eu]

Other sustainable heat sources that can be used in district heating are waste heat from industrial processes, nuclear power stations, power stations and data centers. [relatedproject.eu]

From the central heating plant, the thermal agent is supplied to the buildings via transport pipes. The pipes can be made from different materials, generally plastic or steel depending on the water temperature, and are insulated to reduce the heat losses. Generally the network system has two pipes for supply and return [northsearegion.eu].

The heat exchanger is installed in the end consumer's building and transfers the heat from the district heating pipe network to the client's internal circulation system without a fluid exchange.[danfoss.com]

For industrial processes sometimes the supply temperatures needs to be higher. As a carrier for heat distribution can also be used steam instead of water, but it has higher heat losses.[danfoss.com]

1.4 District cooling

District cooling is the opposite of the district heating, but it works in a similar fashion. It distributes chilled water through an underground pipe network from a central source to the end consumer building for air-conditioning supply or process cooling for industrial purposes. District cooling was developed as a sustainable and more efficient alternative to the conventional air-conditioning system.

Conventional comfort cooling generally runs on individual appliances powered by electricity. With the expansion of traditional air-conditional systems the electricity consumption grows, generating high peak loads in electricity consumption during summertime, putting high pressure on the electricity grid [COWI, 2015].

The first District Cooling System dates back to 1889 at Denver's Colorado Automatic Refrigerator Company where refrigerants were used as a carrier in the cooling process. In Europe, the first DC system was developed in Paris during the sixties, and it began to be widely used in other countries as Sweden, Germany, Italy, etc. In Japan DCS started to develop very fast since 1970 and by 2005 more than 154 district cooling networks were developed.[Gang et al., 2016]

The first DCS in China was implemented in Beijing and it was put in production in 2004. Arab Emirates adopted district cooling since 1999 and currently has 10% of the DC market. [Gang et al., 2016]

Three generations of district cooling were defined based on the technology used. The pipeline refrigeration systems used refrigerants as a medium and consisted of decentralized evaporators and centralized condensers. It was considered to be the first generation of DCS and dated back in the 19th century.[Lund et al., 2014]

In 1960s the second generation was introduced in Paris and Hamburg, using water as a carrier and large mechanical chillers. In 1990s the CFC refrigerants were banned by the Montreal protocol and the third generation of district cooling emerged. It was using mechanical and absorption chillers, cold storages and also water from lakes and streams in the cooling process. [Lund et al., 2014]

A district cooling system has three main components: the cooling plant, the distribution network and the building connection.

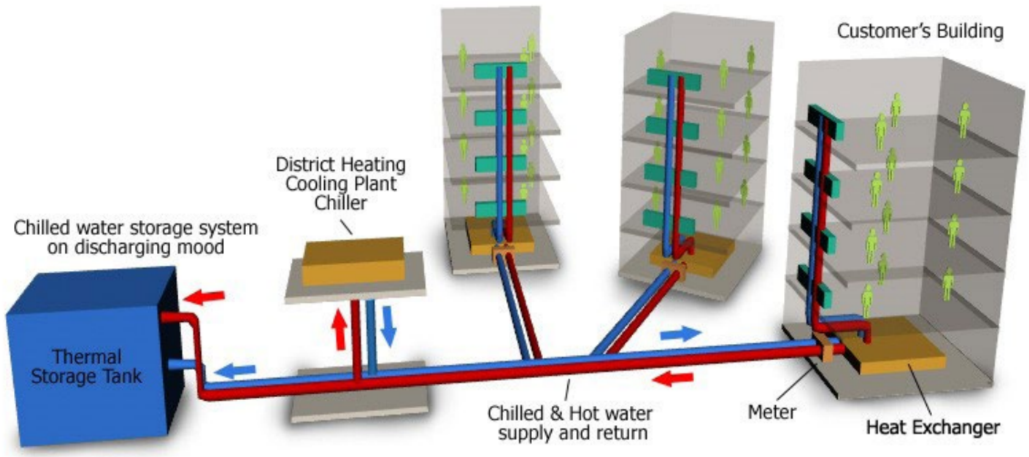

Figure 1.2: District Cooling System *Source: ADB [2017]*

The cooling plant

The cooling plant is the core of a district cooling system. Here the water is chilled using absorption and compression chillers, or sustainable sources like oceans, rivers or deep water lakes. [Calderoni et al., 2019]

Compression chillers are the most commonly used due to their efficiency and flexibility, especially if they have a heat recovery system, where the residual heat can be reused in district heating, increasing this way the energy efficiency of the system.[Calderoni et al., 2019]

Cooling compression chillers are using water to remove the heat from a condenser. The residual heat is rejected in the outside environment, either in bodies of water or atmosphere, using dry or wet cooling towers. In the process of removing the heat, using a wet cooling tower, a heat exchanger helps the water to evaporate in the atmosphere by being sprayed on its surface.[Calderoni et al., 2019]

Absorption chillers take advantage of the fluid properties in the cooling process and are using water-ammonia or lithium bromide-water as absorbent refrigerants. Both solutions are sustainable alternatives to Global Warming Potential refrigerants.[Calderoni et al., 2019]

Natural cooling is another sustainable and cost-effective way of chilling. The water is extracted from natural bodies of water, filtrated and directed in the district cooling system. If the water has a low temperature it can directly be used for chilling or it can be pumped in the compression chillers, increasing efficiency of the system and reducing the production costs. [Calderoni et al., 2019]

The distribution network

The DC distribution network is a closed circuit of conduits composed of two pipes, supply and return. The pipes have a flow temperature of 4-5 degrees for the supply and 12-16 degrees for the return.[districtenergyinitiative.org, 2018]

An efficiency criterion for the cooling network is the water temperature in the return flow, the higher the temperature, the more cost-effective the district cooling network is. But some level of initial investment in the buildings is necessary in order to achieve temperatures higher than 12 degrees like installing chilled beams or bigger air-handling coils.[districtenergyinitiative.org, 2018]

The conduits are insulated and vapor sealed to prevent any heat loss and condensation on the pipes before are buried in the ground. They also have sensors installed that detect if any water pervasions or leakages occur and give early warnings.[districtenergyinitiative.org, 2018]

The building connection

The district cooling conduits are connected with the buildings using two types of connection systems: direct and indirect. In the direct connection the DC water is passing straight into the building's cooling system, while in the indirect system the district cooling system is separate from the building's cooling system and is connected through a heat exchanger that takes care of the cold transfer. [districtenergyinitiative.org, 2018]

Methodology 2

This chapter presents the study area, problem formulation, project objective and the software suite used in developing the project.

2.1 Study area

The proposed study area is the city of Copenhagen, including the suburbs. but a smaller area with a high density of cooling towers was defined for running the experiments, due to extended time periods when running the inferencing.

Figure 2.1: Study area

2.2 Problem formulation

For the utility companies the challenge is to find customers and identify areas with a high cooling demand . As the size and location has an influence on the feasibility of the project, developers must identify the most optimal locations to implement district cooling solutions.

One solution to tackle this challenge is to develop automated mapping tools that identify a cluster of buildings in need for district cooling and estimate the cooling demand for each building. The feasibility of the project would then be derived from the economic difference between conventional cooling and district cooling systems.

2.3 Project objectives

The main objective of the project is to build a model that can be used in the feasibility studies for developing district cooling projects. Using this model, the most suitable areas for developing district cooling projects can be identified and a preliminary design of the pipe infrastructure is developed.

There are three main objectives defined in developing the project: identify a potential market using satellite imagery, define the cooling demand for the area and design the pipe infrastructure.

Based on the project objectives the following research questions are derived:

1. How district cooling potential can be estimated using object detection models?
2. Which object detection model performs better in identifying objects on aerial images?
3. How process automation can be used in assessing the cooling potential?

2.4 Data acquisition and preprocessing

The data used in the project was provided by Kortforsyningen and is freely available on their portal. There are three data layers : the road network, a layer with the building footprints and the aerial images.

The road network and the building footprints layer are in vector format and were clipped on the study area. The bicycle lanes were removed from the dataset since are not used in the project.

The aerial images are in raster format with 12.5 cm per pixel resolution and is divided into a set of small tiles. The tiles were merged into a single mosaic covering the study area.

2.5 Software suite

ArcGIS Pro

ArcGIS Pro is a professional desktop solution provided by Esri widely used GIS. It provides an extensive suite of geoprocessing tools used to manage spatial data and perform spatial analysis.

ArcGIS Python API

ArcGIS Python API is a powerful library used in spatial analysis and data management that enables the users to automate workflows, interact with ArcGIS Online and ArcGIS Enterprise using python scripts. The library is organized into different modules based on functionality and capabilities. [developers.arcgis.com, a]

The arcgis.learn module enables training deep learning models and it integrates with ArcGIS platform to run the model inferencing and consume the training samples. The following dependencies are required to setup the environment for deep learning: pytorch v1.4.0, scikit-image v0.15.0, pillow v6.2.2, libtiff 4.0.10, fastai v1.0.60 and torchvision v0.5.0.[developers.arcgis.com, b]

Anaconda

Anaconda is an open-source package manager that uses command line to install, remove and update dependencies in isolated environments. Since ArcGIS Pro doesn't allow to modify the main python environment, Anaconda was used to clone the environment and install the deep learning dependencies.

Jupyter notebook

Jupyter notebook is a web-based development environment that provides a user interface to support the development process and visualize the results interactively. [https://jupyter.org/]

Theoretical background 3

This chapter supports the implementation phase of the project and will describe different theory concepts, methodologies and underlying principles used throughout the project.

3.1 Machine learning

Machine learning is a branch of artificial intelligence that allows computer systems to learn, adapt and improve from experience using a set of data. [expertAI]

It gives to a computer the power to act without explicitly giving it a set of instructions. Machine learning has a wide range of applicabilities, from speech recognition, object detection to self-driving cars and decoding the human genome [coursera].

Compared to traditional programming which is a manual process where the programmer creates all the rules and strictly defines the functionality of the program, in machine learning the algorithms are automatically derived from the data [LogiAnalytics].

Input + Algorithm = Output (Traditional programming)

Input + Output = Program (Machine learning)

The quality of the data has a huge impact on the machine learning model. If the quality of the input data is bad, even small error can have a big impact, leading to large anomalies. [PotentiaAnalytics]

To understand how machine learning works we first need to know how we feed the data to ML algorithm. The training data is divided into labeled and unlabeled data. Labeled data is a set of sampling data that has one or more characteristics associated (classes). It is a labor-intensive process to label all the sampling data, especially that ML algorithms work with huge amounts of data.[PotentiaAnalytics]

The labeled data must be in a machine-readable format, so several standard formats have been developed, XML or JSON based depending on the libraries, frameworks and algorithms used, like PascalVOC, KITTI Labels, RCNN Masks, MS COCO.[PotentiaAnalytics]

Unlabeled data is a training dataset that it has no characteristics associated, it contains only raw data and typically is used in clustering and anomaly detection tasks. [PotentiaAnalytics]

There are three ways to train a model in ML:

- **Supervised learning** is the most common application in ML and is using labeled data to train the model. The model is trained using a set of data that contains the inputs and the correct outputs, allowing the model to learn in time. The algorithm iterates over the training data a number of times until it achieves a certain level of performance.
- **Unsupervised learning** works with unlabeled data and the goal of this type of algorithm is to discover hidden patterns in massive amounts of data. It requires more training data in comparison with supervised learning.
- **Reinforcement learning** algorithm is inspired from the learning process in human beings and is using the trial-and-error method. On each iteration an interpreter decides if the output is favorable or not. Favorable outputs are reinforced an the non-favorable or the wrong ones are discouraged [PotentiaAnalytics]

3.1.1 Neural Networks

Neural networks are a subdivision of machine learning, their architecture and logic are inspired by the human brain and it simulates the way neurons communicate with each other.[IBMCloudEducation]

The basic computational unit of a neural network is the neuron, also called a node. Neural networks can be composed of thousands, even millions of neurons structured in a series of layers. If all the neurons from each layer are connected with all the other neurons from the adjacent layers, the neural network is fully connected. When a neuron receives an input message it computes the data, if the result is above a certain threshold it gets activated and passes the data to the neurons in the next layer of the network. [IBMCloudEducation]

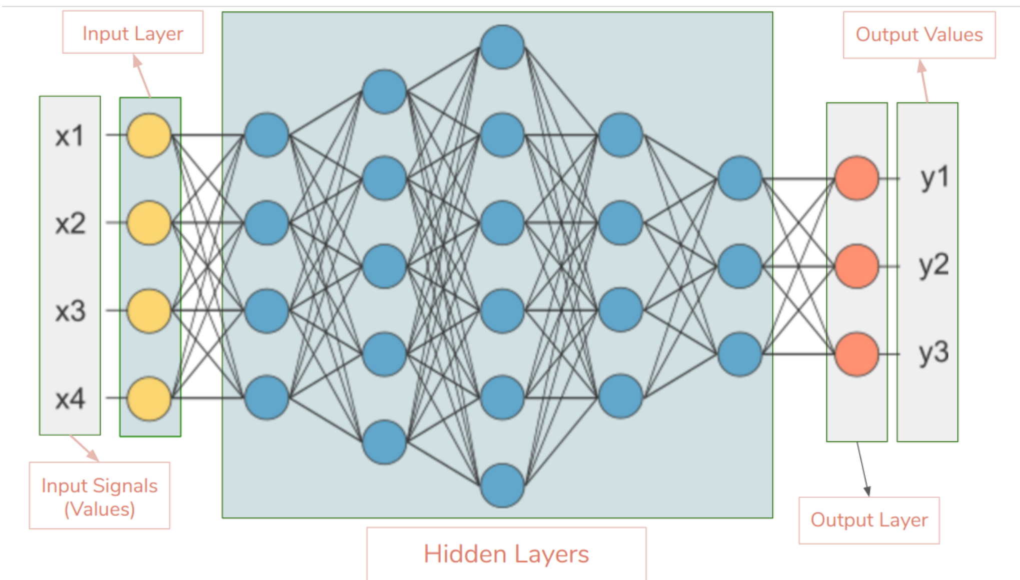

Figure 3.1: Neural Network *Source: hackernoon.com*

The learning mechanism is driven by three main components: the propagation function, connection weight and activation function.[rubikscode.net]

The connection weight simulates the number of neurotransmitters in the brain, in neural networks it represents the importance of the input value of a node. The higher the weight of a neuron is, the greater influence it has on the neuron is passing the meassage to[hackernoon.com].

Usually the weights are determined in the training process using an iterative approach that minimizes the errors by comparing the ground truth or the labeled data with the predicted output. With each iteration the weights are adjusted until the error rate reaches a minimum value [Pantoja et al., 2018].

There are two different types of propagation functions:

- **Forward propagation –** also called inference, is the process of feeding a neural network with input values and getting the predicted values as an output.
- **Back-Propagation –** is used to determine the error rate by comparing the predicted value with the output values [hackernoon.com].

Activation functions are similar with the action-potential in the human brain and it defines if the neuron is going to fire or not. Similarly, in neural networks the activation function establishes if a neuron will pass a message to the next neuron or not [medium.com].

Linear activation is the simplest activation function and is mostly used in regression problems. Neural networks that use this type of activation functions are easy to train but they can not solve complex mapping functions. [towardsdatascience.com, a]

For more complex data structures nonlinear activation functions are more efficient like sigmoid and hyperbolic tangent. Sigmoid activation functions are very popular and transform the input into a value between 0 and 1 and is used for models that predict probabilities as an output. [towardsdatascience.com, a]

Similar to logistic Sigmoid function, another type of nonlinear activation, is hyperbolic tangent function(tanh), which outputs values between -1 and 1. This function has a better predictive performance and is mainly used in classification tasks.[towardsdatascience.com, a]

Machine learning models are often described as black box models because of the hidden layers between the input and output layers. The algorithm is fed with data and outputs the results, but the in-between process is unknown [thelancet.com].

3.1.2 Convolutional Neural Networks

Deep Learning is a subdivision of machine learning and it has a slightly different approach to solve problems. While machine learning algorithms need structured data to train the models and derive the outputs, deep learning models also work with unstructured data. Deep neural networks have a high number of hidden layers compared to conventional neural network which are considered to be shallow. [parsers.me]

A Convolutional Neural Network (CNN) is a Deep Learning model used for processing data, structured in a grid pattern, like images and videos. This type of neural network excels in performing image classification, object detection, edge detection and feature extraction tasks. [towardsdatascience.com, a]

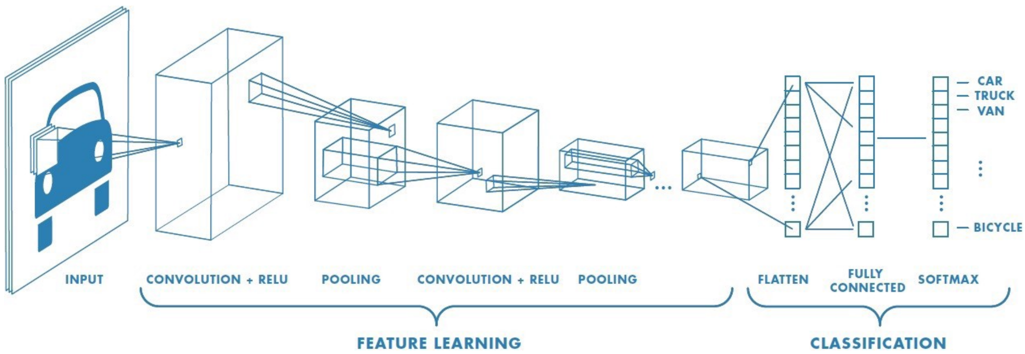

Figure 3.2: Convolutional Neural Network Architecture *Source: towardsdatascience.com [a]*

How CNN work?

An image is interpreted by computers as a matrix where each pixel has a corresponding value in the matrix. A grayscale image is represented as a bidimensional matrix, while a color image is represented by three bidimensional matrices, one for each color channel.[towardsdatascience.com, a]

The image is then passed to an Convolution Layer. The role of a Convolution Layer is to reduce an image to a form that can be processed easier since, high resolution images can be computationally intensive. This process involves a filter or kernel, specialized to extract features, that moves along the image, with a given step size or Stride, until the entire image is traversed. Using this method, the high-level features from the image are extracted without any losses. The output of the convolution operation is called feature map.[towardsdatascience.com, a]

The CNN architecture typically has multiple Convolution Layers, one for detecting low-level features like color or edges and with each layer added it adapts to high-level features. [towardsdatascience.com, a]

For CNN and deep learning the most commonly used activation function is ReLU. It's a very simple function, if it receives any negative inputs, the function returns 0, and for the positive inputs it returns the actual value. Compared to the other activation functions ReLU doesn't activate all the neurons at the same time, which is computationally efficient. [machinelearningmastery.com, b]

Padding is a technique where zero value rows and columns are added on the edge of the matrix (image) in order to fit the center of the kernel on the extremities of the image. Without this technique the feature maps would shrink after each successive convolution operation. [Yamashita et al., 2018]

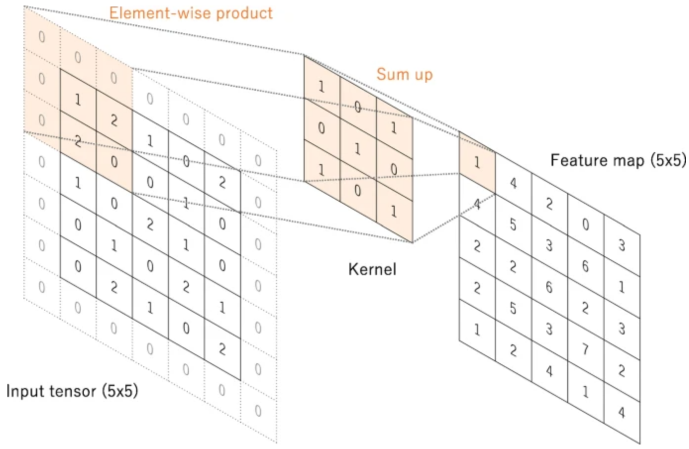

Figure 3.3: Convolution operation *Source: Yamashita et al. [2018]*

Another technique the algorithm is using to increase the performance is Pooling. Pooling is a downsampling operation and its main purpose is to reduce the size of the data. It works in a similar way as the convolutional operation, but the kernel returns either the maximum or the average value from the covered area of the image, discarding the rest of the values.[Yamashita et al., 2018]

Max pooling with a filter size of 2x2 and a stride of 2 is a very common operation which downsamples the image by a factor of 2. It acts as a noise suppressant discarding the unmeaningful data, speeding up the computation this way and reducing overfitting.[Yamashita et al., 2018]

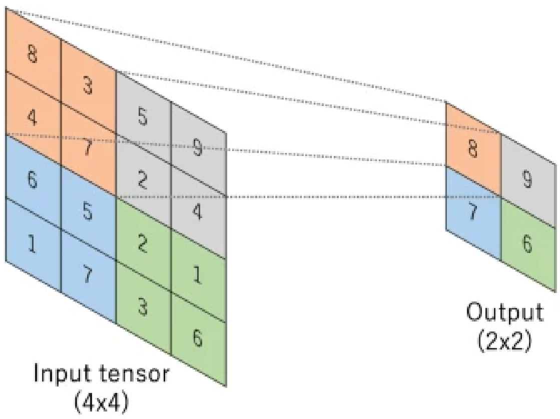

Figure 3.4: Pooling operation *Source: Yamashita et al. [2018]*

Then image is flattened which means that the bidimensional matrix, representing the image in a computer readable form, is converted to a one-dimensional matrix. Each numerical value in the matrix is represented by a neuron. For example, in case of grayscale images, if the image has a resolution of 64x64 pixels, there will be 4096 input neurons in the neural network [towardsdatascience.com, b].

3.1.3 Object detection

Computer Vision is the ability of a computer system to view, understand, and derive meaningful information from visual inputs like videos and digital images. [ibm.com]

Object detection is a computer vision technique used to identify objects and their physical location in images and videos. The output of an object detection model is a bounding box indicating the location of the object, a label specifying to which class it belongs to, and the confidence level of the predicted object. [deepomatic.com]

Object detection models are commonly classified in: top-down and bottom-up detectors. The top-down model works by proposing regions of interest, representing the areas on the image that could potentially contain an object, which are evaluated early in the detection process. These areas are then processed by a neural network that outputs the final bounding boxes and classes for each object alongside with an accuracy indicator. Since the detection is performed in two steps this approach is also known as two-stage detector. [Oksuz et al., 2020]

One-stage detectors work by eliminating the first step and the final prediction is derived straight from the anchors. First the input image is fed to a deep CNN for feature extraction which produces a set of anchors that is compared with the ground-truth or the labeled objects. Another neural network is used to classify the anchors and output the final results.[Oksuz et al., 2020]

There are two ways of training an object detection model:

- **Create an object detector from scratch** – it requires an extensive dataset of labeled images and a manual setup of the neural network which can be time-consuming. But once the network is properly configured the results can be remarkable.
- **Using a pretrained object detector** – it's a much faster and easier way to train a model using a pretrained neural network on a set of images. The model can be customized and fine-tuned for custom objects.[mathworks.com]

3.1.4 Single Shot Detector

The SSD algorithm is a one-stage detector, meaning that it passes only once through the neural network to predict the bounding boxes location and classes. This way it considerably speeds up the processing time being capable to detect objects in real time [machinethink.net].

A second factor that reduces the computational time is that it gets rid of the region proposal network. Other models like R-CNN work by generating region proposals, which basically are areas that potentially contain an object on an image, using a sliding window technique to propose Regions-Of-Interest [jonathan hui.medium.com].

The sliding window method is using a predefined, fixed size window that acts as a classifier on a small portion of an image. After each classification step the window is shifted to different positions on the image with a defined number of pixels or stride. [medium.com]

For each position on the image the algorithm performs a classification operation until the entire image is getting classified. The stride has an important impact on the performance of the algorithm, the smaller the stride is, the bigger the number of windows is to be classified, increasing the computation time. These ROI are then fed into a CNN for feature extraction. [medium.com]

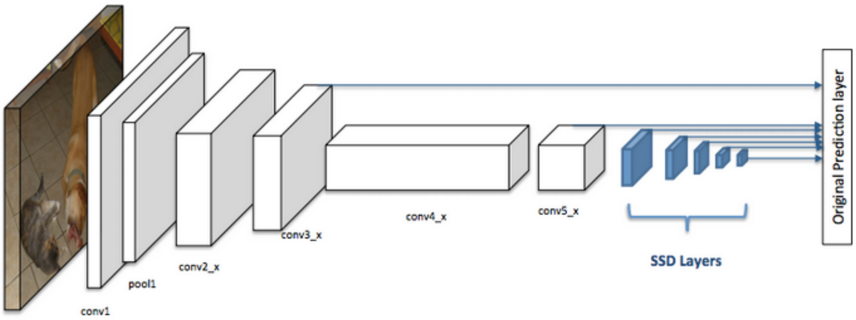

Figure 3.5: Single Shot Detector Architecture *Source: developers.arcgis.com [c]*

The SSD model has two main components: the SSD head and the backbone model. The role of the backbone model is to extract features and usually is a pre-trained neural network on a dataset like ImageNet, where the final fully connected layer was removed. On top of the backbone is added the SSD head, which is a convolutional layer that will make the final prediction.[developers.arcgis.com, c]

The image in SSD is divided in equal parts using a grid system and each cell will be responsible for detecting objects in that particular region. Each cell has pre-defined anchor boxes with a specific width, height and aspect ratio. A matching mechanism is used to pick the one that fits the object. Usually the one with the highest IOU score will be selected to the object's location. [developers.arcgis.com, c]

For evaluating the performance, the algorithm is using the intersection-over-union method, a very popular metric in used object detection, also known as Jaccard index. The underlying principle of this metric is to determine what pertcentage of the predicted bounding box is overlapping the ground-truth or the labeled data. The ideal value of IOU is 100% but anything over 50% is considered to be an countable prediction [giou.stanford.edu].

Another concept used in SSD models is the receptive field which is defined as the region of the input image that produces a feature [distill.pub]. Big objects in images can be hard to detect, receptive fields enables the SSD algorithm to detect objects at different scales.[developers.arcgis.com, c]

3.1.5 YOLOv3

YOLO is a state-of-the-art object detection algorithm that is using a pre-trained deep CNN to identify features in input images. To downsample the feature maps the model is using convolutional layers with a stride value of 2 and no pooling technique to prevent the loss of low-level features. [paperspace.com].

Compared to its previous releases YOLOv3 has an improved structure, is three times faster than SSD with the advantage of having the same accuracy levels. It improved the loss function by replacing the mean square error, instead of the Softmax function, is using logistic regression and the feature extractor network has been extended to 53 convolutional layers.[Ammar et al., 2019]

The model detects objects in four stages: divides the image using a grid, generates a probability map, generates the bounding boxes along with the confidence score and finally outputs the final object. [Ammar et al., 2019]

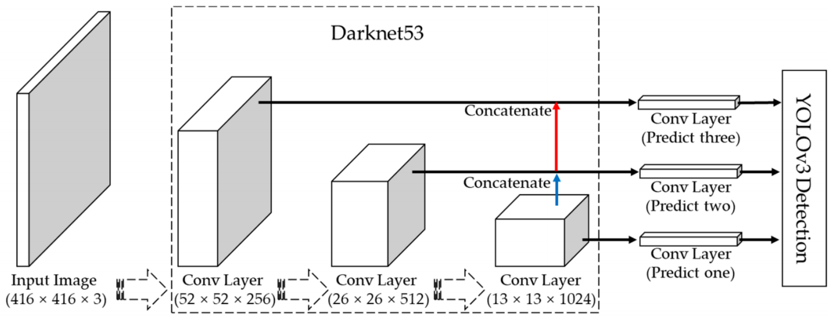

Figure 3.6: YOLOv3 Architecture *Source: Ma et al. [2019]*

The model architecture uses Darknet-53 with 53 convolutional layers as a feature extractor, with 1x1 and 3x3 convolution kernels. First, YOLO scales the original input image to 416x416 pixels. The image is then converted into three feature maps: 13x13, 26x26,52x52. For each cell on the feature map, the algorithm predicts three bounding boxes and based on the IOU score it selects the most suitable one to output the selected object [Ma et al., 2019].

3.1.6 RetinaNet

RetinaNet is a popular one-stage object detection model that works well with satellite imagery because is able to detect high density, small scale objects.[developers.arcgis.com, g]

The RetinaNet architecture has 4 major components:

- The Bottom-up pathway – it calculates the feature maps at different scales
- The Top-down pathway – upsamples the feature maps high pyramid levels merging the to-down and bottom-up layers with the same size
- Subnetwork classifier – for each anchor box it predicts the location and class of an object
- Subnetwork regression – makes a regression of the bounding box [developers.arcgis.com, g]

The backbone network of RetinaNet is Feature Pyramid Network, built on top of ResNet architecture, which is able to output feature maps with proportional sizes at multiple levels of the feature pyramid.[zenggyu.com]

Initially the feature pyramid was formed by subsampled images into lower resolutions, later on being replaced with the pyramidal structure of the feature maps in CNNs.[zenggyu.com]

In practice, the size of the objects in images can vary. The high-level feature maps cover large portions of an image and therefore is suitable to detect large objects. On the other hand, small feature maps are more suitable to detect smaller objects. These feature maps are used in an independent manner for making predictions and providing this way a better performance. [zenggyu.com]

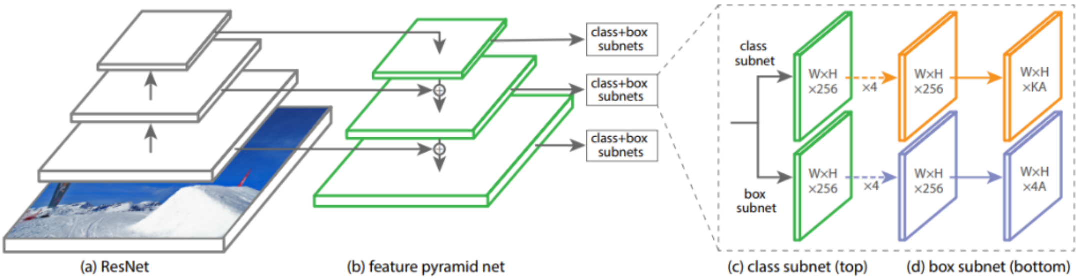

Figure 3.7: RetinaNet model architecture *Source: developers.arcgis.com [g]*

Single stage detectors come with class imbalances problems to be addressed. They are exposed to these imbalances due to the dense sampling of anchor boxes, which can overwhelm the model. Each layer in the pyramid network can have thousands of anchor boxes, most of them will be background classes and only a few of them will be assigned to the labeled objects in the dataset. RetinaNet model uses Focal loss to reduce the losses and correct the misclassified objects [developers.arcgis.com, g].

Focal loss assigns higher weights values on objects harder to classify (partial or noisy objects) and lower weights to objects that are easy to classify [Weng, 2018].

3.1.7 Faster R-CNN

Fast R-CNN is a multi-stage object detector and is an improved version of its previous releases, R-CNN and Fast R-CNN. [developers.arcgis.com, d]

R-CNN, the first version of the model, makes a selection of 2000 region proposals that are converted to feature maps by feeding them to a convolutional neural network. The feature maps are then classified using a Support Vector Machine model and then passed to a regression model to increase the precision of the bounding boxes around the object. [developers.arcgis.com, d]

But this model is very slow and Fast R-CNN came as an improved version of it. Fast R-CNN is feeding the entire image to a CNN, resulting in a combined set of feature maps for all regions of interest. The regions of interest are extracted using an algorithm similar to R-CNN and a pooling layer is used to resize all the feature maps to the same size. Then a regression model is used to output the bounding boxes with a higher precision and the probabilities for each class are determined using a Softmax function. This algorithm increases the computational speed of the algorithm but is not suitable for large datasets.[developers.arcgis.com, d]

Faster R-CNN improved its previous version of the model by introducing a Region Proposal Network (RPN) mechanism. The underlaying principle stays the same, feature maps are produced using a CNN which are then featured for region proposals by the RPN. [developers.arcgis.com, d]

A pooling layer is then used to resize the zones and fed to a fully connected layer, classified using a Softmax function and the bounding boxes are determined using a regressor. In this manner the algorithm speed is close to realtime. [developers.arcgis.com, d]

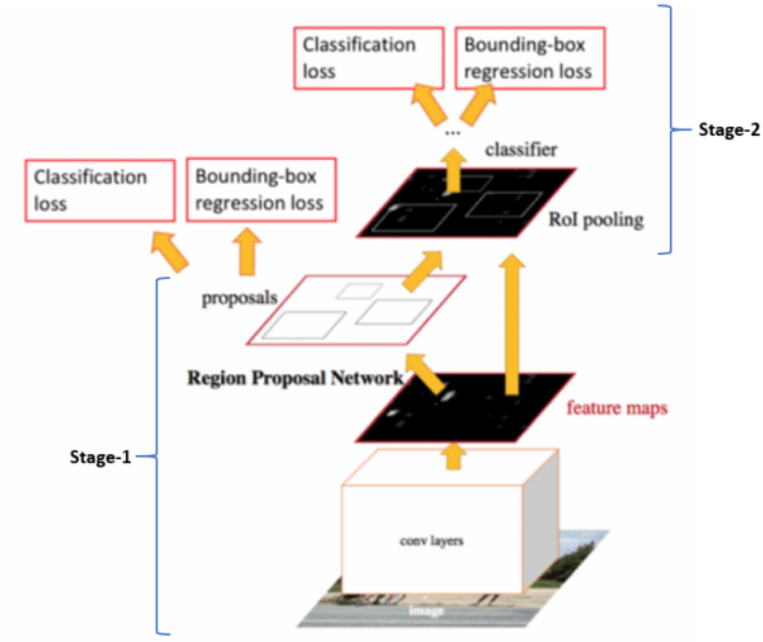

Figure 3.8: Faster R-CNN model architecture *Source: developers.arcgis.com [e]*

The RPN takes as an input an image and it returns a number of regions that potentially contain objects, alongside with a score measuring the probability of a region to have an object. The backbone of the model outputs a feature map used by the RPN to make region proposals.[developers.arcgis.com, e]

By using a sliding window technique, at every position of the window multiple anchor boxes with three different scales and aspect ratios are defined. Each anchor box represents a proposed area and is fed to a fully connected neural network to get the bounding box positions and confidence score. [developers.arcgis.com, e]

3.1.8 Mask R-CNN

Mask R-CNN combines pixel level classification with object detection algorithms to achieve a semantic segmentation of the detected objects.[developers.arcgis.com, e]

In spatial analysis, image segmentation provides solutions to problems that imply defining the extent of the objects detected. It has applicabilities in disaster assessment, building footprint extraction, self-driving cars etc. [developers.arcgis.com, e]

Mask R-CNN works in a similar way with Faster R-CNN, it is also a two-stage detector, but additionally it predicts segmentation masks for each region proposal. In the first stage the algorithm is scanning the input image and generates region proposals. In the second stage it classifies the region proposals, is generating bounding boxes and segmentation masks. [Kaiming He, 2020]

The backbone of the model is a pre-trained CNN, usually ResNet, used for feature extraction. In the first convolutional layers, the low-level features are detected, like corners and edges, followed by the high-level feature detection in the next layers, which are the actual output objects.[engineering.matterport.com]

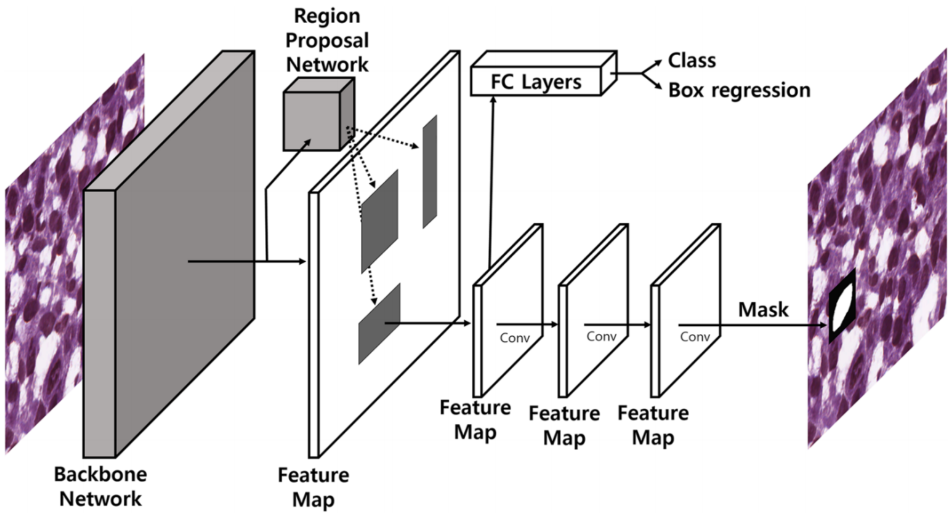

Figure 3.9: Mask R-CNN model architecture *Source: Jung et al. [2019]*

The Feature Pyramid Network improves the feature extraction mechanism by adding an extra pyramid that passes to lower layers the high-level features, allowing access to lower and high-level features on each pyramid level.[engineering.matterport.com]

The RPN is scanning the feature maps using a sliding window, resulting in a set of anchors distributed across the image. For each anchor the algorithm generates a foreground or background class and a foreground anchor.[engineering.matterport.com]

The anchors that are more likely to contain objects are then picked based on the IoU score, refines their size and location, generating the final region proposals that will be passed to the second stage. This process is also referred to as Non-maximum Suppression.[engineering.matterport.com]

In the next stage the proposed regions are classified, and the final bounding boxes are generated for each object. Here a ROI pooling layer is cropping and resizing a part of the feature map to handle variable feature input sizes, the feature map is sampled at differed points and a bilinear interpolation is performed.[engineering.matterport.com]

The segmentation masks are generated at low resolution(28x28 pixels) using a convolutional layer, and during inferencing are scaled up to the size of the bounding box, generating this way the final segmentation mask. [engineering.matterport.com]

4 Implementation

This chapter describes the overall process of implementing the project and it has two main parts, the object detection model and the pipe network design. In the object detection part, the cooling towers are detected using several object detection models, described in Chapter 3. The pipe network design part contains a detailed description of the workflow developed using Model Builder in ArcGIS Pro.

The detected cooling towers are used as an input for the model builder which defines the potential areas where district cooling projects can be implemented, identifies the buildings with a cooling demand located in the area and develops a pipe network that connects the buildings.

A dominant factor in determining the feasibility of a project in a certain area is the distance between the targeted buildings. How sparse or how dense the buildings must be in order for a project to become feasible? The cooling demand plays the main role in defining the extent of the area, the number of buildings included in the project and the density of the buildings in the area. The area of interest can be defined by a small number of buildings in need of cooling, located relatively far away from each other but with a very high cooling demand, like industrial buildings.

But there is also the situation where an area is defined by a big number of buildings with a high density and low cooling demands like residential buildings. The commercial buildings are placed somewhere in the middle between the industrial and residential buildings. Most of the times are targeted for district cooling projects since they have relatively high cooling demands and typically are located in commercial areas with a high number of buildings with the same profile.

Usually the cooling demand is determined based on the type of the building, size, destination and energy profile. The peak energy capacity is estimated based on an annual evaluation of energy consumption and is then adjusted based on one-hour peaks in the previous summer months. [districtenergy.com]

This method of estimating the cooling demand requires detailed data about buildings, that is not always freely available and also not necessary so early in the process. The purpose of the current model is to roughly estimate the cooling demand and make a preliminary feasibility analysis in the targeted areas, which doesn't require such accurate data.

The next step in the process after identifying the buildings and defining the extent of the project is to contact each building owner and gather detailed data about the buildings and the energy consumption. This data will be used to further determine a more precise cooling demand for each building and refine the preliminary estimations.

During this phase other buildings can also be included that were not initially identified with a cooling potential like public institutions and office buildings that don't have an air conditioner system or have it on the building facades or basement.

To estimate the cooling demand the current model is using the detected air conditioner towers, which have been classified into different classes based on the number of fans and a rough estimation of 50 kw cooling capacity was assigned to each fan. This estimation is a mean capacity value for each fan of different air conditioners manufacturers, and it was also reinforced by engineers working in the district cooling, as a general value used in their feasibility studies.[althermo.com]

As the manufacturer of the different cooling towers cannot be precisely determined, all cooling towers fall in the same category, the only criteria that differentiates them is the number of fans. As a result, a building with two cooling towers with 6 cooling fans each has a cooling demand of 600 kw.

Another factor that should be taken in consideration in the feasibility study is how expensive is to develop the pipe network. In district energy usually the initial investment is high, but the maintenance and consumption costs are low. Every client is responsible for the initial investments in the pipe network. The initial investment is then recovered over a period of time due to a reduction in the energy consumption costs.

This initial investment is determined again by the cooling demand and as a rule of thumb, from previous district cooling projects developed in Copenhagen, a value of 1 km of pipe for 500 kw of cooling demand for each building is considered to be feasible. Based on this assumption, for each building in need of cooling in the area of interest, a buffer with a 1km/500 kw ratio was defined.

At the buffer intersection, the highest number of overlapping buffers represents the most suitable area for developing a district cooling project. The total cooling demand for this area can be generated by summing up the cooling demand for each building.

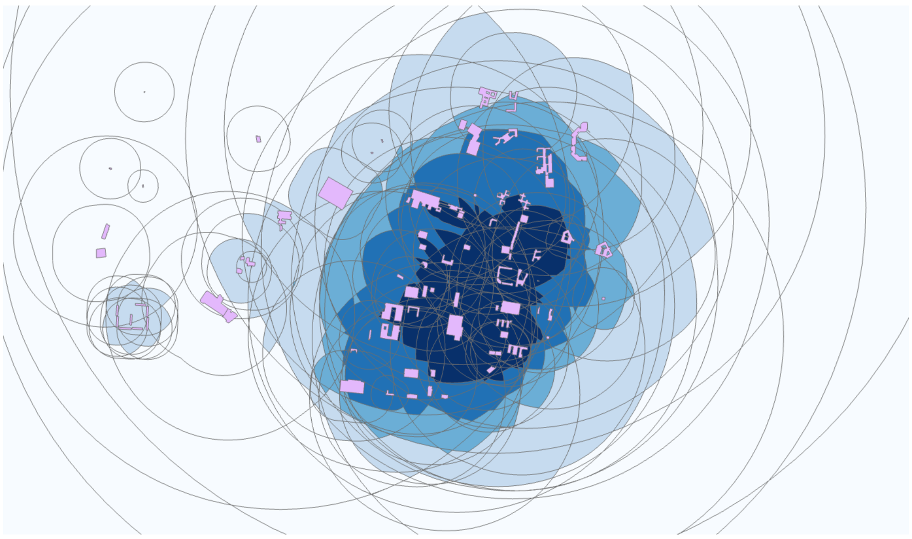

Figure 4.1: Defining potential area

4.1 Object detection

This section describes the process of detecting the cooling towers using several machine learning algorithms used in object detection. The purpose of using different object detection models is to determine which one performs better.

The process starts with creating the training dataset, which in object detection consists in a collection of images where the objects are manually identified and labeled. The labeled images are then exported in a machine-readable format that will be used by the machine learning algorithm to train the model.

Depending on the object detection model the data can be exported in several formats, but the most common formats are Pascal VOC, KITTI labels, MS COCO and RCNN masks. Pascal VOC is an XML based format where the labels and bounding boxes for each training image is generated. It is used in object detection models like Single Shot Detector, RetinaNet, Fast R-CNN, YOLOv3 and it has the following structure:

```xml
<?xml version="1.0"?>
<annotation>
    <filename>000000000.jpg</filename>
    <source>
        <annotation>ESRI ArcGIS Pro</annotation>
    </source>
    <size>
        <width>256</width>
        <height>256</height>
        <depth>3</depth>
    </size>
    <object>
        <name>212</name>
        <bndbox>
            <xmin>155.55</xmin>
            <ymin>138.11</ymin>
            <xmax>217.37</xmax>
            <ymax>229.36</ymax>
        </bndbox>
    </object>
</annotation>
```

The filename tag stores the name of the associated image and the size tag contains information about the size of the training chip and the number of band(in this case 3 for Red, Green and Blue). The object tag describes the labeled object, the coordinates of the bounding box in pixels and the class name.

To generate the training samples, Labeled Objects for Deep Learning tool was used in ArcGIS Pro, which facilitates the data collection process by automatically creating the training chips on the orthophoto images. Usually in object detection the labels are generated on standard images, but since orthophoto images are very big, small snapshot images of the objects are taken with predefined size, called training chips.

In the process of collecting the training sample, the cooling towers were grouped into 18 different classes based on the number fans and the number of rows the cooling fans were placed. For example, the class Model216 is a cooling tower with two rows and 16 ventilators. This categorization was necessary to make an estimation of the cooling capacity for each tower and also to make a better distinction between the cooling towers models.

The object detection models were trained using the Deep Learning module In ArcGIS Python API. Each model was trained for 25 and 50 training epochs using different object detection algorithms.

In total 1092 training samples were collected generating 3112 training samples, 2801 used for training the model and another 311 for validation. The number of samples can be artificially increased by specifying lower value for the stride. A single object can be present in several training chips, the stride value determines how many training chips will be created for a single or a group of objects.

The stride value is defined as the offset value in pixels when the next image chip is created. If the value is equal to half of the training chip size, there will be a 50% overlap between the generated training chips containing the same object and is applied for both X and Y directions. [pro.arcgis.com, a]

Training the model using ArcGIS deep learning module is similar for all object detection models. The next part will describe the scripts used to train the model using Mask RCNN algorithm. For the rest of the models the scripts will be provided in the project attachments as Jupyter Notebooks.

The deep learning is built on top of fastai and pytorch libraries, so it is required to load them before running the script.

```
#Importing the libraries
import arcgis
import arcpy
from arcgis.learn import *
import fastai
import torch
import PIL
import torchvision
from arcgis.learn import prepare_data
from arcgis.learn import MaskRCNN

#Define the project path
path = "C:/Master_thesis/Project/Experiments/6"
#Define the training chip size
chip_size = 256
#Number of batches
batch=4
#Prepare data
data = prepare_data(path=path, chip_size=chip_size, batch_size=batch)
```

The training chips are loaded in batches for processing optimisation, the batch variable stores the number of training chips loaded at once. If the GPU RAM memory is low the batch size must be selected accordingly, otherwise ArcGIS will throw a “CUDA out of memory error”. The batch size also defines how fast the model will process the data, smaller batch sizes may result in longer processing times.

Using prepare_data() the data is optimised for training by applying a set of transformations to the images like flipping, rotating and scaling them so the model is seeing a different image each time. Here the data is also divided into training and validation sets, data structures necessary to load the data into the model are created and memory optimization is performed.[developers.arcgis.com, f]

Another parameter that needs to be defined is the learning rate, which is defined as the step size when the model is trained and controls how much the weights are getting updated with each iteration. [machinelearningmastery.com, a]

Smaller learning rates may result in a higher number of training epoch while larger learning rates will result in fewer training epochs and shorter processing times, but an optimum value needs to be used for good performance. The deep learning module in ArcGIS is using the lr_find() function to determine the optimum value for the learning rates.[machinelearningmastery.com, a]

The training model is built using MaskRCNN() function that takes as an input the training data. The model tends to smooth out the mask boundaries for objects with irregular boundaries which might not be a very precise segmentation. For enhancing the model another neural network layer has been add for a point-based rendering. [developers.arcgis.com, e]

The fit() function is used for training the model and it takes as an input the number of epochs and other optional parameters like tensorboard which enables a more detailed view of the training parameters using Tensorflow.

```
# Single Shot Detector Model
model = MaskRCNN(data)

#Find the learning rate
ssd.lr_find()

#Train the model
model.fit(50, lr=6.309573444801929e-05, tensorboard=True)
```

Training the model for 25 epochs - Model characteristics

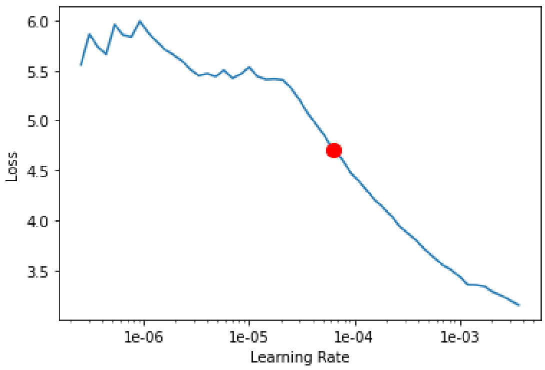

Figure 4.2: Learning rate - MaskRCNN 50 epochs

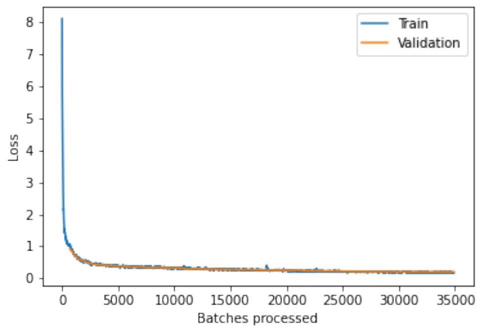

Figure 4.3: Training and validation loss - MaskRCNN 50 epochs

The model performance can be evaluated in two stages: after training the model and after running the object detection tool or the inferencing. During the model training a report is generated specifying the training and validation loss for each epoch. After the training, the results are presented in a graph showing the overall training and validation loss for the entire process.

A model definition file is also generated that contains model parameters, framework used, backbone model, inference function and the average precision score for each class.

Average precision score	
Cooling tower type	Score
Model3	1.0
Model11	0.8333333333333334
Model12	0.8048780487804879
Model13	0.81
Model14	0.6739130434782609
Model15	0.5833333333333334
Model16	0.8518518540594313
Model17	1.0
Model18	1.0
Model22	0.6111111111111112
Model24	0.6964285714285714
Model26	0.7838541669771075
Model28	0.8481481480929587
Model210	0.7631578947368421
Model212	0.6547619047619048
Model214	0.7307692307692307
Model216	0.8500000014901161
Model218	0.5

Table 4.1: Average precision score - Mask RCNN, 50 epochs

After running the inferencing, the model performance can be assessed using Compute Accuracy for Object Detection tool that will generate a table with detailed metrics. The tool works by comparing the ground truth data with the detected objects and the predictions can be true or false.

The model predictions are divided in two classes: positive and negative. For example, a positive class can be " Cooling Tower" and a negative class "No cooling tower". Also, when a prediction is correct is considered to be true and false when is incorrect.[pro.arcgis.com, b]

Considering these categories four possibilities emerge for each prediction compared to the ground truth data:

- **True positive** - the prediction indicates that there is a cooling tower, and is correct
- **False positive** - the prediction indicates that there is a cooling tower, and is incorrect
- **True negative** - the prediction indicates that there is no cooling tower, and is correct
- **False negative** - the prediction indicates that there is no cooling tower and is incorrect [pro.arcgis.com, b]

To determine if a prediction is true or false or false positive, the amount of overlap between the ground truth and predicted objects is calculated using Intersection over Union method. [pro.arcgis.com, b]

The accuracy table contains four different metrics: precision, recall, the F1 score and average precision.

Precision is the ratio between the true positive and the total amount of positive predictions. [pro.arcgis.com, b]

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.1}$$

Recall is defined as the ratio between the true positives and the total amount of detected objects. [pro.arcgis.com, b]

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.2}$$

The F1 score is an accuracy indicator calculated as a weighted average between recall and precision and It takes values between 0 and 1. An F1 score is considered to be good when there are small numbers of false positives and negatives. [pro.arcgis.com, b]

$$Precision = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Average precision indicator calculated across all values of the recall at a given Intersection over Union value. The table below shows the the accuracies for MaskRCNN model trined for 50 epochs.

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.9196	0.9433	0.9313	0.9014	183.0000	16.0000	11.0000
Model24	0.9167	0.8462	0.8800	0.8094	22.0000	2.0000	4.0000
Model28	1.0000	0.8667	0.9286	0.8667	13.0000	0.0000	2.0000
Model12	0.8140	0.9722	0.8861	0.8855	35.0000	8.0000	1.0000
Model13	0.9643	0.9643	0.9643	0.9643	27.0000	1.0000	1.0000
Model14	1.0000	0.9333	0.9655	0.9333	14.0000	0.0000	1.0000
Model212	1.0000	1.0000	1.0000	1.0000	4.0000	0.0000	0.0000
Model16	0.8947	1.0000	0.9444	0.8947	17.0000	2.0000	0.0000
Model15	1.0000	1.0000	1.0000	1.0000	3.0000	0.0000	0.0000
Model26	1.0000	0.9474	0.9730	0.9474	18.0000	0.0000	1.0000
Model214	0.8000	1.0000	0.8889	0.8000	4.0000	1.0000	0.0000
Model18	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model11	0.8889	1.0000	0.9412	0.8889	8.0000	1.0000	0.0000
Model218	1.0000	1.0000	1.0000	1.0000	11.0000	0.0000	0.0000
Model210	1.0000	1.0000	1.0000	1.0000	6.0000	0.0000	0.0000

Table 4.2: Accuracy metrics - Mask RCNN, 50 epochs

4.2 Pipe network design

The previous section describes the cooling towers detection process using several object detection models. This section will emphasize on spatial analysis and describe the workflow that will further develop the model.

The process of defining the potential area was developed using Model Builder in ArcGIS Pro 2.7.2. The input layers in the model are the detected air conditioner towers and another vector layer with the building footprint in the area of interest.

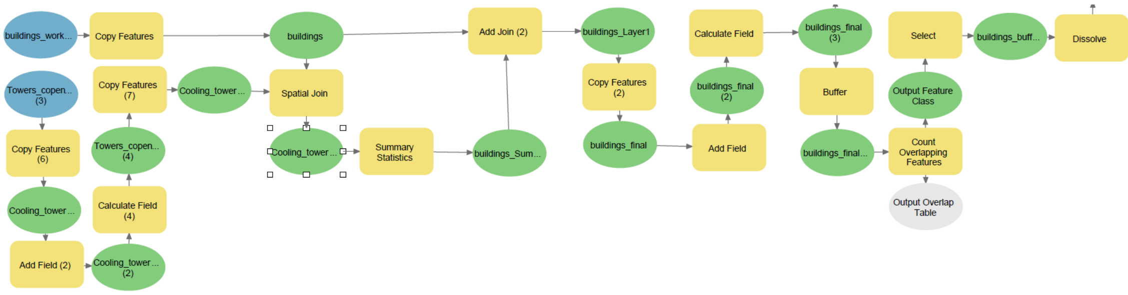

Figure 4.4: Defining potential areas model builder workflow

In the cooling towers layer, a new column in the attribute table will be added that will store the cooling demand based on the number of fans on each feature. The features in this layer have different classes based on the number of fans on each cooling tower. Using Calculate Field Geoprocessing tool, using the class name field each feature will have the cooling demand assigned.

A spatial join assigns the closest building to each cooling tower. The cooling towers can be located on top of the buildings or right next to a building. Using a spatial join, buildings from both situations will be selected. Considering that a building can have one or more cooling towers connected to the air conditioner system, the cooling demand has to be summed up for each building. Summary Statistics does it by summing up the cooling demand based on a unique identifier of each building.

Add Join geoprocessing tool merges the output table resulted by running the Summary Statistics tool with the layer containing the building footprints in the area of interest, based on a common identifier in the joining features. The output of this geoprocessing tool is a feature class that has the cooling demand for each building.

Add Field will add an empty field to the output feature class resulted by running the Add Join tool. This field will store the buffer radius for each building. As previously described the buffer radius represents the feasible amount of pipe used by a building owner to get connected to the district cooling network. The buffer radius is calculated using Calculate Field tool using the following formula:

$$buffer = \frac{demand}{500} * 1000 \tag{4.4}$$

The Buffer tool generates the buffers and using Count Overlapping Features tool, the number of overlaps between the buffers are summed up generating a feature class similar with the one presented in Figure 4.1

The final step of this part of the workflow consists in selecting the features with the highest amount of overlaps and running a Dissolve on the ObjectID field generates the final area of interest.

The second part of the workflow is to generate a preliminary pipe network that will connect all the buildings in need of cooling.

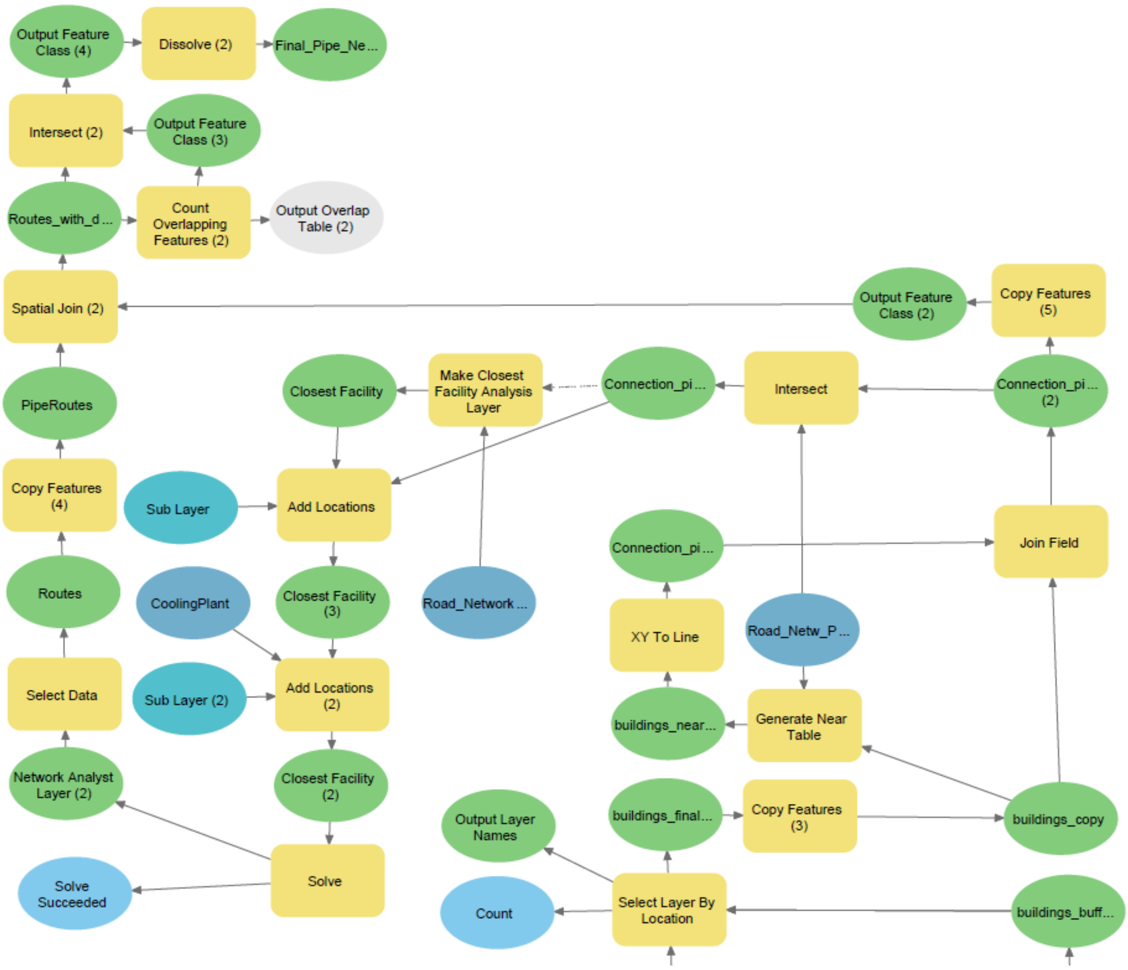

Figure 4.5: Pipe network model builder workflow

The workflow starts with a Select Layer By Location between the area of interest generated in the previous step and the buildings layer with the cooling demand assigned.

Using the features selected in the previous step and the road network layer, the closest road features to the buildings can be identified by running Generate Near Table tool. The generated table contains the Id's of the buildings and the closest road features, also the coordinates of the closest vertex from each feature class. This table can be used as an input for the XY To Line tool that will create a line from each building to the closest road feature representing the connection pipes in the district cooling pipe network.

Join Field will transfer the cooling demand from each building to every connection pipe. An intersection between the connection pipes and the road network will then generate a point feature class used as an input for the next geoprocessing tool.

The pipe network was created using the Network Analyst extension in ArcGIS Pro. This extension is usually used in transportation projects and it allows finding the most optimal route in terms of distance, travel time or costs between a set of geographical location.

Usually the main pipes in district cooling projects are developed along the street network and the tool can be successfully used in the current model since the roads are representing the main pipes.

The tool requires a set of location as an input that will be visited when creating the route. Another requirement for running the tool is the road network generated using Create Network Dataset tool on the road network layer. The input dataset needs to be optimal when creating the network, all the features representing road features must be connected, also each street intersection needs to have a vertex, otherwise it will not be represented as a road intersection. For example, an underground passage or a bridge crossing a street cannot be represented as intersections.

The destination points were generated in the model builder by making an intersection between the connection pipes and the road network. The point feature class are then added as an input to the Make Route Layer using Add Locations tool and the most optimal route can be generated.

Figure 4.6: Optimal route for the pipe network

The pipe network requires the cooling demand for a preliminary design of the pipe network. But using the Make Route Layer generates only one route that will visit each destination point. To create the pipe network, for each pipe a separate route that will connect the destination point with the staring point must be generated. The starting point represents the connection between the pipe network and the cooling plant facility.

Make Closest Facility Analysis tool generates separate routes for each destination point. This type of analysis is usually used for finding the closest facilities to a destination point and generates the most optimal route. For example, it can be used in finding the closest fire station from a fire incident or the closest supply warehouse to a store. In our case it will generate the most optimal routes from each destination point to the connection to the main pipe leading to the chilling plan facility.

The tool requires two inputs, the incident points, represented in this case by the point feature class generated on the intersection of the connection pipes and the street network. Also the facilities represented by the connection point with the main pipe connecting the chilling plant. The connection point needs to be located close to the center of the pipe network so the most optimal routes connecting the incident points can be generated.

If the connection point is located too far from the center of the pipe network, the algorithm might generate routes selecting different road segments based on the shortest travel distance resulting in inefficient and redundant routes. This situation is showcased in Figure 4.5.

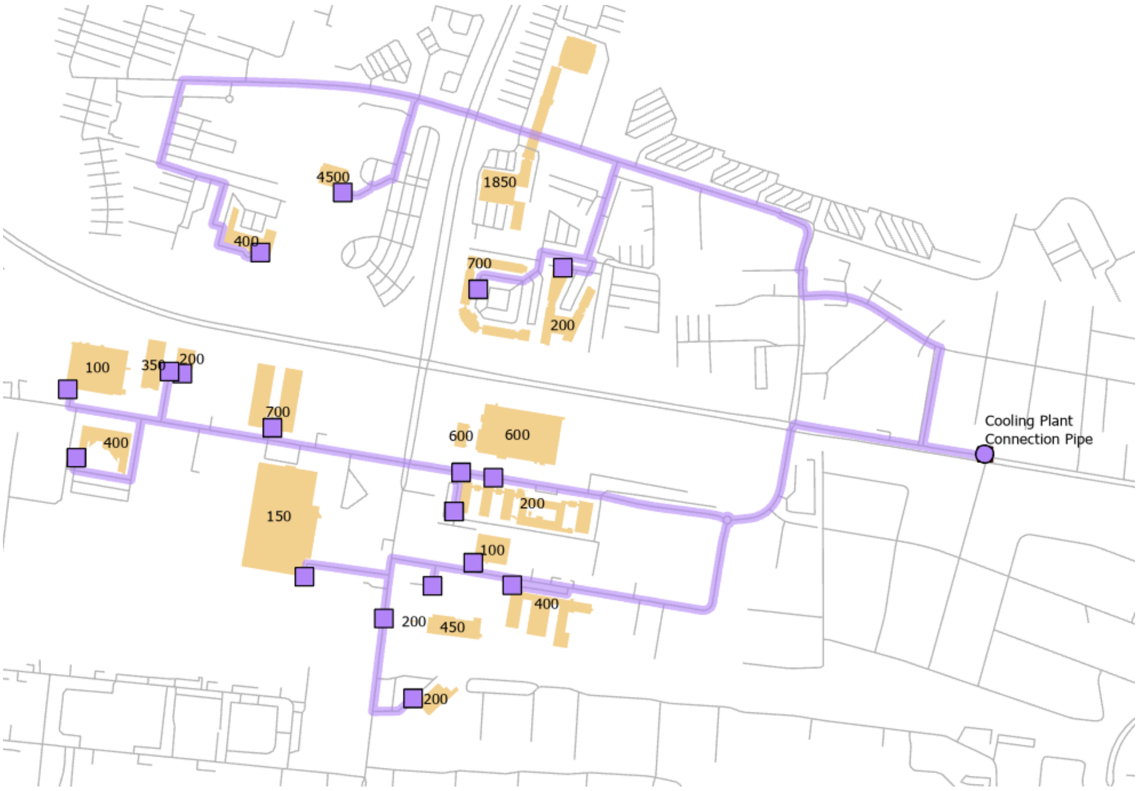

Figure 4.7: Redundant routes

When the routes have been successfully generated a spatial join between the routes and the connection pipes will assign the cooling demand to each route. The diameter of each pipe section is established based on the cooling demand, which is not the same for the entire route. Some of the pipes will overlap on their way to the connection point. For each segment where the routes overlap the cooling demand will be summed up so the pipe diameter can be correctly determined.

At this point in the model each route is represented as a single feature and it needs to be divided into different segments where the routes overlap. Then each overlapping segment must be joined into a single feature and the cooling demands summed up.

Count Overlapping Features will split every route at every intersection with the other routes. The output feature class will be then intersected with the routes with the cooling demand to pass it to each split segment. Using Dissolve all the overlapping features will be joined together into a single feature class and the cooling demand summed up. Figure 4.6 shows the final pipe network with the cooling demands for every building and each pipe segment.

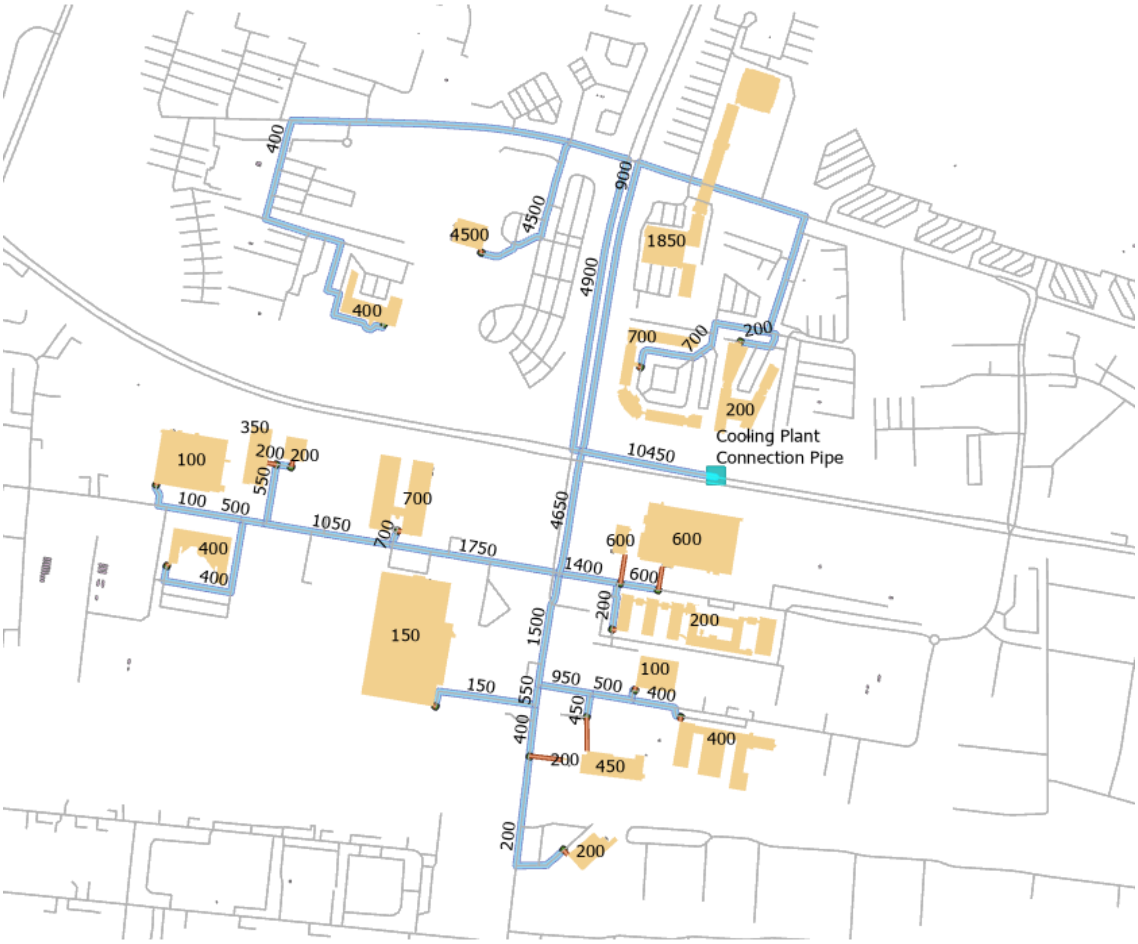

Figure 4.8: Final pipe network

Based on the cooling demand the pipe diameter can be determined using the values from Table 4.3. The table describes the Nominal Diameter(column D_n), the outside diameter of the pipe(column D_y) and the cooling demand for each type of pipe(column Demand)

Using Field Calculator the Nominal Diameter was assigned to each pipe segment depending on the cooling demand by inserting the following script in the Code Block field:

```
def pipeDimensioning(row):
    dimension = None
    if row >= 0 and row < 2:
        dimension = 'DN15'
    elif row >= 2 and row <5.3:
        dimension = 'DN20'
    elif row >= 5.3 and row < 9.9:
        dimension = 'DN25'
    elif row >= 9.9 and row < 19.3:
        dimension = 'DN32'
    elif row >= 19.3 and row < 28:
        dimension = 'DN40'
    elif row >= 28 and row < 53:
        dimension = 'DN50'
    elif row >= 53 and row < 105:
        dimension = 'DN65'
    elif row >= 105 and row < 159:
        dimension = 'DN80'
    elif row >= 159 and row < 319:
        dimension = 'DN100'
    elif row >= 319 and row < 560:
        dimension = 'DN125'
    elif row >= 560 and row < 920:
        dimension = 'DN150'
    elif row >= 920 and row < 1870:
        dimension = 'DN200'
    elif row >= 1870 and row < 3370:
        dimension = 'DN250'
    elif row >= 3370 and row < 5310:
        dimension = 'DN300'
    elif row >= 5310 and row < 6830:
        dimension = 'DN350'
    elif row >= 6830 and row < 9710:
        dimension = 'DN400'
    elif row >= 9710 and row < 13300:
        dimension = 'DN450'
    elif row >= 13300 and row < 17690:
        dimension = 'DN500'
    elif row >= 17690 and row < 28700:
        dimension = 'DN600'
    elif row >= 28700 and row < 42600:
        dimension = 'DN700'
    elif row >= 42600 and row < 61000:
        dimension = 'DN800'
    elif row >= 61000 and row < 78000:
        dimension = 'DN900'
    elif row >= 78000 and row <= 96000:
        dimension = 'DN500'
    return dimension
```

D_n mm	D_y mm	t mm	D_i mm	Velocity m/s	Flow l/s	Flow m^3	Demand MW
DN15	21.3	2	17.3	0.41	0.096	0.347	0.002
DN20	26.9	2	22.9	0.51	0.21	0.755	0.0053
DN25	33.7	2.3	29.1	0.59	0.395	1.422	0.0099
DN32	42.4	2.6	37.2	0.71	0.769	2.77	0.0193
DN40	48.3	2.6	43.1	0.78	1.135	4.09	0.028
DN50	60.3	2.9	54.5	0.91	2.112	7.6	0.053
DN65	76.1	2.9	70.3	1.07	4.17	15.02	0.105
DN80	88.9	3.2	82.5	1.19	6.35	22.9	0.159
DN100	114.3	3.6	107.1	1.41	12.74	45.9	0.319
DN125	139.7	3.6	132.5	1.61	22.2	80	0.56
DN150	168.3	4	160.3	1.82	36.8	132.6	0.92
DN200	219.1	4.5	210.1	2.15	74.5	268.3	1.87
DN250	273	5	263	2.47	134	484	3.37
DN300	323.9	5.6	312.7	2.76	212	763	5.31
DN350	355.6	5.6	344.4	2.93	273	982	6.83
DN400	406.4	6.3	393.8	3.18	388	1395	9.71
DN450	457	6.3	444.4	3.42	531	1911	13.3
DN500	508	6.3	495.4	3.66	706	2542	17.69
DN600	610	7.1	595.8	4.1	1144	4117	28.7
DN700	711.2	8.8	693.6	4.5	1699	6118	42.6
DN800	812.8	8.8	795.2	4.89	2430	8750	61
DN900	914.4	10	894.4	4.95	3110	11197	78
DN1000	1016	11	994	4.95	3842	13829	96

Table 4.3: Pipe dimensions

A preliminary analysis on the total cost of implementing the project can be made based on the total amount of pipe for each diameter. The pipes are manufactured in segments of 6, 12 an 16 meters, dividing the total length of the pipe network to the length of the manufactured pipe segment will result in the total number of pipes that will be used to develop the network. [Logstor].

Results 5

The following chapter presents the results achieved in the implementation phase. The output results of each object detection model will be analyzed alongside with accuracy indicators and performance metrics. The purpose of the analysis is to determine which object detection algorithm performed best, what are the drawbacks and the advantages of each model.

An analysis of the district cooling pipe network model is also included in this chapter, advantages and potential issues or disadvantages will be addressed.

5.1 Cooling towers detection

Single Shot Detector

Overall the model didn't performed well. When trained for 25 epochs the model had very bad results, it predicted 28 objects for confidence levels higher than 50 percent. Not all the predictions were cooling towers, it also found objects that are not even similar to a cooling tower. The predicted classes were mixed up and if the cooling towers were grouped, it found only a single tower.

When the model was trained for 50 epoch the results were still unreliable, it detected 78 objects, not all of them cooling towers. For low confidence levels the classes were mixed up and different other objects were detected. On confidence levels higher than 75%, the class predictions started to match the ground truth data , but groups of cooling towers were classified as a single entity or it found only one cooling tower in locations where there were many.

Training the algorithm with 100 epochs didn't make a major difference taking in consideration that the number of training epochs was double. It found slightly more towers but it was still very unreliable and it couldn't be used.

In previous experiments the model was trained using the same algorithm but the cooling towers were not categorized into different classes. The training dataset contained 943 training samples that generated 2737 training chips, all falling in a single class. The model performed better and it identified a higher number of cooling towers positioned relatively close to each other, which is a positive aspect.

One advantage of this model is that it trains fast, for each epoch took less than 3 minutes and is lightweight, being able to load batches of 64 training chips at a time, compared to the other models that are not able to handle such high amounts. But it couldn't be used in the current model because is very unreliable and it has very bad results.

The accuracies and performance metrics can be found in Appendix A.

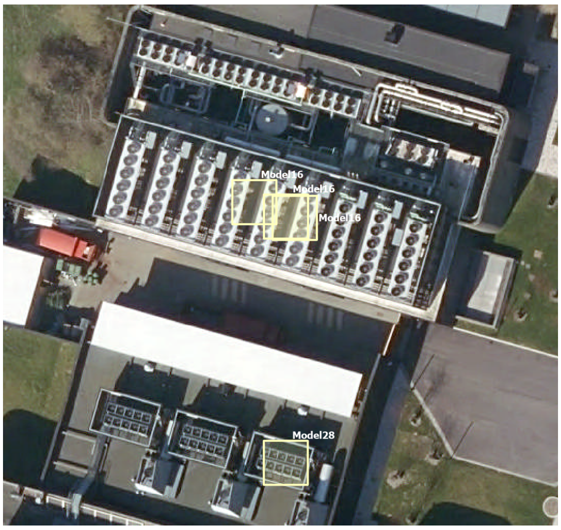

Figure 5.1: Detected cooling towers using SSD

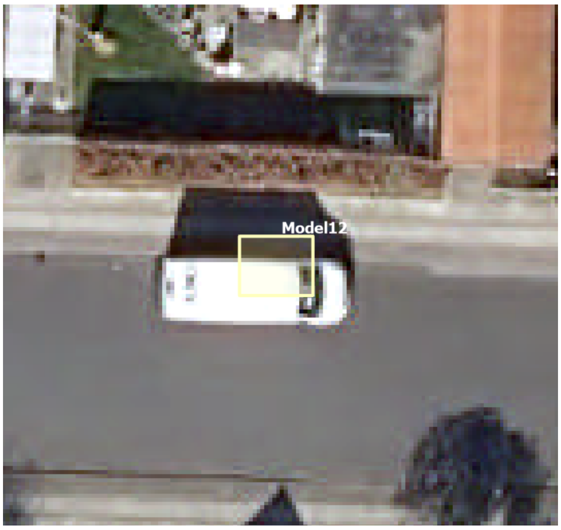

Figure 5.2: Error in object detection

RetinaNet

The model performed slightly better compared with the Single Shot Detector. When trained for 25 epochs. For small confidence levels (50-75) it was able to detect some cooling towers but it wasn't able to classify them correctly. For confidence levels higher than 75%, the detected towers had the correct classes, however the number of towers were very small compared with the ground truth.

Better results were achieved when the model was trained with 50 epochs, resulting in a reduction in the number of misinterpreted objects. Most of the objects were cooling towers but the object class names were not matching the ground truth data.

The model couldn't be used due to the fact the fact that not all the towers were detected and the classes were not correctly interpreted leading to high errors in the cooling demand estimations.

The accuracies and performance metrics can be found in Appendix B.

YOLOv3

Training the model for 25 epochs outputs relatively good results in finding cooling towers, however is not very precise in classifying them. The detected towers have very low confidence levels, are mixed up with other similar objects, and very few of them had confidence levels higher then 80%, but even then it couldn't predict the correct classes.

Training the model for 50 epochs generated better results, for confidence levels higher than 80%, 50 cooling towers were detected and classified correctly compared to 7 in the previous case. For confidence levels between 50% - 80% the towers were mixed up with other objects, but the vast majority of them had the correct classes. For confidence levels less than 50%, lots of cooling towers were detected and a big number of them had the correct classes, but they were mixed up with different other objects with the same confidence levels and they couldn't be filtered out.

The accuracies and the training and validation loss can be found in Appendix C.

Fast RCNN

The model performs good at detecting cooling towers. When was trained for 25 epochs the model performed good, it found the majority of the cooling towers, most of them were classified correctly. The problem in using it is that the cooling towers are mixed up with other objects even on high confidence levels and they can't be filtered out.

When trained for 50 epochs the model performed much better resulting in fewer objects detected but it was more accurate, filtering out the unwanted objects. More than 50% of the detected objects had accuracy levels higher than 85% and these objects were correctly classified, and they were not mixed up with other irrelevant objects. For lower accuracy levels the cooling towers were mixed up also with other objects.

Compared with all the other object detection models used up to this point Fast RCNN performed the best but is still unreliable and it requires cleaning the data manually. Probably increasing the number of training samples, also the training time would increase the accuracy of the model but is computationally expensive. To train the model, for each training epoch it took about 15 minutes, much more compared to the other models, and with bigger training data sets also the training times increase.

A drawback of using this algorithm is that when the cooling towers are located close to each other, the model doesn't detect all of them. Another issue is that the cooling towers are not always vertically or horizontally oriented. Sometimes are oriented on a 45 degree angle and it was noticed that in these situations the algorithm doesn't perform very well.

The model is also computationally expensive and it might take longer times to train the model. One of the ways to increase the model performance is collecting high amounts of training samples, covering as many different situations as possible, like towers with different orientation, situated in the shadow of the buildings or covered by objects. Depending on the size of the training set the training times might increase significantly.

The accuracies and the training and validation loss can be found in Appendix D.

Figure 5.3: Detected cooling towers using Fast RCNN model

Mask RCNN

Mask RCNN had the best performance in between all the models. It is very accurate not only in detecting the cooling towers but also in classifying them.

In the first place the model was trained with only 15 epochs and it had very good outputs detecting 155 towers from 194, and the majority of the towers have been classified correctly. But still 44 other objects were predicted as cooling towers, most of them having similar patterns to a cooling tower.

Increasing the number of training epochs had even better results, detecting 183 cooling towers and only 16 other objects. The cooling towers have been correctly classified with very high confidence levels, above 90% and the vast majority having around 99%. The false positives can be easily identified and removed since they have very low confidence levels.

Training the model takes very long times, more than 30 minutes for a single epoch, but taking in consideration the high accuracies the objects were classified for relatively low number of training epochs is very convenient. The process of labeling the training data is time consuming and very tedious. Using the Mask RCNN model, the time for data collection is considerably reduced since it has very good results with a relatively small number of training samples.

Due the fact that the model is classifying the objects very accurate and the misclassified objects can be easily filtered out, it is very reliable. Another positive aspect of the model is that it performed well in detecting and classifying objects located close to each other, which other object detection models couldn't.

The accuracies and the training and validation loss can be found in Appendix E.

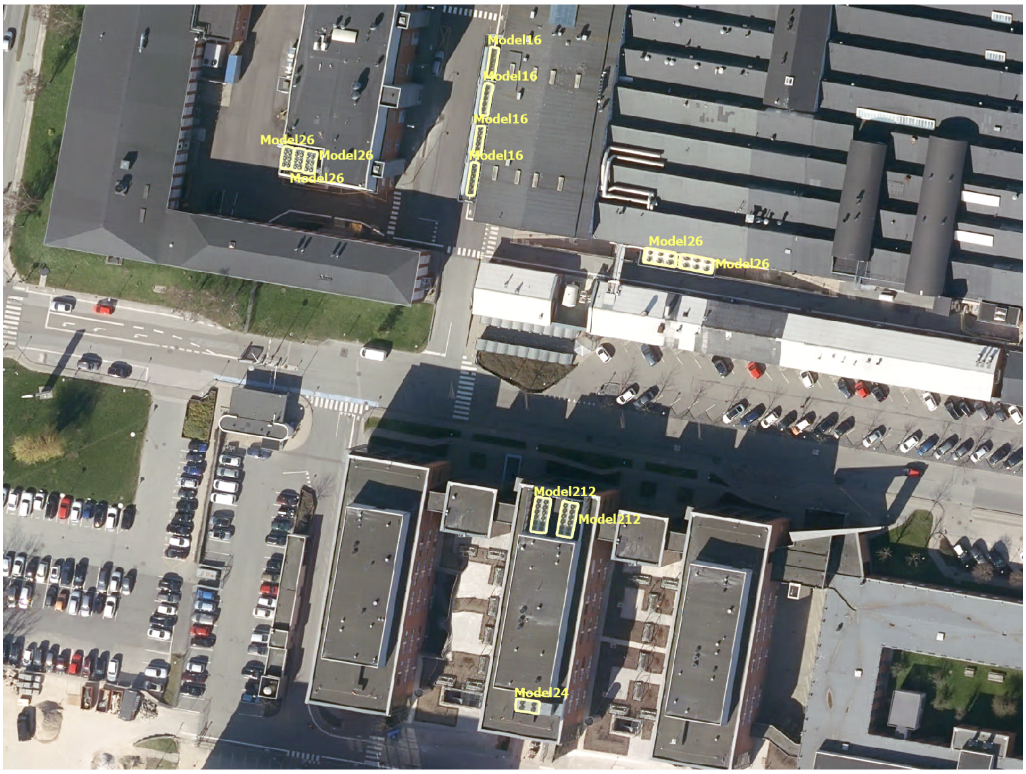

Figure 5.4: Detected cooling towers using Mask RCNN model

5.2 Pipe infrastructure design

Although the model for designing the pipe network aims for a fully automated process there still is some degree of manual work involved. The connection with the main pipe leading to the district cooling plant needs to be carefully chosen, otherwise it can result in redundant and inefficient routes.

An important role in designing the routes it plays the street network. The roads are represented in a dataset with a center line, but bigger roads with more than one lane for each driving direction are represented with multiple lines, one for each direction.

For designing the pipe infrastructure, the road network is used because in most of the cases the district cooling pipes are following the street infrastructure. When the algorithm is calculating the routes, it might choose two different centerlines which in reality belong to the same road, resulting in redundant routes. Cleaning up the data can solve this issue but for large areas can be time consuming.

Depending on the cooling demand the potential district cooling areas can be divided in three main categories with high, medium and low demands. Once a district cooling project is developed, it can easily expand to the other areas.

The pipe dimensions calculated in the model is just a rough estimation of the total costs involved, there are several hydraulic parameters that need to be considered when the pipe infrastructure is designed, but these calculations are made in the next phase of the project.

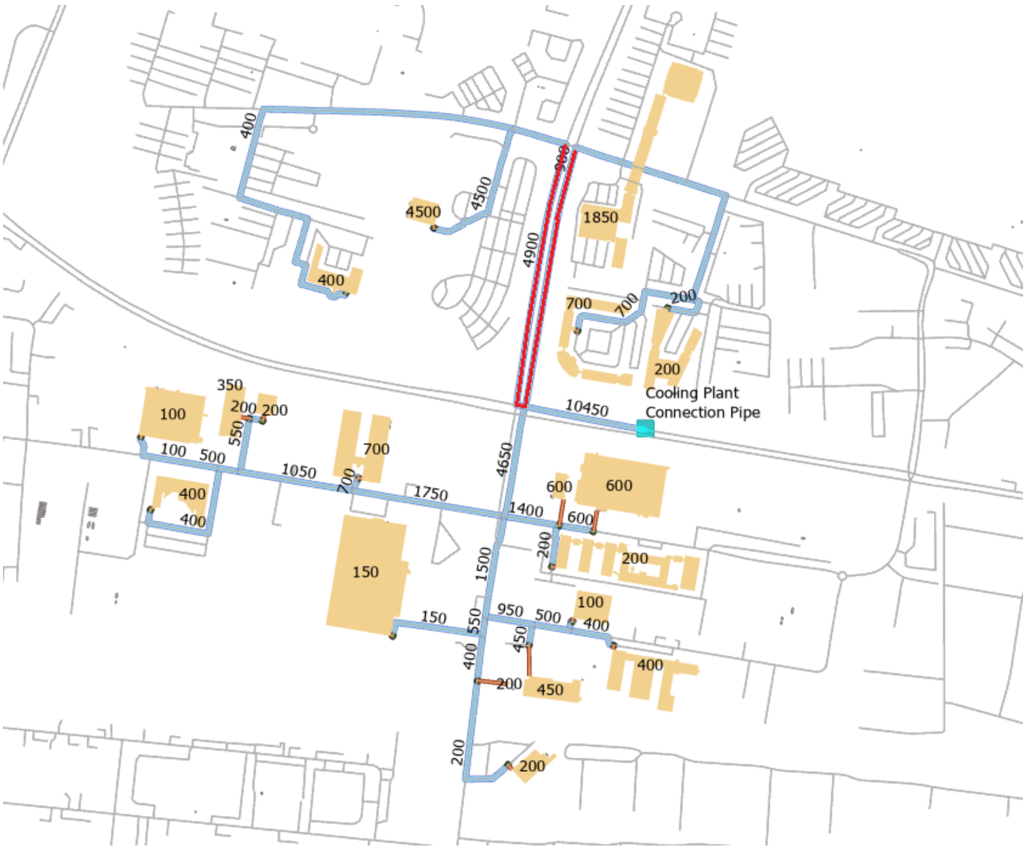

Figure 5.5: Redundant route

Discussion 6

This chapter will provide the answer for the research questions and different approaches in developing the model will be discussed.

1. How district cooling potential can be estimated using object detection models?

Using object detection, the cooling towers could by located and classified in satellite images. The accuracy the cooling towers were identified depends on the image resolution, what object detection model was used, the number and quality of training samples.

Estimating the cooling potential was based on the number and density of the buildings with a cooling demand in an area. The cooling demand for each building was estimated by summing up the cooling capacity of each tower located on top or next to a building. Based on the cooling demand a potential area was defined and the cooling potential determined.

2. Which object detection model performs better in identifying objects on aerial images?

In the development phase of the project 5 different object detection models were used on a testing area with the same training dataset, each model generating different results. Mask RCNN had the best results in detecting the cooling towers with high accuracy and classifying them correctly.

Fast RCNN also generated good results for the objects detected with a high accuracy but not all cooling towers fall in this category. For lower accuracies other objects were detected as cooling towers and it wasn't possible to filter them out.

RetinaNet, Single Shot Detector and YOLOv3 generated unreliable results and they couldn't be used.

3. How process automation can be used in assessing the cooling potential?

Using the model builder in ArcGIS Pro the workflow for the cooling potential assessment could be automated. Using the trained model developed in the object detection phase can automatically detect the cooling towers in the area of interest and used as an input in the model builder along with the road network and the buildings layer to identify the potential areas and estimate the cooling potential.

The model also generates the pipe infrastructure in the area that can be used to make a rough estimation of the total costs involved in developing a district cooling project in the area.

Other approaches in model development

Several approaches were considered in developing the model. Initially the cooling towers were detected without dividing them into different categories. The cooling demand would be then estimated using the volume of the buildings.

The volume of the buildings was determined using a Digital Elevation Model(DEM) and a Digital Surface Model(DSM) and using raster calculator the DEM was subtracted from the DSM generating a raster containing the height of the objects on the terrain surface.

Using the output raster, the volume of each building could be estimated. But the cooling demands require also other parameters be taken into account and other approaches had to be considered.

In another experiment the individual ventilators on the cooling towers were used as training samples. The model was trained using Mask RCNN and 2445 samples were collected generating 1102 training chips. Considering the simple patterns in the training set the model performed relatively good, but not all the ventilators could be detected, and similar objects were also classified as ventilators.

Figure 6.1: Single ventilator detection

7 Future Development

For evaluating the cooling demand detailed information about the buildings is required like the energy profile, constructions materials, size and destination that are not always available. [districtenergy.com] The advantage of using aerial images in assessing the cooling demand has the advantage of availability. Orthophoto images are easy to find and cheap, sometimes even free.

Therefore, the model can be scaled up and used on a global level. The only difference between different countries is the cooling tower manufacturer and capacity. Depending on the area, different sets of training samples must be collected for training the model. But if the similarities between the cooling towers in different areas can be identified, for each zone the model can be replicated for each training set.

The road infrastructure provided by Open Street Maps can be used for developing the pipe network. Some level of data preprocessing might be necessary to create the network dataset, automating this process would greatly improve the model.

The model has the potential of being further developed into a web application. The object detection model and the model builder can be published on an ArcGIS Enterprise Portal and integrated into a web map. However, the drawback of turning it into a web app is the long time it takes for inferencing. A potential solution would be to isolate the inferencing component and use the detected cooling towers as an input layer for assessing the potential areas.

Another component that can be developed is identifying potential areas for the cooling plant placement. Several factors could influence the location of the cooling plant facility like the distance from the area of interest, the proximity to lakes, rivers and sea, the property prices and other potential areas with cooling demand.

A multi-criteria analysis can support in making these decisions and the most suitable areas can be mapped. A significant weight would have the distance from the free cooling resources, as they can significantly reduce the energy costs and are sustainable alternatives for the conventional cooling methods.

8 Conclusion

The aim of this thesis is to develop an automated model for assessing district cooling potential using computer vision. Several experiments were conducted using different object detection models, the performance was observed and analyzed in order to choose the most suitable one to meet the project objectives.

An important aspect in evaluating the models was using the same training datasets and testing area. Another criterion in assessing the models was the accuracy metrics generated by comparing the results with the ground truth. It was noticed that models performed differently in the same conditions generating very different results.

In the analysis it was observed that the machine learning algorithm performed better in identifying isolated cooling towers compared with the situations where the towers were grouped together.

Another issue was to filter out the noise, or the other objects classified as cooling towers. At low confidence levels it wasn't possible to make a clear distinction between the misclassified objects and the cooling towers based on accuracy. However, the Mask RCNN model overcame these issues, it was able to generate masks for each individual tower. The misclassified objects had very low confidence levels and could be easily excluded.

The number and diversity of the training samples had a significant weight in getting better and more accurate predictions. Increasing the number of training epochs also generated better results up to a point but in some situations it doesn't necessarily help and it can have the opposite effect. The model can be overtrained, and it can't generalize patterns, resulting in inaccurate predictions.

Using the ArcGIS Python API was easy to use and it had good result, but it has its limitations. Having a better control over the machine learning algorithms might generate better results.

Using the model builder, the process of defining the cooling potential and building the pipe infrastructure could be automated, but there still is some manual work involved. Choosing the best location for the connection with the main pipe leading to the district cooling plant is necessary. Also the road network needs to be adapted for the current model to avoid redundant routes.

Although is not fully automated the model has a great potential, it can be further developed and turned into a fully functional application. An advantage of this model is the small number of variables taken into account for assessing the cooling potential. The cooling towers capacity can provide a good estimation of the cooling demands for early stages in district cooling projects.

Bibliography

ADB, 01 2017. ADB. *District Cooling in the People's Republic of China*, 2017. [ONLINE] Available at: `https://www.adb.org/sites/default/files/publication/222626/district-cooling-prc.pdf`.

althermo.com. althermo.com. *Cooling Demand Explanation.* [ONLINE] Available at: `https://www.althermo.com/products/dmi/`.

Ammar et al., 2019. Adel Ammar, Anis Koubaa, Mohanned Ahmed and Abdulrahman Saad. *Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3.* doi: 10.20944/preprints201910.0195.v1.

Bennett, 2017. Lauren Bennett. *Deforestation and climate change.* A publication of climate institute, 1400.

calderdale.gov.uk. calderdale.gov.uk. *Heating technologies in District Heating.* [ONLINE] Available at: `https://www.calderdale.gov.uk/v2/residents/environment-planning-and-building/sustainability/district-heat`.

Calderoni et al., May 2019. Marco Calderoni, Sophie Dourlens-Qaranta, Bishnu B. Sreekumar, Zia Lennard, Miika Rämä, Krzysztof Klobut, Zao Wang, Xiaojian Duan, Yin Zhang, Joakim Nilsson and Lars Hargö. *Sustainable District Cooling Guidelines.* 2019. [ONLINE] Available at: `https://cris.vtt.fi/en/publications/sustainable-district-cooling-guidelines-iea-dhcchp-annex-xii-repo`.

CEWEP, 12 2019. CEWEP. *District Heating. Consultation to Inform a Policy Framework for the Development of District Heating in Ireland.* [ONLINE] Available at: `https://assets.gov.ie/77662/207f6ccf-b496-41bc-99ea-59cbffa559d9.pdf`.

coursera. coursera. *What is Machine Learning? A Definition.* [ONLINE] Available at: `https://www.coursera.org/learn/machine-learning`.

COWI, 07 2015. COWI. *Booklet 3 - District Cooling*, 2015. [ONLINE] Available at: `http://basrec.net/wp-content/uploads/2015/01/Booklet-3-District-Cooling.pdf`.

Danfoss. Danfoss. *District energy: the way towards energy efficiency and climate change mitigation.* [ONLINE] Available at: `https://www.danfoss.com/en/about-danfoss/articles/dhs/district-energy-the-way-towards-energy-efficiency-and-climate-change-mitigation/`.

danfoss.com. danfoss.com. *How does district heating work?* [ONLINE] Available at: `https://www.danfoss.com/en-in/about-danfoss/our-businesses/heating/knowledge-center/heating-school/how-does-district-heating-work/`.

deepomatic.com. deepomatic.com. *What is object recognition and how you can use it.* [ONLINE] Available at: `https://deepomatic.com/en/what-is-object-recognition-and-how-you-can-use-it`.

developers.arcgis.com, a. developers.arcgis.com. *Overview of the ArcGIS API for Python.* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/overview-of-the-arcgis-api-for-python/`.

developers.arcgis.com, b. developers.arcgis.com. *ArcGIS API For Python.* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/install-and-set-up/`.

developers.arcgis.com, c. developers.arcgis.com. *How single-shot detector (SSD) works?* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/how-ssd-works/`.

developers.arcgis.com, d. developers.arcgis.com. *Faster R-CNN Object Detector.* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/faster-rcnn-object-detector/`.

developers.arcgis.com, e. developers.arcgis.com. *How Mask R-CNN Works?* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/how-maskrcnn-works/`.

developers.arcgis.com, f. developers.arcgis.com. *Object Detection Workflow with arcgis.learn.* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/object-detection/`.

developers.arcgis.com, g. developers.arcgis.com. *How RetinaNet works?* [ONLINE] Available at: `https://developers.arcgis.com/python/guide/how-retinanet-works/`.

distill.pub. distill.pub. *Computing Receptive Fields of Convolutional Neural Networks.* [ONLINE] Available at: `https://distill.pub/2019/computing-receptive-fields/`.

districtenergy.com. districtenergy.com. *Cooling Demand Explanation.* [ONLINE] Available at: `http://www.districtenergy.com/wp-content/uploads/2016/09/CDEFY17.pdf`.

districtenergyinitiative.org, 12 2018. districtenergyinitiative.org. *RESULTS FROM A TECHNo-ECONOMIC ASSESSMENT OF THE POTENTIAL FOR DISTRICT COOLING IN THANE, INDIA*, 2018. [ONLINE] Available at: `https://www.districtenergyinitiative.org/sites/default/files/Thane%20District%20Cooling%20Pre-Feasibility%20Studies_0.pdf`.

EESI. EESI. *Climate Change.* [ONLINE] Available at: `https://www.eesi.org/topics/climate-change/description`.

engineering.matterport.com. engineering.matterport.com. *Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow.* [ONLINE] Available at: `https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46`.

EPA. EPA. *What Is CHP?* [ONLINE] Available at: `https://www.epa.gov/chp/what-chp/`.

Eurostat. Eurostat. *Share of energy products in total energy available.* [ONLINE] Available at: `https://ec.europa.eu/eurostat/cache/infographs/energy/bloc-2a.html`.

expertAI. expertAI. *What is Machine Learning? A Definition.* [ONLINE] Available at: `https://www.expert.ai/blog/machine-learning-definition/`.

Gang et al., 2016. Wenjie Gang, Shengwei Wang, Fu Xiao and Dian ce Gao. *District cooling systems: Technology integration, system optimization, challenges and opportunities for applications.* Renewable and Sustainable Energy Reviews, 53, 253–264. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2015.08.051. [ONLINE] Available at: `https://www.sciencedirect.com/science/article/pii/S1364032115009144`.

giou.stanford.edu. giou.stanford.edu. *Generalized Intersection over Union.* [ONLINE] Available at: `https://giou.stanford.edu/`.

GRIMMOND, 2007. SUE GRIMMOND. *Urbanization and global environmental change: local effects of urban warming.* The Geographical Journal, 173(1), 83–88. doi: https://doi.org/10.1111/j.1475-4959.2007.232_3.x. [ONLINE] Available at: `https://rgs-ibg.onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-4959.2007.232_3.x`.

hackernoon.com. hackernoon.com. *Neural Networks.* [ONLINE] Available at: `https://hackernoon.com/everything-you-need-to-know-about-neural-networks-8988c3ee4491`.

https://jupyter.org/. https://jupyter.org/. *Project Jupyter.* [ONLINE] Available at: `https://jupyter.org/`.

IBMCloudEducation. IBMCloudEducation. *Neural Networks.* [ONLINE] Available at: `https://www.ibm.com/cloud/learn/neural-networks`.

ibm.com. ibm.com. *Computer vision.* [ONLINE] Available at: `https://www.ibm.com/topics/computer-vision`.

jonathan hui.medium.com. jonathan hui.medium.com. *SSD object detection: Single Shot MultiBox Detector for real-time processing.* [ONLINE] Available at: `https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac`

Jung et al., 10 2019. Hwejin Jung, Bilal Lodhi and Jaewoo Kang. *An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images.* BMC Biomedical Engineering, 1. doi: 10.1186/s42490-019-0026-8.

Kaiming He, Feb 2020. Piotr Dollar Ross Girshick Kaiming He, Georgia Gkioxari. *Mask R-CNN.* IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 42(2). [ONLINE] Available at: `https://ieeexplore-ieee-org.zorac.aub.aau.dk/stamp/stamp.jsp?tp=&arnumber=8372616&tag=1`.

LogiAnalytics. LogiAnalytics. *What is Machine Learning? A Definition.* [ONLINE] Available at: `https://www.logianalytics.com/predictive-analytics/machine-learning-vs-traditional-programming/`.

Logstor. Logstor. *Product Catalogue.* [ONLINE] Available at: https://www.logstor.com/media/6506/product-catalogue-uk-202003.pdf.

Lund et al., 2014. Henrik Lund, Sven Werner, Robin Wiltshire, Svend Svendsen, Jan Eric Thorsen, Frede Hvelplund and Brian Vad Mathiesen. *4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems.* Energy, 68, 1–11. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2014.02.089. [ONLINE] Available at: https://www.sciencedirect.com/science/article/pii/S0360544214002369.

Ma et al., 2019. Haojie Ma, Yalan Liu, Yuhuan Ren and Jingxian Yu. *Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3.* Remote Sensing, 12(1), 44. doi: 10.3390/rs12010044.

machinelearningmastery.com, a. machinelearningmastery.com. *Understand the Impact of Learning Rate on Neural Network Performance.* [ONLINE] Available at: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/.

machinelearningmastery.com, b. machinelearningmastery.com. *A Gentle Introduction to the Rectified Linear Unit (ReLU).* [ONLINE] Available at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

machinethink.net. machinethink.net. *One-stage object detection.* [ONLINE] Available at: https://machinethink.net/blog/object-detection/.

Madlener and Sunak, 2011. Reinhard Madlener and Yasin Sunak. *Impacts of urbanization on urban structures and energy demand: What can we learn for urban energy planning and urbanization management?* Sustainable Cities and Society, 1(1), 45–53. ISSN 2210-6707. doi: https://doi.org/10.1016/j.scs.2010.08.006. [ONLINE] Available at: https://www.sciencedirect.com/science/article/pii/S2210670710000077.

mathworks.com. mathworks.com. *What Is Object Detection?* [ONLINE] Available at: https://www.mathworks.com/discovery/object-detection.html.

medium.com. medium.com. *Understanding Activation Functions in Neural Networks.* [ONLINE] Available at: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0.

northsearegion.eu. northsearegion.eu. *Heating technologies in District Heating.* [ONLINE] Available at: https://northsearegion.eu/media/7117/energielandschap-toolkitwarmtenetten-a4lig-bw-cc18_en_web-2.pdf.

Oksuz et al., Mar 19 2020. Kemal Oksuz, Baris C. Cam, Sinan Kalkan and Emre Akbas. *Imbalance Problems in Object Detection: A Review.* IEEE Transactions on Pattern Analysis and Machine Intelligence, PP. [ONLINE] Available at: https://www-proquest-com.zorac.aub.aau.dk/scholarly-journals/imbalance-problems-object-detection-review/docview/2381626929/se-2?accountid=8144.

Pantoja et al., 09 2018. Maria Pantoja, Anahid Behrouzi and Drazen Fabris. *An Introduction to Deep Learning*. Concrete International, 40(9), 35–41. ISBN 01624075. [ONLINE] Available at: https://www-proquest-com.zorac.aub.aau.dk/trade-journals/introduction-deep-learning/docview/2100824712/se-2?accountid=8144. Copyright - Copyright American Concrete Institute Sep 2018; Last updated - 2019-07-02.

paperspace.com. paperspace.com. *How to implement a YOLO (v3) object detector from scratch in PyTorch: Part 1*. [ONLINE] Available at: https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/.

parsers.me. parsers.me. *Deep learning Machine learning: what's the difference?* [ONLINE] Available at: https://parsers.me/deep-learning-machine-learning-whats-the-difference/.

PotentiaAnalytics. PotentiaAnalytics. *What is Machine Learning? A Definition*. [ONLINE] Available at: https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/.

pro.arcgis.com, a. pro.arcgis.com. *Export Training Data For Deep Learning*. [ONLINE] Available at: https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/export-training-data-for-deep-learning.html.

pro.arcgis.com, b. pro.arcgis.com. *How the Compute Accuracy For Object Detection tool works*. [ONLINE] Available at: https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.html.

relatedproject.eu. relatedproject.eu. *Heating technologies in District Heating*. [ONLINE] Available at: http://www.relatedproject.eu/heating-technologies-in-district-heating/.

Rezaie and Rosen, 2012. Behnaz Rezaie and Marc A. Rosen. *District heating and cooling: Review of technology and potential enhancements*. Applied Energy, 93, 2–10. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2011.04.020. [ONLINE] Available at: https://www.sciencedirect.com/science/article/pii/S030626191100242X. (1) Green Energy; (2)Special Section from papers presented at the 2nd International Enery 2030 Conf.

rubikscode.net. rubikscode.net. *Neural Networks*. [ONLINE] Available at: https://rubikscode.net/2018/01/15/how-artificial-neural-networks-learn/.

Sorknæs et al., 2015. Peter Sorknæs, Henrik Lund and Anders N. Andersen. *Future power market and sustainable energy solutions – The treatment of uncertainties in the daily operation of combined heat and power plants*. Applied Energy, 144, 129–138. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2015.02.041. [ONLINE] Available at: https://www.sciencedirect.com/science/article/pii/S0306261915002184.

thelancet.com. thelancet.com. *Opening the black box of machine learning*. [ONLINE] Available at: https://www.thelancet.com/journals/lanres/article/PIIS2213-2600(18)30425-9/fulltext.

towardsdatascience.com, a. towardsdatascience.com. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* [ONLINE] Available at: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`.

towardsdatascience.com, b. towardsdatascience.com. *The Most Intuitive and Easiest Guide for Convolutional Neural Network.* [ONLINE] Available at: `https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480`.

UNEP. UNEP. *DISTRICT ENERGY IN CITIES.* [ONLINE] Available at: `https://c2e2.unepdtu.org/wp-content/uploads/sites/3/2016/01/districtenergyreportbook-0.pdf`.

UnitedNations, 2018. UnitedNations. *World Urbanization Prospects: The 2018 Revision*, 2018. [ONLINE] Available at: `https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf`.

Weng, 2018. Lilian Weng. *Object Detection Part 4: Fast Detection Models.* lilianweng.github.io/lil-log. [ONLINE] Available at: `http://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html`.

Sven Werner. District Heating and Cooling. In Cutler J. Cleveland, editor, *Encyclopedia of Energy*, pages 841–848. Elsevier, New York, 2004. ISBN 978-0-12-176480-7. doi: https://doi.org/10.1016/B0-12-176480-X/00214-X. [ONLINE] Available at: `https://www.sciencedirect.com/science/article/pii/B012176480X00214X`.

Yamashita et al., 2018. Rikiya Yamashita, Mizuho Nishio, Richard Kinh Do and Kaori Togashi. *Convolutional neural networks: an overview and application in radiology.* Insights into Imaging, 9(4), 611–629. doi: 10.1007/s13244-018-0639-9.

zenggyu.com. zenggyu.com. *RetinaNet Explained and Demystified.* [ONLINE] Available at: `https://blog.zenggyu.com/en/post/2018-12-05/retinanet-explained-and-demystified/#fn3`.

SSD performance metrics A

Single Shot Detector - 25 training epochs

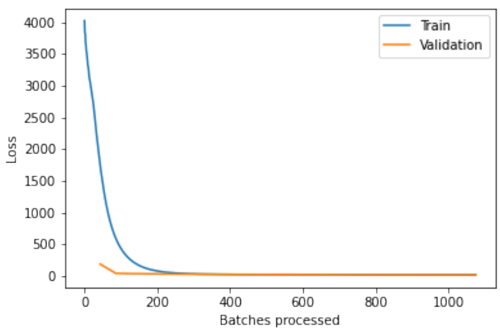

Figure A.1: Training and Validation loss - SSD 25 epochs

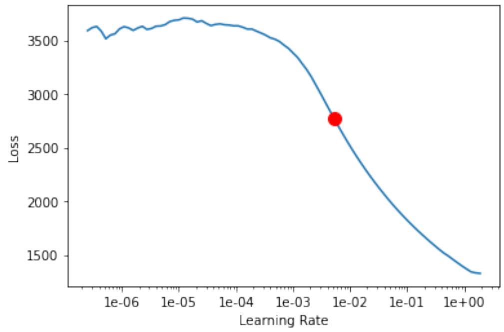

Figure A.2: Learning rate - SSD 25 epochs

Average precision score	
Cooling tower type	Score
Model11	0.0
Model12	0.34309496849776977
Model13	0.1305803619325161
Model14	0.05263157933950424
Model15	0.0714285746216774
Model16	0.2672727223056732
Model17	0.0
Model18	0.0
Model210	0.1913461595343855
Model212	0.13261648521201153
Model214	0.0
Model216	0.20606061617533378
Model218	0.0
Model22	0.0
Model24	0.21522367388805663
Model26	0.2545095695862405
Model28	0.09901800515121018
Model3	0.0

Table A.1: Average precision score - SSD, 25 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.6786	0.1546	0.2519	0.1282	19.0000	9.0000	164.0000
Model24	1.0000	0.3077	0.4706	0.3077	4.0000	0.0000	18.0000
Model28	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	15.0000
Model12	0.8571	0.1944	0.3170	0.1865	6.0000	1.0000	29.0000
Model13	0.5714	0.2500	0.3478	0.1905	4.0000	3.0000	21.0000
Model14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model16	1.0000	0.1176	0.2105	0.1176	1.0000	0.0000	15.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000
Model26	0.5000	0.2105	0.2963	0.1053	3.0000	3.0000	15.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model11	1.0000	0.2500	0.4000	0.2500	1.0000	0.0000	6.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	11.0000
Model210	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	6.0000

Table A.2: Accuracy metrics - SSD, 25 epochs

Performance metric for SSD - 50 epochs

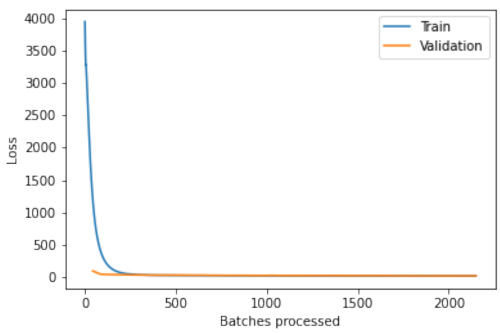

Figure A.3: Training and Validation loss - SSD 50 epochs

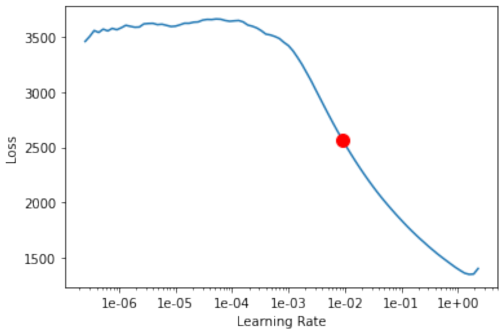

Figure A.4: Learning rate - SSD 50 epochs

Average precision score	
Cooling tower type	Score
Model11	0.0
Model12	0.39020856236757906
Model13	0.28445763140916824
Model14	0.13157894276082516
Model15	0.1666666689727987
Model16	0.5190724186069957
Model17	0.0
Model18	0.0
Model210	0.28098291799298725
Model212	0.33938170967483394
Model214	0.0833333358168602
Model216	0.4545454680919647
Model218	0.0
Model22	0.0
Model24	0.4233333355188371
Model26	0.3550293426337281
Model28	0.21798773399877414
Model3	0.0

Table A.3: Average precision score - SSD, 50 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.5385	0.3196	0.4011	0.2339	42.0000	36.0000	132.0000
Model24	1.0000	0.4615	0.6316	0.4615	8.0000	0.0000	14.0000
Model28	0.5000	0.0667	0.1176	0.0667	1.0000	1.0000	14.0000
Model12	0.4091	0.3056	0.3498	0.2156	9.0000	13.0000	25.0000
Model13	0.6923	0.5357	0.6040	0.4703	9.0000	4.0000	13.0000
Model14	0.5000	0.2667	0.3478	0.1333	2.0000	2.0000	11.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model16	0.7500	0.2941	0.4225	0.2794	3.0000	1.0000	12.0000
Model15	1.0000	0.3333	0.5000	0.3333	1.0000	0.0000	2.0000
Model26	0.3750	0.4737	0.4186	0.2594	6.0000	10.0000	10.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model11	0.6000	0.5000	0.5455	0.3000	3.0000	2.0000	4.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	11.0000
Model210	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	6.0000

Table A.4: Accuracy metrics - SSD, 50 epochs

RetinaNet performance metrics B

RetinaNet - 25 training epochs

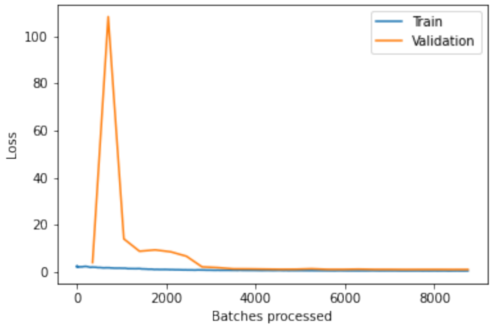

Figure B.1: Training and Validation loss - RetinaNet 25 epochs

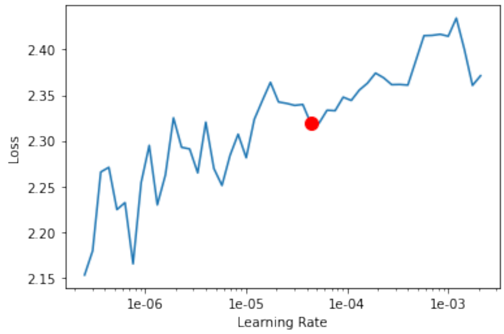

Figure B.2: Learning rate - RetinaNet 25 epochs

Average precision score	
Cooling tower type	Score
Model11	0.0
Model12	0.39020856236757906
Model13	0.28445763140916824
Model14	0.13157894276082516
Model15	0.1666666689727987
Model16	0.5190724186069957
Model17	0.0
Model18	0.0
Model210	0.28098291799298725
Model212	0.33938170967483394
Model214	0.0833333358168602
Model216	0.4545454680919647
Model218	0.0
Model22	0.0
Model24	0.4233333355188371
Model26	0.3550293426337281
Model28	0.21798773399877414
Model3	0.0

Table B.1: Average precision score - RetinaNet, 25 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.2427	0.1289	0.1684	0.0524	25.0000	78.0000	169.0000
Model24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	26.0000
Model28	0.5000	0.2667	0.3478	0.1524	4.0000	4.0000	11.0000
Model12	0.2857	0.1667	0.2105	0.0488	6.0000	15.0000	30.0000
Model13	0.2727	0.3214	0.2951	0.2152	9.0000	24.0000	19.0000
Model14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	4.0000
Model16	0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	17.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000
Model26	0.2000	0.3158	0.2449	0.1155	6.0000	24.0000	13.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	8.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	11.0000
Model210	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	6.0000
Model216	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000	0.0000
Model17	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000

Table B.2: Accuracy metrics - RetinaNet, 25 epochs

RetinaNet - 50 training epochs

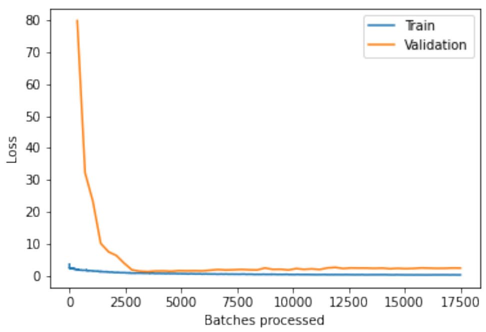

Figure B.3: Training and Validation loss - RetinaNet 50 epochs

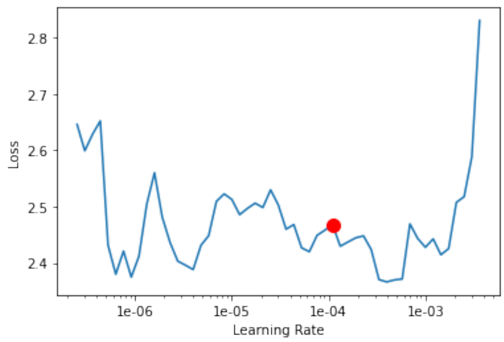

Figure B.4: Learning rate - RetinaNet 50 epochs

Average precision score	
Cooling tower type	Score
Model11	0.0
Model12	0.06031067404282259
Model13	0.013020833488553762
Model14	0.0
Model15	0.0
Model16	0.0
Model17	0.0
Model18	0.0
Model210	0.006410256840097606
Model212	0.0
Model214	0.0
Model216	0.09090909361839294
Model218	0.0
Model22	0.0
Model24	0.08113488450854478
Model26	0.04498044472926016
Model28	0.0
Model3	0.0

Table B.3: Average precision score - RetinaNet, 50 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.3333	0.1186	0.1749	0.0458	19.0000	38.0000	171.0000
Model24	0.3077	0.3077	0.3077	0.1674	8.0000	18.0000	18.0000
Model28	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.0000
Model12	0.3636	0.3056	0.3321	0.1159	8.0000	14.0000	25.0000
Model13	1.0000	0.0357	0.0690	0.0357	1.0000	0.0000	27.0000
Model14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	15.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	17.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000
Model26	0.2000	0.1053	0.1379	0.0211	1.0000	4.0000	17.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model11	1.0000	0.1250	0.2222	0.1250	1.0000	0.0000	7.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	11.0000
Model210	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	6.0000

Table B.4: Accuracy metrics - RetinaNet, 50 epochs

YOLOv3 performance metrics C

YOLOv3 - 25 training epochs

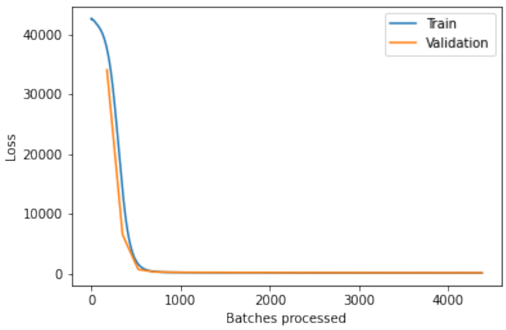

Figure C.1: Training and Validation loss - YOLOv3 25 epochs

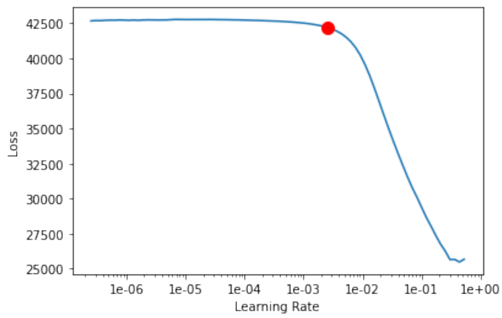

Figure C.2: Learning rate - YOLOv3 25 epochs

Average precision score	
Cooling tower type	Score
Model11	0.25014049145073125
Model12	0.5736615529969149
Model13	0.23490502079948783
Model14	0.15246449931273398
Model15	0.1428571492433548
Model16	0.24153846249366406
Model17	0.0
Model18	0.0
Model210	0.20459534441168747
Model212	0.26119814275047215
Model214	0.030303032109231687
Model216	0.5666666587193809
Model218	0.0
Model22	0.0
Model24	0.20251658360578384
Model26	0.36915978526662707
Model28	0.3096531805998566
Model3	0.0

Table C.1: Average precision score - YOLOv3, 25 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.1289	0.1907	0.1538	0.0695	37.0000	250.0000	157.0000
Model24	0.2500	0.0769	0.1176	0.0513	2.0000	6.0000	24.0000
Model28	0.1364	0.2000	0.1622	0.0600	3.0000	19.0000	12.0000
Model12	0.1328	0.4722	0.2073	0.1995	17.0000	111.0000	19.0000
Model13	0.1714	0.2143	0.1905	0.1720	6.0000	29.0000	22.0000
Model14	0.0000	0.0000	0.0000	0.0000	0.0000	5.0000	15.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000	4.0000
Model16	0.0000	0.0000	0.0000	0.0000	0.0000	5.0000	17.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	3.0000
Model26	0.1556	0.3684	0.2188	0.1131	7.0000	38.0000	12.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model11	0.1000	0.1250	0.1111	0.0208	1.0000	9.0000	7.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	5.0000	11.0000
Model210	0.3333	0.1667	0.2222	0.0833	1.0000	2.0000	5.0000
Model216	0.0000	0.0000	0.0000	0.0000	0.0000	11.0000	0.0000
Model3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Model17	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	0.0000

Table C.2: Accuracy metrics - YOLOv3, 25 epochs

Performance metric for YOLOv3 - 50 epochs

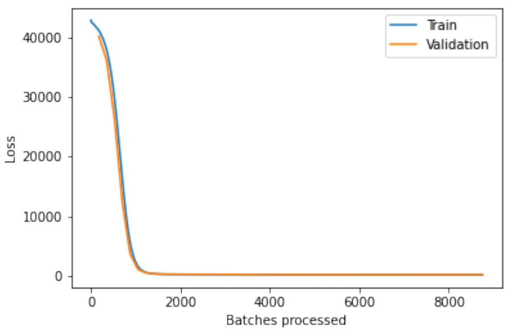

Figure C.3: Training and Validation loss - YOLOv3 50 epochs

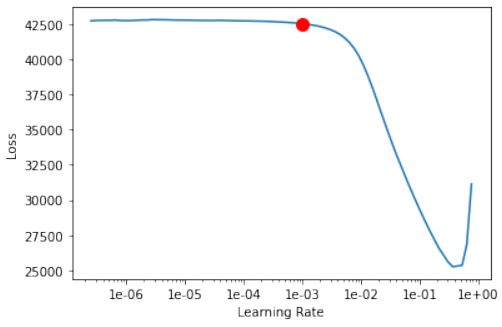

Figure C.4: Learning rate - YOLOv3 50 epochs

Average precision score	
Cooling tower type	Score
Model11	0.36038961419553495
Model12	0.6787235347206062
Model13	0.6226160656660795
Model14	0.2741241135344139
Model15	0.370890027423159
Model16	0.446245988496484
Model17	0.0
Model18	0.1666666716337204
Model210	0.29552172639532204
Model212	0.5488960100889653
Model214	0.3834175088439724
Model216	0.702479353620987
Model218	0.0
Model22	0.3888888971673117
Model24	0.4394222375500352
Model26	0.5540438953136961
Model28	0.6007778079231376
Model3	0.0

Table C.3: Average precision score - YOLOv3, 50 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.2240	0.3660	0.2779	0.1606	71.0000	246.0000	123.0000
Model24	0.2813	0.3462	0.3103	0.2518	9.0000	23.0000	17.0000
Model28	0.3571	0.3333	0.3448	0.1667	5.0000	9.0000	10.0000
Model12	0.2500	0.5556	0.3448	0.3241	20.0000	60.0000	16.0000
Model13	0.3556	0.5714	0.4384	0.3252	16.0000	29.0000	12.0000
Model14	0.1667	0.1333	0.1481	0.0333	2.0000	10.0000	13.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	4.0000
Model16	0.1765	0.1765	0.1765	0.0496	3.0000	14.0000	14.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	3.0000
Model26	0.2340	0.5789	0.3333	0.3612	11.0000	36.0000	8.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	5.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model11	0.1111	0.5000	0.1818	0.2917	4.0000	32.0000	4.0000
Model218	0.0000	0.0000	0.0000	0.0000	0.0000	4.0000	11.0000
Model210	0.1429	0.1667	0.1538	0.0833	1.0000	6.0000	5.0000
Model216	0.0000	0.0000	0.0000	0.0000	0.0000	9.0000	0.0000
Model3	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	0.0000

Table C.4: Accuracy metrics - YOLOv3, 50 epochs

FastRCNN performance metrics D

FastRCNN - 25 training epochs

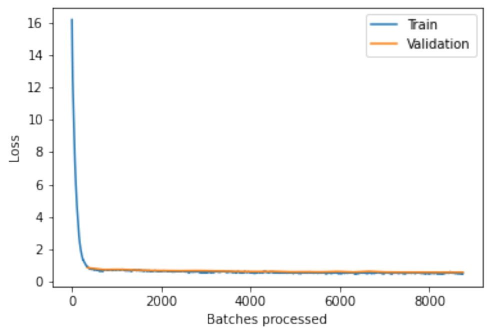

Figure D.1: Training and Validation loss -FastRCNN 25 epochs

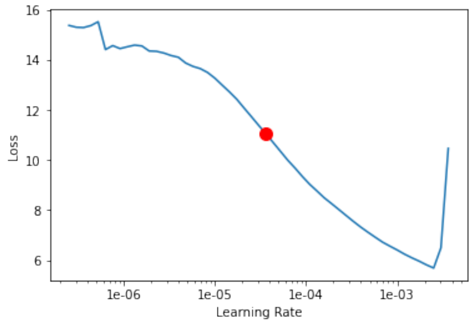

Figure D.2: Learning rate - FastRCNN 25 epochs

Average precision score	
Cooling tower type	Score
Model11	0.4851911407926348
Model12	0.7814678362288117
Model13	0.6782179661095142
Model14	0.47007331119864304
Model15	0.6281933223628862
Model16	0.8471578812254101
Model17	0.0
Model18	0.3333333432674408
Model210	0.3991344009032538
Model212	0.6586580699990296
Model214	0.5205357209912389
Model216	0.7272727489471436
Model218	0.0
Model22	0.633333351214727
Model24	0.6786311216501595
Model26	0.798130469877206
Model28	0.7559977575991541
Model3	0.0

Table D.1: Average precision score - FastRCNN, 25 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.3673	0.6495	0.4693	0.4124	126.0000	217.0000	68.0000
Model24	0.8000	0.7692	0.7843	0.7275	20.0000	5.0000	6.0000
Model28	0.3500	0.4667	0.4000	0.2473	7.0000	13.0000	8.0000
Model12	0.3368	0.8889	0.4885	0.6727	32.0000	63.0000	4.0000
Model13	0.3137	0.5714	0.4051	0.2961	16.0000	35.0000	12.0000
Model14	0.3529	0.4000	0.3750	0.1974	6.0000	11.0000	9.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	4.0000
Model16	0.7143	0.5882	0.6452	0.4752	10.0000	4.0000	7.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	3.0000
Model26	0.3077	0.8421	0.4507	0.5249	16.0000	36.0000	3.0000
Model214	0.1429	0.2500	0.1818	0.2500	1.0000	6.0000	3.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	1.0000
Model11	0.2188	0.8750	0.3500	0.6321	7.0000	25.0000	1.0000
Model218	0.7778	0.6364	0.7000	0.4949	7.0000	2.0000	4.0000
Model210	0.2667	0.6667	0.3810	0.3611	4.0000	11.0000	2.0000
Model216	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000

Table D.2: Accuracy metrics - FastRCNN, 25 epochs

FastRCNN - 50 training epochs

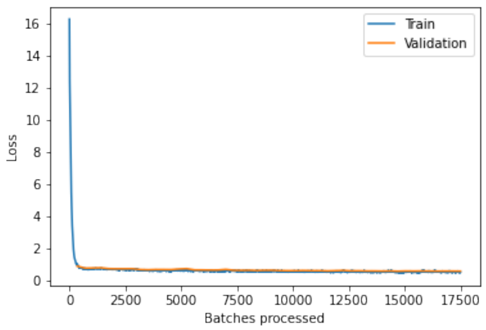

Figure D.3: Training and Validation loss -FastRCNN 50 epochs

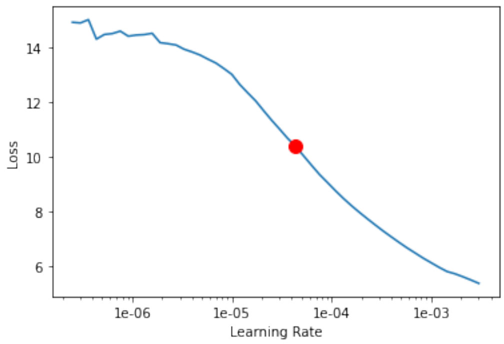

Figure D.4: Learning rate - FastRCNN 50 epochs

Average precision score	
Cooling tower type	Score
Model11	0.7013888768851757
Model12	0.7611841088937155
Model13	0.6817452441900969
Model14	0.6413466857640913
Model15	0.7752173754073084
Model16	0.8766666609048848
Model17	0.6666666865348816
Model18	0.6666666865348816
Model210	0.62532661758544
Model212	0.7714049848561331
Model214	0.6666666865348816
Model216	0.7272727489471436
Model218	0.0
Model22	0.6666666865348816
Model24	0.6634920837387206
Model26	0.8007608483845878
Model28	0.8263052183752748
Model3	0.0

Table D.3: Average precision score - FastRCNN, 50 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.4659	0.5979	0.5237	0.4219	116.0000	133.0000	78.0000
Model24	0.8182	0.6923	0.7500	0.6505	18.0000	4.0000	8.0000
Model28	0.6923	0.6000	0.6429	0.4728	9.0000	4.0000	6.0000
Model12	0.4426	0.7500	0.5567	0.6160	27.0000	34.0000	9.0000
Model13	0.4000	0.5714	0.4706	0.3627	16.0000	24.0000	12.0000
Model14	0.2857	0.4000	0.3333	0.2000	6.0000	15.0000	9.0000
Model212	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	4.0000
Model16	0.3571	0.2941	0.3226	0.1337	5.0000	9.0000	12.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	3.0000
Model26	0.6400	0.8421	0.7273	0.7665	16.0000	9.0000	3.0000
Model214	0.0000	0.0000	0.0000	0.0000	0.0000	5.0000	4.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model22	0.2000	1.0000	0.3333	1.0000	1.0000	4.0000	0.0000
Model11	0.3333	0.8750	0.4828	0.7012	7.0000	14.0000	1.0000
Model218	0.7500	0.5455	0.6316	0.4091	6.0000	2.0000	5.0000
Model210	0.5556	0.8333	0.6667	0.6926	5.0000	4.0000	1.0000

Table D.4: Accuracy metrics - FastRCNN, 50 epochs

Mask RCNN performance metrics E

Mask RCNN - 15 training epochs

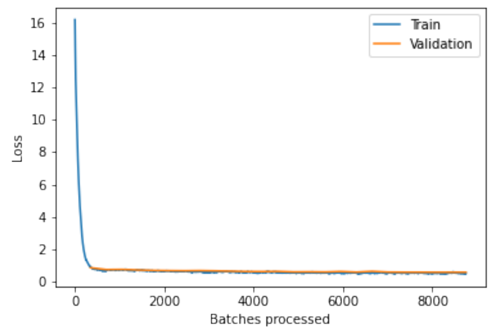

Figure E.1: Training and Validation loss -Mask RCNN 25 epochs

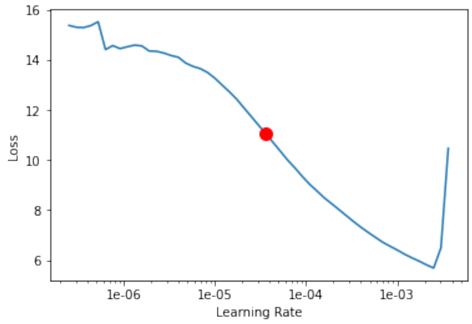

Figure E.2: Learning rate - Mask RCNN 25 epochs

Average precision score	
Model3	0.0
Model11	0.6851851873927646
Model12	0.7439024390243902
Model13	0.71
Model14	0.46376811680586444
Model15	0.38461538461538464
Model16	0.5333333373069763
Model17	0.975
Model18	0.3333333333333333
Model22	0.4444444444444444
Model24	0.6428571428571429
Model26	0.7109375
Model28	0.7203703713085916
Model210	0.5526315789473685
Model212	0.4369047624724252
Model214	0.4722222238779068
Model216	0.7250000014901161
Model218	0.0

Table E.1: Average precision score - Mask RCNN, 15 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.7789	0.7990	0.7888	0.7543	155.0000	44.0000	39.0000
Model24	0.9091	0.7692	0.8333	0.7619	20.0000	2.0000	6.0000
Model28	1.0000	0.6000	0.7500	0.6000	9.0000	0.0000	6.0000
Model12	0.6792	1.0000	0.8090	0.9305	36.0000	17.0000	0.0000
Model13	0.8182	0.9643	0.8852	0.9643	27.0000	6.0000	1.0000
Model14	0.7857	0.7333	0.7586	0.6222	11.0000	3.0000	4.0000
Model212	1.0000	0.5000	0.6667	0.5000	2.0000	0.0000	2.0000
Model16	1.0000	0.7059	0.8276	0.7059	12.0000	0.0000	5.0000
Model15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	3.0000
Model26	0.6800	0.8947	0.7727	0.8892	17.0000	8.0000	2.0000
Model214	0.6667	0.5000	0.5714	0.5000	2.0000	1.0000	2.0000
Model18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Model11	0.5000	0.8750	0.6364	0.6500	7.0000	7.0000	1.0000
Model218	1.0000	0.8182	0.9000	0.8182	9.0000	0.0000	2.0000
Model210	1.0000	0.5000	0.6667	0.5000	3.0000	0.0000	3.0000

Table E.2: Accuracy metrics - Mask RCNN, 15 epochs

Mask RCNN - 50 training epochs

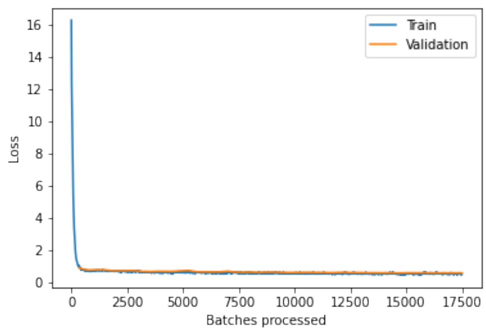

Figure E.3: Training and Validation loss -Mask RCNN 50 epochs

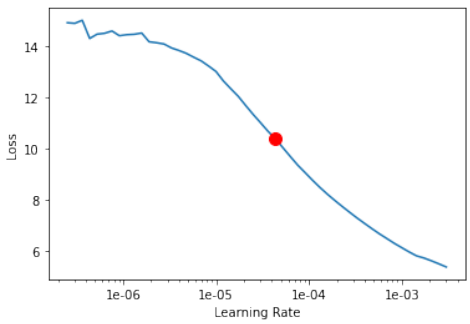

Figure E.4: Learning rate - Mask RCNN 50 epochs

Average precision score	
Model3	1.0
Model11	0.8333333333333334
Model12	0.8048780487804879
Model13	0.81
Model14	0.6739130434782609
Model15	0.5833333333333334
Model16	0.8518518540594313
Model17	1.0
Model18	1.0
Model22	0.6111111111111112
Model24	0.6964285714285714
Model26	0.7838541669771075
Model28	0.8481481480929587
Model210	0.7631578947368421
Model212	0.6547619047619048
Model214	0.7307692307692307
Model216	0.8500000014901161
Model218	0.5

Table E.3: Average precision score - Mask RCNN, 25 epochs

IoU >= 0.5000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.9196	0.9433	0.9313	0.9014	183.0000	16.0000	11.0000
Model24	0.9167	0.8462	0.8800	0.8094	22.0000	2.0000	4.0000
Model28	1.0000	0.8667	0.9286	0.8667	13.0000	0.0000	2.0000
Model12	0.8140	0.9722	0.8861	0.8855	35.0000	8.0000	1.0000
Model13	0.9643	0.9643	0.9643	0.9643	27.0000	1.0000	1.0000
Model14	1.0000	0.9333	0.9655	0.9333	14.0000	0.0000	1.0000
Model212	1.0000	1.0000	1.0000	1.0000	4.0000	0.0000	0.0000
Model16	0.8947	1.0000	0.9444	0.8947	17.0000	2.0000	0.0000
Model15	1.0000	1.0000	1.0000	1.0000	3.0000	0.0000	0.0000
Model26	1.0000	0.9474	0.9730	0.9474	18.0000	0.0000	1.0000
Model214	0.8000	1.0000	0.8889	0.8000	4.0000	1.0000	0.0000
Model18	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.0000
Model22	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
Model11	0.8889	1.0000	0.9412	0.8889	8.0000	1.0000	0.0000
Model218	1.0000	1.0000	1.0000	1.0000	11.0000	0.0000	0.0000
Model210	1.0000	1.0000	1.0000	1.0000	6.0000	0.0000	0.0000

Table E.4: Accuracy metrics - Mask RCNN, 50 epochs

F Model builder workflow

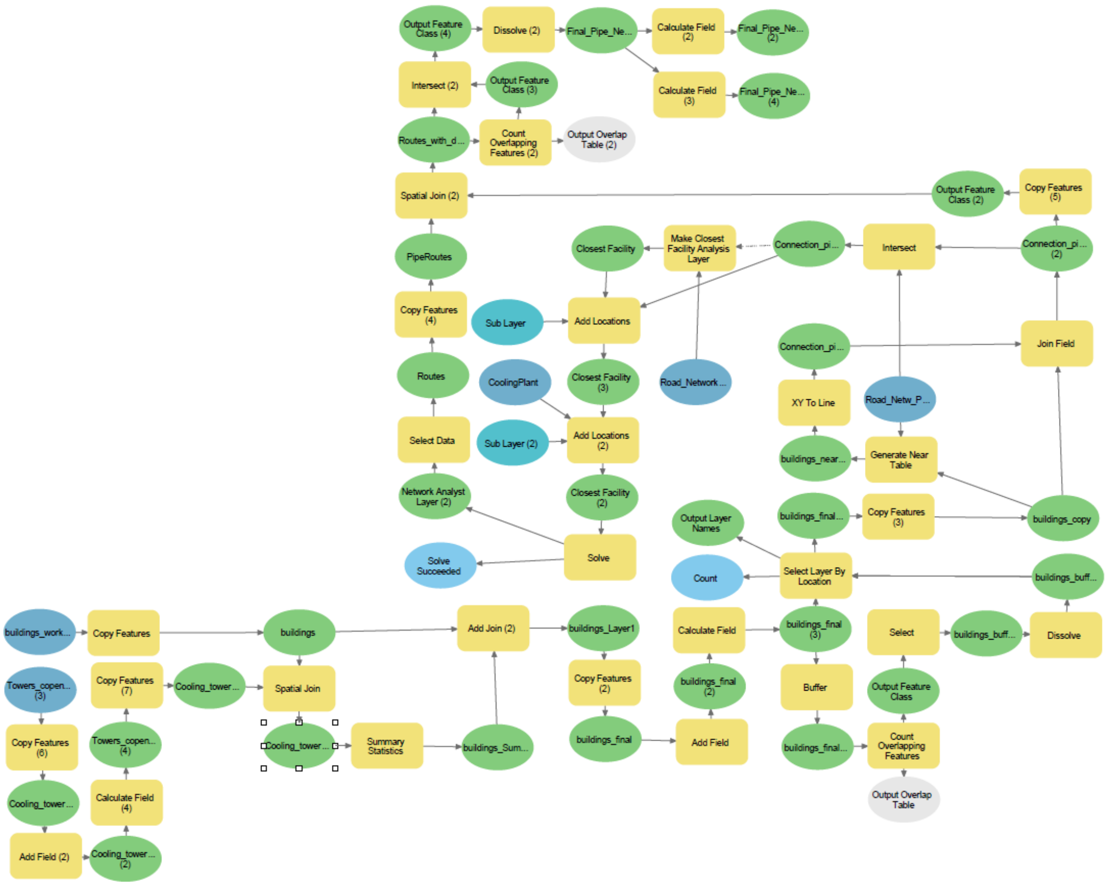

Figure F.1: Model builder workflow