Project Internship

(ProjektPraktikum)

|  |  |
| --- | --- |
| Enter English title here  Enter German title here | |
|  | |
| Student: | Yousif Mohammed Elsheikh Mohammed |
| Matriculation number: | 413278 |
| Supervisor (Company): | Mr. Keven Kemmer |
| Company: | Würth Group |
| Period of internship: | 01.03.2022 – 01.09.2022 |
|  | |
| October 2022 | |
| **Confirmation of the Company**  We hereby confirm that we have checked and approved the contents of this internship report.  Name and Signature of an authorized representative of the company and stamp of the company[[1]](#footnote-1) | |

*Blank page*

*This page is intentionally left blank*

Task Description

|  |
| --- |
| Task description of the project internship (Projektpraktikum) for Yousif Mohammed Elsheikh Mohammed , Matr. No. 413278 |
| Enter English title here  Enter German title here |
| *The task description of the project internship, given by the supervisor in the company, has to be inserted in this field (the text has to be formatted with the style "CVT Text"). The fields in the header are adopted from the title page after updating the document. The date at the end of the task has to be inserted manually.* |
|  |

Declaration of authorship

|  |  |  |
| --- | --- | --- |
| I, Yousif Mohammed Elsheikh Mohammed | | Matr. No. 413278 |
| declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material, which has been cited either literally or indirect from the used sources.  Enter date here  (Yousif Mohammed Elsheikh Mohammed ) | | |
| Declaration of agreement to plagiarism assessment  The reports to be reviewed are transmitted to the service URKUND, where they are checked for consistency with external sources and permanently stored in the database maintained by URKUND exclusively for the University of Kaiserslautern for the purpose of comparison with future reports to be reviewed. | | |
| I, Yousif Mohammed Elsheikh Mohammed | Matr. No. 413278 | |
| declare that I agree that the report I have submitted and written, including my personal data, may be permanently stored in the database maintained by URKUND for the University of Kaiserslautern for the purpose of checking internship reports for plagiarism.  The agreement to the permanent storage of the text and the disclosure of personal data is voluntary. The agreement to the storage and use of personal data can be revoked at any time by declaration for the future.  Enter date here  (Yousif Mohammed Elsheikh Mohammed ) | | |

Table of Contents

[List of Figures II](#_Toc114493869)

[List of Tables III](#_Toc114493870)

[List of Abbreviations IV](#_Toc114493871)

[List of Formula Symbols V](#_Toc114493872)

[1 Introduction 1](#_Toc114493873)

[2 Notes on working with the style sheet 6](#_Toc114493874)

[3 Formatting and content 7](#_Toc114493875)

[3.1 General formatting and structure 7](#_Toc114493876)

[3.2 Bullet points and numerations 7](#_Toc114493877)

[3.3 Figures and tables 7](#_Toc114493878)

[3.4 Formulas and references 9](#_Toc114493879)

[3.5 Overview of styles 10](#_Toc114493880)

[4 Literature and citation 12](#_Toc114493881)

[4.1 References 12](#_Toc114493882)

[4.2 How to cite 13](#_Toc114493883)

[4.2.1 Reference in the text 13](#_Toc114493884)

[4.3 Structure of the reference list 14](#_Toc114493885)

[5 Summary and Outlook 16](#_Toc114493886)

[References 17](#_Toc114493887)

[A Appendix 18](#_Toc114493888)



[Abbildung 1: Universal Robot Manipulator [https://www.universal-robots.com/products/] 3](#_Toc114666246)

[Figure 7‑1: This is an example for a figure 27](#_Toc114666247)

[Figure 7‑2: Using the labeling function for figures in Word 27](#_Toc114666248)

[Figure 7‑3: Inserting of cross-references in Word (Part 1) 28](#_Toc114666249)

[Figure 7‑4: Inserting of cross-references in Word (Part 2) 29](#_Toc114666250)

[Figure 7‑5: Inserting of footnotes in Word 29](#_Toc114666251)

[Figure A‑1: Figure 1 in appendix A 37](#_Toc114666252)

[Figure A‑2: Figure 2 in appendix A 37](#_Toc114666253)

List of Tables

[Table 3‑1: This is an example for a table 8](#_Toc51264753)

[Table 3‑2: Overview of all available styles 10](#_Toc51264754)

[Table 4‑1: Author abbreviations 12](#_Toc51264755)

[Table 4‑2: Special rules for author abbreviations 12](#_Toc51264756)

[Table 4‑3: Year of publication 13](#_Toc51264757)

[Table 4‑4: Different types of literature 14](#_Toc51264758)

[Table A‑1: Table 1 in appendix A 18](#_Toc51264759)

[Table A‑2: Table 2 in appendix A 18](#_Toc51264760)

List of Abbreviations

|  |  |
| --- | --- |
| CVT | Commercial Vehicle Technology |
| TUK | Technische Universität Kaiserslautern |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

List of Formula Symbols

|  |  |  |
| --- | --- | --- |
| F | Force | N |
| m | Mass | kg |
| a | Acceleration | m/s² |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

The problem of object detection and classification is an inherent issue in Computer Vision. IBM defines Computer Vision as “a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action.” [IBM22]

To address this problem, earlier approaches included feature extraction to identify objects, which are time consuming and highly manual. A later approach that emerged is CNNs (Convolutional Neural Networks) which is a more scalable approach that leverages methods from Linear Algebra, such as Matrix Multiplication to identify patterns in an image.

Within the standard 2D object detection, the information and the features used include edges and RGB colors [IBM22] which abstracts the features (corners and edges as well as RGB channels). To position these objects, which is a task under the umbrella of object detection, a bounding box is regressed, as follows up, an image would be an input and the output if a box or a box coordinates that are around or correspond to this object (a car in an image for example) [PS22].

## 3D Object Detection

Recently, higher level applications like scene understanding and object positioning require a richer form of information such as Point Clouds, in which the regression 3D box provides not only viewpoint information, but also information about the position of the object in the 3D space [PS22].

3D information is available as a set of vectors containing the x, y and z pairs representing the vertices that make up the point cloud. As the data in the pointcloud is random, and is hard to learn from, many proposals preprocess the points to voxel grids as views before network consumption which renders the data voluminous as well as introducing transformations that may change the original data [QSM17]. [ZT18] is an example of such approaches. Other appraches include working on the 2D images using CNNs to extract features, as the work with 2D images has been far more extensive than working with point clouds. After that, a 2D box is regressed to give the object’s position, which is further relayed to obtain the box’s 3D coordinates or the object’s z position, an example would be the works of [HZC17] that only directly works with image data and throws out the coordinates of the bounding box to further be processed and extract the depth data.

Other approaches propose working directly on the pointcloud, feeding it to the network and performing the classification and/or the detection task [QSM17; QYS17; QLH19]. Though the nature of the pointcloud is random and lacks order, these approaches, almost all of them, make use of learning the pointcloud features with a modification in the network, specifically, incorporating symmetrical functions to preprocess the data for learning allowing for the learning to take place from directly consuming the pointcloud [QSM17; QYS17] and they oftentimes employ the architecture of PointNet++ as a backbone to the actual network [ZSY20; QSM17; QYS17; QLH19].

## Overview

In chapter 2 the main problem will be discussed as well as the issues related to 3D object detection in general, chapter 3 will go through project specific literature, adapted proposals and current state of the art. In chapter 4 the proposed methodology and its implementation will be discussed and chapter 5 will deliver the the results obtained and chapter 6 will conclude the report with a discussion and future work (method to improve the results).

# Problem Statement

## Problem Description:

The main goal is to detect and localize 3D structures (KLTs) such that the information is available to a robot manipulator equipped with a gripper and a 3D Stereo-camera. The structures are set upon one another and have different categories depending on the objects inside, in which case the objects vary from C-Parts (fittings, screws, etc.) to dampers and other automotive parts. End goal is to develop a proof of concept, utilizing tools such as Deep Learning and 3D Object Detection algorithms to confirm the validity of the approach and culminate an in-house knowledge about the approaches available and their viability.

## Setting

The setting incorporates a manipulator, an end-effector with a greifer and an intel RealSense D435, the information from the camera are extracted, processed by a computer which throws out data fed to the Universal Robot manipulator as commands, which after receiving the position and class information acts accordingly. Figure 2-1 shows the robot manipulator and Figure 2-2 shows the Intel RealSense Depth (stereo-vision) camera, Figure 2-3 illustrates in more detail the different components of the depth camera.

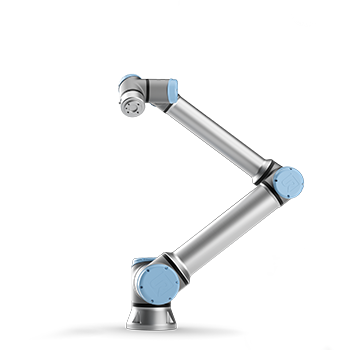


Figure : Universal Robot Manipulator, courtesy of [Uni22]



Figure 2-2: Intel Realsense D435 Depth Camera, as seen in [Rea22]

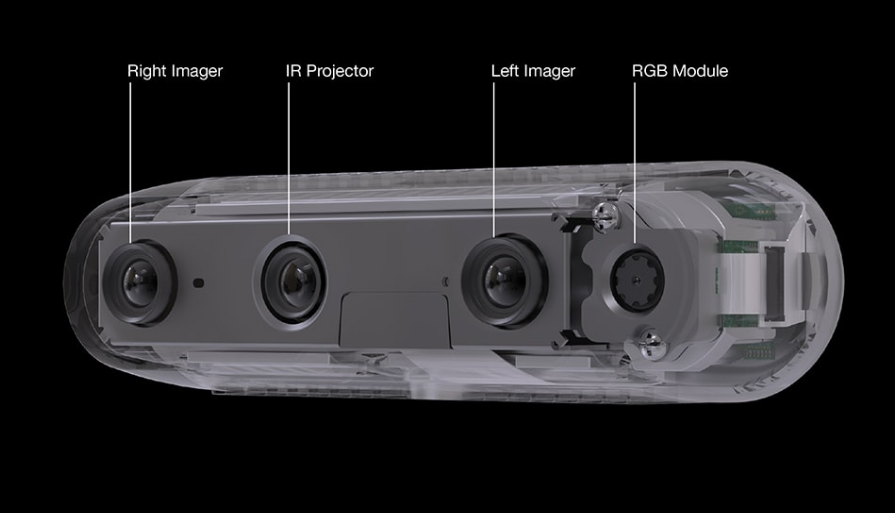


Figure 2-3: Intel Realsense D435 components, courtesy of [Rea22]

## Targeted Detection Pipeline:

The detection pipeline includes gathering the data, investigating the state-of-the-art, passing the state of the art to the data gathered, such that the targeted Network is able to learn from this data, which is shown briefly in Figure 2-4.

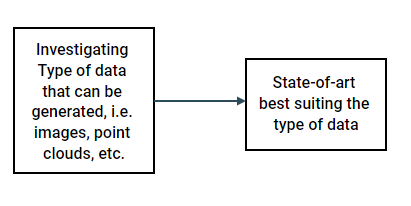


Figure 2-4: Targeted Detection Pipeline

After training the network, and for the purpose of testing the employed the state-of-the-art, testing data is prepared in a similar manner as the training data and the necessary code refactoring and edits to the code are carried out, proper metrics are then set, such that in the end, weakness points are determined, and necessary improvements carried out. Figure 2-5 shows a more detailed illustration of the data flow within the process.

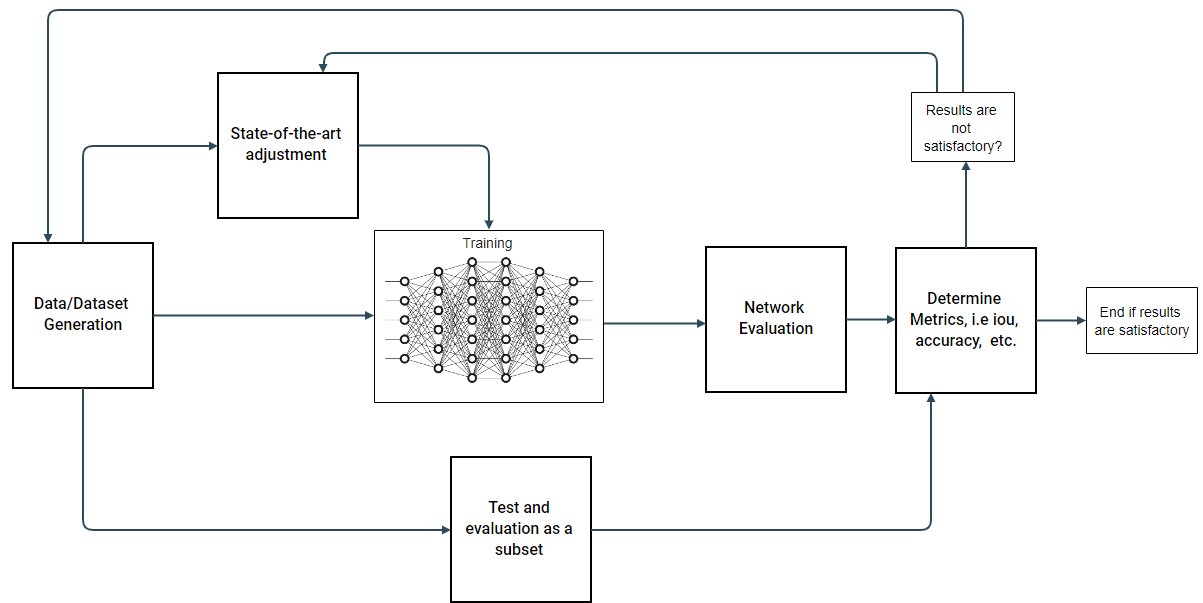


Figure 2-5: Training,Testing, Dataset, Algorithm and Dataflow

After the training has been carried out, the checkpoint where the network has stopped training, i.e. the last training epoch, is saved and an inference is carried out using new novel data. Additionally the network, is the state-the-art allows it, can be tested in real-time. This can be seen in Figure 2-6.

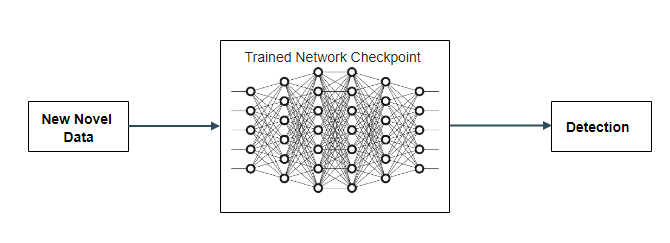


Figure 2-6: Inference on the saved Checkpoint

# State-of-the-Art, Deep Learning and Transfer Learning

## Deep Learning

Deep Learning is a subset of Machine Learning. Simply put, Machine Learning is defined as learing from experience, according to [ZLL21, p. 33]. Machine Learning houses several key components that describe the Machine Learning system [ZLL21, p. 35]:

* The Data that is utilized for learning and building the network parameters.
* The Model that includes data transformation techniques.
* The Objective Function that appraises the model’s performance on the data.
* The Algorithm that updates the model’s parameters given the dataset.

### Data

In order to make the concept understandable and abstract, no matter what the type of the data is, a numerical representation is required. As follows up, for each data sample there exists a set of „features“, based on these features the model is trained, tested and deployed for predition. The aim when using supervised learning is to predict attributes that are not present in the data, these attributes are conventionally called „labels“. [ZLL21, p. 35]

For instance, when working with images, the photo pixel values represent the features, depending on the size of the picture this „vector“ might differ in size (i.e. 200 x 200 x 3 = 120000 vector elements, where 200 are the horizontal pixels and the vertical and 3 is the number of channels, in this case RGB, Red Green Blue). On the other hand the category to which a cetrain image belongs (dog or cat in the picture) represents the label. [ZLL21, p. 35-36]

The amount of data corresponds in most cases proportionatly to the network performance in predicing in new data. This however depends on other factors, such as the data quality and weather the data is right or not. Finally, the data must be representative of the calsses that need to be represented for instance. [ZLL21, p. 35-36]

### Model

Zhang et al. defines the model as „denoting the computational machinery for ingesting data of one type, and spitting out predictions of a possibly different type“ [ZLL21, p. 36]. The interest is typically in statistical models that can „represent“ data and large amounts of data particularly, and hence are more complex in their computational nature as simple models for instance [ZLL21, p. 36]. This comlexity and ability to learnig form data is what gived Deep Learning its edge compared to other Machine Learning techniques [ZLL21, p. 36].

### Objective Function

To define the objective function one needs to first define the learing process, which indicate in its most basic meaning improving something (a task, a function etc.) over time [ZLL21, p. 36]. When one does so, a measure is necessary to tell when the model is updated during this learning process weather this update makes the model better or worse [ZLL21, p. 36].

In formal terms and within the context of Machine Learing and optimization this measure is described by utilitzing the objective function. The lesser the value this funtion spits out oft he model, the better the model is. These functions are known also conventionally in Machine Learning and optimization as „loss functions“. Depending on the type of the problem, being Regression or Classification an example would be square error or error rate respectively. [ZLL21, p. 36-37]

Depending on the value of the loss function the model parameters are updated accordingly until a satisfactory value of the loss function is reached, this indicates the end of training and updating the model’s parameters and the model is saved. [ZLL21, p. 36-37]

### Optimization Algorithms

After preparing the data, determining the model in which the data will train as well as determinig a suitable loss function that best help best determine the model’s parameters, the task is to find a way to establish this optmization relationship between the models parameters and the corresponding value of the loss function. There are several optmization algorithms in machine learning and deep learning that does this job of comparing the result oft he lossw function and updated the parameters such that this value will be minimized, an example of this class of algorithms is gradient descent. This algorithm checks the derivitave direction of the loss function and pertrubes the model parameters in the direction that drives the derivative to the negative direction or the less direction. [ZLL21, p. 37]

### Supervised Learning

Supervised Learning is when the given task, or dataset for that matter, contains both the labels as well as the features, and the goal is to train the model for the model to predict labels given a new dataset set of features (of course given that the new dataset has a similar structure as the one used to train the model). [ZLL21, p. 37]

Figure 3-1 illustrated the process of supervised learning given the model training inputs, labels as well as inputs and outputs to the model.

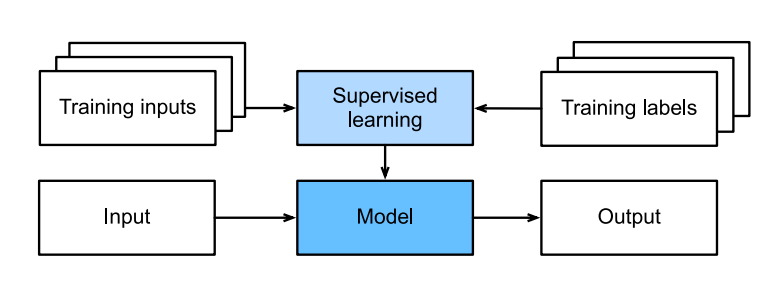


Figure 3-1: Supervised Learning, courtesy of [ZLL21]

### Definition

Deep Learning differs from the other methods in that it exploits very powerful models that house successive data transformations. These successions are tied together end-to-end in a top to bottom manner, from which the name „Deep Learning“ emerges. [ZLL21]

### Regression vs Classification

Problems in Deep Learning are two kinds, first one is a Regression Problem, in which data is used to approximate the data trend that is within in a form of a „Function“. This function in this case is the Deep Neural Network. The goal of doing so is to establish this function, train and test it with the given data so as to perform possibly predictions for new data that is from the same data class of the data used tot rain the network. [ZLL21]

Second Problem is the Classification Problem in which the data is in the form of classes. The network is trained and is to produce a function, similar to the Regression Problem, but in this case the function outputs the prediction for classes rather the predition of a datapoint. [ZLL21]

### Softmax Regression

Conventionally, the Machine Learning community set the problems within the field to two categeories, hard assignments of examples to class and soft assignments, assessing probabilies. This is however confusing at times. To elaborate on the classification problem and remove any confusion, a closer look into the classification problem is required. [ZLL21, p. 135]

As the classification data are missing in order the usage of one-hot encoding comes in handy. This is a convention in statistics in which a vector has elements set at 0 and the one element with 1 represents the class. [ZLL21, p. 135]

Considering the following [ZLL21, p. 136]:

where each of represents a distinct class in a 3 class group

is the output.

### Linear Models

As the model should throw out probabilities of the classes of interenst, a multi-output model is needed, hence, the amout of affine functions should be just as much as the number of classes. Assuming there are 3 classes and 4 features and denoting the weights with and the biases with , the targeted linear model becomes [ZLL21, p. 136-137]:

The corresponding Neural Network diagram is shown in Figure 3-2.



Figure 3-2: Architecture of 4 features and 3 classes, courtesy of [ZLL21]

To ensure nonnegativity, an exponential function is used, this function is called Softmax and can be defined by [ZLL21, p. 137]:

Where [ZLL21, p. 137];

### Convolutional Neural Networks

In the context of images, images can be seen as an array, in which the elements are the pixel intensity values as well as the RGB channels. The network input is irrespective of the order of the features and when dealing with imaga data this necessarily ignores the fact that there could be some kind of a relationship between neighboring pixels, hence there is a set of information lost and can be leveraged when dealing with image based datasets. CNNs are designed specifically to leverage this type of data, which has proven a significant modific ation compared to the aforementioned approach. Additionally CNNs provide more convenience when it comes to computational efficiency owing this to the fact that they require less parameters compared to fully connected layers that require a huge amout of parameters. CNNs do not offer only this advantage when it comes to computational efficiency, but they are also better parallelized when it comes to applying them using a GPU. [ZLL21, p. 228]

Furthermore, a most important feature of Convnets is their invariance, i.e. their ability to give robust results, irrespective of the position or orientation of an object in an image. Classical classifiers suffer from this disadvantage, for instance, when an object is flipped, the results are usually not reliable. If, for example, the network learns from a cars dataset that usually a car is usually present in a road, it should be able to detect or classify for that matter the car that isn’t on a different terrain that is present, say, on the edge oft he picture, or is in a flipped position. This problem is elemenated with Convnets as they are spatially invariant, making it possible to learn meaningful representations from fewer parameters. [ZLL21, p. 228]

In mathematics, taking and as function, convolving these two funtions is defined by the integral [ZLL21, p. 232]:

When dealing with discrete functions, which is the case for image data, the integration reduces to a summation [ZLL21, p. 232]:

Finally, performing this summation along the image rows and columns of the image, i.e. along the rows and columns [ZLL21, p. 232]:

Cross-Correlation

With reference to the aforementioned definition and usage of CNNs, in an image the convolution is a combination between an input tensor and a kernel, i.e., both of these are convolved with one another to produce an output tensor through the cross-correlation, this is demonstrated in Figure 3-3. [ZLL21, p. 234-235]

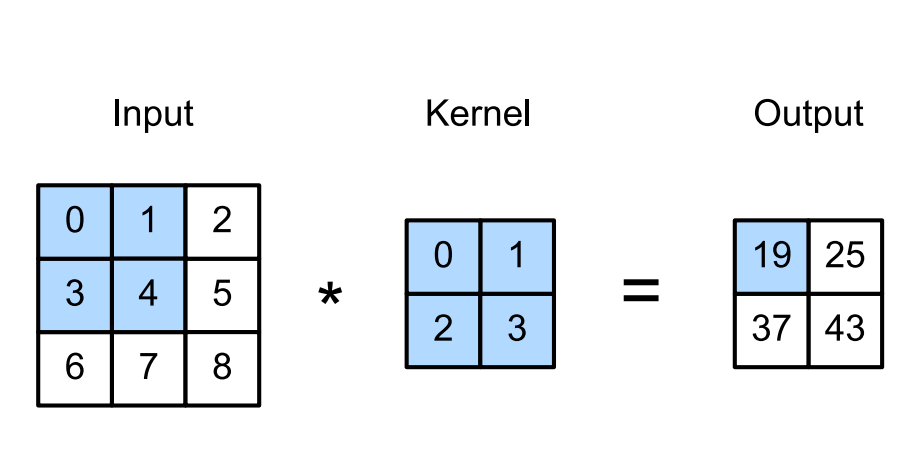


Figure 3-3: Cross Correlation applid to imges, as seen in [ZLL21]

## Performance Measures of Deep Learning and Computer Vision-based Models

### Precision, Recall and Accuracy

### Precision can be defined as [OD08]:

### Recall is defined as [OD08]:

And accuracy is hence [OD08]:

Where;

: is the number of true positives in the validation dataset

: is the number of true negatives in the validation dataset

: is the number of false positives in the validation dataset

: is the number of false negatives in the validation dataset

### Intersection over Union (IoU)

Intersection over Union Indicates the amount to which the Predicted Bounding Box corresponds tot he Ground Truth Bounding Box when validating the model through the validation dataset [Sha22]:

## MobileNets

### MobileNet-V1

MobileNets employ a set of depth-wise separable convolutions, with the main point of reducing the computation in the first few layers [HZC17]. The structure is mainly depthwise separable convolutions, except for the first layer which is a full convolutional network. This structure is carried out to ensure the networks ability to be employed on mobile and embedded devices carrying out inference/realtime inference on the spot, considering the limited set of computational and memory available in such devices [HZC17]. Figure 3-4 is an example of vanilla convolutions.

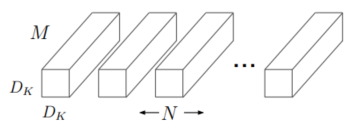


Figure 3-4: Convolution kernels [HZC17]

In MobileNets the depthwise seperable convolutions are applied in 2 steps [HZC17]:

* A single Dk x Dk x 1 input filter to each channel [HZC17]. As illustrated in Figure 3-5

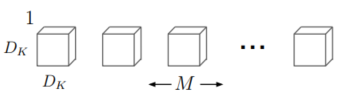


Figure 3-5: Channel input filter [HZC17]

* A pointwise M x 1 x 1 convolution to combine the outputs of the previous filter [HZC17], this can be seen in Figure 3-6

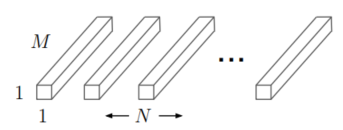


Figure 3-6: M x 1 x 1 convolutions [HZC17]

MobileNets which apply 3x3 depthwise seperable convolutions use 8-9 times less computational power, comprated to regular convolutions with a slight reduction in accuracy [HZC17].

The work of MobileNets is based on [Cho17], which introduces the depthwise seperable convolutions, Figure 3-7 illustrates the general structure applying both depthwise and pointwise convolutions.

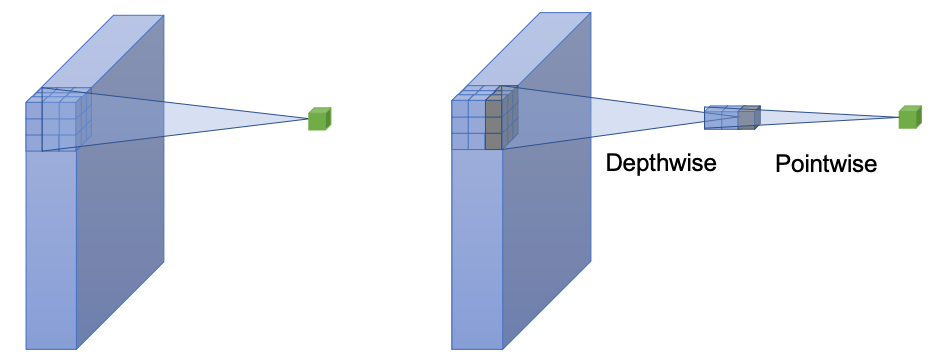


Figure 3-5 demonstrates the architecture of MobileNetV1 according to [KAK22]

Figure 3-6 illustrated the structure of MobileNet V1 according to [KAK22]

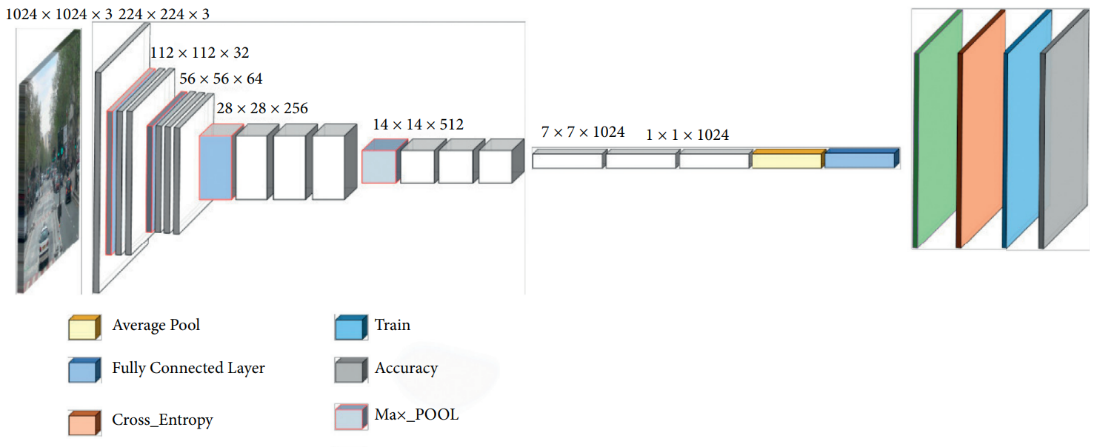


Figure 3-6: Mobilenet V1 architecture [KAK22]

### MobileNet-V2

The structure of MobileNetV2 differs from that of MobileNetV1. It is based on the concept of inverted residuals, where a shortcut connection is established between bottleneck layers. Bottleneck layers, known as projection layers, and as the name suggests, reduce multi-layer tensors. An intermediate layer is also introduced, it uses the lightweight depthwise convolutions to filter features as a source of non-linearity. Additionally, the authors remove non-linearities in the narrow layers and the reasoning behind it  is to maintain the features that are representative. [SHZ18]

Figure 3-7 demonstrated the general architecture of MobileNetV2 according to [Hol22]

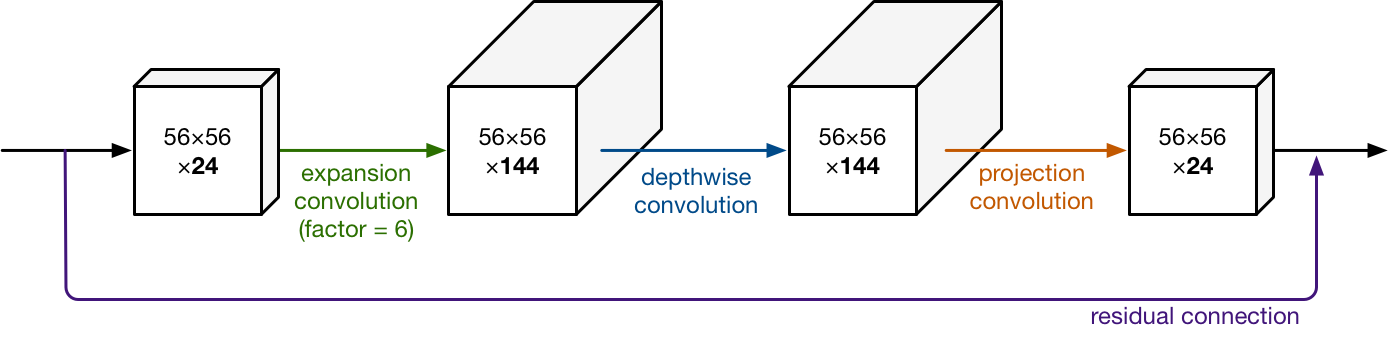


Figure 3-7: MobileNetV2 general architecture according to [Hol22]

## Transfer Learning

As Olivas et al. put it: “Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned.” [OGM09]

Transfer learning is effective and desired when the features learned from one set are general. [OGM09]

Figure 3-8 illustrates the process of transfer learning from [Bha22].

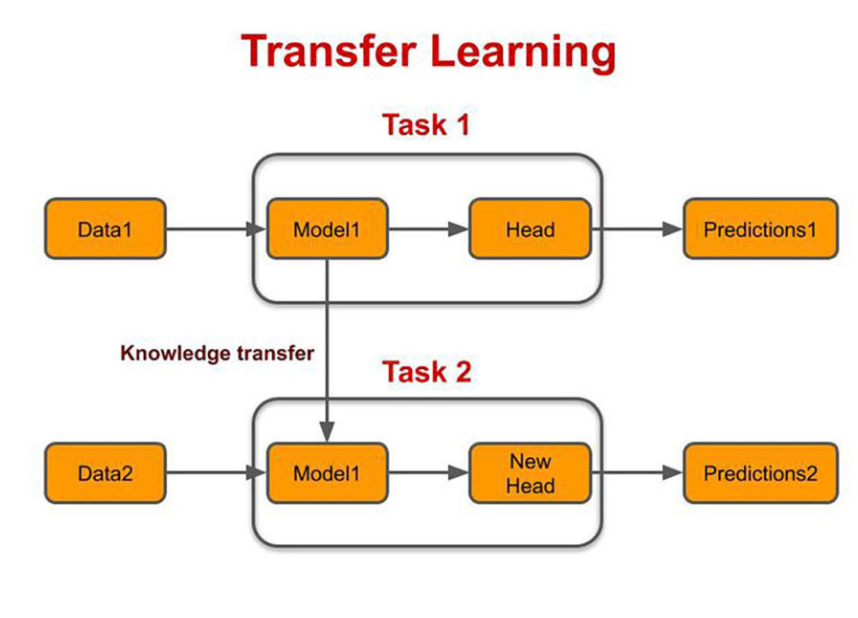


Figure 3-8: the process of transfer learning according to [Bha22]

## VoteNet

VoteNet leverages PointNet++, as PointNet++ processes the point cloud as it is and learns features in the point cloud structures without having to convert it to regular structures. This consequently indicates that the information is taken as is, rather than partly losing information in the conversion process.

A standard approach to apply PointNet++ to propose 3D bounding boxes is to follow suit with the standard box proposal methods by proposing the boxes that has dense objects within. This approach is not favourable in 3D proposals for a few reasons, firstly, the inherent sparsity in point clouds and secondly, depth sensors tend to capture only the surface of the objects in the scene, so the centers of the objects are usually spatially in a void place in that point cloud. Herein comes VoteNet to leverage Hough Voting and customize to establish a relationship between the centers and the points, which after aggregation are used to generate bounding box proposals. [QLH19]

The input is first sampled through the backbone network a set of seed points is sampled, from which feature are derived and votes are generated. The votes, as aforementioned are targeted to indicate an object’s bounding box’s center location and to generate the bounding box proposals they are clustered and aggregated through a learning module. [QLH19]

### PointNet

PointNet is a unified approach that consumes point clouds directly as inputs and outputs labels and/semantic labels, labels are generated per point in the point set. The authors of pointnet argue that it has a simple architecture. Each point is processed individually as a coordinate (*x; y; z*), the paper leverages the use of symmetric functions, such as max pool, and including an MLP (Multi Layer Perceptron). The network then learns special relationships between interesting points in the given input set, the selection grounds are embedded in the network parameters. The final layers are fully connected layers that aggregate the learned features into a ‘global’ descriptor for the shapes in the case of shape classification and the point labels in the case of segmentation. [QSM17]

Figure 3-9 illustrates the architecture of PointNet according to [QSM17]

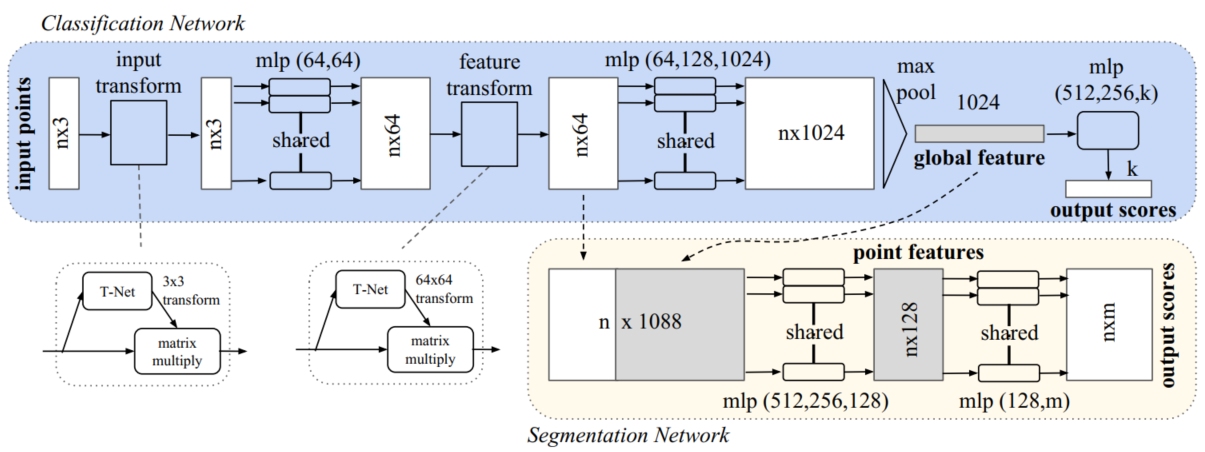


Figure 3-9: Architecture of PointNet [QSM17]

### PointNet++

One of the problems that lead to the inception of PointNet++ is that PointNet does not capture local structures into features, it rather constructs the relationship function based on the whole consumed point cloud, regardless of the individual and distinct geometries embedded within the point cloud. CNNs takes datapoints in grid-like shapes and is able to capture features in different scales and among different resolutions. The field receptivity of the neurons are proportional to weather the levels are low or high, the authors of PointNet++ argue that abstracting local patterns in a more explicit way allows for better generalization regarding the network parameters when the network is shown net data. [QYS17]

PoitNet++ hence processes points hierarchically, the points are partitioned according to a defined distance metric, and the features are extracted from these points in a manner similar to that of CNNs, these features are then aggregated further into a larger point feature hierarchy, and so on the process is undertaken from the bottom up until the whole point cloud is enveloped in-whole. [QYS17]

The generation of the overlapping in the point set is first by selecting centroids of the areas of interest with the aid of the FPS (Farthest Point Sampling) algorithm [QYS17].

Figure 3-10 demonstrated the architecture of PointNet++ according to [QYS17]

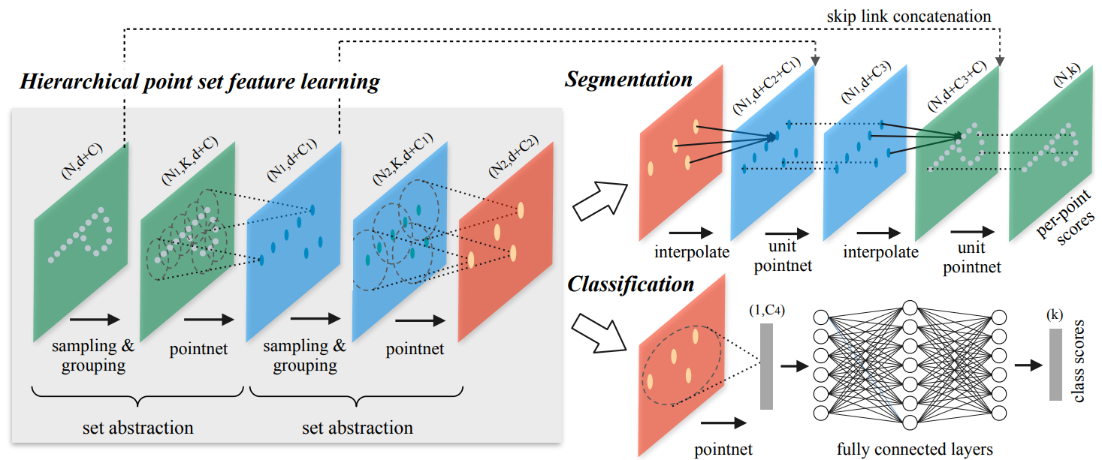


Figure 3-10: Architecture of PointNet++ [QYS17]

Hierarchical Point Set Feature Learning

As aforementioned and in contrast to PointNet that lumps features from point sets using a max pool operation, PointNet++ does this feature lumping in a bottom-up approach in what is called hierarchical feature learning. [QYS17]

As seen in Figure 3-10 in the feature abstraction layers 3 types of processing take place, firstly sampling the points (sampling layer), grouping points (grouping layer) and the PointNet layer. [QYS17]

The points are processed in these layers as follows: centroids are selected from the point set via a query algorithm, in PointNet++ this is FPS (Farthest Point Sampling). After that regions around these centroids are selected via the grouping layer/process. Finally, the PointNet layer then aggregates local features from these set of local points. [QYS17]

*Sampling Layer:*  Given the point set: {*x*1, *x*2,…, *xn*} as input to the network, the FPS algorithm is deployed to select a subset: {*xi*1, *xi*2, …, *xim*} such that *xij* that is the most distant point form the set: {*xi*1, *xi*2, …, *xim*}. The advantage of FPS over random sampling is that the point set is included as a whole and all of the regions of the point set are represented.  Another key feature that sets PointNet++ ahead of CNNs with reggard to data representation is that CNNs as put by the PointNet++ authors: “In contrast to CNNs that scan the vector space agnostic of data distribution, our sampling strategy generates receptive fields in a data dependent manner.” [QYS17]

*Grouping Layer:* The input to the grouping layer is the point set of size *N ×* (*d* + *C*) as well the set of centroids that the sampling layer outputs. The output is a group of points of size *N’ × K ×* (*d* + *C*), where N’ is the number of centroids and K is the number of points within the centroid’s neighborhood. The Ball Query algorithm is deployed to limit the points in the centroid’s neighborhood number of points. [QYS17]

*PointNet Layer:* the input is *N’* local regions of points with data size *N’×K×*(*d*+*C*), the features of the local regions are encoded and the layers throws out an output of size: *N’ ×* (*d* + *C’’*). [QYS17]

Multi Scale Resolution:

It is not uncommon for point clouds to have varying densities, not only within the same point cloud, but also compared to other point clouds in the same dataset.  This is a problem, as it can be more difficult to match sparse and dense point clouds, i.e., features learned from dense point clouds/areas cannot be generalized to sparse ones/areas. [QYS17]

This is where Multi Scale Resolution comes to play, as shown in Figure 3-11.

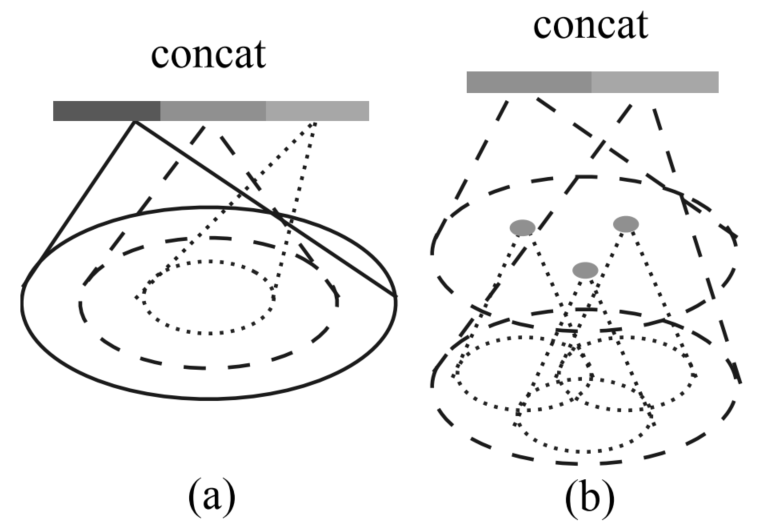


Figure 3-11: Multi Scale Resolution, as seen in [QYS17]

It is used to capture the features in multiple scales, mainly done by applying grouping layers that would then be the input to PointNet but this is done in different scales. The features are then concatenated, solving the dense vs. sparse point cloud problem. [QYS17]

### Hough Voting and the Hough Transform:

Introduced in the late 1950s, it is originally used to detect patterns in a set of points in some arbitrary coordinate system, but extends this pattern detection to peak detection in the parametric space.

It achieves this by finding the parameters that correspond to these patterns [IK88]. The patters are transformed into another space in which the derived features will be mapped in a compact way [IK88]. Hence the HT transforms the detection problem into a space where they can more easily solved problem of peak detection [IK88].

 Furthermore, in the context of object detection the Hough Voting or the Hough Transform is used to detect abnormal or complex objects in image patches. [LLS08] introduced implicit shape methods, other work used it for plane extraction from 3D point clouds and pose estimation [QLH19]. Hough Voting has also been combined with learning techniques [QLH19].

Deep Hough Voting

VoteNet leverages Hough Voting in 2 distinct ways, the first one is by establishing that voting based detection is more compatible with point sets that are sparse, as the Region Proposal Networks (RPN) will have to extra computations in void or sparse areas. Secondly, the information is aggregated and is built up from small sets up to engulfing the whole point set (point cloud). [QLH19]

To adapt the concept of Hough Voting into a 3D state-of-the-art VoteNet follows the following approach [QLH19]:

Firstly, interest points are selected in line with the aforementioned information. The Votes are then generated via a learning module the is part of the larger neural network. These votes are then aggregated and ‘bad’ votes are then excluded. Finally using these aggregated features and votes objects are proposed, the proposal is in the form of location of the object in the particular coordinate frame, the dimensions and orientation of the bounding box that are generated via the bounding box and finally, the semantic class. [QLH19]

Figure 3-12 illustrated the process of Deep Hough Voting on the SUNrgbd dataset. Figure 3-13 illustrates the architecture of VoteNet.



Figure 3-12: Deep Hough Voting on SUNrgbd [QLH19]

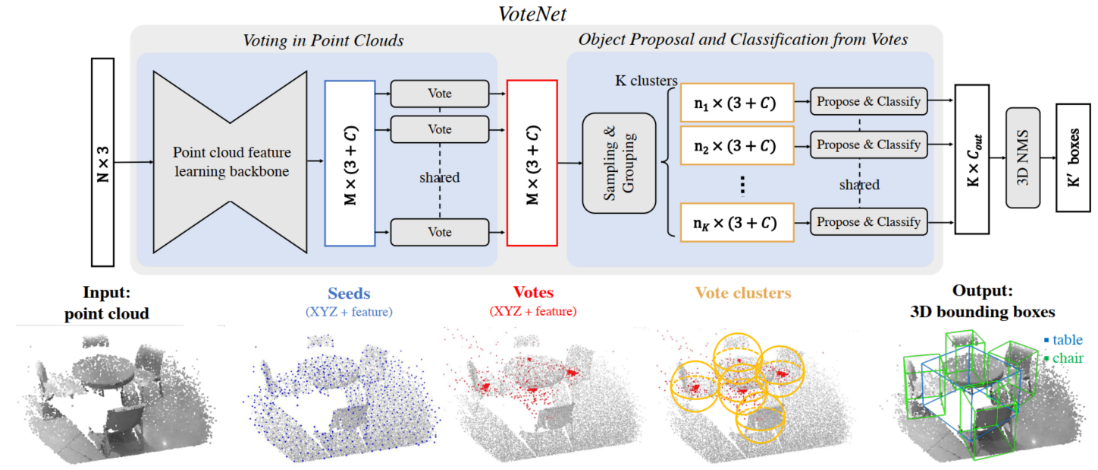


Figure 3-13: architecture of VoteNet [QLH19]

Training Process and Forward Propagation

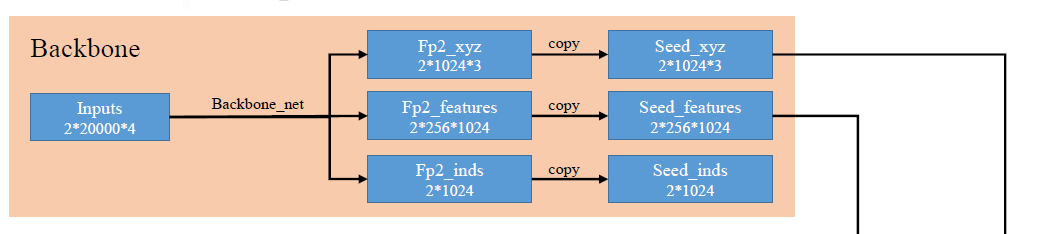


Figure 3-14: Backbone Network[Hal22]

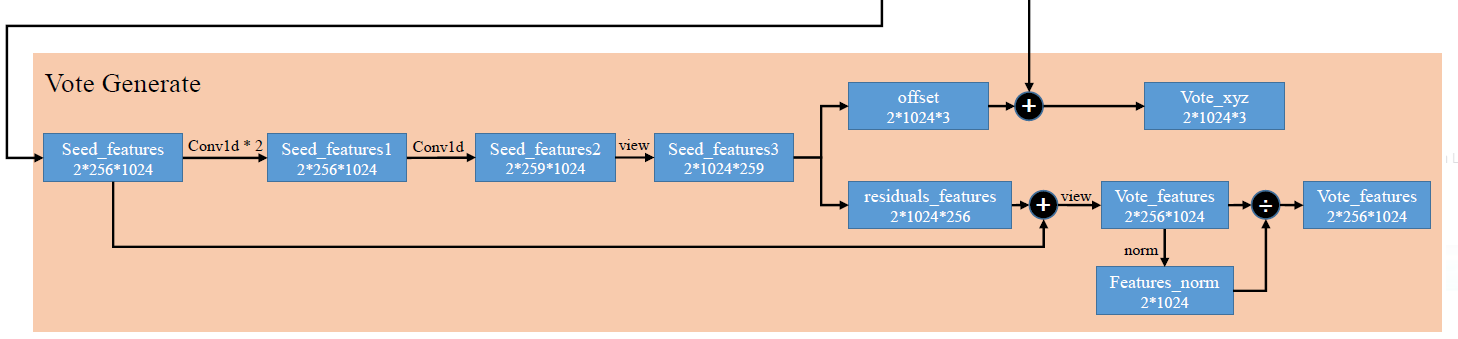


Figure 3-15: Vote generation [Hal22]

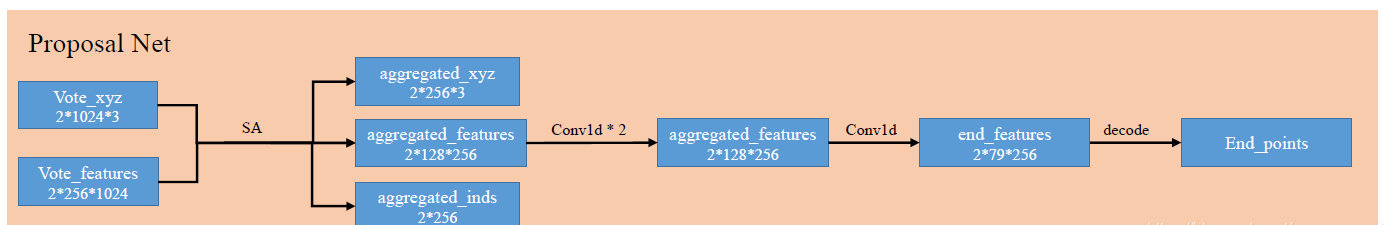


Figure 3-16: Proposal net [Hal22]

The specific forward calculation process is shown in the above figure.

* Backbone: BackBone consists of Pointnet++, including 4 SA layers, 2 FP layers, and the final output Seed\_inds is an indication of Seed\_xyz's position in Inputs. As seen in Figure 3-14.
* Vote Generate: use Conv1d on Seed\_features to get offsets and residuals\_features of vote, then add them with xyz and features of seed to get vote\_xzy and vote\_features, and finally normalize vote\_ features is normalized. As seen in Figure 3-15.
* Porposal Net: the Vote\_xyz and Vote\_feature are then passed through another layer of SA, and then the final result is obtained by using Conv1d on the feature, and the variables for calculating the loss are constructed by decode. This is illustrated in Figure 3-16.

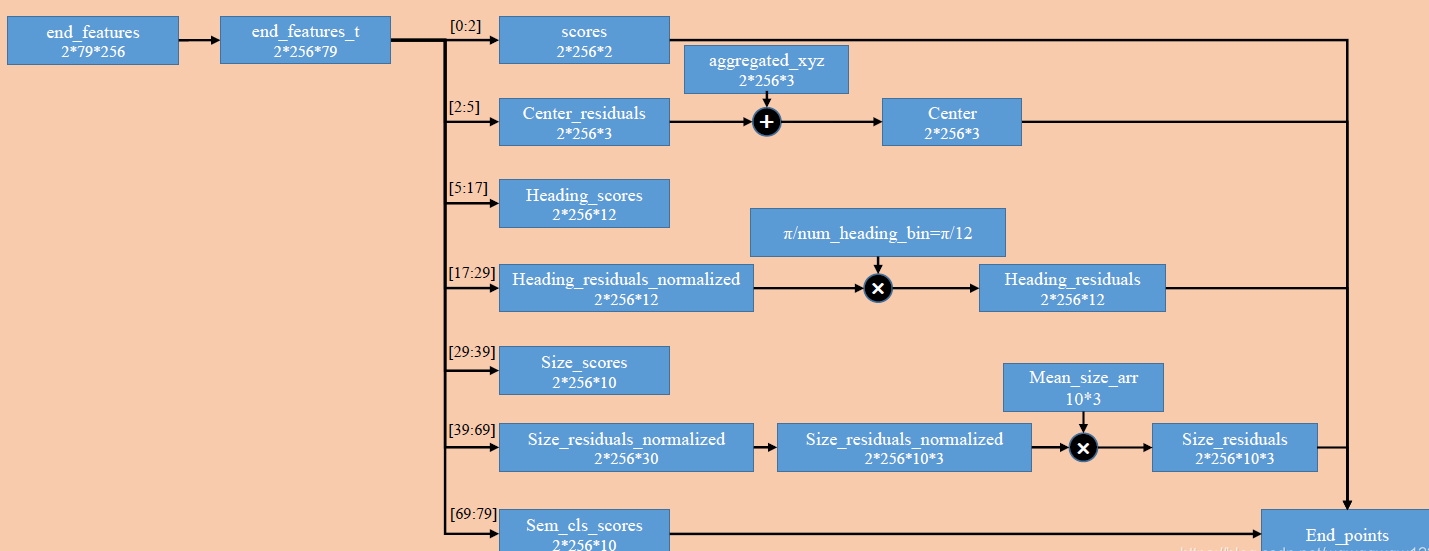


Figure 3-17: Decoding net

Decode process does not generate gradients, there is no convolution layer, just the final result of the resolution into the calculation of loss variables, the specific operation is seen in Figure 3-17.

The max-pooled features are further processed by MLP2 with output sizes of 128, 128, 5+2NH+4NS+NC where the output consists of 2 objectness scores, 3 center regression values, 2NH numbers for heading regression (NH heading bins) and 4NS numbers for box size regression (NS box anchors) and NC numbers for semantic classifications.

* Calculating Loss

The composition of Loss is divided into several parts and is rather complex. The below figures show how the Vote and the Semantic losses are calculated. Figure 3-18 and Figure 3-19 illustrated the calculation of the vote and the semantic class losses respectively.

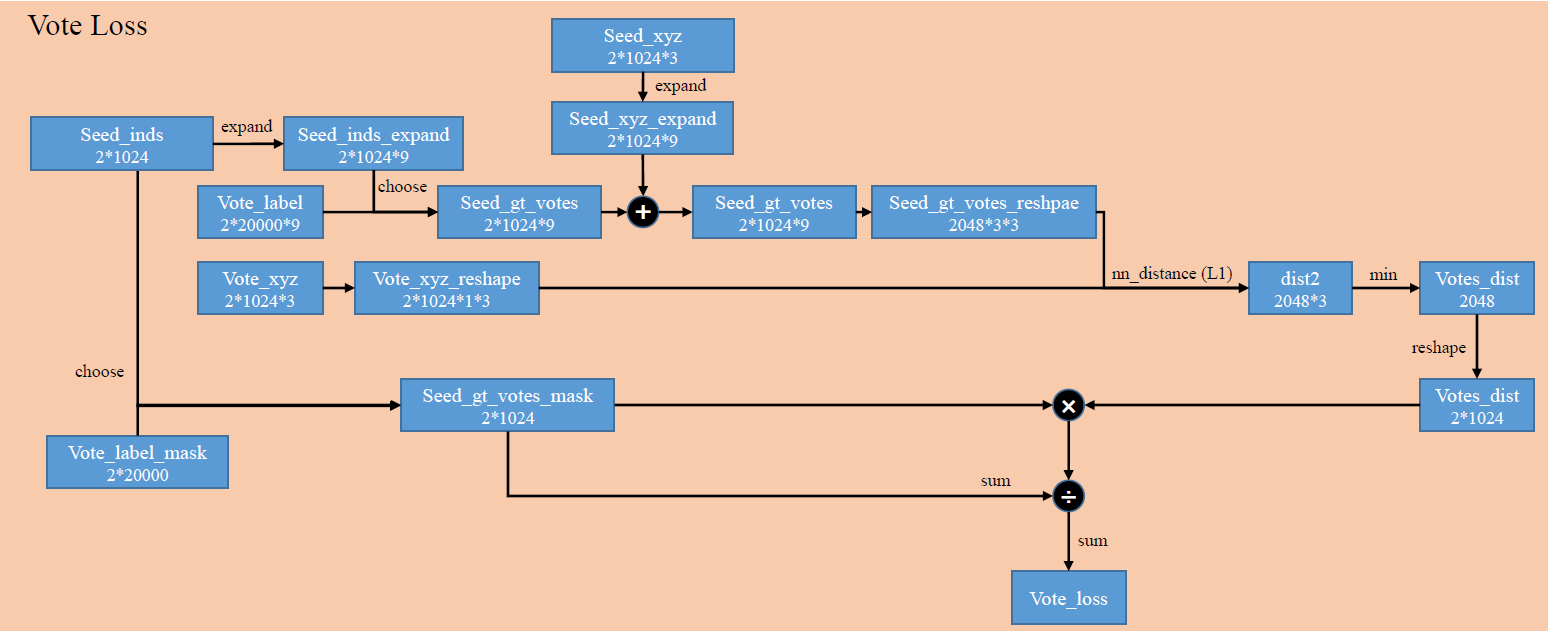


Figure 3-18: Vote loss [Hal22]

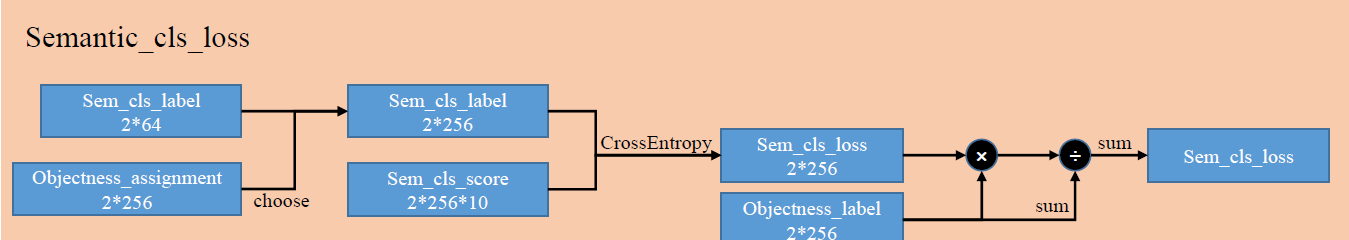


Figure 3-19: Semantic class loss [Hal22]

# Implementation

In this chapter the implementation details will be illustrated as well as dataset preparation, training and results.

To achieve good results on a state-of-the art, a careful implementation of the code to fit a certain application is of the essence, this includes diving into the code base to understand and verify the results on benchmark datasets to consequently training on a custom dataset.

## Preparation and Input Data

The input data to the network is in the form of 3D point clouds with x, y and z coorinates, the blow figures (Figure 4-1 and Figure 4-2) shows how the data looks like after being strapped out of the color (the Inel RealSense camera usually output the taken point cloud in the form of colored point cloud, but this is unncessary as theb proposed State-of-the-Art consumes the point cloud directly).

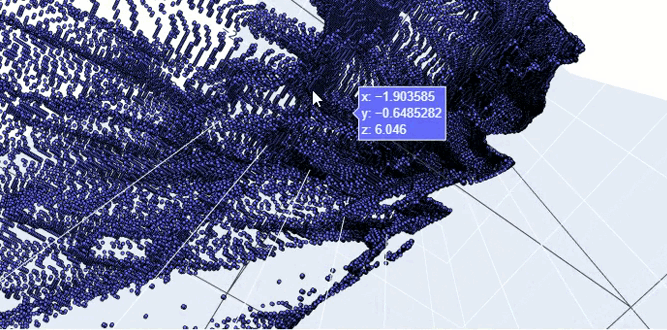


Figure 4-1: Depth data in the Cartesian coordinate system

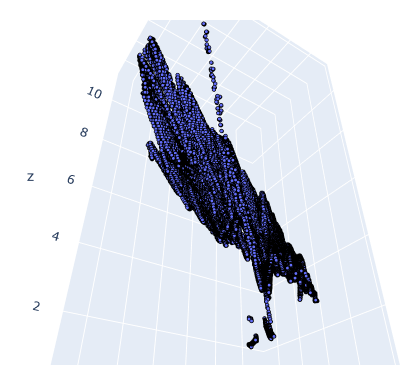


Figure 4-2: Full depth data Point Cloud

## 2D to 3D Methods

These methods try to leverage the use of both 2D and 3D, harnessing the well-established knowledge in 2D Computer Vision with Convolutionsl Neural Networks (CNNs) and uses point cloud coordinates to determine the position.

### MobileNet-V1 and OpenCV

The idea is to utilize a trained MobileNet-V1, use the checkpoint to generate bounding box proposals and then leverage the x and y coordinates of the bounding box to re-reflect those on the point cloud, setting an approximate location to where the points constituting the object in the point cloud are.

This is then followed by a depth calculation, which in this case calculated as the average z-coordinate of all of the isolated points deemed belonging to the object by the 2D network. This is carried out with the aid of OpenCV’s “DNN” module, the depth is then calculated using the cameras pipeline and the information is combined into an OpenCV label, Figure 4-3 and Figure 4-4 demonstrates an example of the results as a depiction of real-time detection. The OpenCV Label can be described as follows:

* Person: which ist he main label as a an inference result
* 78.92% represents the confidence of the inference for the case
* 0.34 is the distance from the camera in meters: in this case the reading is accurate

as the image exists in a single depth level, for real objects the point cloud data retrieved from x and y coordinates can be unordered and existent in various, and possible far away, z coordinate level.



Figure 4-3: person is the label, the percentage is the confidence of the prediction and the last 0.34 is the distance from the camera in meters.

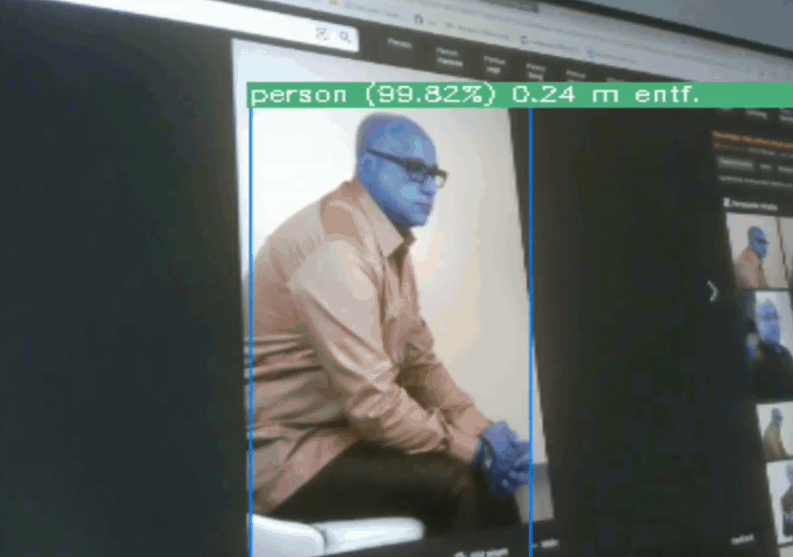


Figure 4-4: a person’s label as an inference result with the label, bounding box, confidence (score) and distance from the camera in meters.

### MobileNet-V2

The dataset is prepared from 2D images that are labeled in .xml format, the network is then trained to these images and labels and the checkpoint saved for inference. The 2D bounding box information is just as MobileNet-V1 used to retrieve the depth information using the corresponding z-coordinate. In the case of MobileNet-V2 this is done via the Tensorflow API (used for training as well as retrieving information such as labels, confidence for inference) and custom functions that leverage the output information from MobileNet-V2, combines it with the camera pipeline (pyrealsense) and throws out, in addition to labels and confidence scores, the distance to the object.

The network is not trained from scratch, as this will be extremely lengthy until meaningful results are reached, therefore Transfer Learning is deployed in this case. Particularly, a MobileNet-V2 that is trained on the COCO dataset, which is constituted from more than 91 classes that generate ‘general’ features. After setting the checkpoint to a trained COCO dataset, the last layer of the network is fine-tuned, i.e. the weights and biases are reset. The network is then trained in this setting on the net custom dataset, and as far as 2D methods go the results obtained are good.

### VoteNet, BoxNet, PointNet and PointNet++ :

To investigate how PointNet processes the data for classification an example is taken into consideration. As PointNet consumes a subsampled version of the original point cloud. Figure 4-5 illustrated the process of classifying an object by establishing geometric relations through PointNet’s symmetric functions.

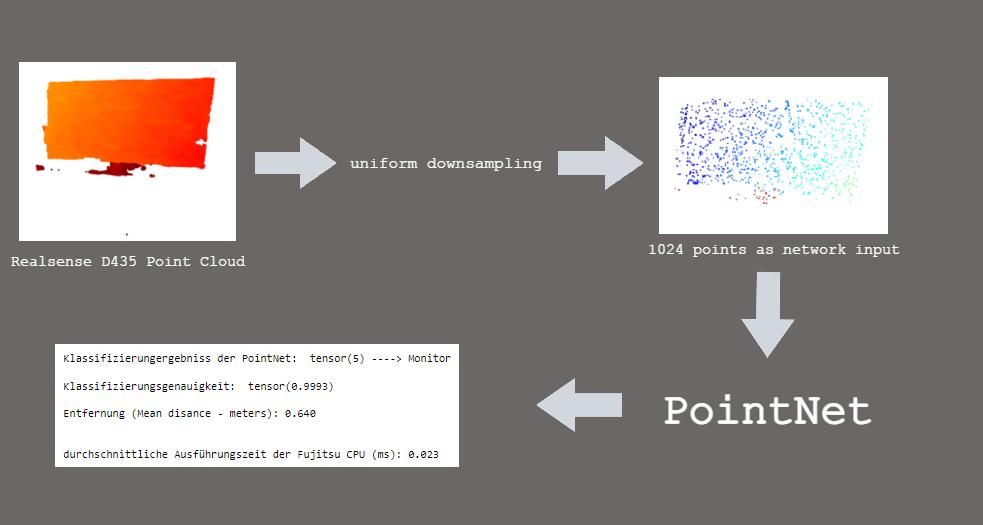


Figure 4-5: Classification process with PointNet

To deploy this on a larger scale, particularly, in point sets with several objects as well as to leverage this method for object detection PointNet++ is utilized as a backbone for most of 3D detection techniques that consumes the point set in an ‘as is’ fashion such as VoteNet.

An example of a pretrained VoteNet is shown in Figure 4-6, this network is trained with the benchmark SUNrgbd dataset (a dataset of indoor items), the point cloud in Figure 4-6 is one example of the validation set of the SUNrgbd dataset.

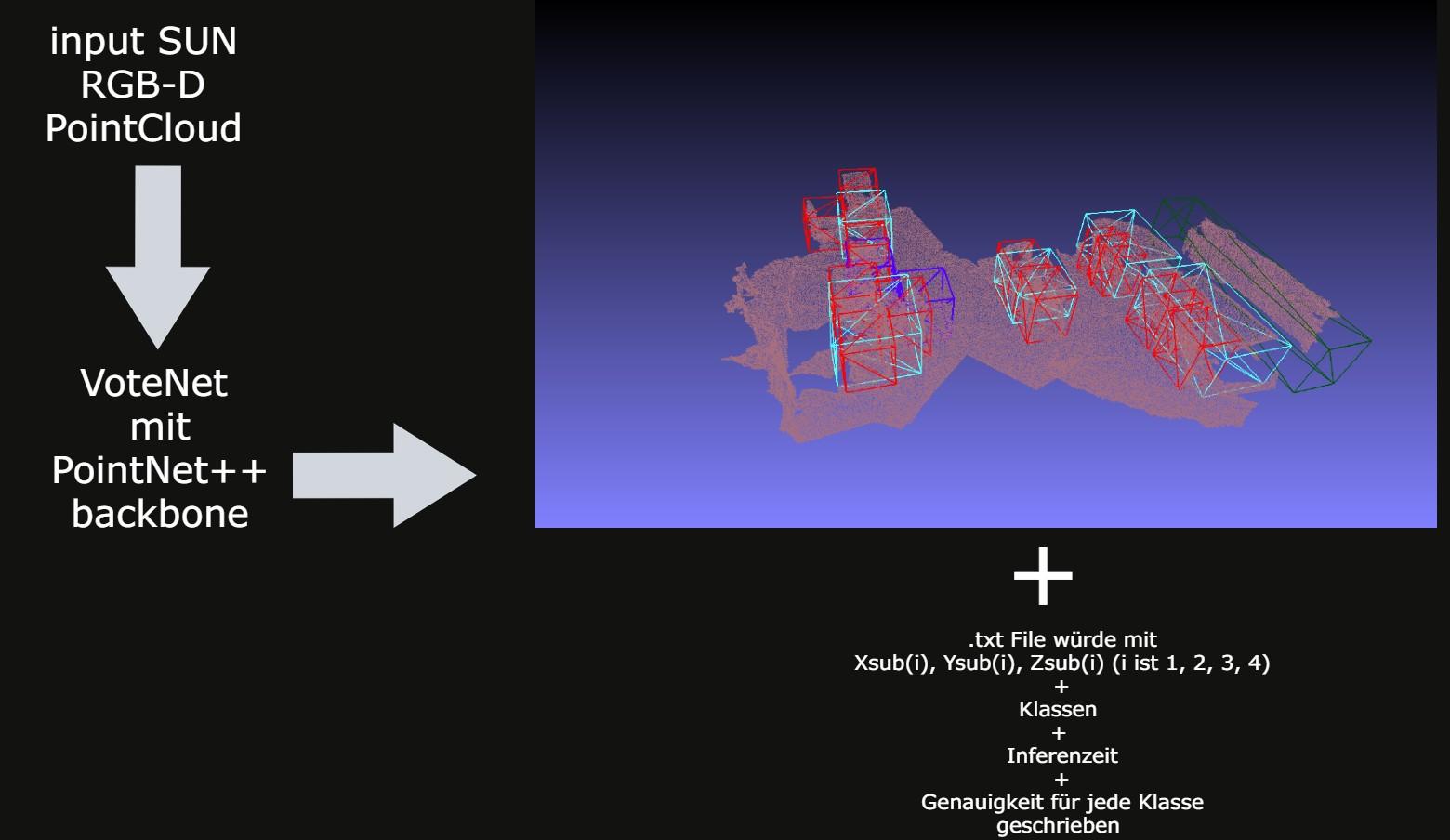


Figure 4-6: Point Cloud from the SUNrgbd’s validation set

the point cloud in Figure 4-7 is taken through the RealSense, i.e., the network has not seen the point cloud before and Figure 4-7 constitutes an inference example on a custom point cloud.

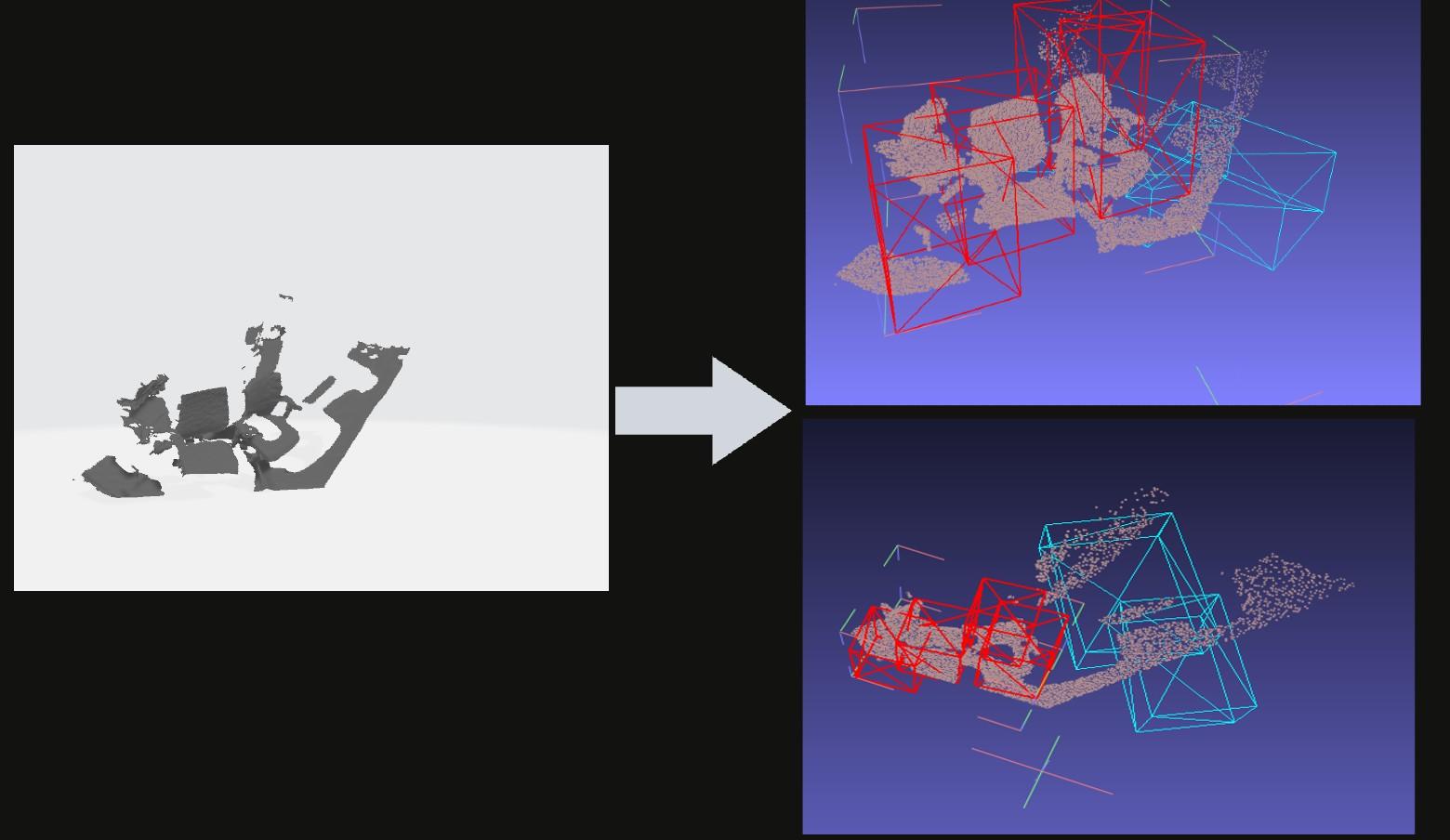


Figure 4-6: Point Cloud taken by the RealSense depth camera and set as an input to the model trained on SUNrgbd dataset

For the custom dataset and to annotate the data, LabelCloud is used, which is an Open Source software that generates 3D bounding boxes and labels for point clouds. The coordinates and the center of the bounding box are calculated in the camera coordinate frame and are then fed to the network as Ground Truth bounding boxes. Figure 4-8 illustrates an example snapshot of the Software and the point cloud.

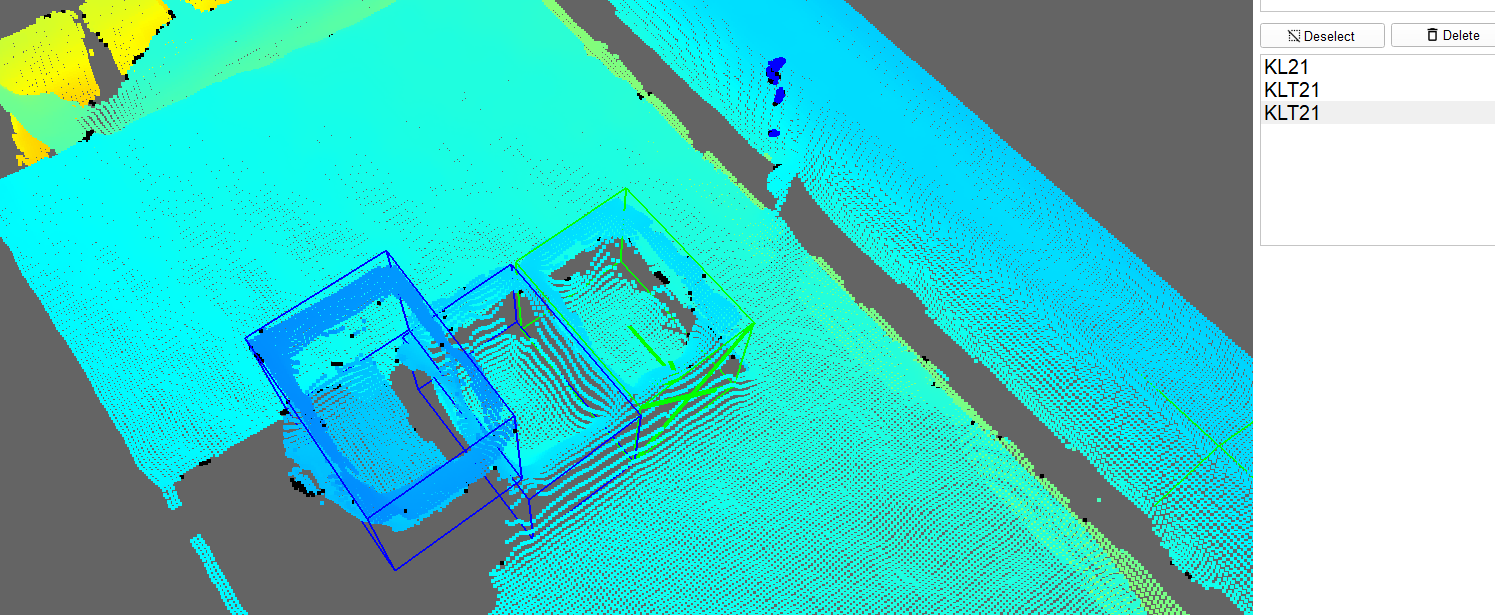


Figure 4-8: LabelCloud: the software used to generate labels for the point clouds

# Results

## MobilenetV2:

Table (1) demonstrates the results of the training.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | IoU | area | Detections (Maximum number) |  |
| Average Precision (AP) | 0.50:0.95 | All | 100 | = 0.915 |
| Average Recall (AR) | 0.50:0.95 | all | 100 | = 0.901 |

Table (1)

The loss fluctuates between 1.4 and 2.0 and settles at 1.4 in epoch ~50000

Table. (2) and Fig. (18) demonstrates the training summary of VoteNet:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # Epochs | subsampled points | batch | accuracy | box loss | mean loss | vote loss |
| 180 | 20000 | 10 | 0.997882 | 0.165145 | 3.764862 | 0.149045 |
| 1000 | 35000 | 8 | 0.997606 | 0.163451 | 3.843967 | 0.150954 |

As only preliminary results were required to establish the validity of the concept, comparing VoteNet to a benchmark is out of the scope of the project, it is however useful to see how votes affect the results. A proposal is to train BoxNet on the same dataset, since BoxNet does not leverage the votes and only utilizes the features obtained from PointNet++, the following shows how the 2 networks compare to one another after several training rounds:

BoxNet gives a 2.4-2.7 mean loss when trained from 180-700 epochs

VoteNet gives 3.5-3.8 mean loss when trained from 180-1000 epochs

Table (3)

This is a preliminary indication that BoxNet performs better on the given dataset. An example of a pretrained BoxNet on the custom dataset is illustrated in Fig. (18) as well as Fig. (19):

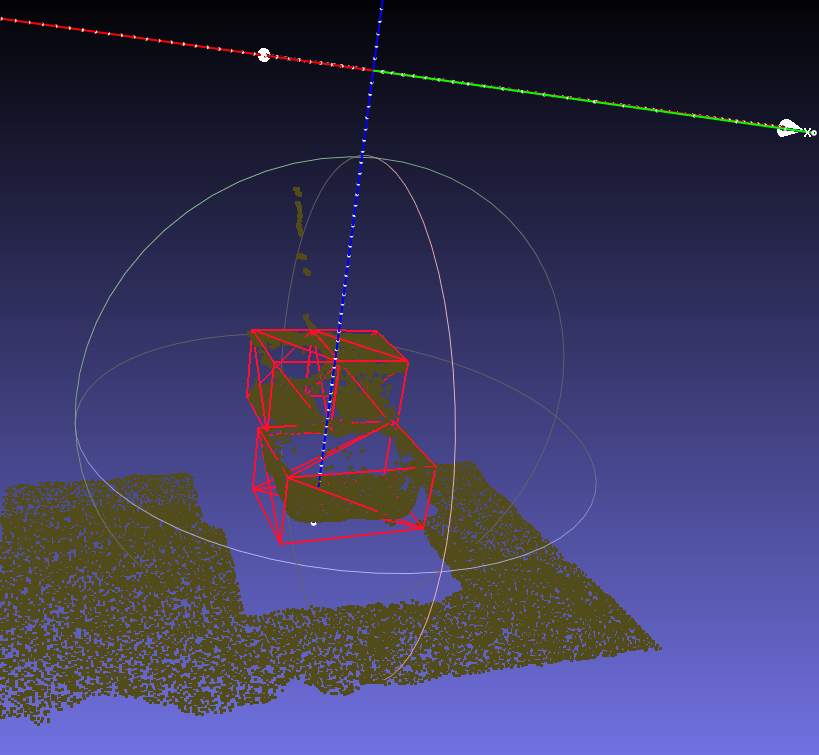


Fig. (18)

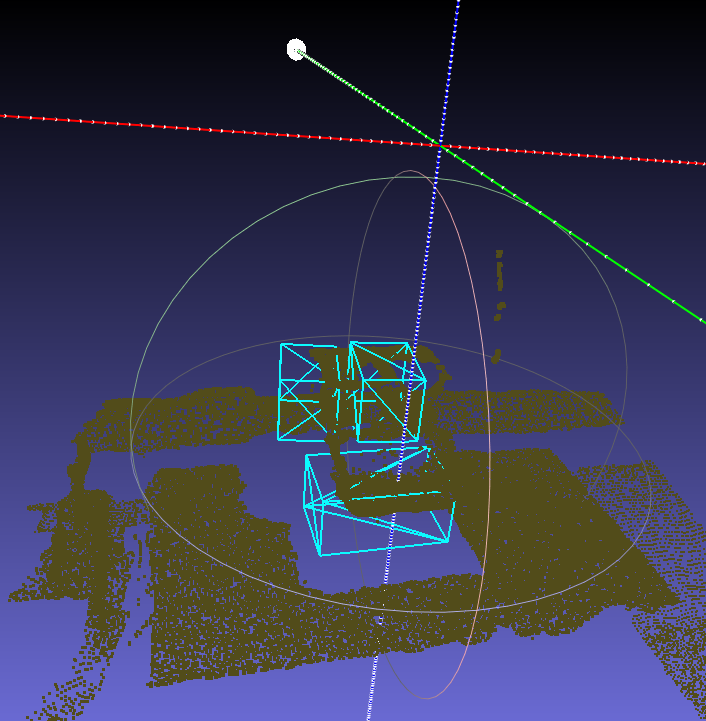


Fig. (19)

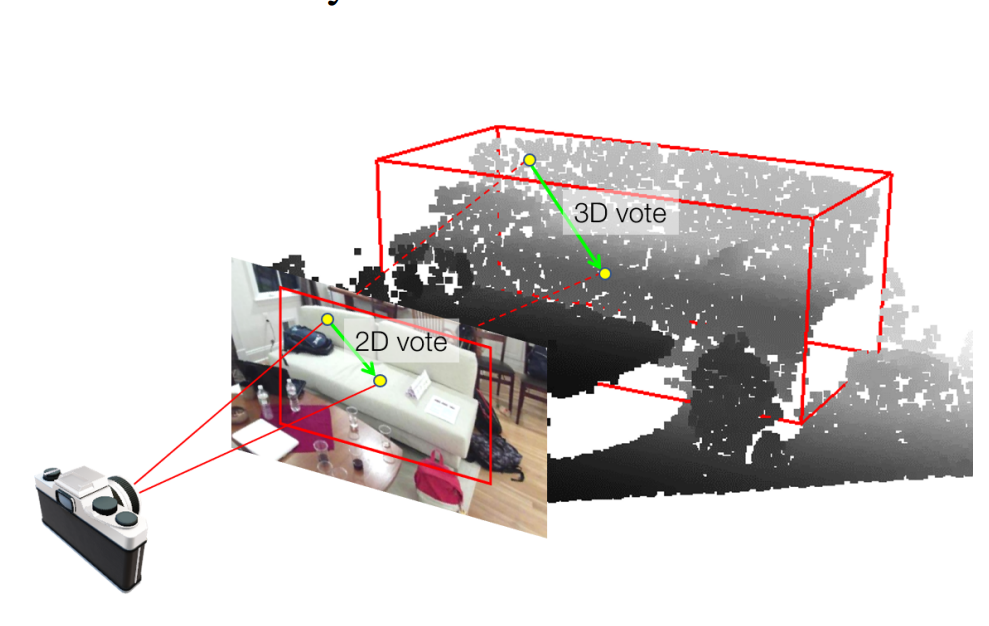
# Conclusion and Future Work

## Model

### Improving VoteNet

In this work the models employed were VoteNet and MobileNet V2, with a special focus on directly consuming the point cloud in VoteNet. VoteNet as aforementioned builds the relationship between the points in 3D space using PointNet++ and procedes to stack higher dimensional features, i.e. votes, learns these relationships and assigns the network parameters accordingly. This is carried out without regard to other data that may be captured by the camera, particularily, RGB channels data as well as image data. VoteNet in this regard consumes only the geometrical information embedded in the point cloud, that is, the x, y and z coordinates, rendering color and image information unused. A better model will try to not only leverage depth, but also image and color data.

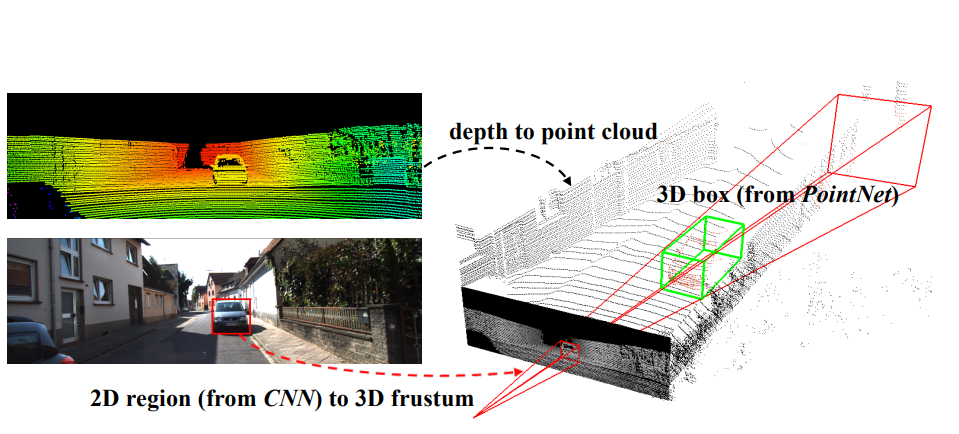
On the other hand ImVoteNet uses both the information from the RGB channel as well as the depth information to abstract features. ImVoteNet carries this out by explicitly extracting 2D geometric as well as semantic information, these extracted features are then fused with the 3D point cloud information with the aid of the camera parameters. Additionally and to improve the fusion of the data a multi-tower training scheme, Fig. () illustrates the basic structure of ImVoteNet, demonstrating how it can be applied to the case at hand.



Furthermore, other methods that also leverages both the RGB as well as the depth information include FrustumNet. FrustumNet uses 2D data to precisely localize the 3D bounding boxes, similar to VoteNet, FrustumNet consumes the point cloud directly and processes the 2D image seperately. Assigning objects in 3D space is a challenging, FrustumNet leverage the 2D information, establishes a relationship between the 2 representations and then uses that information to precisely loaclize the 3D bounding boxes in 3D space. The network assumes the existence of a projection matrix and is carried out in 3 parts:

* Frustum proposal
* 3D instance segmentation
* And 3D amodal bounding box estimation

Fig. () illustrates how the 2D image can be used to provide a precise way of the 3D bounding box proposal.

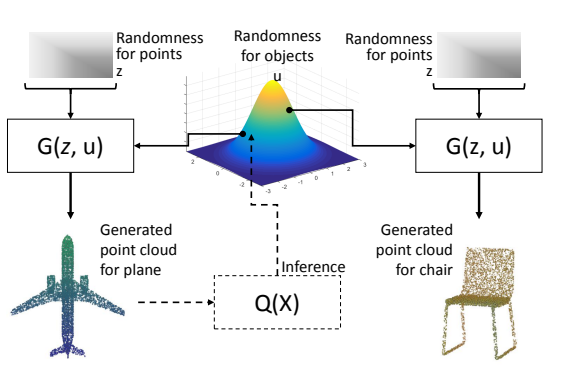


## Dataset

A main problem that prevented more accurate results, in addition to the information loss suggested in the previous section, is the dataset generation and the fact that the generated dataset did not include enough data nor features to fully make use of VoteNet. The process of setting the equipment, point cloud generation and labeling this point cloud consumes much time and it thus becomes more expensive as the dataset gets larger. To solve the problem of dataset size, Generative Adversarial Networks can be used. GANs are a set of Neural Networks that learns complex data distributions, encodes this data and then with the aid of Conditional GANs regenerates a similar set of data according to the calss assigned by the user. The vanilla GAN is not suitable to handle point clouds, they are more popular with 2D images. PC GAN however, proposes a method to learn from point clouds.

This will allow for a more dynamic choice of dataset size, reducing the effort that comes with physically generating a dataset. Fig. () shows the structure of PC-GAN.

(it is self-explanatory that distinct datasets have to be in part generated to then be expanded with GANs)



Furthermore and to diversify the dataset, providing more and different features for learning, a combination of a synthetic as well as a real dataset. This can in turn be provided to the PC-GAN network.

## Transfer Learning

Applyin transfer learning in the context of 3D data was not advised by the authors of VoteNet, this is most probably due to the fact that VoteNet has been trained on datasets that are not general enough to abstract global features that are reusable to a different dataset. To confirm this, this approach can be employed, proven that the training checkpoint comes from a general enough 3D dataset.

# Notes on working with the style sheet

This style sheet shows you the format of the report and provides also some information regarding the structure and the content. The introduction is followed by the main part of the scientific report. In this style sheet the chapters 2 - 4 and the appendix contain notes and examples on how to use the style sheet. The content in these chapters must be removed for your report. Your report ends with a summary and a outlook.

The fields "English title", "German title", "Student", “Matriculation number”, “Supervisor (Company)” and “Period of internship” on the front cover must be filled in. Then, the document has to be updated (mark the whole text "Ctrl + A" and then update with "F9"). Further fields (e. g. header of the task description) will be filled in automatically. In general, the fields to be filled out are red and in the case of fields that are automatically updated, black. Optionally, a logo of the company can be inserted on the front cover. If this option is not used, be sure that the placeholder is not printed (this case can occur with old WORD versions). Furthermore, the current date must be added in the two declarations. Before printing the work, all fields must be updated and checked again.

Further information about the project internship (Projektpraktikum) and the scientific report can be found in the OLAT course of the Graduate School CVT.

# Formatting and content

This chapter describes the formatting and other guidelines to write your report.

## General formatting and structure

Use only the font Arial. You can use up to 7 main chapters and subchapters of different levels to structure your report. The heading on the highest level must be formatted with the style "CVT H1" (12pt, bold). A new chapter of the first headline level always starts on a new page. The second level must be formatted with the style "CVT H2" (11pt, bold) and the third level with the style "CVT H3" (11pt, normal). This is the last level that appears in the table of contents. When using subchapters of any headline level, always use **at least two subchapters**. The fourth headline level must be formatted with the style "CVT H4" (11pt, underlined). More than four levels of headline are not allowed. **An introductory text must always be used between two heading levels** to explain the content of the chapter and you need some text at the end of the chapter witch a transition to the next chapter. A chapter must be at least half a page long. Shorter chapters are not allowed.

Normal text has to be formatted with the style "CVT Text" (11pt, normal, 1.3 - 1.5 times line spacing). Hyphenation should be activated. Formulas and calculations should be written in Arial italics. Formula symbols and abbreviations must be provided with the SI units used. General abbreviations (for example "e. g.") need not be listed in the list of abbreviations.

The typographical mistakes of "Widows and orphans" (see [[Dud17](#_CTVL001253c2400f08241bc983787c540ff0978)]) must be avoided. In the continuous text, this is guaranteed by the activated paragraph control in Word. In all other cases this must be observed manually (e. g. for headings).

## Bullet points and numerations

Bullet points and numerations are not allowed! You have to describe everything in continuous text!

## Figures and tables

Used figures should be inserted in the highest possible quality. When using figures created in other Microsoft programs, the figures should be inserted as "extended metafiles" (vector graphics). All figures should be formatted with the style "CVT Fig.".

Figures and tables must be mentioned at the correct position in the text. The reference must be linked to the figure or table via a cross-reference in Word (e. g. Figure 4‑1 or Table 4‑1; see also chapter 4.4).



Figure 7‑1: This is an example for a figure

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

Table 4‑1: This is an example for a table

(According to [[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)])

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

The labels of figures and tables can be created using the labeling function of Word (see Figure 4‑2).

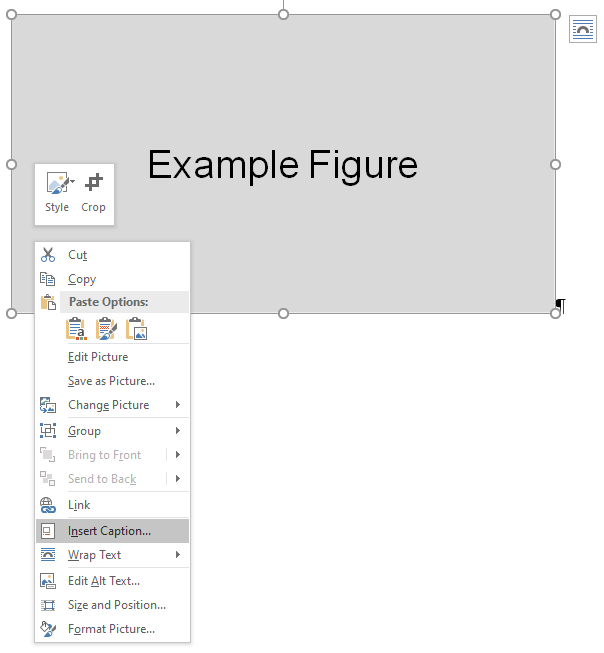


Figure 7‑2: Using the labeling function for figures in Word

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

The label text has to be formulated in a way that it can be understood by itself and has to be formatted with the style "CVT Fig. label" or "CVT Tab. label". A tab stop must be inserted between the figure no. or table no. and the labelling text instead of a space. Sources for figures and tables must always be mentioned but may not be listed in the list of figures or tables. For this purpose, a „style separator“ must separate the label text and the source (Ctrl + Alt + Enter) (see e. g. Figure 4‑1).

Figures and tables should be created by yourself. Attention should be paid to an attractive and uniform layout. The text in figures and tables should correspond to the style "CVT Text". The font size may be slightly smaller than in the continuous text (but not larger!). The original sources or sources containing the used content should be indicated by the expression "(According to [source])" (see Table 4‑1). Figures from sources must either be cited correctly (compare UrhG § 51; e. g. citations of figures for illustrative purposes are not permitted) or the permission (license) of the copyright owner is required (not necessarily the author, as the rights may have been transferred).

After tables, insert a blank line with the style "CVT Text" (Not for figures!). References for figures and tables must be inserted according to the scheme presented in chapter 5 in the figure label. The list of figures and tables is generated automatically (after updating).

## Formulas and references

**Formulas** are to be numbered continuously in the text. Analogous to figures and tables, the text must refer to the corresponding formula (see equation (1)).

|  |  |
| --- | --- |
|  | (1) |

The used formula symbols and abbreviations shall be listed alphabetically in the order upper case Latin, lower case Latin, upper case Greek, lower case Greek and, if applicable, sorted by subject groups in the list of formula symbols. The dimensions must be conform to the SI system of units. The list of formula symbols must be created manually.

**References** in the text to other chapters, figures, tables and formulas must be cross-referenced in the document (see Figure 4‑3 and Figure 4‑4).



Figure 7‑3: Inserting of cross-references in Word (Part 1)

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

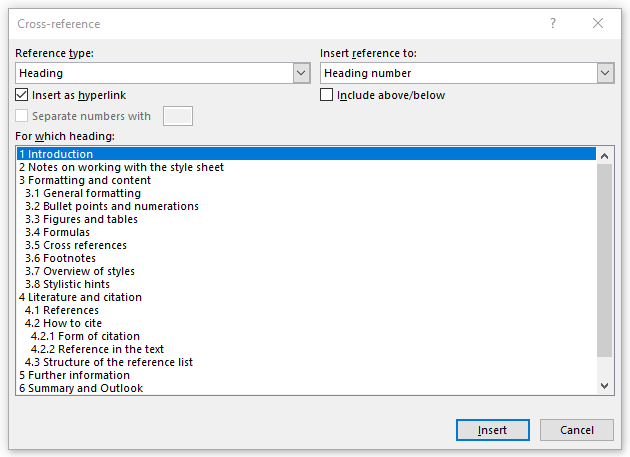


Figure 7‑4: Inserting of cross-references in Word (Part 2)

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

**Footnotes** can be used for further annotations to make the text more readable, but they should be used carefully. Footnotes must be created using the footnote function in Word (see Figure 4‑5) [[2]](#footnote-2).

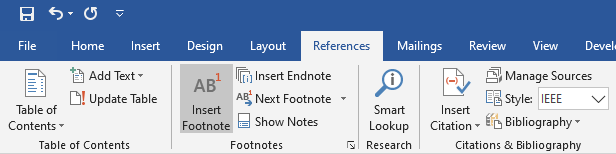


Figure 7‑5: Inserting of footnotes in Word

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

## Overview of styles

Table 4‑2 shows an overview of all possible styles and indicates their intended use.

Table 4‑2: Overview of all available styles

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

| **Name** | **Usage** |
| --- | --- |
| CVT B1 | Bullet list (Level 1) |
| CVT B2 | Bullet list (Level 2) |
| CVT B3 | Bullet list (Level 3) |
| CVT E1 | Enumeration (Level 1) |
| CVT E2 | Enumeration (Level 2) |
| CVT E3 | Enumeration (Level 3) |
| CVT H1 | Headline (Level 1) |
| CVT H2 | Headline (Level 2) |
| CVT H3 | Headline (Level 3) |
| CVT H4 | Headline (Level 4) |
| CVT Appendix H1 | Headline Appendix (Level 1) |
| CVT Appendix H2 | Headline Appendix (Level 2) |
| CVT Appendix H3 | Headline Appendix (Level 3) |
| CVT Project internship | - |
| CVT Fig. | Figures |
| CVT Fig. label | Label of figures |
| CVT Tab. label | Label of tables |
| CVT Text | Text |
| CVT H without No. | Headlines without numeration (e. g. list of figures) |
| CVT H without No.\_NotInTOC | Headlines without numeration, which should not appear in the TOC (Task description, Declaration and TOC) |



# Literature and citation

Works in the literary, scientific and artistic domain are protected in Germany by copyright and related rights. All rights (e. g. the right to reproduce and distribute) are owned by the author (but the rights of use may be transferred). However, exceptions have been created for scientific purposes (copyright restrictions). One possibility for this is the right of citation (§51 UrhG). The following chapters describe how to use citations in the text. For the requirements for the use of citations, however, please consult § 51 UrhG. The University Library also offers information and various courses on this topic on its homepage. Furthermore, you can learn more about this topic in the seminar “Scientific writing and publishing”.

## References

The used references have to be mentioned directly after a citation. If possible, the original sources should be cited. No sources should be cited without mentioning them in the text! For your scientific report, you have to use citation keys whose structure is described below. References in the text are indicated by a citation key, which consists of a name abbreviation and the year of publication, followed by the page number. Both are written in square brackets (e. g. [Kun16a] or [MKS16]). The page number is written inside the square brackets and is separated from the source by a comma (e. g. [[MKS16a](#_CTVL001b63a94de90a34e9890cd57278002b1b1" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 56]).

Name abbreviation

The structure of the name abbreviation depends on the number of authors (see Table 5‑1). There are special rules for sources without authors, which can be found in Table 5‑2.

Table 5‑1: Author abbreviations

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

|  |  |  |
| --- | --- | --- |
| **Number of authors** | **Example for authors** | **Name abbreviation** |
| 1 | Müller, Kurt | Mül |
| 2 | Müller, Kurt; Kunze, Heinrich | MK |
| 3 | Müller, Kurt; Kunze, Heinrich; Schmitt, Monika | MKS |
| >3 | Müller, Kurt; Kunze, Heinrich; Schmitt, Monika; Adam, Mark | MKS |

Table 5‑2: Special rules for author abbreviations

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

|  |  |  |
| --- | --- | --- |
| **Author** | **Example** | **Name abbreviation** |
| VDI Guidelines | VDI 2221 | VDI |
| DIN Standards | DIN 8580 | DIN |
| ISO Standards | ISO 9001 | ISO |
| EN Standards | EN 60204-1 | EN |
| Patent specification | DE102006013662A1 | DE |

Year of publication

The year of publication must be cited according to Table 5‑3.

Table 5‑3: Year of publication

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

|  |  |  |
| --- | --- | --- |
|  | **Example year** | **Example year of publication** |
| One publication of one author in one year | 2009 | 09 |
| Several publications of one author in the same year | 2009 | 09a, 09b, 09c, … |

Please pay attention when writing and referencing to the quality of the sources. If possible, avoid websites and use high-quality references such as conference or journal papers or dissertations. A citation of Wikipedia is generally NOT allowed.

## How to cite

This chapter explains how to cite and how to insert a reference.

### Reference in the text

With the **citation form**, a distinction can be made between a direct and an indirect citation. In the case of a direct citation, the text is taken literally and placed in quotation marks. The following applies:

* Errors of any kind in the cited text must be accepted, but may be indicated by the expression "[sic]" (lat.: sīc erat scriptum, engl.: as it was written) directly after the error.
* The omission of parts from the cited text must be indicated by the expression "[...]".
* Direct citations should only be used for important statements or definitions.

In the case of an indirect citation, the content of the cited text must be reproduced in your own words (paraphrase). When mentioning the authors in the text, only the first author with the addition “et al.” should be mentioned. (e. g. „Müller et al. verified that… “ [[MKS16a](#_CTVL001b63a94de90a34e9890cd57278002b1b1" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 25]). In general, the indirect citation is preferable. For both forms of citation, the source must always be mentioned.

If the source refers to a sentence, the source is located within the sentence (e. g. „Text [[MKS16a](#_CTVL001b63a94de90a34e9890cd57278002b1b1" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)].“). However, if the source refers to a whole paragraph, the source is written at the end of the paragraph (e. g. „Text. [MKS16]“). Several sources must be indicated within square brackets and separated by a semicolon (e. g. [[MKS16a](#_CTVL001b63a94de90a34e9890cd57278002b1b1" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.); [Mül16d](#_CTVL001d011b957c51f4bd7910632a49ff92a98" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]).

Important statements in the work must always be supported by several sources. The primary sources, i.e. the original sources and non-recited citations, must be used.

## Structure of the reference list

In the reference list, the references mentioned in the text are listed in alphabetical order of the citation key. Here a distinction is made between the document types shown in Table 5‑4.

Table 5‑4: Different types of literature

[[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.)]

| **Document types** | **Example** |
| --- | --- |
| Book (monograph), 1 author | [[Mül16d](#_CTVL001d011b957c51f4bd7910632a49ff92a98" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 15] |
| Book (monograph), 2 authors | [[MK16](#_CTVL001e64e2b70126a4ecaa267d56ce2ea7933" \o "[MK16]Müller, Kurt; Kunze, Heinrich: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 14-20] |
| Book (monograph), 3 authors | [[MKS16a](#_CTVL001b63a94de90a34e9890cd57278002b1b1" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 2-3] |
| Book (monograph), >3 authors | [[MKS16b](#_CTVL0018a646d34e77849bf80e30afef9cf7772" \o "[MKS16]Müller, Kurt; Kunze, Heinrich; Schmitt, Monika; Klein, Manfred: Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 2-3] |
| Edited Book (to be avoided) | [[Mül16c](#_CTVL0010d15640224bd416683550c4fde579fee" \o "[Mül16]Müller, Kurt (Hrsg): Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, 2016.), p. 44-46] |
| Chapter in an edited Book | [[Kun16a](#_CTVL0017f9eddde7d7e450eb77d85ea3655d0dc" \o "[Kun16]Kunze, Heinrich: Mustertitel. In: Müller, Kurt (Hrsg): Mustertitel - Musteruntertitel. X. Auflage, Musterort: Musterverlag, XY - XY, 2016.), p.1-5] |
| Website | [[Mül16a](#_CTVL001ae362dea7ebe4612b3b1a6b57e5af804" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. URL: http://www.musterinternetseite.de - 03.06.2016.)] |
| Conference presentation or paper (without conference proceedings, otherwise like a chapter in an edited book) | [[Mül16e](#_CTVL0012dbbb12d28964ec4a8978a7c284fc12f" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. Musterkonferenz XY, Musterort, 2016.), p.45] |
| Dissertation (thesis) | [[Mül16f](#_CTVL001ac135d76d7fc4587a8800081295e89a0" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. Dissertation, Musterhochschule, Musterort, 2016.), p.78] |
| Journal article | [[Mül16b](#_CTVL001a49eea604b5d4413ac6abb0c8c2cd97c" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. In: Musterzeitschrift, XXX (2016), YY, XY-XY.), p.7-8] |
| Standards and guidelines | [[VDI93](#_CTVL001ab859f76c6124d25b071762f5894633d" \o "[VDI93]VDIRichtlinie, 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Berlin: Beuth-Verlag, 1993.), p.4] |
| Gray literature (to be avoided) | [[Mül16g](#_CTVL001c1979c349a5642798c97008465ef8291" \o "[Mül16]Müller, Kurt: Mustertitel - Musteruntertitel. Musterinstitution, Musterort, 2016.), p.25] |
| Personal communication | [[Kun16b](#_CTVL001a356a74d89714e048082df46ed742f42" \o "[Kun16] Kunze, Heinrich: Mustertitel. Im Gespräch mit Müller, Kurt. Musterort 01.01.2016.), p.9] |
| Patent | [[Kun17](#_CTVL0019408c32f071343d8851309bbbdd18c42" \o "[Kun17]Kunze, Heinrich: Beispielpatent. Erfinder: Müller, Kurt. Anmeldedatum: 01.01.2017, DE000000000000A1, 01.06.2017.), p.5] |

The citation of an entire edited book should be avoided. Instead, the respective chapter in the edited book should be cited.

The structure of the references in the reference list for each document type can be found in the exemplary reference list at the end of this document. Thereby, the example in Table 5‑4 refers to the corresponding entry.

Please note the following for the references in the reference list:

Authors

* Authors are always mentioned without their academic degree (e. g. Dr.-Ing.) and their title. Titles of nobility should be mentioned.
* In case of missing authors, the respective institution, authority, etc. is mentioned as author.
* In the case of edited books, only the first three editors should be mentioned. Further editors are indicated with the addition "et al.". For all other document types, all authors must be mentioned.

Title

* If available, the subtitle should also be mentioned.

Edition

* The edition must always be indicated. If possible, the most recent edition should be used.
* The edition must be uniformly mentioned by the expression “x. edition” in the reference list (Incorrect expressions are e. g. „x. ed.“ or „x., updated edition“).

Place and year of publication and publisher

* The place, year and publisher of a publication must always be mentioned if this is required by the respective document type.
* If the place, year and publisher of publication are missing, the expressions "w. pl.“ (without place), "w. y.“ (without year) and "w. pu.“ (without publisher) can be used. If possible, the year of publication should be approximated.
* In the case of two places of publication, the two locations are separated by a comma (e. g. "Berlin, Heidelberg"). If there are more than two places of publication, only the first place should be mentioned with the addition "et al." (e. g. "Berlin et al.").
* In the case of several publications from the same publisher, ensure that the publisher is always mentioned in the same way (e. g. uniformly "Carl Hanser Verlag" instead of "Carl Hanser" and "Hanser Verlag").

# Summary and Outlook

(Max. 4 pages).

# 

# References

[QSMK17] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. . "Pointnet: Deep learning on point sets for 3d classification and segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652-660. 2017.

[QYS17] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." *Advances in neural information processing systems* 30 (2017).

[QLH19] Qi, C. R., Litany, O., He, K., & Guibas, L. J. "Deep hough voting for 3d object detection in point clouds." In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277-9286. 2019.

[HZCB17] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[ZSY20] Zhang, Z., Sun, B., Yang, H., & Huang, Q. "H3dnet: 3d object detection using hybrid geometric primitives." In *European Conference on Computer Vision*, pp. 311-329. Springer, Cham, 2020.

[ZT18] Zhou, Y., & Tuzel, O. "Voxelnet: End-to-end learning for point cloud based 3d object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490-4499. 2018.

[IBM22] IBM: Convolutional Neural Networks URL: <https://www.ibm.com/cloud/learn/convolutional-neural-networks#toc-types-of-c-yL2bT7qZ> – 01.09.2022

[PS22] Pepik, B.; Stark, M.: 3D Object Detection. URL: <https://www.mpi-inf.mpg.de/news/spotlights/understanding-images-videos/3d-object-detection/> - 01.09.2022

[Uni22] Universal Robots: Products. URL: <https://www.universal-robots.com/products/> - 01.09.2022

[Rea22] Intel RealSense: D435. URL: [https://www.intelrealsense.com/depth-camera-d435 - 01.09.2022](https://www.intelrealsense.com/depth-camera-d435%20-%2001.09.2022)

[ZLL21] Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. "Dive into deep learning." Berkely: *arXiv preprint arXiv:2106.11342*, 2021.

[SHZ18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520) 2018.

[OGM09] Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., & Serrano, L. (Eds.) *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI global, 2009.

[Bha22] Bhavsar, P.: An Ultimate Guide to Transfer Learning in NLP, URL: <https://www.topbots.com/transfer-learning-in-nlp/> - 15.09.2022

[Hal22] Hal, K.: VoteNet. URL: [https://blog.csdn.net/wqwqqwqw1231/article/details/103374515 - 15.09.2022](https://blog.csdn.net/wqwqqwqw1231/article/details/103374515%20-%2015.09.2022)

[Cho17] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251-1258. 2017.

[KAK22] Kadam, K. D., Ahirrao, S., & Kotecha, K. "Efficient approach towards detection and identification of copy move and image splicing forgeries using mask R-CNN with MobileNet V1." *Computational Intelligence and Neuroscience* 2022.

[Hol22] Hollemans, M.: Mobilenet version 2, URL: <https://machinethink.net/blog/mobilenet-v2/> - 15.09.2022

[IK88] Illingworth, J., & Kittler, J. "A survey of the Hough transform." *Computer vision, graphics, and image processing* 44, no. 1 (1988): 87-116.

[LLS08] Leibe, B., Leonardis, A., & Schiele, B. "Robust object detection with interleaved categorization and segmentation." *International journal of computer vision* 77, no. 1 (2008): 259-289.

[OD08] Olson, David L.; and Delen, Dursun (2008); Advanced Data Mining Techniques, Springer, 1st edition (February 1, 2008), page 138, [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [3-540-76916-1](https://en.wikipedia.org/wiki/Special:BookSources/3-540-76916-1)

[Sha22] Shah,D.: Mean Average Precision Explained: Everything you need to know. URL:<https://www.v7labs.com/blog/mean-average-precision#:~:text=Average%20Precision%20is%20calculated%20as,mAP%20varies%20in%20different%20contexts>. – 01.10.2022

[Dud17] Duden: Duden Online Wörterbuch. URL: http://www.duden.de/ - 16.05.2017.

[Kun16a] Kunze, Heinrich: Example\_Title. In: Müller, Kurt (ed.): Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, pp - pp, 2016.

[Kun16b] Kunze, Heinrich: Example\_Title. Personal Communication with Müller, Kurt. Example\_place, 01.01.2016.

[Kun17] Kunze, Heinrich: Example\_Patent\_title. Inventor: Müller, Kurt. Date: 01.01.2017, DE000000000000A1, 01.06.2017.

[MK16] Müller, Kurt; Kunze, Heinrich: Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, 2016.

[MKS16a] Müller, Kurt; Kunze, Heinrich; Schmitt, Monika: Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, 2016.

[MKS16b] Müller, Kurt; Kunze, Heinrich; Schmitt, Monika; Klein, Manfred: Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, 2016.

[Mül16a] Müller, Kurt: Example\_Title - Example\_Subtitle. URL: http://www.example\_url.de - 03.06.2016.

[Mül16b] Müller, Kurt: Example\_Title - Example\_Subtitle. In: Example\_journal, Example\_ Volume (2016), Example\_issue\_number, pp-pp.

[Mül16c] Müller, Kurt (Hrsg): Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, 2016.

[Mül16d] Müller, Kurt: Example\_Title - Example\_Subtitle. X. edition, Example\_place: Example\_publisher, 2016.

[Mül16e] Müller, Kurt: Example\_Title - Example\_Subtitle. Example\_conference, Example\_place, 2016.

[Mül16f] Müller, Kurt: Example\_Title - Example\_Subtitle. Dissertation, Example\_university, Example\_place, 2016.

[Mül16g] Müller, Kurt: Example\_Title - Example\_Subtitle. Example\_institution, Example\_place, 2016.

[VDI93] VDI Guideline 2221 Part 1: Design of technical products and systems. Berlin: Beuth-Verlag, 2019.

1. Appendix

The headings of the Appendix shall be created using the styles "CVT Appendix H1", "CVT Appendix H2" and " CVT Appendix H3". The figures and tables shall be numbered according to the scheme "A-1, A-2, A-3, B-4, B-5, B-6 etc.". Due to the different numbering of the figures and tables in the appendix, the headings may not be inserted as in the previous part of this report, but must be copied from the following examples. Otherwise, this will result in incorrect representations.

* 1. Appendix Example
     1. Appendix Example

Table A‑1: Table 1 in appendix A

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



Figure A‑1: Figure 1 in appendix A

Table A‑2: Table 2 in appendix A

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



Figure A‑2: Figure 2 in appendix A

* + 1. Appendix Example
  1. Appendix Example

1. The signatory's affiliation with the company must be confirmed by a company stamp. This can be omitted if the signature matches the signature on the internship certificate. [↑](#footnote-ref-1)
2. This is an example for a footnote [↑](#footnote-ref-2)