TECHNISCHE UNIVERSITÄT CHEMNITZ

Driving Scenario Based Traffic Sign Recognition using Image processing /Machine Learning

Internship Research Report

Chair of Computer Engineering

Dept. of Computer Science

Submitted by: Abdulrahman Sweidat
MTR Nr.: 513012
Submission date: 22.03.2020

Supervising tutor: Prof. Dr. W. Hardt
                    Julkar nine

Acknowledgment

Foremost, I have to express my sincere gratitude to my technical supervisor Mr. Julkar nine who also was my advisor, my mentor and my beacon for the guidance whenever I was stuck during my internship. His patience, motivation, enthusiasm and immense knowledge was a lighthouse for me to finish my internship within time and efficiency. Moreover, his feedback on how to approach my report has helped me in writing my report. I could not have imagined having a better advisor and mentorship for my internship.

Besides my advisor, I would like to thank the rest of my project partner at IKON project in Laubusch especially: Khubaib Abbasi for his great work with me and insightful comments, and invaluable suggestions during the project.

I would also like to thank my course director, Prof. Dr. Wolfram Hardt, who allowed me to do my mandatory research internship in International project in Laubusch (IKON Project).

I thank my friends at TU Chemnitz for their support and motivation throughout my education. In particular, I am grateful to Rahul for enlightening my path with his professional expertise and guiding me toward the completion of the master's program.

Last but not least, I would like to thank my family: my dad, my mom for supporting me spiritually throughout my life.

Abstract

The motivation behind this internship at IKON Project in laubusch [1] was to support students for their successful study. Main aim of the project is to give opportunity to students to work in a team at international level and get experience with practical tasks. My task was assigned to design system to detect and recognize traffic sign which used in Autonomous vehicles during autonomous driving in the artificial intelligence domain that should be efficient and could be merged into other ADAS system. Firstly, I had to perform the R&D to grab the core knowledge and prepare the DS for the project. I had a task to analyses different dataset for Traffic signs and their features to extract the useful data that can be ingested into the current pipeline of the product, After the data ingestion into the pipeline it will be processed by machine learning techniques by using their build tools and algorithms and get the final intelligent classifier model to recognize desired results. One of the main goals of this AI model is to make is robust, easily maintainable, for that purpose I used the computer vision and Deep learning techniques to combine all components of the application together as much as possible and dockerize each component to make it easy to migrate to any setup and configure on suitable hardware like raspberry pi in the future when the need arises. As an intern, it was important to get familiarized with tools and technologies being adopted by our development team before starting the project. The agile method of software development called Scrum was employed in project management. This methodology being iterative development inspired the swift and flexible response to any changes. Additionally, Test-driven development was deployed by writing the unit and integration tests. During the course of the internship, all the assigned tasks were executed with positive results in various phases. The comprehensive overview of this internship is depicted in different sections of this report.

Keywords: SDLC, R&D, ADAS, DS, CV, DL, Agile, Scrum, ML, raspberry pi, AI.

Contents

List of Figures

List of Abbreviation

PO   Product Owner

SM   Scrum Master

GUI   Graphical User Interface

TS   Traffic Signs

IDE   Integrated Development Environment

SDLC Software Development Life Cycle

US   United States of America

EU   European Union

AI   Artificial intelligence

TSD Traffic Sign Detection

DS   Dataset

ML   Machine learning

DL   Deep Learning

CNN Convolutional Neural Network

TDD Test Driven Development

R andD Research and Development

TSR   Traffic Sign Recognizing

TU   Technische Universität

1 Introduction

TU Chemnitz is one of the biggest German universities and IKON project [2] is an international project combined between TU Chemnitz in Germany and Czech Technical University in Prague and other companies from industry side. The main aim of the project is to give opportunity to students to work in a team at international level and get experience with practical tasks. IKON project which enables it to get knowledge and experience from business with a lot of companies around Germany and international students and professors and some of them have excellent reputation in the world and their names are issued for the public, my task was to download the information from US and EU sanctions library and fetch the most effective research related to my project names of most research and later check them with our database whether we are doing any business with them or not. The data and researches were not prepared and unable to be read by machine unless customized and summarize we applied summarized a lot of data collection and research related to my task , and parsers to get meaningful data and information. After successful data extraction, following datasets were created.

1. German Traffic Sign Dataset Benchmark (GTSDB).

These were further refined using data cleaning techniques so that parallelism could be arranged between them. The datasets were quite large most of them reaching 50K records each. We modified the existing tool called Excel sheet and other websites according to our needs so that we could obtain the desired result in the minimal time possible, which is further discussed how we did it. Once the basic dataset was ready, it was modularized into Deep learning Model as classifier to making decision which traffic is this; this also ensures the stability, maintainability, and reliability of trained model. Further, the application was automated through the computer vision techniques (TS detection) and Artificial Intelligences (Neural Network Model Recognizer) scripting which didn't involve any human involvement and ran as a sequence mode. After the successful development and testing of the application, the application now automatically generates a detection of TS from videos and Recognize this TS by CNN model which is trained already before to achieve the result in high accurate . Furthermore, the application works good in raspberry pi 3 which we used it in our project , which means it is a great way and easy to configure in other hardware which using in ADAS applications. Throughout the SDLC cycle, the Test Driven Development approach was carried out. Unit testing and Integration Testing were carried out for each task and process. The application was documented properly and thoroughly throughout the SDLC cycle on the Confluence.

2 Project overview

The Internship and our project are not just coding, not just documentation, not just thinking or writing algorithms. All researches projects goes through multiple phases to accomplish what was really required of it and thus called SDLC. We will be discussing the phases of the internship project and also the tasks that were performed during those phases.

2.1 Data collection

To start working on the project I needed some input or dataset to achieve the project and go through processes and I would find on different websites on the internet in different document formats but I have chosen one website called German Traffic sign Dataset Benchmark (GTSDB)[3]. Once the data set was available and installed, I analyzed the dataset and started to look for meaningful and useful data that could be ingested into our pipeline, in the website which we followed you will find more than 43 TS in Germany and just we have chosen 4 TS because there is a limitation for hardware and time.

2.2 Data preparation and Extraction

Once the data was analyzed and prepared, it became easier to fetch it from the documents using different techniques. Realizing what was needed to be done is the real headache once that was clear I started doing it and finding different ways to do it. Following is how I extracted the meaningful data which we call it later CSV files Preparation.

As they were multiple sources where data need to be extracted from it. We applied different techniques to get the data from it

2.2.1 Data Parsing

Different data parsers were created according to document specifications that were used online website called "zamzar"[4] to convert dataset from different ways into CSV files which is required to accomplish our application .

2.3 Data Preprocessing

After data extraction, many different unstructured datasets [11] were created. So the next step was to make datasets structured and improve the quality of the data

by applying data cleaning techniques [9]. When the required quality of data was achieved, the next step was to ingest that data into the system. With the help to be easier using and fetching in TS detection and TS recognizing model was modified according to our requirement to get the desired result. This needed a lot of refactoring; changing the CSV files structure and data insights.

2.4 Traffic Sign Detection

the traffic sign detection. It describes the fetching and characterize of the road signs, the requirements and difficulties behind road signs detection is how to deal with outdoor images, and the different techniques but we used normal way to detect the signs we call it the image segmentation based on normally matching with images and template which we have in our dataset and extract these are template (TS) from images. the color analysis, shape analysis in our project is neglect. Although image processing plays a central role in the road signs recognition, especially in color analysis, but unfortuinalty there is to many problems regarding the stability of the received information of colors, variations of these colors with respect to the daylight conditions and we will see next how it is detecting from images [2].

2.4.1 Detection algorithm

Template Matching is a method for searching and finding the location of a template image in a larger image. OpenCV comes with a function cv2.matchTemplate() for this purpose. It simply slides the template image over the input image (as in 2D convolution) and compares the template and patch of input image under the template image. Several comparison methods are implemented in OpenCV[6].

2.5 Traffic Sign Recognizing

Traffic sign recognizing is one of the most techniques in ADAS , after we detect the signs from the images in previous step ,now we have to know which kind of sign is this for example "speed limit", "stop" or any traffic sign which assigned in the road for these step we have to use something called machine learning –Deep learning which is a part of machine learning , in the deep learning techniques we used convolutional neural network (CNN), it is an algorithm we used it to train the model by images or any complicated datasets , from these algorithm we can build a model and trained this model to classify which sign is this[7] .

2.5.1 Convolutional Neural Network (CNN)

It is one of the most impressive forms of ANN architecture; it is Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture, offers a simplified. method of getting started with ANNs it consists of four main layers [7]:

1- Input Layer: which hold pixel value for input images.
2- The convolutional layer : determined output neurons.
3- The pooling layer: perform down sampling the spatial input and reducing number of parameter.
4- The fully-connected layers: this is the final layer and it determined the output result shape from the activation to be used in classification.

2.6 Predicting Results and Testing

In predicting results stage we combine two previous algorithms together "Detecting and recognizing algorithms" we can detect the road signs and recognizing it by model for example the following signs:

Real Image	Detection Signs	Recognize Signs
		00000.ppm (36, 42, 3) <class 'numpy.ndarray'> Priority at next intersection elapsed time 43.114745400000004
		00005.ppm (115, 113, 3) <class 'numpy.ndarray'> No overtaking

Figure1: TSD && TSR [1]

2.7 Hardware side Configuration on Raspberry Pi 3B+

The Raspberry Pi is a small single computer; it has two model A and B models. Model B became far more popular than anticipated, selling outside its target market for uses such as robotics, Automotive and Automobile. It does not has peripherals like keyboards it has a different specification like 1GB RAM and 16GB SSD card Unfortunately it has limitation to configure a big application, in our project we used raspberry Pi 3 B+.

Figure1: Raspberry Pi 3 B+[1]

2.8 Compare Result Between PC VS Raspberry Pi 3B+

Hardware platform	Detection Framework	mAP	FPS	Test Time Per Image	Size Of Model	Number of objects	Implement Type
PC	CNN	46.3	0.1	48 sec	8 MB	4500	PC
Raspberry PI 3B+	CNN	86.1	0.1	82 sec	8MB	4500	Raspberry pi

Figure1: Comparing results PC Vs raspberry PI

2.9 Documentation

One of the important steps in our internship project is well-documented project. When required, the team can easily understand the project and continue with the project with the moment go. The following documentation was added during the Internship at Confluence.

1. Results of the R&D

2. A detailed description of algorithms was created on the confluence which contained information and comments about what and how the functions are works and which inputs are needed for them to get the result from the system.

3. The Technical instruction set of the project and system manual.

4. Backup of whole system codes and libraries.

3 Software Development Life Cycle - SDLC

In order to make the product more efficient, scalable, and maintainable and reduce the development time and cost, many professional tools were used throughout the development process. The description of these tools with respect to the work done during the internship will be provided in this section.

3.1 Anaconda

Anaconda Distribution tool is open-source community development framework and it has more than 20 million users worldwide. It is the tool is using by data scientists machine learning engineers who want to use Python or R for scientific computing projects it has a lot of development IDEs for example Spyder IDE ,Jupyter Notebook…etc .
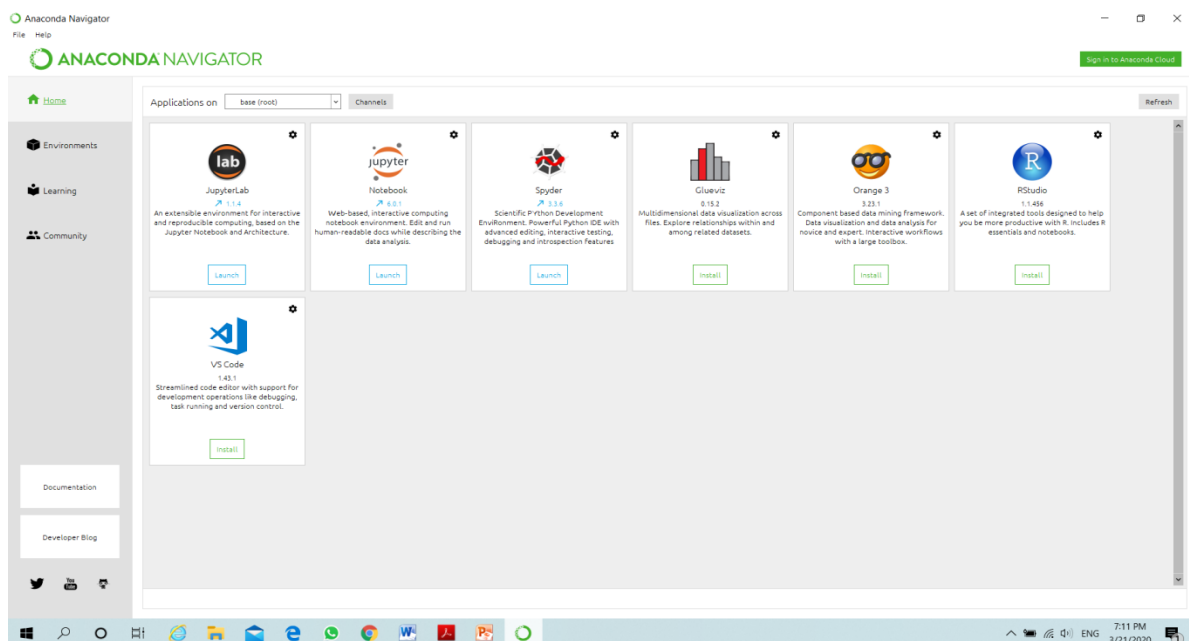

Figure 3.1: Working with Anaconda

3.2 Spyder IDE 3

Spyder IDE is a development tool kit used to development python application, it is really excellent and easy to use tool for this reason I have chosen this tool in my project.
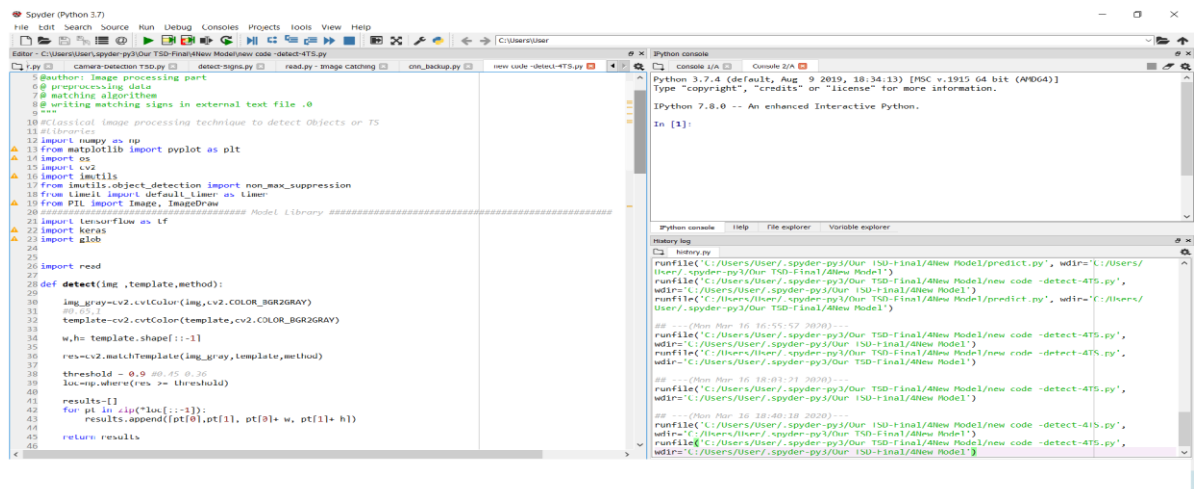

Figure 3.2: Working with Spyder IDE

3.3 Jupyter Notebook

Jupyter notebook is another development tool kit we used it to build a machine learning model or we call it classifier because it easy to test and run environment as the following:
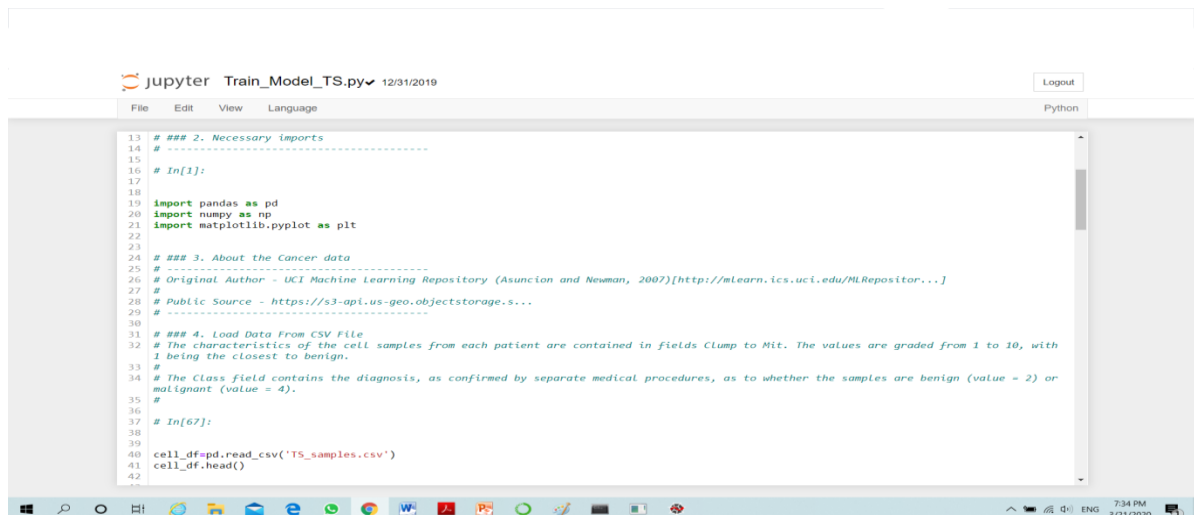

Figure 3.3: Working with Jupyter Notebook

4 Language and Frameworks

In our internship was done the Application development in one main Language we call it Python were used in the Internship. Python was used for developing the core application. Further different frameworks and libraries of python language were used according to the project requirements. This section will give a detailed description of the libraries and frameworks used during the internship.

4.1 Python

Python[8] is a high-level object-oriented scripting language. Python syntax is similar to English keywords, which are easy to interpret and has fewer syntax complications compare to any other programming language. With some past experience in python and the reason stated above made me opt for Python for the core app. It comes with a lot of amazing libraries. A few of the libraries I used are listed below.

4.1.1 Tensorflow

Tensorflow [9] is an interface for ML algorithms and an implementation for executing AI algorithms.it can executed with little or no change on a wide variety of heterogeneous systems, ranging from mobile devices such as phones and tablets .The system is flexible and can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models, and it has been used for conducting research and for deploying machine learning systems into production.

4.1.2 Keras

Keras [10] is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

4.1.3 OpenCv
OpenCV (Open Source Computer Vision Library), library of programming functions mainly aimed at real-time computer vision,OpenCV supports some models from deep learning frameworks like Tensor Flow, Torch, PyTorch ...etc [6].

4.1.4 Imutils

It is a is library allows developers to detecting objects form images and find features for templates.

4.1.5 matplotlib

It is a is library allows developers to showing dataset as a visual results in mathematical model and most of the time it is used by data scientist .

4.2 System Architecture

Traffic sign detection and recognizing is one of the most subsystem in ADAS in autonomous vehicles, so this system is divide into two parts : the first part is Detection part and the second part is recognizing part which we used Deep learning techniques as we mentioned before .

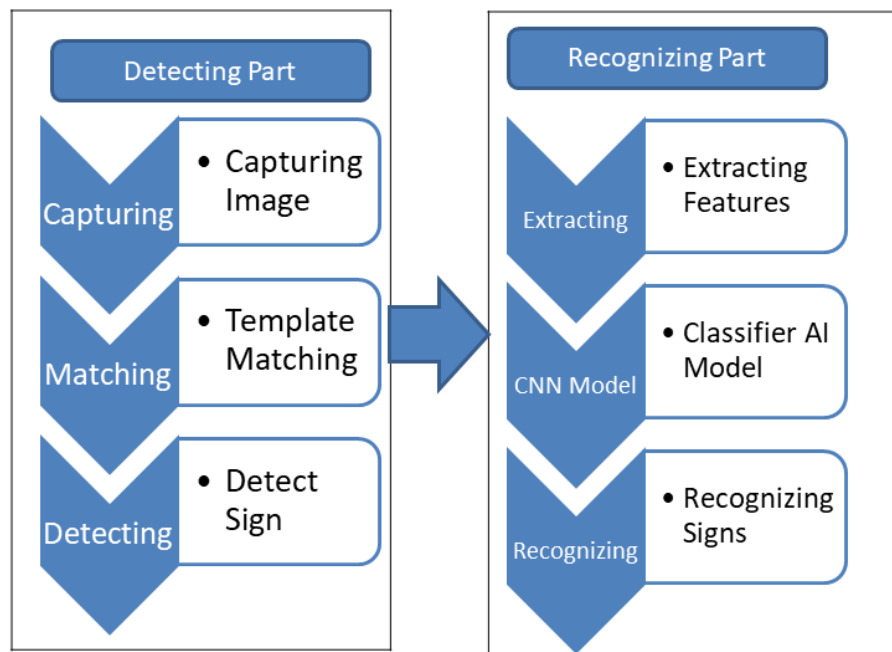In this section we will show the architecture for the whole system TSR.

Figure 4.1: TSR Architecture

5 Development Methodology

During the internship, both the Test-Driven Development (TTD) and Agile Development were used. This section will give a brief overview of TTD, Agile Development, and the Scrum Framework used in the project.

5.1 Test Driven Development - TDD

TTD[3] is a software development technique that turns the requirement of the software into a test case by relying on the repetition of a very short development cycle. Along with the agile development process, TTD has become an accepted industry practice in developing quality software. TTD was used throughout our project to help build an application with minimum bugs. The repetitive process of a TTD[5][8] is as follows:

1. A test case is written based on the software requirement

2. Run and it to see if it fails or not.

3. Correct the code based on the errors identified in the test run.

4. This process is repeated until all the software requirements are satisfied.

5.2 Testing Techniques used in the TDD

Unit Testing

The goal of this is to ensure that each unit/component of the software is performing according to the requirements of the software.

Integration Testing

This testing is done to identify if any errors exist in the interaction of different units in the software.

System Testing

This is the macro level testing and is done to assess whether the entire system fulfills all the software requirements.

5.3 Agile Development

Agile development defines an approach to developing software, which emphasizes incremental delivery, team collaboration, continual planning, and continual learning along the way instead of delivering all of these towards the end. The aim of Agile is to keep the process lean and crate only minimum viable products (MVPs) which then go through multiple iterations before anything is finalized. Feedback is incorporated continuously and at each stage. The overall process is quite dynamic, where every unit is working towards achieving the end goal.

5.4 Scrum

Scrum[17] is agile that allows teams to work together to solve complex problems. It promotes self-learning through your own mistakes and looking at your team's wins and losses to consistently improve.

5.4.1 Scrum Roles

A Scrum Team consists of three major players: Product owner, the Team and a Scrum Master. The Product Owner (PO)in our internship project is "Technical Supervisor" who is basically the interface between the stakeholders and the development team. PO envisions the product and using back- logs, assigns the development team its tasks. The responsibility of Return on In- vestment (ROI) of the product falls on the PO. The Scrum Master (SM), on the other hand is responsible for removing any obstacles or obstructions for the team to deliver the final product(our team). It oversees and provides whatever help necessary to the Team and the PO to be successful by helping the product group learn and apply Scrum in developing a successful product. The Team (Our team) is the one who does all the manual labor. The Team consists of multiple individuals who all work together to deliver the final product. The Team has individuals of much expertise such as soft- ware engineers, architects, programmers, analysts, system admins, QA experts, etc. They are also accountable for Sprint backlog which is in our project the Scrum master, Team developer both are both conducted and integrated into our Team.

5.4.2 Benefits of Scrum

Firstly, the Scrum Framework is simple and easy to understand. It has a semi-prescriptive approach that allows companies to integrate their individual talent and skills into the process. The Scrum Framework streamlines the process and makes complex tasks manageable by assigning clear roles and instructions. There is transparency and a sense of collective ownership that acts as a motivation for everyone on the project to deliver their best work. The ability to see the result of your hard work is another benefit of this framework.

6 Conclusion

This internship has been a great learning curve and working at TU Chemnitz especially in IKON Project which is international project- for the past 6 months has given me great insight into the norms of the German IT industry that they expect from the fresh graduates like me in the future. Most importantly working there allowed students to polish development and communication skills in English. I am very thankful to my supervisor for their intuitive knowledge sharing and generosity throughout my internship. With respect to the technical aspects, my supervisor gave me a free hand to choose whatever technologies that I felt comfortable with it. As someone having no prior professional experience. Which made it easy for me to settle in the internship project. The agile method of development was adopted, which I was not familiar with before this internship. I Found out what is agile methodology and why we use it as a substantial solution in order to manage or reduce the complexity of software development. In general, all the assigned tasks were implemented with positive results. The key things learned during the past 6 months are task organization, time management, task planning, communication, and self-motivation.

Bibliography

[1] Technival University of Chemnitz –IKON International internship Project. www.tu- hemnitz.de/informatik/studium/master/ase/forschungspraktikum.php.en

[2] Hasan FLEYEH1, Mark DOUGHERTY2 ," ROAD AND TRAFFIC SIGN DETECTION AND RECOGNITION".

[3] Burris, J.W.: Test-driven development for parallel applications. in 2017 Second International Conference on Information Systems Engineering (ICISE), pp. 27–31. (2017), https://ieeexplore.ieee.org/document/8241215

[4] Conversion tool website for CSV file and Data Preprocessing and Preparation, https://www.zamzar.com/ [Access 20.03.2020]

[5] Farcic, V.: Test driven development (tdd): Example walk- through. Technology Conversations, 20-Dec-2013 (12-Nov-2019), https://technologyconversations.com/2013/12/20/ test-driven-development-tdd-example-walkthrough/

[8] GeeksforGeeks: Python language introduction. Online (06-Nov-2015), https: //www.geeksforgeeks.org/python-language-introduction/

[6] Open CV Libraries and Platform and methods , ethods,https://docs.opencv.org/4.2.0/d1/dfb/intro.html [Access -20.03.2020]

[7] Keiron O'Shea1 and Ryan Nash2 ,Introduction to Convolutional Network. https://www.researchgate.net/publication/285164623 [Access,20.03.2020]

[8] Hamad, M.M., Jihad, A.A.: An enhanced technique to clean data in the data warehouse. in 2011 Developments in E-systems Engineering (2011), https: //ieeexplore.ieee.org/docume nt/6149998

[9] Kumar, V., Khosla, C.: Data cleaning-a thorough analysis and survey on un-structured data. in 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence) (2018), https://ieeexplore.ieee.org/document/8442950

[10] Learnpython.org: Pandas basics - learn python - free interactive python tutorial. Online (12-Nov-2019), https://www.learnpython.org/en/Pandas_Basics

[11] Scrum.org: Scrum. Online (12-Nov-2019),https://www.scrum.org/index

[12] Stančin, I., Jović, A.: An overview and comparison of free python libraries for data mining and big data analysis. in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectron- ics (MIPRO), pp. 977–982 (2019), https://ieeexplore.ieee.org/document/ 8757088

[13] Tutorialspoint: Unix makefile tutorial - tutorialspoint. Online (12-Nov-2019), https://www.tutorialspoint.com/makefile/index.htm