GRADUATE PROJECT

TO OBTAIN THE BACHELOR DEGREE IN MATHEMATICAL AND COMPUTER SCIENCE (MCS)

Applying AI Technique in Traffic/Road Signs Recognition
Horizontal Road Signs

Internship carried out from the 16th of May 2022 until the 4th of July 2022

Realized by :

Barras Mouad
Maghrane Wail

Defended before the examination committee on July 4, 2022:

Mr ABDELWAHED El Hassan	*Professor at FSSM, Cadi Ayyad University*
Mr QAZDAR Aimad	*Professor at FSSM, Cadi Ayyad University*
Mr Atheer Al-Tameemi	*Professor at DHBW*

SESSION YEAR : 2021-2022

Acknowledgements

*"First and foremost, we want to thank **ALLAH** for giving us the courage and patience to accomplish this work."*

Our heartfelt gratitude goes to our Dear Parents for their moral support and encouragement.

Our gratitude goes to our Professors at the Semlalia Faculty of Sciences for their contributions to our bachelor formation, and we especially thank our supervisors, Mr EL HASSAN Abdelwahed , Mr QAZDAR Aimad , and Mr Atheer Al-Tameemi (Professor at DHBW), for their precious recommendations and the trust they placed in us, as well as their availability, help, and good advice.

We would like to thank all the people who, directly or indirectly, have contributed to the accomplishment of this work.

Finally, we would like to thank the members of the jury for attending this defense.

Abstract

This work is a component of the final project required by FSSM University and DHBW University for the Bachelor of Science in Computer Science.

In this topic, we will treat the difficulties with detecting road markings, which addressing the shortcomings , such as the absence of challenging scenarios, the prominence given to lane markings, the lack of an evaluation script, the absence of annotation formats, and lower resolutions. we adopted on the road marking dataset presented by CeyMo , which had presented a dataset captured variety of traffic, lighting, and weather situations. in order to support a wide variety of road marking recognition methods .

This collection of road markings also includes annotations in the form of polygons, bounding boxes, and pixel-level segmentation masks.

Additionally, we divided our dataset into a training set and a testing set. Furthermore, we compared the effectiveness of object detection-based algorithms for the task of detecting road markings.

We came in real time during the deployment phase. Then taking into consideration all of its benefits and drawbacks.

Contents

List of Figures

Chapter 1

General presentation of the project

In this chapter, we will define the project's context, pose the problem, and provide the objectives of our work. At the end, We shall introduce the organizational framework for the work we employed for our project.

1.1 Introduction

In order to complete our final year project and obtain a Bachelor of Science in Computer Science, We carried out this two-month internship in cooperation with Cadi Ayyad University(UCA) and Baden-Württemberg Cooperative State University (DHBW) , and it was overseen by a teaching staff from each university.

The Baden-Württemberg Cooperative State University (Duale Hochschule Baden-Württemberg/DHBW) is the first higher education institution in Germany to combine on-the-job training with academic studies, thus achieving a close integration of theory and practice, both of which are components of cooperative education. DHBW is one of the largest higher education institutions in the German Federal State of Baden-Württember.

The Faculty of Science Semlalia, is one of the main institutions of Cadi Ayyad University, and it was established in 1978. Since then, it has experienced a remarkable dynamism in terms of training, scientific research, and cooperation. Thus, it has created a place for choice in the national higher education scene.It trains the majority of postgraduate students in Computer Science and offers academic degrees (MA and PhD) as well as higher vocational training certificates.

The subject of "Horizontal traffic signs" discuss a many problems, which human drivers cause accidents because they are unable to notice or comprehend road signals and the role of autonomous car is to solve this problems.

1.2 Context and Problem

Artificial intelligence and machine learning have a wide range of exciting applications in our daily lives. Whether we're attempting to read our emails, locate music or movies, or obtain driving directions, AI assists us in every area of our life.

An autonomous vehicle, sometimes referred to as a driverless vehicle, is one that can run and carry out critical tasks without the assistance of a driver and relies on deep learning, a kind of artificial intelligence, to detect its surroundings. An AI system developed by the technology company Nvidia enables vehicles to "look, understand, and learn, enabling them to navigate a practically unlimited range of conceivable driving conditions". It is anticipated that the company's AI-powered technology, which is now present in Toyota, Mercedes-Benz, Audi, Volvo, and Tesla vehicles will completely change the way people drive and eventually allow for self-driving cars.

1.3 Challenges and Objectives

As newcomers to the field of artificial intelligence, we initially faced some difficulties that we had to overcome in order to complete this project. The first of these was that we lacked the fundamentals to understand the project, and despite the fact that this is a recent complex project that requires significantly more work than most, we didn't have the fundamentals to understand it.

Our objectives were to make the most of the chance to put the abilities we acquired during the two-month internship to use. and the chance to develop those abilities, learn more about the workplace, and get insight from our supervisor's knowledge and suggestions. However, we made an effort to make sure that our goals were things that we could influence, that they were SMART (Specific, Measurable, Achievable, Realistic, Time-bound), and that we did not overlook include any of our personal touches.

1.4 Methodology

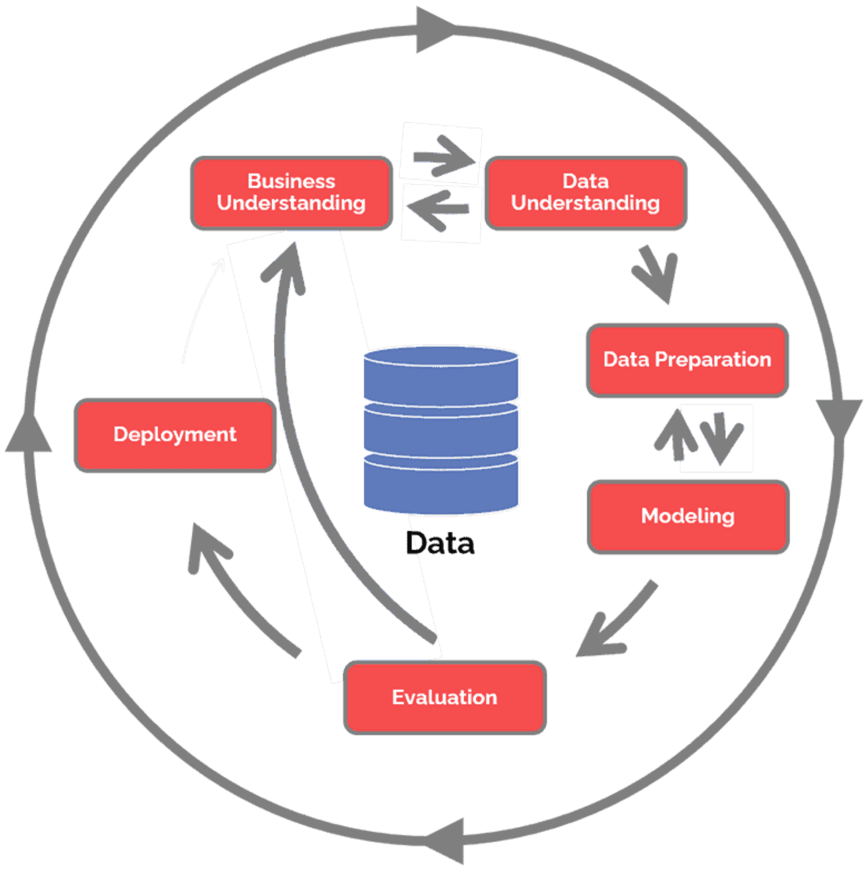


Figure 1.1: CRISP-DM Diagram

To structure our article, we provide the CRISP-DM (CRoss Industry Standard Process for Data Mining) methodology1.1, which is a process model that serves as the cornerstone for a data science process. It is split into different stages:

- Business understanding:

  Understands the project's business objectives and requirements, then turns this knowledge into a data mining issue description and a preliminary approach designed to meet the objectives indicated below:

  - What the client really wants to accomplish?
  - Uncover important factors (constraints, competing objectives)

- Data understanding :

  We start with data collection and then go on to activities to familiarize ourselves with the data, find data quality concerns, discover preliminary levels of insights into the data, or detect attractive subsets to generate hypotheses about hidden information.

- Data preparation:

  It encompasses all operations necessary to generate the final dataset from raw data. Converting raw data into analytical datasets is crucial. The quality of cleaned data will influence model performance. Tasks for data preparation are likely to be conducted several times and in no particular order. Tasks include table, record, and attribute selection, as well as data processing and purification for modeling tools.

- Modeling :

  Here you'll likely build and assess various models based on several different modeling techniques.

- Evaluation :

  Whereas the Modeling phase's Assess Model task focuses on technical model evaluation, the Evaluation phase focuses on which model best satisfies the business's needs and what to do next.

- Deployment:

  where data mining pays off. In this final phase of the Cross-Industry Standard Process for Data Mining (CRISP-DM) process, it doesn't matter how brilliant your discoveries may be, or how perfectly your models fit the data, if you don't actually use those things to improve the way that you do business.

Chapter 2

Autonomous car uses emerging technology

2.1 Autonomous car

An autonomous car is a vehicle that can sense its environment and operate without human involvement. A human passenger is not needed to take control of the car at any time, nor is a human passenger needed to be present in the vehicle at all. An autonomous car can travel everywhere and a standard car can go and do everything that a skilled human driver can do.

To execute software, Autonomous cars rely on sensors, actuators, machine learning algorithms, and advanced processors.

Some example of Autonomous cars:

Tesla is an automobile company that popularized autonomous cars which features an autonomous system that keeps the Model S in its designated lane of travel, while the Full Self-Driving Capability allows for lane changes (if the sensors indicate it is safe), automated parking assistance, and a mode to call the car from a parking place.

Mercedes-Benz is a technological powerhouse among autonomous cars. Vehicles can communicate with one another and provide alerts about prospective road hazards or slick road conditions.

Ford Fusion It has intelligent cruise control and a front collision warning system with a brake that keeps a predefined space between it and the vehicles ahead of it. Similarly, the Fusion offers an active park assist system with forwarding sensors, which allows the car to park automatically.

TXAI is a taxi with an **AI twist**, TXAI is the first-ever driverless taxi. Powered by Geospatial intelligence and layered with AI, TXAI provided full-stack software and hardware solutions, as well as operating and monitoring systems.

2.2 Technology Implemented

2.2.1 Big Data

Big data is described as data that is more diversified, arrives at a greater velocity,and arrives in higher numbers. These are also referred to as the three Vs.

Generally defined, big data refers to bigger, more complex data collections, especially those from new data sources. These massive data volumes are just beyond the capabilities of traditional data processing techniques.

However, these massive volumes of data may be used to find solutions to previously intractable business problems. Recently, Big data have become more and more popular in self-driving cars. Big data for autonomous vehicles is what allows sensors to be used.An autonomous vehicle will be worthless on the road if it does not have access to a consistent and reliable stream of self-driving car data - it will not know what to do with the data it gets. A connected automobile without data is like a newborn that pushes its fingers into sockets, grabs a knife, or attempts to catch a spark because it doesn't realize it's harmful. However, data alone cannot assist the automobile in performing its function. So, if data represents "what to learn," algorithms represent "how to learn." Such as computer vision algorithms, deep learning, and machine learning, to name a few.

2.2.2 Computer Vision

Machine vision (computer vision) is an umbrella term for a set of techniques used to interpret image-based data by computers. While image processing has been possible for a long time, computers were unable to interpret those images in any way. Simple heuristics were used to deliver semi-intelligent ways of processing an image – for example, gamma analysis enabled cameras to check if a particular area had been breached. But not much more.

When using an artificial neural network, the capabilities of machines have risen to levels as unseen before. The machine can now understand the context of a scene and filter out the noise to process only the most important information.

For example, a child running into the street or a car nearby would be considered more important than, say, a cloud in the sky or a petrol station on the horizon.

But the sole fact that computer vision in an automotive context can tell the difference between a car and a human or a tree and a building does not mean that it can rival the perceptive skills of a human driver. This ability is only a prelude for more sophisticated technologies to arrive.

2.2.3 Machine learning and Deep learning

Machine learning is a subset of artificial intelligence. It focuses on enhancing the way a machine executes a certain task. The most crucial component is that learning implies that the system moves beyond the training data. A computer using machine learning algorithms may use induction to create knowledge structures In other words, machine learning and artificial intelligence may succeed where traditional programming fails.

Machine learning is now one of the trendiest technologies for autonomous driving. Particularly, Deep learning is becoming more popular. Deep learning is a class of machine learning that focuses on computer learning from real-world data using feature learning.

2.2.4 Internet of things (IoT)

The internet of things, or IoT, is a network of connected computing devices, mechanical and digital machine, objects, animals, or people that may exchange data across a network without needing human-to-human or human-to-computer contact.

A thing in the internet of things can be a person driving a car equipped with sensors that inform the driver when tire pressure is low, or any other natural or man-made object that can be issued an Internet Protocol (IP) address and can send data across a network.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business.

Chapter 3

Horizontal Traffic Signs

3.1 Introduction

Horizontal traffic signs are also called the road markings which are those that help to control the traffic with the paint and other materials in the road surface. The significance of road markings is essential in terms of driving and helping traffic participants in the possible directions of travel.

This section draws attention to the anomalies of traffic lanes and road markings currently used. In many circumstances, human drivers cause accidents because they are unable to notice or comprehend road signals. Weather, ambient conditions, and traffic scenarios all have an impact on recognizability, making it harder for autonomous systems to recognize.

To avoid making mistakes, Collect circumstances that create issues with system environment detection. [5]

3.2 The problem to be solved

Usually, Circumstances in the environment significantly influence the visibility of the signs. So The natural wear and fault of the signs makes it more difficult for recognition systems and drivers to detect them.

However, recognition systems can only solve situations for which they have been trained and prepared for. [5]

It may result in an accident in many cases like:

3.2.1 The visibility of the road markings is not clear in the rain for detection by recognition systems and drivers

Figure 3.1

3.2.2 Use two types of traffic lanes and cause misunderstandings

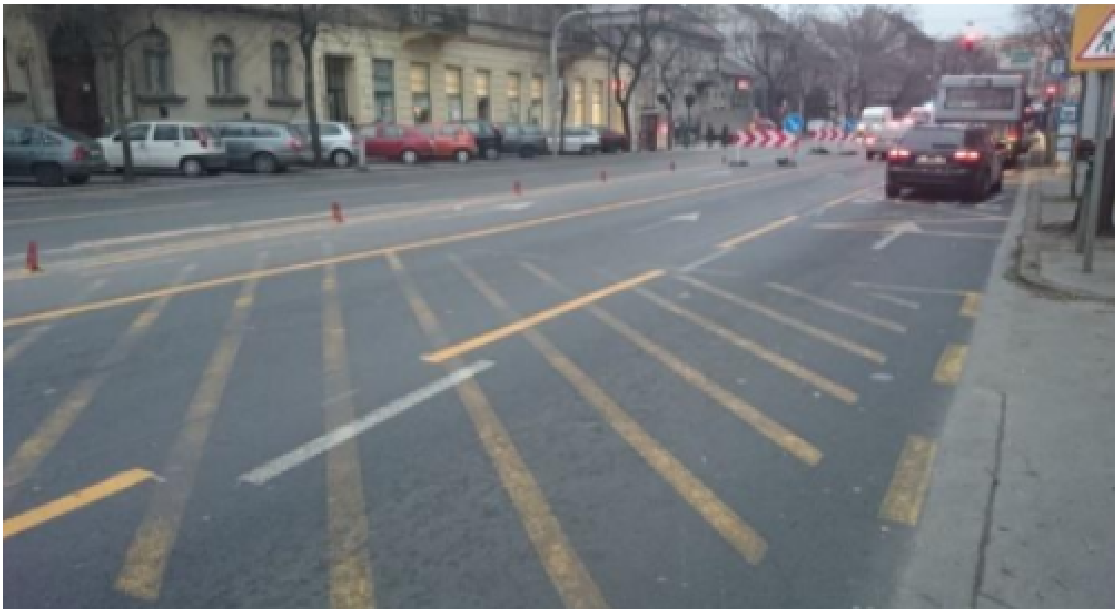


Figure 3.2

3.2.3 The sun's rays may obscure the visibility of the road markings

Figure 3.3

3.2.4 Road markers may be difficult to discern at night

Figure 3.4

3.3 Technologies

Most road markings detection research relies on image processing techniques supplemented with machine learning algorithms.

The usual detection process includes image pre-processing, regions of interest (ROI) generation, feature extraction, and classification using machine learning algorithms. [3]

However, this may lead to poor performance since the accuracy of road markings recognition is directly related to the performance of lane detection.

Maximally stable extremal regions (MSER) [8] are used as possible candidate regions in [12] , and histogram of oriented gradients (HOG) feature descriptors are used to build a template pool for each class. [3]

The MapNet DNN model included in the NVIDIA DRIVE Software 10.0 version is capable of detecting painted lane line markings (solid/dash lines, intersection entry/exit lines, road edges), painted road markings (for example arrows, Junction Box, and STOP text...). [1]

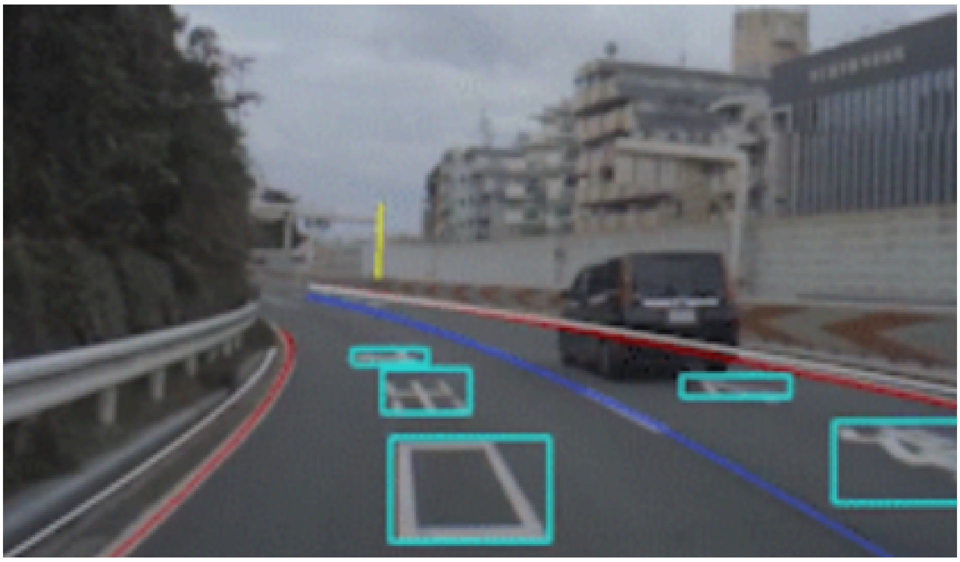


Figure 3.5

3.4 Related Work

In this project, we examine the publicly available road marking detection datasets, evaluate it in different algorithms and implementations used for the road marking detection challenge, and compare the models used with each others.

Afterwards, we run it through real-time detection tests, noting all of its advantages and limitations.

Chapter 4

The Dataset

4.1 Source

For the dataset we use a novel benchmark dataset for road markings detection called CeyMo [4]. This includes a wide range of challenging urban, sub-urban, and rural road scenarios. The dataset contains 2887 total pictures with a resolution of 1920 × 1080 and 4706 road markings instances divided into 11 classes. Normal, crowded, dazzling light, night, rain, and shadow are the six categories.

Figure 4.1: This Figure shows the Challenging scenarios present in the CeyMo dataset. (a) Dazzle light (b) Shadow (c) Rain (d) Night (e) Occlusion (f) Deteriorated road markings

4.2 Data statistics

4.2.1 Data annotation

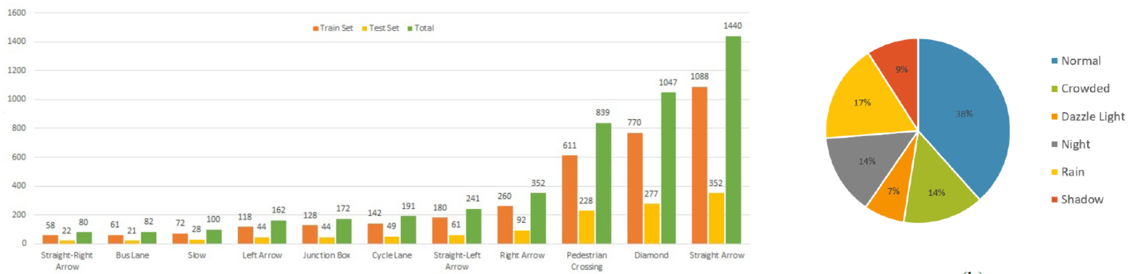
Annotations for road markings are available in three formats: polygons, bounding boxes, and pixel-level segmentation masks. Polygon annotations in JSON format, bounding box annotations in XML format, and segmentation masks in PNG format.

(a) Image (JPG) (b) Polygon (JSON) (c) Bounding Box (XML) (d) Segmentation Mask (PNG)

Figure 4.2: Annotation formats provided with the CeyMo dataset

4.2.2 Data separation

This new benchmark is constituted of 2887 images with a resolution of 1920 × 1080. The dataset is separated into the train set which contain 2099 images and the test set which contain 788 images, respectively.



(a) (b)

Figure 4.3: The frequency of each class in the dataset is shown in the column graph (a), while the proportion of each situation in the test set is shown in the pie chart (b)

The next Table 4.1 shows the number of instances in each of the 11 road marking classes covered by the benchmark:

Table 4.1: Number of instances for each class in the dataset.

Road Marking Class	Train Set	Test Set	Total
Straight Arrow	1088	352	1440
Left Arrow	118	44	162
Right Arrow	260	92	352
Straight-Right Arrow	58	22	80
Diamond	770	277	1047
Pedestrian Crossing	611	228	839
Junction Box	128	44	172
Slow	72	28	100
Bus Lane	61	21	82
Cycle Lane	142	49	191
Total	3488	1218	4706

The 788 images in the test set are divided into 6 categories including normal and 5 challenging scenarios: crowded, dazzle light, night, rain and shadow. The number of images and the proportion of each category are shown in Table **??** .

Scenario	No. of Images	Percentage
Normal	303	38.45
Crowded	109	13.83
Dazzle Light	56	7.11
Night	110	13.96
Rain	136	17.26
Shadow	74	9.39

Table 4.2: Number of images for each scenario in the test set

4.2.3 Data augmentation

we employed bounding boxes using data augmentation ,which was done to increase the quantity and diversity of data , To generate new images from the CeYMO dataset, a few data augmentation techniques such as blur, brightness, and gray scale. The datasets were also uploaded to the roboflow web tool for data augmentation, preprocessing, and train-test split.

4.3 Conclusion

Though detecting and recognizing road markings is an important task, it is frequently under-researched, due to the lack of publicly available datasets and limitations in existing datasets. Other issues include an undefined train-test split and unavailability of an evaluation script. Therefore, We chose this dataset because it has the highest availability of a difficult. assignment due to occlusions, lighting variations, shadows, changing weather conditions, and deterioration of road signs over time. The following table 4.3 compares the CeyMo dataset to publicly accessible road marking detection datasets.

Table 4.3: Comparison of CYMO dataset with publicly available road marking detection datasets

Dataset	Year	Images	Classes	location	Annotation Format
Road Marking	2012	1443	11	USA	Bounding Box annotations (TXT)
TRoM	2017	712	19	China	Pixel-level annotations (PNG)
VPGNet	2017	21097	17	Korea	Pixel-level and Grid-level annotations (MAT)
CeyMo	2021	2887	11	Sri Lanka	Polygon annotations (JSON) Bounding Box annotations (XML) Pixel-level segmentation masks (PNG)

Chapter 5

Machine vision and Machine learning algorithms

5.1 Machine Vision Algorithms

Healthcare, agriculture, automotive, and security are just a few of the industries where computer vision algorithms are used. recently, a lot of effort has gone into developing frameworks, toolkits, and software libraries. Machine vision technologies offer image-based examination and analysis for sush as difficult challenges.

Here are some machine vision algorithms:

5.1.1 SIFT Algorithm

SIFT or the scale-invariant feature transform algorithm is used to detect also describe the local features in a digital image. It finds important locations and provides them with numerical data, commonly referred to as descriptors utilized for object identification and recognition. The descriptors derived by SIFT are robust to rotation, lighting, and perspective as well as invariant against the modification of pictures, which causes the image to appear different even when it contains the same items.

SIFT is a 4-Step computer vision algorithm:

Scale-space Extrema Detection: In this stage, the algorithm looks for prospective interest spots by searching the entire image scales and positions using a difference-of-Gaussian (DoG) function. These points are oriented and scale-invariant.

Keypoint Localization: Based on their stability metrics, a thorough model is fitted at each possible location to establish the position and scale of key points.

Orientation Assignment: Orientations are assigned to each keypoint based on local image gradient directions. The assigned orientation, scale, and location of each feature in the image are used in all future operations on the image, which are invariant to any transformations.

Keypoint Descriptor: Around each keypoint, the local image gradients are measured at the chosen scale. These gradients are converted into a form that enables both local shape distortion and significant changes in illumination.

5.1.2 Viola-Jones Algorithm

Paul Viola and Michael Jones, two computer vision researchers, created the Viola-Jones object detection algorithm in 2001 to address the issue of face detection, but it can also be trained to recognize a variety of object classes in images in real-time. For a given dataset, this method takes a while to train, yet it can recognise faces in real time with amazing speed and accuracy. Using Haar-like characteristics, the Viola-Jones algorithm can identify faces in images.

5.1.3 Kalman Filter

One of the most intriguing areas of computer vision research is obstacle detection. It requires keeping track of and predicting the movement of items. When it comes to computer vision applications like object tracking, prediction, and corrective tasks, the Kalman filter has long been considered to be the best choice. The Kalman filter is used in practical contexts such as robotics, medicine, defense imagery and video, public and private security, and navigation and positioning systems.

5.1.4 YOLO Algorithm

Object detection in computer vision and graphics involves detecting various objects in digital images and videos. YOLO or 'You Only Look Once' is an algorithm that provides real-time object detection using neural networks. This algorithm is known for its speed and accuracy. The algorithm can be used to detect people, animals, etc... Depends of the datasets.

YOLO performs real-time object detection using CNNs, or convolution neural networks. The CNN model applies bounding boxes for the detected objects in an input image and predicts the class probabilities for the objects that are discovered. As implied by the name, the approach only needs one forward propagation through the model to recognize objects in an input image and estimate their locations.

5.2 Machine Learning Algorithms

Machine learning algorithms may be used on IIoT to enjoy the benefits of cost reductions, increased efficiency, and enhanced performance. We have all witnessed the advantages of machine learning techniques in the recent past. It can handle big and complicated data sets in order to extract interesting patterns or trends, such as anomalies. Machines are required to evaluate information quickly and to make choices when it reaches a certain level. There are several machine learning techniques that aid in data analysis in industrial IOT devices.

ML algorithms can be divided into supervised or unsupervised learning:

5.2.1 Supervised Learning

A sort of ML approach called supervised ML algorithms may be used to generate new data using labeled data and predict future events or labels based on what has already been learnt. When learning in this manner, a supervisor (labels) is there to direct or correct. The learning method is used to forecast the output values for this initial analysis after the known training set. The learning system's output may be compared to the actual output; if discrepancies are found, they can be fixed, and the model can be adjusted accordingly.

5.2.1.1 Linear Regression

it can be used to predict real values (cost of something, number of customers, total sales, etc...). Based on continuous variable(s), This best fit line is known as regression line and represented by a linear equation:

$$Y = a * X + b$$

In this equation:

Y – Dependent Variable

a – Slope

X – Independent variable

b – Intercept

These coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

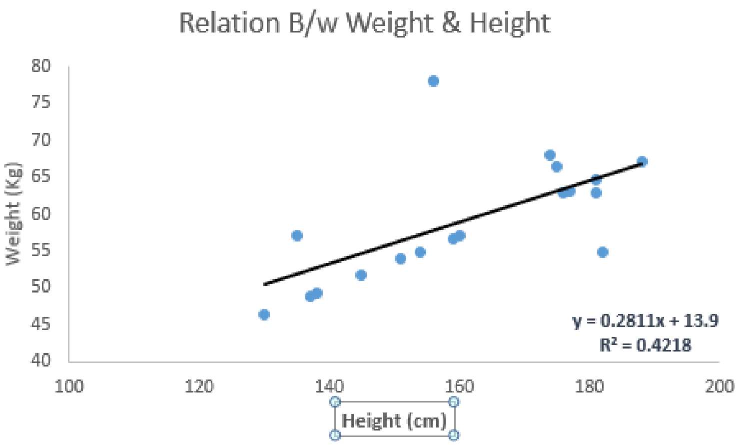


Figure 5.1: Linear Regression

5.2.1.2 Decision Tree

It is a supervised learning method that is mostly used to solve classification issues.

It works for both categorical and continuous dependent variables, which is surprising. We divide the population into two or more homogenous sets using this approach.

This is done to create as many separate groups as feasible based on the most relevant attributes/independent variables. For further information, see Decision Tree Simplified:

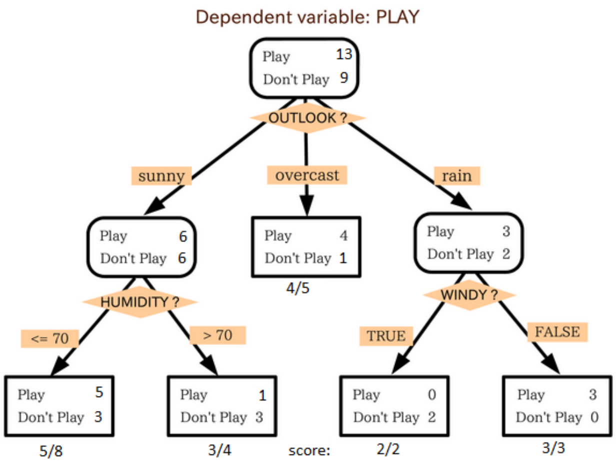


Figure 5.2: Decision Tree

5.2.1.3 SVM (Support Vector Machine)

It is a method of classification. In this approach, each data item is plotted as a point in n-dimensional space (where n is the number of features), with the value of each feature being the value of a certain coordinate.

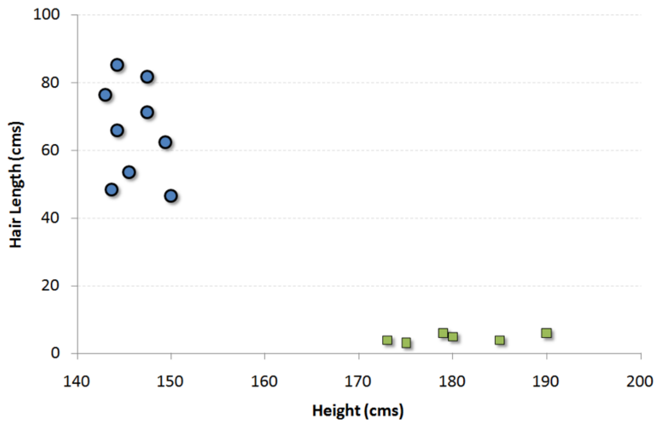


Figure 5.3: Linear Regression

5.2.2 Unsupervised Learning

Unsupervised machine learning algorithms: In these, a supervisor is not present to direct or correct. When there is unlabeled or unclassified data available to train the system, this kind of learning method is applied.

The system explores the data in a way that allows it to make inferences (rules) from datasets and to characterize hidden structures from unlabeled data, even when it does not define the ideal output.

5.2.2.1 Dimensionality Reduction Algorithms

How did you find highly significant variables in a data set of 1000 or 2000? In such circumstances, the dimensionality reduction algorithm, in combination with other algorithms such as Decision Tree, Random Forest, SVM, Linear Regression, Identify based on correlation matrix, missing value ratio, and others, might be useful.

Dimension Reduction is the technique of transforming a set of data with large dimensions into data with smaller dimensions while still conveying similar information concisely.

These strategies are often used to get better features for a classification or regression challenge while tackling machine learning issues. [10]

5.2.2.2 Clustering

Clusteringis the process of grouping similar data sets together (based on defined criteria). It is essential for segmenting data into groups and analyzing each data set to find patterns.

5.3 Algorithms used

We tried many algorithms, including YOLO v3, V4 and v4 tiny , V5, and mobilenetssd v2, but the best results came from YOLO V5 and YOLO v4 tiny , as well as ssd mobilenet V2.

Now we will provide a brief explanation of these algorithms.

5.3.1 YOLO V5

Yolo is a state-of-the-art, real-time object detector, and Yolov5 is based on Yolov1-Yolov4. Continuous improvements have made it achieve top performances on two official object detection datasets: Pascal VOC (visual object classes) [7] and Microsoft COCO (common objects in context) [6].

YOLOv5 is distinct from earlier versions. Instead of Darknet, it employs PyTorch. It makes use of CSP-Darknet53 as its backbone. This backbone solves the repetitive gradient information in large backbones and integrates gradient change into feature map that reduces the inference speed, increases accuracy, and reduces the model size by decreasing the parameters. It employs a path aggregation network (PANet) as neck to boost the information flow. PANet adopts a new feature pyramid network (FPN) that includes several bottom ups and top down layers. This increases the model's propagation of low-level features. PANet enhances localisation in lower layers, which improves the localization accuracy of the object. In addition, the head in YOLOv5 is the same as YOLOv4 and YOLOv3 which generates three different output of feature maps to achieve multi scale prediction. It also aids in the efficient prediction of tiny to large objects in the model. The image is sent via CSPDarknet53 for feature extraction before being passed to PANet for feature fusion. Finally, the YOLO layer generates the results. The architecture of the YOLOv5l algorithm is seen in Figure5.4. The Focus layer [9] is evolved from YOLOv3 structure. It replaces the first three layers of YOLOv3 and create a single layer in YOLOv5. Additionally, here Conv denotes a convolution layer. C3 is composed of three convolution layers and a module cascaded by various bottlenecks. Spatial pyramid pooling (SPP) is a pooling layer that is used to eliminate the network's fixed size limitation. Upsample is used in upsampling the previous layer fusion in the nearest node. Concat is a slicing layer that slices the preceding layer. The final three Conv2d are detection modules that are employed in the network's head.

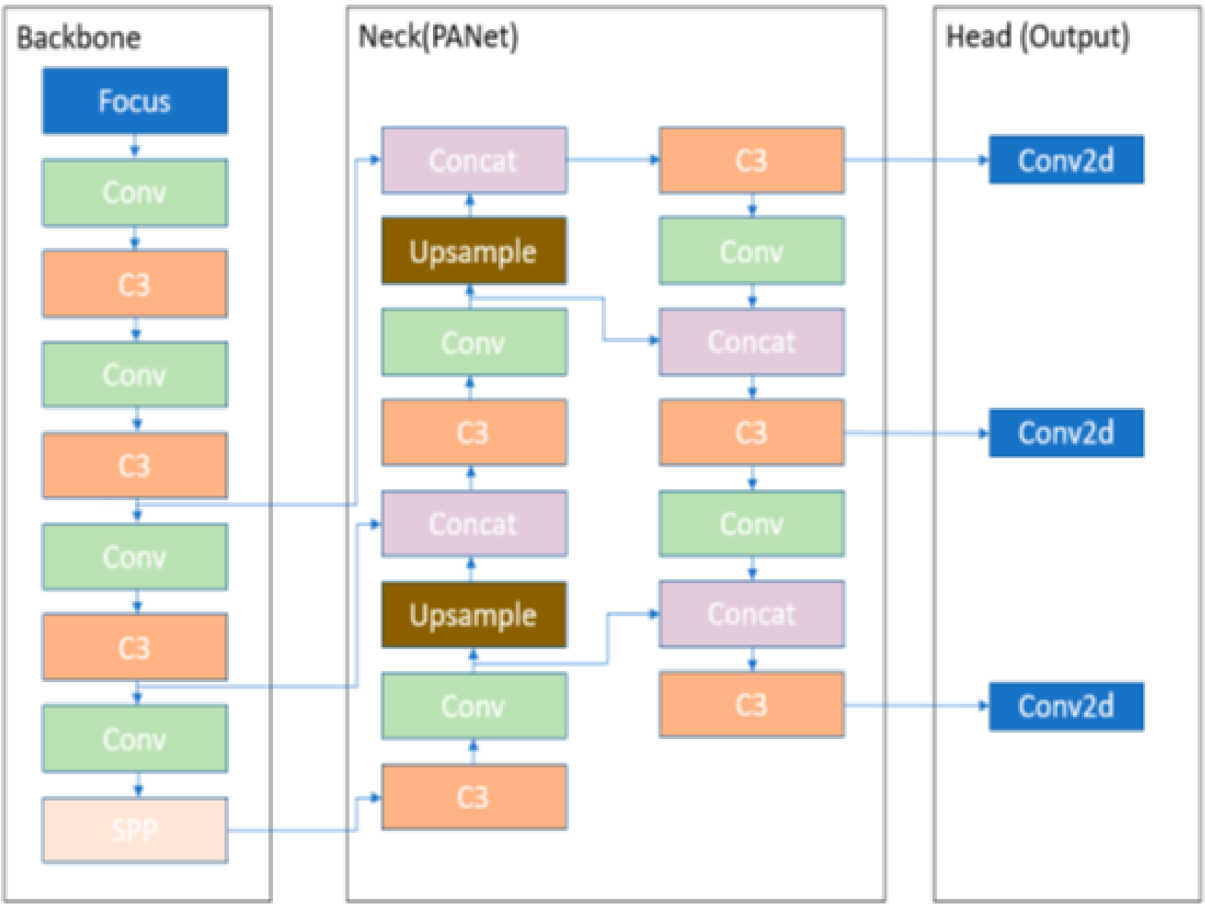


Figure 5.4: YOLOv5 Architecture

5.3.2 Yolo v4 tiny

YOLOv4-tiny is a compact version of YOLOv4 that is meant to train on devices with little CPU capacity. Its model weights are around 16 gigabytes in size, allowing it to train on 350 images in one hour utilizing a Tesla P100 GPU. On the Tesla P100, YOLOv4-tiny has an inference time of 3 ms, making it one of the quickest object detection models available. To attain these high speeds, YOLOv4-Tiny makes many modifications to the original YOLOv4 network. To begin with, the number of convolutional layers in the CSP backbone is reduced to 29 pretrained convolutional layers. Furthermore, the number of YOLO layers has been lowered from three to two, and there are fewer anchor boxes for prediction.

5.3.3 SSD Mobilenet V2

The overall tendency is that computer vision models are becoming deeper and more complicated in order to obtain higher precision.. However, these advancements increase size and latency and cannot be used in systems with computational challenges. MobileNet comes in useful in these situations. This architecture was created specifically for mobile and embedded applications that demand high speed Its first version (MobileNetV1) had a deeply detachable fold that decreased model size and network complexity to a reasonable level for low-throughput applicationsAs a result, the second edition of the MobileNet family, known as MobileNetV2, has an inverted residual structure for much improved modularity. This has contributed to the elimination of non-linearities in tight layers, resulting in improved performance for earlier applications. Around the time that the first version of MobileNet was introduced, Google released Single Shot Detector (SSD) for applications that rely heavily on speed and accuracy alike. As the name suggests, SSD recognized many items in a picture in a single shot. MobileNet is a model that offers decent speed, the only downside is its accuracy. SSD was particularly valuable for the model because of its potential to enhance accuracy while maintaining model speed. The SSD algorithm was developed to be compatible with a variety of networks, including YOLO, MobileNet, and the VGG architecture. Hence, MobileNet was incorporated into SSD for superior performance and was named Mobile Net-SSD. This integrated architecture is shown in Figure 5.5

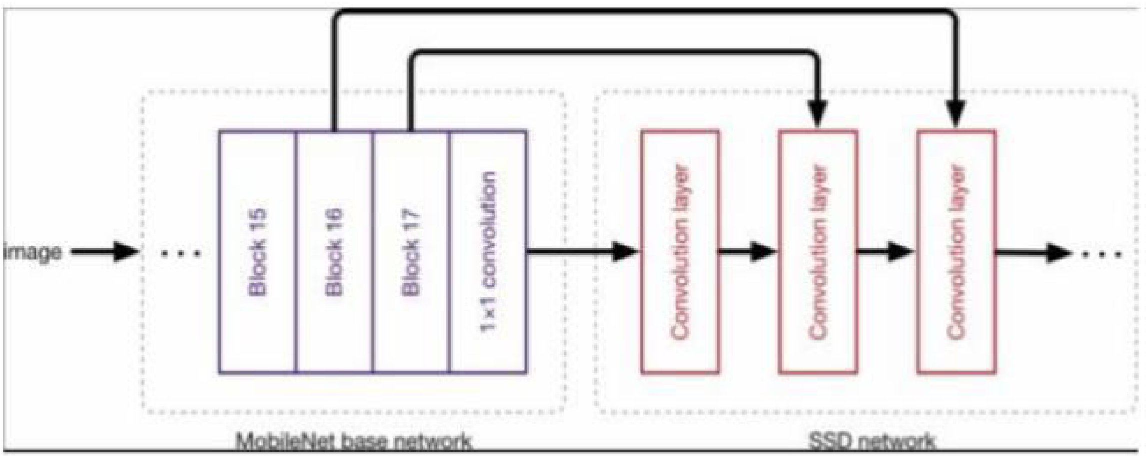


Figure 5.5: Mobilenet SSD v2 Architecture

Chapter 6

Implimentation and Evaluation

6.1 Implimentation

6.1.1 Importing Dataset

The dataset was downloaded to colab using the roboflow generated url as a zip folder. The zip folder has been unzipped and deleted.

6.1.2 Cloning YOLOv5 repository

YOLOv5 repository was cloned from GitHub using 'git clone <url>'.
!git clone 'https://github.com/ultralytics/yolov5.git'

6.1.3 Training YOLOv5 model with CeYMO dataset

The model's training was conducted using CeyMo road marking signs datasets in a free Integrated Development Environment (IDE) called Google Collaboratory (Google Colab). Tesla P100 is utilized from google colab, which is Google provides to support research and learning about Machine Learning (ML) .

In this stage, learnable weights and biases are assigned to objects in preprocessing frames. Then,the YOLO V5 , mobilenet SSD V2 and YOLO V4 tiny algorithms are applied with initial parameters. see Tab 6.1 , Tab 6.3 and Tab 6.2.

Table 6.1: Training parameter settings(YOLO V5)

Epochs	500
Batch Size	16
Momentum	0.937
Weight decay	0.0005
Thresh	0.4

Table 6.2: Training parameter settings (YOLO V4 TINY)

Iteration	6000
batch	64
momentum	0.9
learning rate	0.00261
decay	0.0005

Table 6.3: Training parameter settings(MobileNet SSD V2)

Number of training steps	30000
Number of evaluation steps	50

6.2 Evaluation

6.2.1 Evaluation Metrics

Precision gives how many of the correctly predicted cases actually turned out to be positive. The proportion is calculated with the formula shown in Equation 6.1.

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

TP: True Positive
TF: True Negative
FP: False Positive
FN: False Negative

Recall gives how many of the actual positive cases the model is able to predict correctly, It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

F1-score is a metric which takes into account both precision and recall and is defined as follows:

$$F1 - Score = \frac{2 * TP}{2 * TP + FP + FN} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.3}$$

Furthermore, mAP (mean average precision) is computed using Eq. 6.4. In the detection model, GIoU quantifies the overlap between the bounding box of predicted and the real object's ground truth bounding box. Every value of the IoU threshold provides a different mAP. Therefore, this value must be specified. If IoU is compared to a specific threshold, the detection may be valid or wrong. To test detection model performance using a series of experiments, we test the performance of the trained system using GIoU thresholds of 0.5 and 0.5:0.95.

$$mAP = \sum_{N}^{i=1} AP_i \tag{6.4}$$

6.2.2 YoloV5

Actually, before we arrived at the ultimate solution, we tried three times to get a satisfactory result.

In our initial attempt, we utilized bounding boxes in our datasets.

Graph 6.3 depicts the outcome of those measures for CeYmo road marking signs across 500 epochs

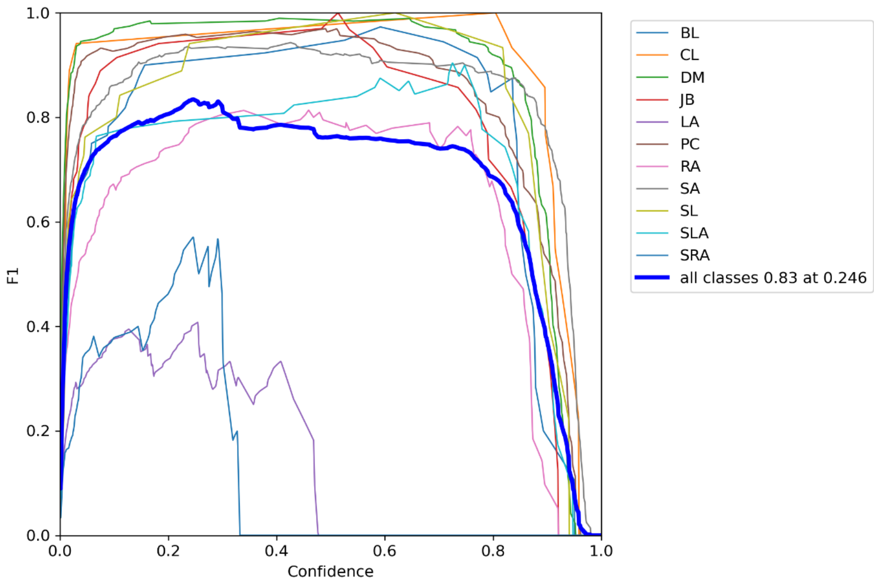


Figure 6.1: F1-Score of YOLOV5 for CYMO dataset

The highest F1 score value is 0,83, which corresponds to a confidence value that optimizes recall and precision is 0.24. This result is quite poor in terms of object prediction confidence.

So we tried again using image augmentation (Data augmentation) , and the results are presented in Graph 6.3 with 500 epochs.

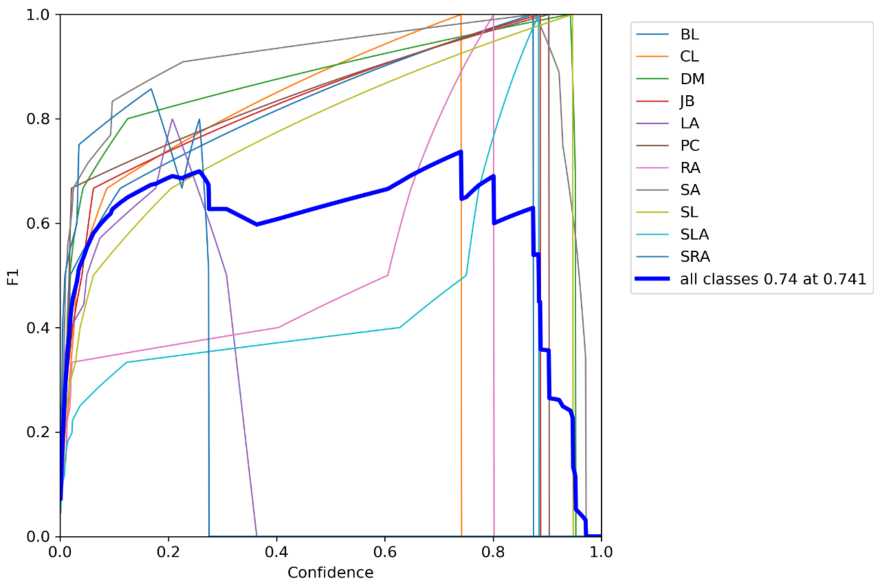


Figure 6.2: F1-Score of YOLOV5 for CYMO dataset

The F1 score is 0.74, which corresponds to a confidence level of 0.741. This result is acceptable, but it is not ideal because the F1 score is low.

we employed the polygon in our datasets for the final effort and achieved the outcome shown in the graph below with 500 epochs. the result is in the graph below:

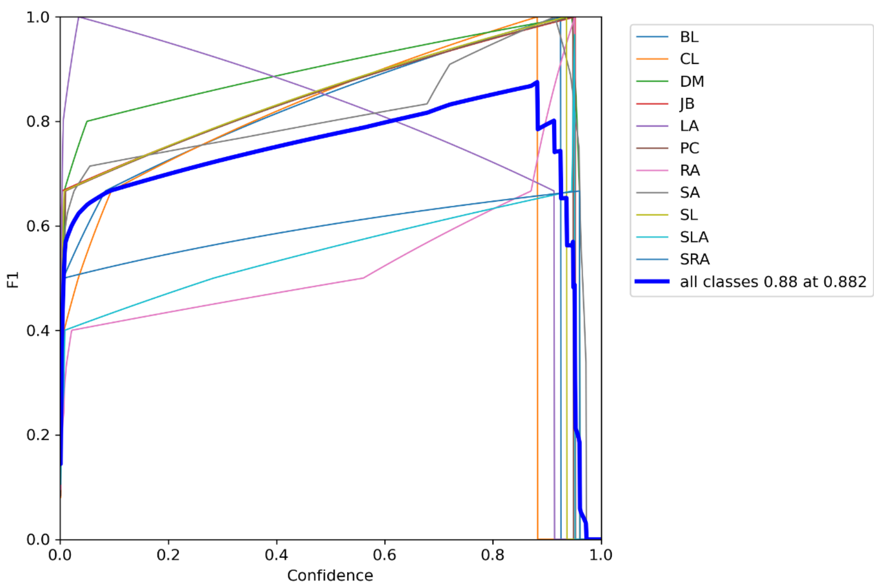


Figure 6.3: F1-Score of YOLOV5 for CYMO dataset

The confidence value is 0.882, which corresponds to the maximum F1 score value (0.88), and this is a satisfactory result for now utilizing YOLO V5 with 500 epochs.

6.2.3 MobileNet SSD V2

We also utilized this model for training, but we didn't find any results in terms of metrics, therefore we compared the images of this algorithm to other models employed.

6.2.4 YoloV4 Tiny

In addition to the above approaches, we employed the Yolo v4 tiny model to analyze our datasets. After 30000 iterations, we obtained 0.86 in F1 score, 0.84 in precision, 0.84 in recall, and a mean average precision (mAP@0.5) of 92.12

Chapter 7

Deployment

7.1 Results

In this section, we present the qualitative and quantitative results we obtained. The performance of Yolo v5 and Yolo v4 tiny object detection models on our CeyMo road marking benchmark dataset is presented in Table 7.1 .

Table shows the F1-score values and Mean Average precision of each model for the test set.

Table 7.1: Performance of Yolo v5 and Yolo v4 tiny

Model	mAP	F1-Score
YOLOV5	0.965	0.88
YOLOV4 TINY	0.9212	0.86

Qualitative results obtained by our two object detection models are visualizedin Figure 7.3, along with the input images for the six categories in the test set.

Figure 7.1: Visualization of road marking detection results on the CeyMo road marking dataset

After comparing all of the models in terms of F1 score and mean average accuracy (mAP), as well as images for mobileNet SSD V2, In terms of table 7.1, the YOLO V5 model outperforms the YOLO V4 TINY model in terms of mAP and F1 score, and it outperforms the Mobilenet SSD V2 in terms of precision of the road markings shown in Figure 7.3.

It can be observed that the YOLO V5 model perform better, especially in challenging scenarios shown in figure 7.1.

Figure 7.2: Bounding boxes generated by Yolo v5

7.2 Real time

Real-time object detection is a method for identifying things in video. Various network architectures have been suggested and published over the years, including EfficientDet, YOLOv4, and YOLOv5.

Our model is ready for inference after we achieved satisfactory training results. The input for inference used during judgment is webcam. The following detection command is used for inference for real-time testing.

python detect.py –weights best.pt –conf 0.8 –source 0

source — input path (0 for webcam)

weights — weights path

conf — confidence threshold

Inference results are automatically saved to the defined folder.

The following image displayed a capture from a live video taken in real-time using the YOLO V5 algorithm.

Figure 7.3: Real time detection

7.2.1 Limitation

Due to the weakness of our platforms for computing, which consists of an Intel Core i5-6200U CPU and Intel HD Graphics 520 GPU, and a Intel Core i7-3537U CPU, Intel HD Graphics 4000 and NVIDIA GeForce GT 720M , when I apply detection command to use for inference for testing in real time, the video renders successfully, but the detection was in very low FPS (frame per second) because our CPUs usage goes up to almost 100 percent, while our GPUs is not used in task at all.

So we looked for a solution in some references, and we found Yolo v5 in real time it runs perfectly in FPS using Nvidia Tesla T4 [2].

Conclusion

The CeyMo road marking dataset was published for the aim of identifying road markings. The new benchmark dataset contains 2887 images of 4706 individual road marking samples from 11 separate road marking classes, all captured in a range of traffic, lighting, and weather circumstances. The dataset is separated into two parts: the train set and the test set, which contain 2099 and 788 images, respectively.

The YOLO V5, MOBILENET ssd V2, and YOLO V4 TINY were used to evaluate the detection performance of object detectors for road markings. In the result above 7.1 7.1, Yolo v5 fared better than MobileNet ssd V2 and YOLOV4 TINY in terms of performance.

We also attempted real-time with Yolo v5, but were thwarted by the limitations of our processing systems, which refused to accept it.

Perspective

There are still a lot of things that can be done to improve road marking detection with YOLOv5. We are currently working technique Inverse perspective transfom (IPT) which to obtain a bird's eye view of the road area. IPT reduces perspective deformation of the captured images and it also removes a larger area of the background and the road markings become more prominent in the resultant image.

These object detector models output the road marking detections as bounding boxes on the inverse perspective transformed image. These bounding box detections are transformed to the original image domain as 4-sided polygons using the inverse of the IPT.

However, we have yet to deploy simulation and embedded systems, and we are confident in taking the next step.

Bibliography

[1] Nefi Alarcon. Drive labs: Detecting road markings and landmarks with high precision.

[2] R Iyer, P Shashikant Ringe, R Varadharajan Iyer, and K Prabhulal Bhensdadiya. Comparison of yolov3, yolov5s and mobilenet-ssd v2 for real-time mask detection. *Artic. Int. J. Res. Eng. Technol*, 8:1156–1160, 2021.

[3] Oshada Jayasinghe, Sahan Hemachandra, Damith Anhettigama, Shenali Kariyawasam, Ranga Rodrigo, and Peshala Jayasekara. Ceymo: See more on roads - a novel benchmark dataset for road marking detection. `https://github.com/oshadajay/CeyMo/`, 2021. [Online; accessed 2022].

[4] Oshada Jayasinghe, Sahan Hemachandra, Damith Anhettigama, Shenali Kariyawasam, Ranga Rodrigo, and Peshala Jayasekara. Ceymo: See more on roads-a novel benchmark dataset for road marking detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3104–3113, 2022.

[5] Henrietta Lengyel and Zsolt Szalay. Iop conf. ser.: Mater. sci. eng. 448 012046, 2018.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Everingham Mark and S Ali Eslami. M, van gool luc, williams christopher k, i, winn john, and zisserman andrew. *The pascal visual object classes challenge: A retrospective*, 111:98–136, 2015.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla. *Robust wide baseline stereo from maximally stable extremal regions.* 2004. 22:761–767, 2004.

[9] Upesh Nepal and Hossein Eslamiat. Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. *Sensors*, 22(2):464, 2022.

[10] Sunil Ray. *Beginners Guide To Learn Dimension Reduction Techniques.* — July 28, 2015.

[11] From Wikipedia. the free encyclopedia. `https://en.wikipedia.org/wiki/Lidar`.

[12] T. Wu, A. Ranganathan, and A practical. *system for road marking detection and recognition.* In IEEE Intelligent Vehicles Symposium (IV), pages 25–30., 2012.