A project report on

VIDEO BASED ADAPTIVE CRUISE CONTROL IN ADVANCED DRIVER ASSISTANCE SYSTEMS

Submitted in partial fulfilment for the award of the degree

of

MTech Integrated Software Engineering

by

GUNJI JAYA LAKSHMI (17MIS0023)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SITE

MAY 2022

VIDEO BASED ADAPTIVE CRUISE CONTROL IN ADVANCED DRIVER ASSISTANCE SYSTEMS

Submitted in partial fulfilment for the award of the degree

of

MTech Integrated Software Engineering

by

GUNJI JAYA LAKSHMI (17MIS0023)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SITE

MAY 2022

DECLARATION

I hereby declare that the thesis entitled “VIDEO BASE ADAPTIVE CRUISE CONTROL IN ADVANCED DRIVER ASSISTANCE SYSTEMS” submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bonafide work carried out by me under the supervision of Sureshu Macherla.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date:30/5/2022 **Signature of the Candidate**

Internship completion certificate

CERTIFICATE BY THE EXTERNAL GUIDE

This is to certify that the project report entitled "**VIDEO BASED ADAPTIVE CRUISE CONTROL IN ADVANCED DRIVER ASSISTANCE SYSTEMS**" submitted by GUNJI JAYA LAKSHMI (17MIS0023) to Vellore Institute of Technology in partial fulfillment of the requirement for the award of the degree of MTech Integrated Software Engineering is a record of bonafide work carried out by him under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the External Supervisor

ABSTRACT

Automobiles are the foundation of the next generation of mobile-connected devices, with rapid advances being made in autonomous vehicles. Autonomous application solutions are partitioned into various chips, called SoCs (systems on a chip). These chips connect sensors to actuators through interfaces and high-performance ECUs (electronic controller units). Self-driving cars use a variety of these applications and technologies to gain 360-degree vision, both near (in the vehicle's immediate vicinity) and far. That means hardware designs are using more advanced process nodes to meet ever-higher performance targets while simultaneously reducing demands on power and footprint.

Almost all vehicle accidents are caused by human error, which can be avoided with Advanced Driver Assistance Systems (ADAS). The role of ADAS is to prevent deaths and injuries by reducing the number of car accidents and the serious impact of those that cannot be avoided.

These lifesaving systems are key to ensuring the success of ADAS applications, incorporating the latest interface standards and running multiple vision-based algorithms to support real-time multimedia, vision co-processing, and sensor fusion subsystems. The "Smart Phonezation" of ADAS applications is the beginning steps to realization of autonomous vehicles.

ACKNOWLEDGEMENT

I would like to express my gratitude to **Dr. G. Viswanathan** Chancellor, **Dr. G. Sekar Vishwanathan** Vice president, **Dr. Rambabu Kodali** Vice Chancellor, **Dr. S. Narayanan** Pro-vice Chancellor, and **Dr. Sumathy S**, Dean, School of Information Technology and Engineering, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to **Dr. SHANTHARAJAH S P**, Head of the Department, Software and Systems Engineering, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date:30/5/2022 **Gunji Jaya Lakshmi**

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 BACKGROUND

The Bosch Group is a leading global supplier of technology and services. It employs roughly 395,000 associates worldwide (as of December 31, 2020). The company generated sales of 71.5 billion euros in 2020. Its operations are divided into four business sectors: Mobility Solutions, Industrial Technology, Consumer Goods, and Energy and Building Technology. As a leading IoT provider, Bosch offers innovative solutions for smart homes, Industry 4.0, and connected mobility. Bosch is pursuing a vision of mobility that is sustainable, safe, and exciting. It uses its expertise in sensor technology, software, and services, as well as its own IoT cloud, to offer its customers connected, cross-domain solutions from a single source. The Bosch Group's strategic objective is to facilitate connected living with products and solutions that either contain artificial intelligence (AI) or have been developed or manufactured with its help. Bosch improves quality of life worldwide with products and services that are innovative and spark enthusiasm. In short, Bosch creates technology that is "Invented for life."

1.2 MOTIVATION

Bosch is motivated by the desire to develop products that are "Invented for life," that spark enthusiasm, that improve quality of life, and that help conserve natural resources. Their "We are Bosch" mission statement reflects this. It summarizes our values, our strengths, and our strategic orientation. The mission is based on seven central values, which shape our corporate culture – ranging from a focus on the future and earnings to cultural diversity. Responsibility and sustainability are part of this set of values and thus of our actions.

1.3 PRODUCTS AND SERVICES

Bosch develops innovative solutions that facilitate new mobility offerings. Whether for private or commercial vehicles, multimodal transportation services, fleet management, or smart transport infrastructure, Bosch brings together vehicle technology, the data cloud, and services to offer complete mobility solutions. Bosch offers you individual solutions for many homes to make life a bit easier every day. Bosch also offers innovative products and services for industry and trades. The way the internet of things is transforming the world offers great opportunities for many sectors such as manufacturing, mobility, energy management, and more. Software and service platforms from Bosch provide a solid foundation for installing and operating tailor-made solutions. Discover how Bosch software solutions help companies improve the way they run their businesses and bring new revenue channels and products to life.

1.4 OBJECTIVES

The main objective of this project is to monitor all the Autonomous vehicle using the Video based Adaptive Cruise Control (VACC). VACC can automatically accelerate, slow down, and at times stop the vehicle, depending on the actions of other objects in the immediate area with the help of the video sensors present in the vehicle. This is particularly helpful on the highway, where drivers can find it difficult to monitor their speed and other cars over a long period of time. This mainly focuses to reduce the human errors while driving.

Chapter-2

Tasks Performed

2.1 TASK-1:

Image cleaning for training of the system: Here in this task, I was assigned to do all the cleaning process of raw pictures data so that it could be used by the final system for implementing the algorithm. The following steps were involved during this process:

- **Step-1:** Here in this step all the captured frames were taken from the database and downloaded to the local drive.
- **Step-2:** As all the captured frames were in the infrared picture format a special software known as Irfanview64 was installed and setup successfully.
- **Step-3:** After getting all the image folders and the software installed in the local device, all the images were loaded into the software.
- **Step-4:** Once all the images are loaded, we will start with the classification of images as good and bad into the separate path folders.
- **Step-5:** After getting all the good and bad images, all the good images will be sent to the further training of the system.

2.2 TASK-2:

Animal detection in the captured frame: Here in this task, there will be a raw data in the form of captured images from the video sequences obtained by the system. From those frames we need to detect the animal presence. The following are the steps to be followed:

- **Step-1:** Firstly need to gather all the captured image folders from the source to the local drive
- **Step-2:** Once all the images are downloaded now we need to install a Bosch built software called as labelImg.
- **Step-3:** labelImg is a software by which we can exactly point the animal in the given image frame by getting the pointed animal's frame details and its all other factors like type of animal into a json file.
- **Step**-4: That obtained json file and all the details of json file will be further used for the training of the system.

- **Step**-5: That json file is automatically generated by the software when we start pointing out the animal in the particular picture frame.
- **Step-6:** Each picture gets there respective unique json files created and saved once we point the animal and save that in the software.
- **Step-7:** All the above steps are to be followed to get the perfect coordinates of the animal present in the frame.

2.3 TASK-3:

Analysis of MATLAB sequences for target object selection: In this task we will be analyzing the MATLAB sequences using PlotStr which is the main used library for ADAS. The following are the steps to be followed:

- **Step-1:** At first to start with the analysis of the sequences we need to get the knowledge of the basic theory behind the object selection.
- **Step-2:** There are many algorithms involved in the target object selection process. Some of them are:
    - Algo-1: Imitating the object that is present in front of our autonomous vehicle.
    - Algo-2: Here we use the concept of parallel line traffic and the imitating algorithm also.
- **Step-3:** Considering the scenarios around our vehicle system will be selecting any of the both algorithms and follows that accordingly.
- **Step-4:** Now after getting all the theory concepts of the object selection we will start with the analyses of the video sequences obtained.
- **Step-5:** For this analysis after installing the MATLAB software we need to load all the necessary libraries of the PlotStr.
- **Step-6:** Once successful setup is done now, we need to first load the sample signals and all other data to it and then later we need to load our particular MATLAB video sequence and start analyzing the data in that video file.
- **Step-7:** Mainly we need to concentrate on the vehicle losing its target object, going out of the sight to capture a target object etc., all these kinds of things are to be noted.
- **Step-8:** All this noted data will be used to reduce the data lose and maintain exact accuracy.

Chapter 3

Technologies Used

3.1 MACHINE LEARNING:

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information.

Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

This machine learning tutorial gives you an introduction to machine learning along with the wide range of machine learning techniques such as Supervised, Unsupervised, and Reinforcement learning. You will learn about regression and classification models, clustering methods, hidden Markov models, and various sequential models.

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of Machine Learning.

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced **by** Arthur Samuel in 1959.

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**.** The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning:

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Characteristics of Machine Learning:

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Machine learning Life cycle:

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. But how does a machine learning system work? So, it can be described using the life cycle of machine learning. Machine learning lifecycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.

Machine learning life cycle involves seven major steps, which are given below:

- **Gathering Data**
  Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.
- **Data preparation**
  After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.
- **Data Wrangling**
  Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.
- **Analyse Data**
  The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the

determination of the type of the problems, where we select the machine learning techniques such as **Classification**, **Regression**, **Cluster analysis**, **Association**, etc. then build the model using prepared data, and evaluate the model. Hence, in this step, we take the data and use machine learning algorithms to build the model.

- **Train the model**

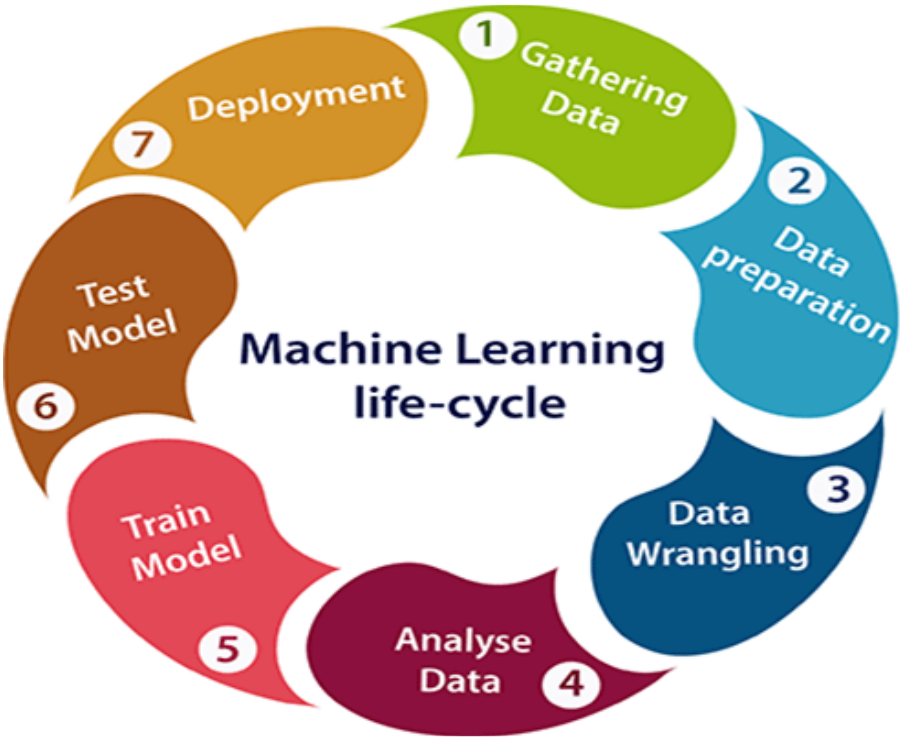
  Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem. We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and features.

- **Test the model**

  Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it. Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

- **Deployment**

  The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.



Classification of Machine Learning:

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**

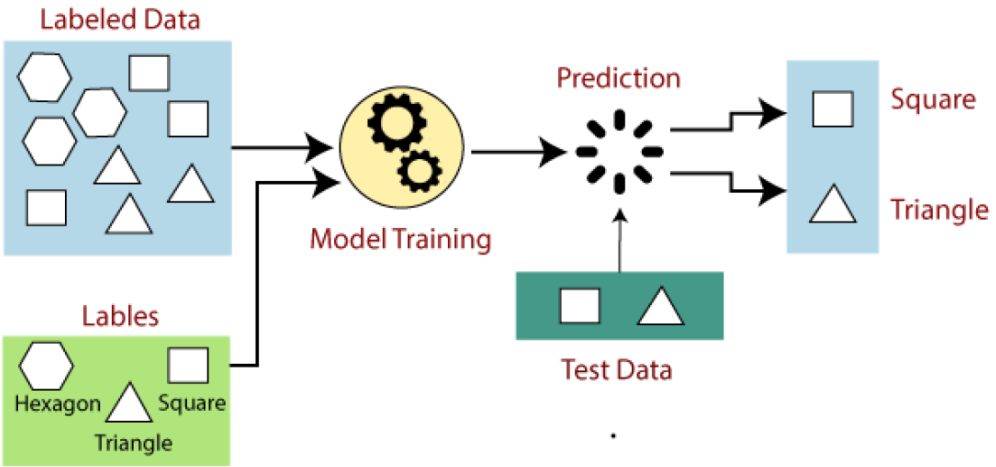
Supervised Machine Learning:

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is **to find a mapping function to map the input variable(x) with the output variable(y).**

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

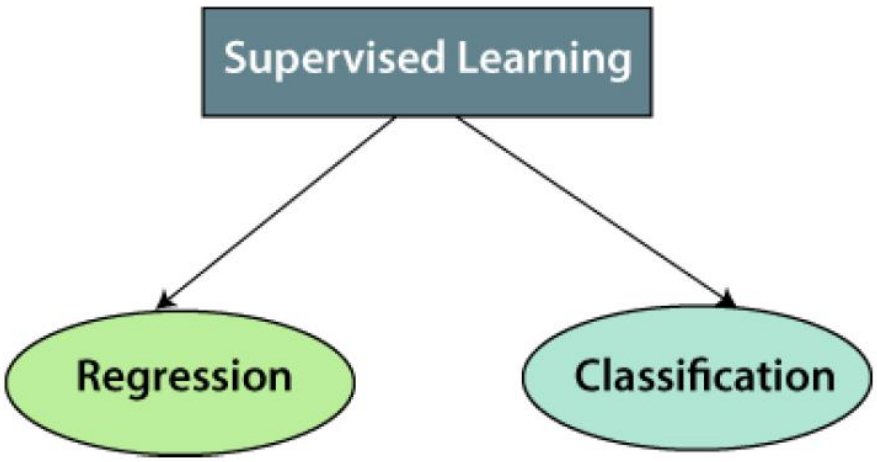


Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems



1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Tress
- Non-Linear Regression
- Bayesian Line Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machines

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output based on prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

Unsupervised Learning:

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format**.**

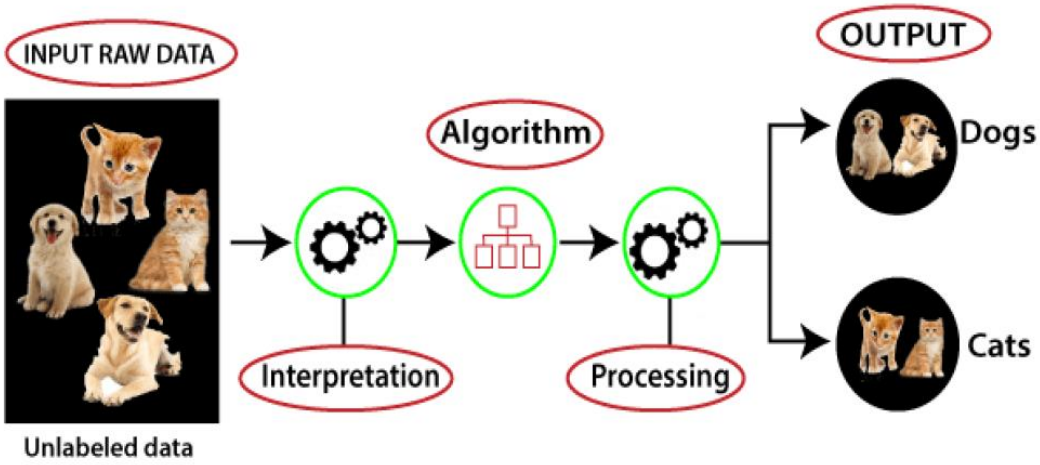
Why Unsupervised Learning?

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

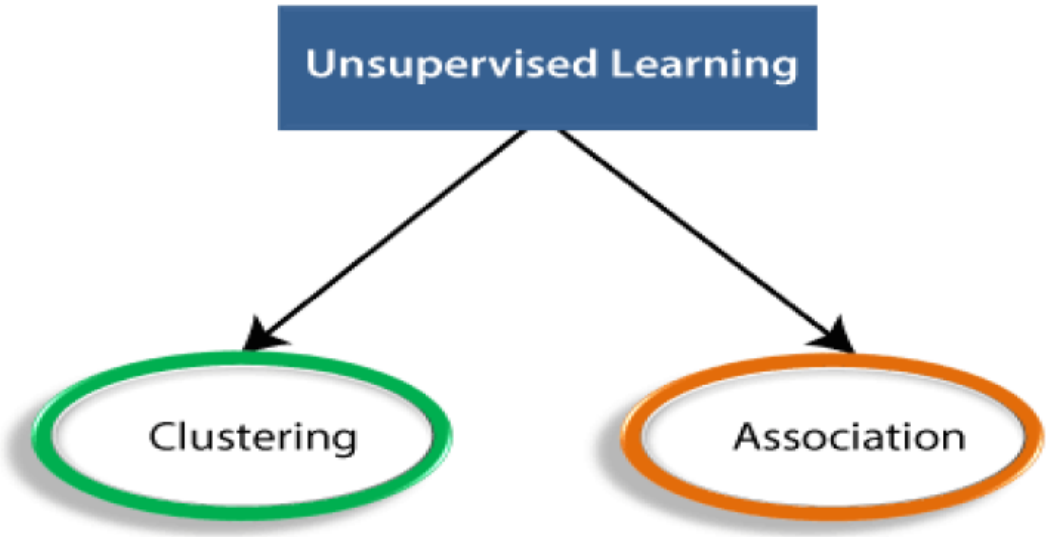
Working of Unsupervised Learning:

Working of unsupervised learning can be understood by the below diagram:



Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



1.Clustering:

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

2.Association:

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Unsupervised Learning algorithms:

Below is the list of some popular unsupervised learning algorithms:

- K-means Clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

Advantages of Unsupervised Learning:

- Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
- Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

3.2 ARTIFICIAL INTELLIGENCE:

In today's world, technology is growing very fast, and we are getting in touch with different new technologies day by day. Here, one of the booming technologies of computer science is Artificial Intelligence which is ready to create a new revolution in the world by making intelligent machines. The Artificial Intelligence is now all around us. It is currently working with a variety of subfields, ranging from general to specific, such as self-driving cars, playing chess, proving theorems, playing music, Painting, etc.

AI is one of the fascinating and universal fields of Computer science which has a great scope in future. AI holds a tendency to cause a machine to work as a human.

Artificial Intelligence is composed of two words Artificial and Intelligence, where Artificial defines "man-made," and intelligence defines "thinking power", hence AI means "a man-made thinking power."

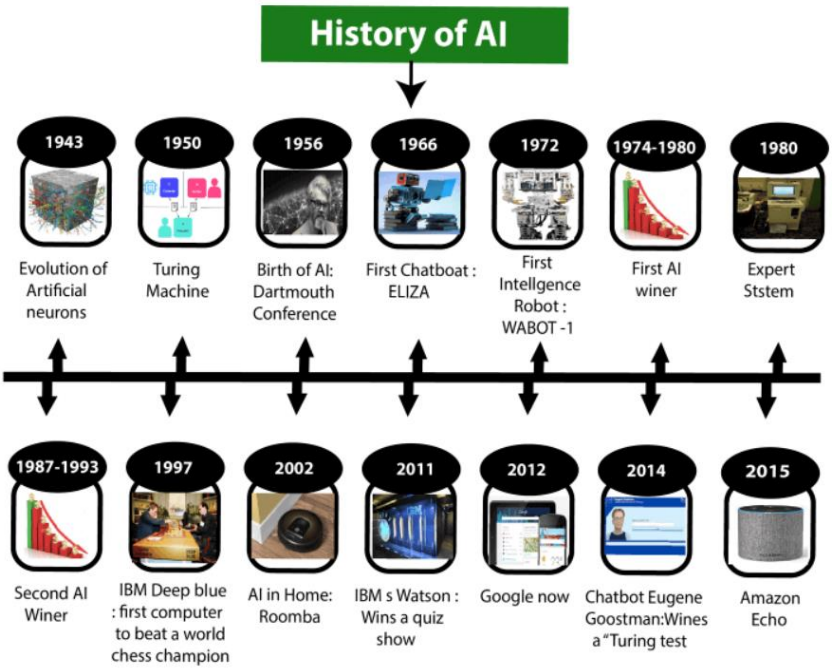
Artificial Intelligence exists when a machine can have human based skills such as learning, reasoning, and solving problems.

With Artificial Intelligence you do not need to preprogram a machine to do some work, despite that you can create a machine with programmed algorithms which can work with own intelligence, and that is the awesomeness of AI.

It is believed that AI is not a new technology, and some people says that as per Greek myth, there were Mechanical men in early days which can work and behave like humans.

History of Artificial Intelligence:

Artificial Intelligence is not a new word and not a new technology for researchers. This technology is much older than you would imagine. Even there are the myths of Mechanical men in Ancient Greek and Egyptian Myths. Following are some milestones in the history of AI which defines the journey from the AI generation to till date development.



Maturation of Artificial Intelligence (1943-1952)

- **Year 1943:** The first work which is now recognized as AI was done by Warren McCulloch and Walter pits in 1943. They proposed a model of artificial neurons**.**
- **Year 1949:** Donald Hebb demonstrated an updating rule for modifying the connection strength between neurons. His rule is now called Hebbian learning**.**
- **Year 1950:** The Alan Turing who was an English mathematician and pioneered Machine learning in 1950. Alan Turing publishes "Computing Machinery and Intelligence" in which he proposed a test. The test can check the machine's ability to exhibit intelligent behavior equivalent to human intelligence, called a Turing test.

The birth of Artificial Intelligence (1952-1956)

- **Year 1955:** An Allen Newell and Herbert A. Simon created the "first artificial intelligence program which was named as "Logic Theorist". This program had proved 38 of 52 Mathematics theorems and find new and more elegant proofs for some theorems.
- **Year 1956:** The word "Artificial Intelligence" first adopted by American Computer scientist John McCarthy at the Dartmouth Conference. For the first time, AI coined as an academic field.

At that time high-level computer languages such as FORTRAN, LISP, or COBOL were invented. And the enthusiasm for AI was very high at that time.

The golden years-Early enthusiasm (1956-1974)

- **Year 1966:** The researchers emphasized developing algorithms which can solve mathematical problems. Joseph Weizenbaum created the first chatbot in 1966, which was named as ELIZA.
- **Year 1972:** The first intelligent humanoid robot was built in Japan which was named as WABOT-1.

The first AI winter (1974-1980)

- The duration between years 1974 to 1980 was the first AI winter duration. AI winter refers to the time period where computer scientist dealt with a severe shortage of funding from government for AI researches.
- During AI winters, an interest of publicity on artificial intelligence was decreased.

A boom of AI (1980-1987)

- **Year 1980:** After AI winter duration, AI came back with "Expert System". Expert systems were programmed that emulate the decision-making ability of a human expert.
- In the Year 1980, the first national conference of the American Association of Artificial Intelligence was held at Stanford University**.**

The second AI winter (1987-1993)

- The duration between the years 1987 to 1993 was the second AI Winter duration.
- Again, Investors and government stopped in funding for AI research as due to high cost but not efficient result. The expert system such as XCON was very cost effective.

The emergence of intelligent agents (1993-2011)

- **Year 1997:** In the year 1997, IBM Deep Blue beats world chess champion, Gary Kasparov, and became the first computer to beat a world chess champion.
- **Year 2002:** for the first time, AI entered the home in the form of Roomba, a vacuum cleaner.
- **Year 2006:** AI came in the Business world till the year 2006. Companies like Facebook, Twitter, and Netflix also started using AI.

Deep learning, big data and artificial general intelligence (2011-present)

- **Year 2011:** In the year 2011, IBM's Watson won jeopardy, a quiz show, where it had to solve the complex questions as well as riddles. Watson had proved that it could understand natural language and can solve tricky questions quickly.
- **Year 2012:** Google has launched an Android app feature "Google now", which was able to provide information to the user as a prediction.
- **Year 2014:** In the year 2014, Chatbot "Eugene Goostman" won a competition in the infamous "Turing test."
- **Year 2018:** The "Project Debater" from IBM debated on complex topics with two master debaters and also performed extremely well.
- Google has demonstrated an AI program "Duplex" which was a virtual assistant and which had taken hairdresser appointment on call, and lady on other side didn't notice that she was talking with the machine.

Now AI has developed to a remarkable level. The concept of Deep learning, big data, and data science are now trending like a boom. Nowadays companies like Google, Facebook, IBM, and Amazon are working with AI and creating amazing devices. The future of Artificial Intelligence is inspiring and will come with high intelligence.

Why Artificial Intelligence?

Following are some main reasons to learn about AI:

- With the help of AI, you can create such software or devices which can solve real-world problems very easily and with accuracy such as health issues, marketing, traffic issues, etc.
- With the help of AI, you can create your personal virtual Assistant, such as Cortana, Google Assistant, Siri, etc.
- With the help of AI, you can build such Robots which can work in an environment where survival of humans can be at risk.
- AI opens a path for other new technologies, new devices, and new Opportunities.

Goals of Artificial Intelligence:

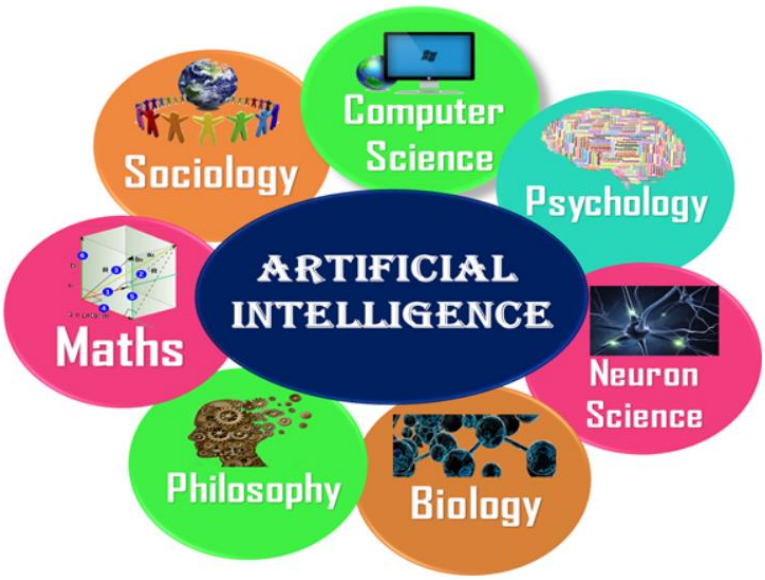
- Replicate human intelligence
- Solve Knowledge-intensive tasks
- An intelligent connection of perception and action
- Building a machine which can perform tasks that requires human intelligence such as:
    - Proving a theorem
    - Playing chess
    - Plan some surgical operation
    - Driving a car in traffic
- Creating some system which can exhibit intelligent behavior, learn new things by itself, demonstrate, explain, and can advise to its user.

What Comprises to Artificial Intelligence?

Artificial Intelligence is not just a part of computer science even it's so vast and requires lots of other factors which can contribute to it. To create the AI first we should know that how intelligence is composed, so the Intelligence is an intangible part of our brain which is a combination of Reasoning, learning, problem-solving perception, language understanding, etc.

To achieve the above factors for a machine or software Artificial Intelligence requires the following discipline:

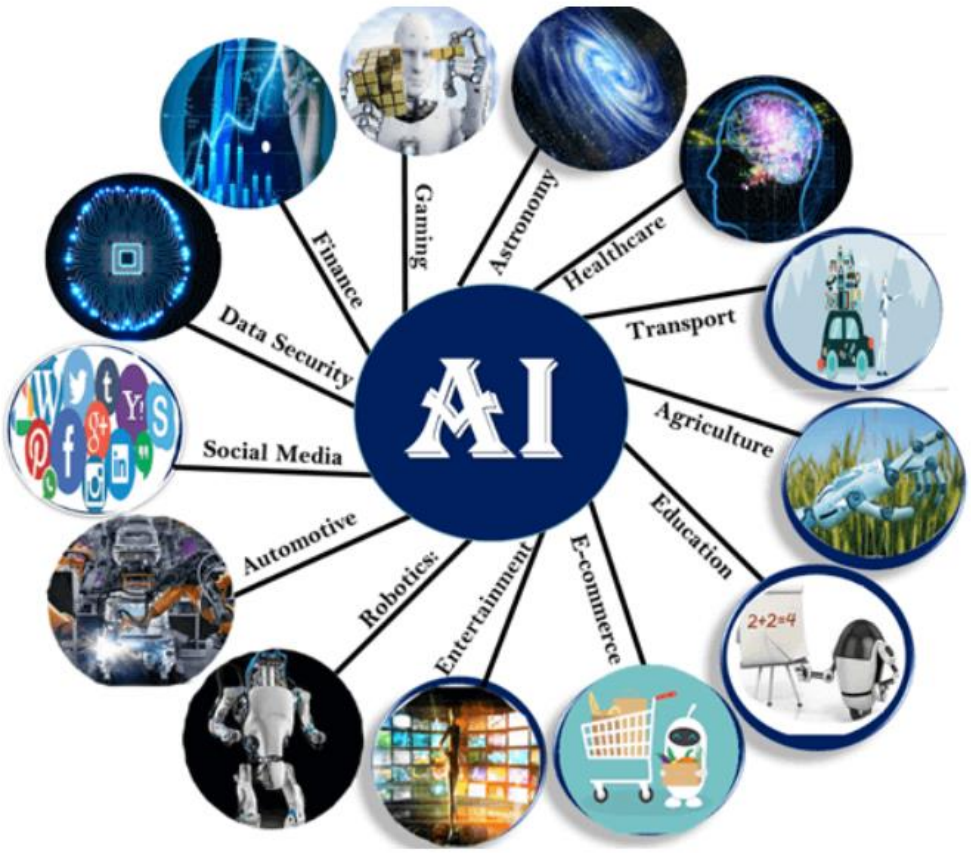
- Mathematics
- Biology
- Psychology
- Sociology
- Computer Science
- Neurons Study
- Statistics



Application of AI

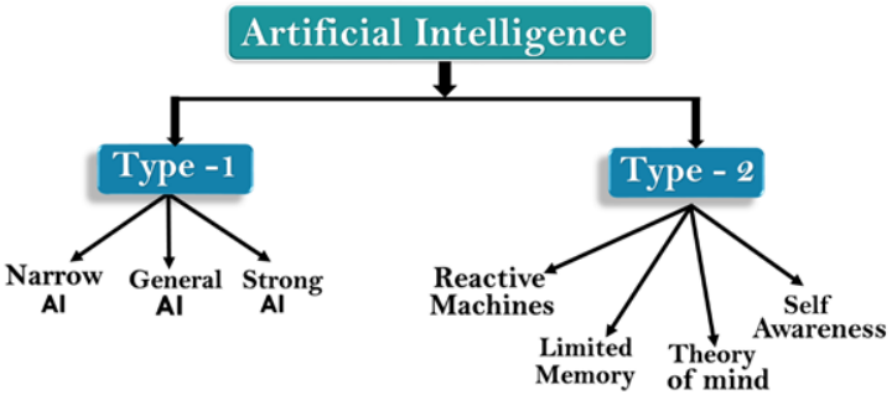
Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc. AI is making our daily life more comfortable and faster.

Following are some sectors which have the application of Artificial Intelligence:



Types of Artificial Intelligence:

Artificial Intelligence can be divided in various types, there are mainly two types of main categorization which are based on capabilities and based on functionally of AI. Following is flow diagram which explain the types of AI.



AI type-1: Based on Capabilities

1. Weak AI or Narrow AI:

- Narrow AI is a type of AI which is able to perform a dedicated task with intelligence. The most common and currently available AI is Narrow AI in the world of Artificial Intelligence.
- Narrow AI cannot perform beyond its field or limitations, as it is only trained for one specific task. Hence it is also termed as weak AI. Narrow AI can fail in unpredictable ways if it goes beyond its limits.
- Apple Siriis a good example of Narrow AI, but it operates with a limited pre-defined range of functions.
- IBM's Watson supercomputer also comes under Narrow AI, as it uses an Expert system approach combined with Machine learning and natural language processing.

- Some Examples of Narrow AI are playing chess, purchasing suggestions on e-commerce site, self-driving cars, speech recognition, and image recognition.

2. General AI:

- General AI is a type of intelligence which could perform any intellectual task with efficiency like a human.
- The idea behind the general AI to make such a system which could be smarter and think like a human by its own.
- Currently, there is no such system exist which could come under general AI and can perform any task as perfect as a human.
- The worldwide researchers are now focused on developing machines with General AI.
- As systems with general AI are still under research, and it will take lots of efforts and time to develop such systems.

3. Super AI:

- Super AI is a level of Intelligence of Systems at which machines could surpass human intelligence, and can perform any task better than human with cognitive properties. It is an outcome of general AI.
- Some key characteristics of strong AI include capability include the ability to think, to reason,solve the puzzle, make judgments, plan, learn, and communicate by its own.
- Super AI is still a hypothetical concept of Artificial Intelligence. Development of such systems in real is still world changing task.

Artificial Intelligence type-2: Based on functionality

1. Reactive Machines

- Purely reactive machines are the most basic types of Artificial Intelligence.
- Such AI systems do not store memories or past experiences for future actions.
- These machines only focus on current scenarios and react on it as per possible best action.

- IBM's Deep Blue system is an example of reactive machines.
- Google's AlphaGo is also an example of reactive machines.

2. Limited Memory

- Limited memory machines can store past experiences or some data for a short period of time.
- These machines can use stored data for a limited time period only.
- Self-driving cars are one of the best examples of Limited Memory systems. These cars can store recent speed of nearby cars, the distance of other cars, speed limit, and other information to navigate the road.

3. Theory of Mind

- Theory of Mind AI should understand the human emotions, people, beliefs, and be able to interact socially like humans.
- This type of AI machines are still not developed, but researchers are making lots of efforts and improvement for developing such AI machines.

4. Self-Awareness

- Self-awareness AI is the future of Artificial Intelligence. These machines will be super intelligent, and will have their own consciousness, sentiments, and self-awareness.
- These machines will be smarter than human mind.
- Self-Awareness AI does not exist in reality still and it is a hypothetical concept.

Advantages of Artificial Intelligence

Following are some main advantages of Artificial Intelligence:

- **High Accuracy with less errors:** AI machines or systems are prone to less errors and high accuracy as it takes decisions as per pre-experience or information.
- **High-Speed:** AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.

- **High reliability:** AI machines are highly reliable and can perform the same action multiple times with high accuracy.
- **Useful for risky areas:** AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.
- **Digital Assistant:** AI can be very useful to provide digital assistant to the users such as AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
- **Useful as a public utility:** AI can be very useful for public utilities such as a self-driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

Disadvantages of Artificial Intelligence

Every technology has some disadvantages, and thesame goes for Artificial intelligence. Being so advantageous technology still, it has some disadvantages which we need to keep in our mind while creating an AI system. Following are the disadvantages of AI:

- **High Cost:** The hardware and software requirement of AI is very costly as it requires lots of maintenance to meet current world requirements.
- **Can't think out of the box:** Even we are making smarter machines with AI, but still they cannot work out of the box, as the robot will only do that work for which they are trained, or programmed.
- **No feelings and emotions:** AI machines can be an outstanding performer, but still it does not have the feeling so it cannot make any kind of emotional attachment with human, and may sometime be harmful for users if the proper care is not taken.
- **Increase dependency on machines:** With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.
- **No Original Creativity:** As humans are so creative and can imagine some new ideas but still AI machines cannot beat this power of human intelligence and cannot be creative and imaginative.

3.3 Advanced Driver Assistance System (ADAS)

Almost all vehicle accidents are caused by human error, which can be avoided with Advanced Driver Assistance Systems (ADAS). The role of ADAS is to prevent deaths and injuries by reducing the number of car accidents and the serious impact of those that cannot be avoided.

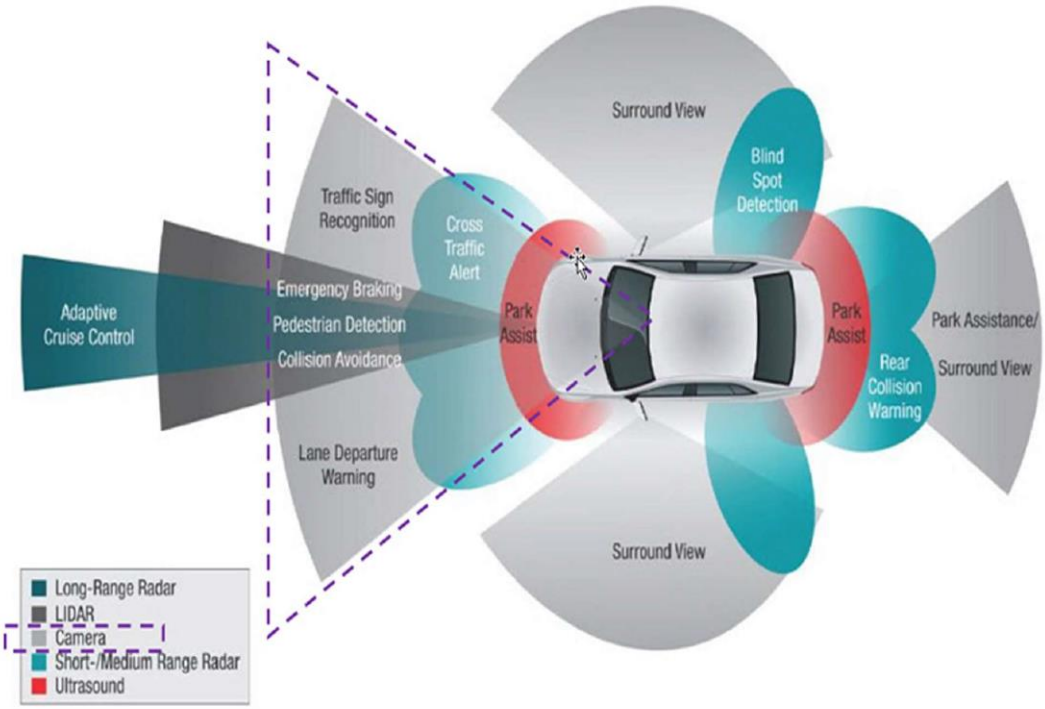
Essential safety critical ADAS applications include:

- Pedestrian detection/avoidance
- Lane departure warning/correction
- Traffic sign recognition
- Automatic emergency braking
- Blind spot detection

These lifesaving systems are key to ensuring the success of ADAS applications, incorporating the latest interface standards and running multiple vision-based algorithms to support real-time multimedia, vision co-processing, and sensor fusion subsystems. The “SmartPhonezation” of ADAS applications is the beginning steps to realization of autonomous vehicles.

How does ADAS work?

Automobiles are the foundation of the next generation of mobile-connected devices, with rapid advances being made in autonomous vehicles. Autonomous application solutions are partitioned into various chips, called SoCs (systems on a chip). These chips connect sensors to actuators through interfaces and high-performance ECUs (electronic controller units). Self-driving cars use a variety of these applications and technologies to gain 360-degree vision, both near (in the vehicle’s immediate vicinity) and far. That means hardware designs are using more advanced process nodes to meet ever-higher performance targets while simultaneously reducing demands on power and footprint.



ADAS Applications

Significant automotive safety improvements in the past (e.g., shatter-resistant glass, three-point seatbelts, airbags) were passive safety measures designed to minimize injury during an accident. Today, ADAS systems actively improve safety with the help of embedded vision by reducing the occurrence of accidents and injury to occupants. The implementation of cameras in the vehicle involves a new AI function that uses sensor fusion to identify and process objects. Sensor fusion, similar to the human brain process information, combines large amounts of data with the help of image recognition software, ultrasound sensors, lidar, and radar. This technology can physically respond faster than a human driver ever could. It can analyze streaming video in real time, recognize what the video shows, and determine how to react to it.

Some of the most common ADAS applications are:

- **Adaptive Cruise Control:** Adaptive cruise control (ACC) is particularly helpful on the highway, where drivers can find it difficult to monitor their speed and other cars over a long period of time. Advanced cruise control can

automatically accelerate, slow down, and at times stop the vehicle, depending on the actions of other objects in the immediate area.

- **Glare-Free High Beam and Pixel Light:** Glare-free high beam and pixel light uses sensors to adjust to darkness and the vehicle's surroundings without disturbing oncoming traffic. This new headlight application detects the lights of other vehicles and redirects the vehicle's lights away to prevent other road users from being temporarily blinded.
- **Adaptive Light Control:** Adaptive light control adapts the vehicle's headlights to external lighting conditions. It changes the strength, direction, and rotation of the headlights depending on the vehicle's environment and darkness.
- **Automatic Parking:** Automatic parking helps inform drivers of blind spots, so they know when to turn the steering wheel and stop. Vehicles equipped with rearview cameras have a better view of their surroundings than traditional side mirrors. Some systems can even complete parking automatically without the driver's help by combining the input of multiple sensors.
- **Autonomous Valet Parking**: Autonomous valet parking is a new technology that works via vehicle sensor meshing, 5G network communication, with cloud services that manage autonomous vehicles in parking areas. The vehicle's sensors provide the vehicle with information about where it is, where it needs to go, and how to get there safely. All this information is methodically evaluated and used to perform drive acceleration, braking, and steering until the vehicle is safely parked.
- **Navigation System**: Car navigation systems provide on-screen instructions and voice prompts to help drivers follow a route while concentrating on the road. Some navigation systems can display exact traffic data and, if necessary, plan a new route to avoid traffic jams. Advanced systems may even offer Heads Up Displays (HuD) to reduce driver distraction.
- **Night Vision:** Night vision systems enable drivers to see things that would otherwise be difficult or impossible to see at night. There are two categories of night vision implementations: Active night vision systems project infrared light, and passive systems rely on the thermal energy that comes from cars, animals, and other objects.

- **Blind Spot Monitoring:** Blind spot detection systems use sensors to provide drivers with important information that is otherwise difficult or impossible to obtain. Some systems sound an alarm when they detect an object in the driver's blind spot, such as when the driver tries to move into an occupied lane.
- **Automatic Emergency Braking:** Automatic emergency braking uses sensors to detect whether the driver is in the process of hitting another vehicle or other objects on the road. This application can measure the distance of nearby traffic and alert the driver to any danger. Some emergency braking systems can take preventive safety measures, such as tightening seat belts, reducing speed and adaptive steering to avoid a collision.
- **Crosswind Stabilization:** This relatively new ADAS feature supports the vehicle in counteracting strong crosswinds. The sensors in this system can detect strong pressure acting on the vehicle while driving and apply brakes to the wheels affected by crosswind disturbance.
- **Driver Drowsiness Detection:** Driver drowsiness detection warns drivers of sleepiness or other road distractions. There are several ways to determine whether a driver's attention is decreasing. In one case, sensors can analyze the movement of the driver's head, and heart rate to determine whether they indicate drowsiness. Other systems issue driver alerts similar to the warning signals for lane detection.
- **Driver Monitoring System**: The driver monitoring system is another way of measuring the driver's attention. The camera sensors can analyze whether the driver's eyes are on the road or drifting. Driver monitoring systems can alert drivers with noises, vibrations in the steering wheel, or flashing lights. In some cases, the car will take the extreme measure of stopping the vehicle completely.
- **5G and V2X:** This hot new 5G ADAS feature, with increased reliability and lower latency, provides communication between the vehicle and other vehicles or pedestrians, generally referred to as V2X. Today, millions of vehicles connect to cellular networks for real-time navigation. This application will enhance existing methods and the cellular network to improve situational awareness, control or suggest speed adjustments to account for traffic

congestion, and update GPS maps with real-time updates. V2X is essential to support over-the-air (OTA) software updates for the now-extensive range of software-driven systems in cars, from map updates to bug fixes to security updates and more.

Why is ADAS important?

According to the August 2016 Traffic Safety Facts Research Note by the National Highway Traffic Safety Administration (NHTSA), The Nation lost 35,092 people in crashes on U.S. roadways during 2015. This 7.2% increase was the largest percentage increase in nearly 50 years. An analysis revealed that about 94% of those accidents were caused by human error, and the rest by the environment and mechanical failures.

The opportunity to reduce car accidents is making automotive ADAS even more critical. Automatic emergency braking, pedestrian detection, surround view, parking assist, driver drowsiness detection, and gaze detection are among the many ADAS applications that assist drivers with safety-critical functionality to reduce car accidents and save lives.

The future of ADAS:

The increasing amount of automotive electronic hardware and software requires significant changes in today's automobile design process to address the convergence of conflicting goals:

- Increased reliability
- Reduced costs
- Shorter development cycles

The trend is shifting from distributed ADAS electronic controller units (ECUs) to a more integrated ADAS domain controller with centralized ECUs. This means that we are currently at what SAE International designates as Level 2 (Partial Driving Automation), where the vehicle can control both steering and accelerating or decelerating but falls short of self-driving because a human sits in the driver's seat

and can take control of the car at any time. Shifting toward fully autonomous cars-vehicles capable of sensing their environment and operating without human involvement-requires an increase in the electronic architecture of these vehicles.

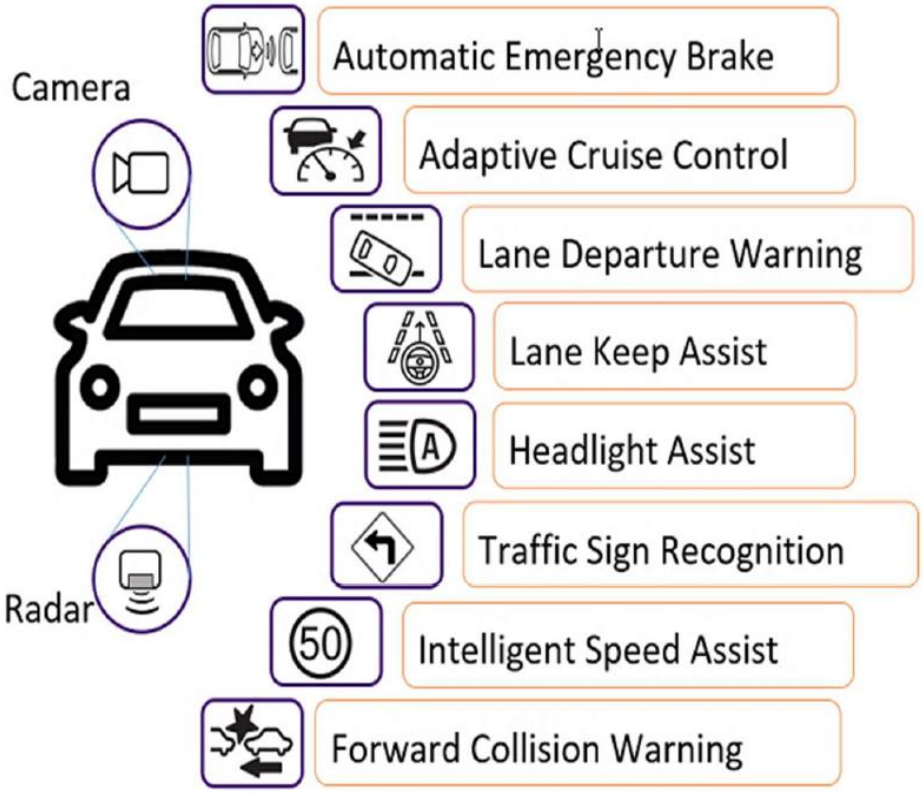
With the increase in electronic architecture comes an increase in the volume of data. To handle this data, the new integrated domain controllers require higher computing performance, lower power consumption, and smaller packaging.

The adoption of 64-bit processors, neural networks and AI accelerators to handle the high volume of data requires the latest semiconductor features, semiconductor process technologies, and interconnecting technologies to support ADAS capabilities.

The reduction of electronic modules leads to centralized computing architectures, requiring critical automotive building blocks, including processors with vision processing capabilities, neural networks, and sensor fusion. All while addressing the need for quality, safety and security.

Every aspect of the car is designed to be more connected, requiring subsystem and SoC designers to expand the scope of safety measures beyond the traditional steps taken to ensure physical safety. Applying the latest embedded computer vision and deep learning techniques to automotive SoCs brings greater accuracy, power efficiency, and performance to ADAS systems.

Popular ADAS Camera Functions in Market:

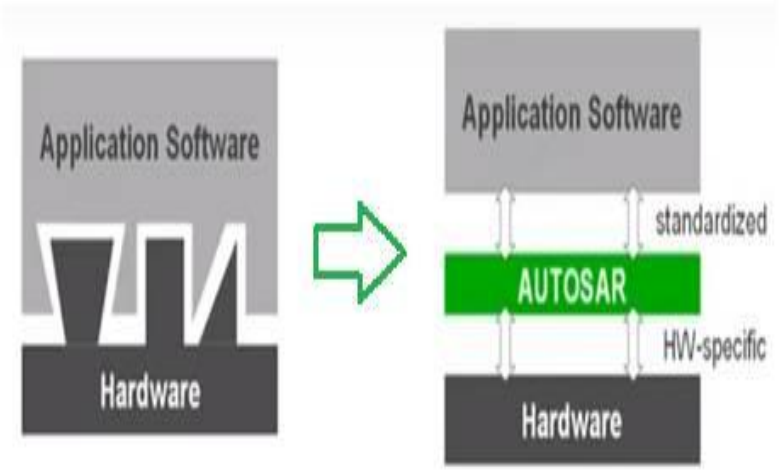


3.4 AUTOSAR:

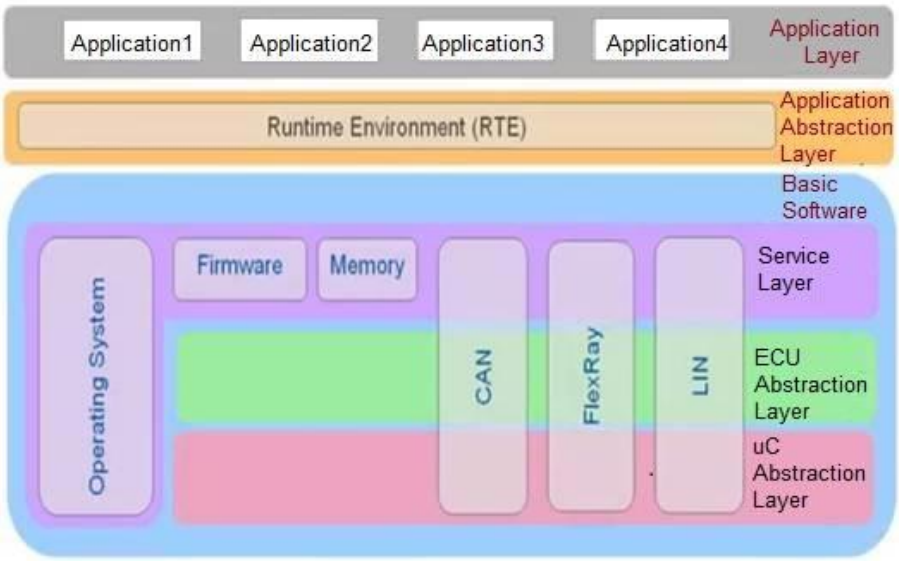
AUTomotive Open System ARchitecture (**AUTOSAR**) is a development partnership of automotive interested parties founded in 2003. It pursues the objective to create and establish an open and standardized software architecture for automotive electronic control units (ECUs). Goals include the scalability to different vehicle and platform variants, transferability of software, the consideration of availability and safety requirements, a collaboration between various partners, sustainable use of natural resources, and maintainability during the product lifecycle

Some more points:-

- AUTOSAR is an alliance of more than 150 companies of automotive manufacturers and automotive suppliers.
- It is short form of AUTomotive Open System Architecture.
- The aim of AUTOSAR is to establish an open and standardized automotive software architecture.
- It simplifies automotive software development.
- AUTOSAR helps in managing product modifications, reliable upgrades, reusability and scalability in software development.



- Due to innovations in automotive electronics industry, software of ECU (Electronic Control Unit) is highly depend upon hardware. This leads to higher costs and more efforts for relocation of functions between ECUs.
- AUTOSAR introduces standardized layer between application software and ECU (i.e. hardware). This makes software independent from chosen microcontroller and OEM. This simplifies development processes and helps in reuse of application software.



APPLICATION LAYER

The application layer consists of interconnected software components. A software component is a reusable self-contained artifact which implements a function with given properties. These software components encapsulate parts of application and communicate over hardware independent bus called as VFB ("Virtual Functional Bus"). If software components are tied to different ECUs, then they can communicate via their respective RTEs and basic software layers. In this case, RTE implements VFB on a dedicated ECU. The "Basic software" offers various service functions for application developers.

Virtual Functional Bus

The virtual functional bus is the abstraction of the AUTOSAR Software Components interconnections of the entire vehicle. The communication between different software components and between software components and its environment (e.g. hardware driver, OS, services, etc.) can be specified independently of any underlying hardware (e.g. communication system). The functionality of the VFB is provided by communication patterns.

RTE (RUNTIME ENVIRONMENT)

- All interaction between AUTOSAR Software Components is routed through the AUTOSAR Runtime Environment. The AUTOSAR Interface specification assures the connectivity.
- The AUTOSAR Runtime Environment (RTE) acts as a system level communication center for inter- and intra-ECU information exchange. • The RTE is the runtime representation of the Virtual Function Bus for a specific ECU.
- The RTE provides a communication abstraction to AUTOSAR Software Components providing the same interface and services for inter-ECU (using CAN, LIN, Flexray, MOST, etc.) or intra-ECU communication.
- As the communication requirements of the software components are application dependent, the RTE needs to be tailored.
- The RTE shall support multiple instantiations of software components.

BASIC SOFTWARE

- An AUTOSAR Software Component is not allowed to access Basic Software directly.
- Basic Software is the standardized software layer, which provides services to the SW Components. It does not fulfill any functional job and is situated below the AUTOSAR Runtime Environment. It contains standardized components and ECU specific components.
- The standardized components include following.
    - services including diagnostic protocols; NVRAM, flash and memory management.
    - Communication the communication framework (e.g. CAN, LIN, FlexRay...), the I/O management, and the network management.
- The ECU specific components include following.
    - Operating system
    - Microcontroller abstraction
    - Complex Device Drivers

AUTOSAR Communication Patterns:

Elementary communication patterns

- Client-Server
- Sender-Receiver

Interfaces specify

- what information sender receiver communication transports
- which services with which arguments can be called by client-server

The formal description of the interface is in the software component template, including also data types that can be used and interface compatibility. The detailed behavior of a basic communication pattern is specified by attributes. With those attributes e.g. the length of data queues and the behavior of receivers (blocking, non-blocking, etc.) and senders (send cyclic, etc.) can be defined.

Client-Server communication:

The server is a provider and the client is a user of a service. The client initiates the communication, requesting that the server performs a service, transferring a parameter set if necessary. The server waits for incoming communication requests from a client, performs the requested service and dispatches a response to the client's request. The direction of initiation is used to categorize whether an AUTOSAR Software Component is a client or a server. A single component can be both a client and a server depending on the software realization. After the service request is initiated and until the response of the server is received from the client can be- blocked (synchronous communication).

Sender-Receiver communication:

Model for the asynchronous distribution of information where a sender distributes information to one or several receivers. The sender is not blocked (asynchronous) and neither expects nor gets a response from the receivers (data or control flow), the sender just provides the information and the receivers decides autonomously when and how to use it. It is the responsibility of the communication infrastructure to distribute the information. The sender does not know the identity or the number of receivers.

What Is The Need Of AUTOSAR?

Listed below are the problems faced in conventional ways of writing software for ECUs:

- Embedded systems is a vast field having n number of semiconductor manufacturers, hardware and software platforms which can be selected based on application requirements. Due to such varieties, the development effort is tough and the portability of code is hard which further increases the development cost.
- A automotive is a complex machine which consists of n number of small embedded systems called Electronic control Unit(ECUs) so maintenance and development of code for such controllers is not easy. Further complexity is increased if different ECUs use different MCUs for meeting cost requirements, then each ECU will have different software as hardware platforms will be different.
- To partially standardize things, sometimes there is also a need to develop and follow custom created standard (Custom standard means to develop a protocol for communication which is agreed by all ECUs in network) to communicate with other ECUs. This is the conventional way of writing software which is very hard to maintain and has very less chance of code portability or reuse ability.
- A automobile has n number of parts which are manufactured by different companies called Tier 1 companies which supplies the parts to OEMs like BMW, Volkswagen, etc. Today almost all mechanical parts are becoming intelligent by adding ECU in them to increase control and efficiency. So those ECUs also need to have a common way of communication to communicate with the ECUs of OEM for this again a custom standard needs to be implemented and maintained.

Hence a standardized software development infrastructure was needed to solve these problems and AUTOSAR solves this very well.

AUTOSAR uses a layered architecture which has different layers dedicated to perform different operations and abstraction. The application code is fully portable as AUTOSAR is designed in such a way that the application code is written independent of the hardware so the same application code can run on different hardware platforms. AUTOSAR has a layer dedicated to support hardware functionalities called MCAL (Micro controller abstraction) layer which has drivers for accessing the underlying hardware peripherals of MCU. As AUTOSAR provides standard way of communication, ECUs can communicate with each other irrespective of ECU developer (whether OEM or Tier1) and hence there is no need to maintain custom standard of communication. ECUs utilizing AUTOSAR can communicate with each other irrespective of underlying differences in hardware. Mostly chip manufacturers provides MCAL layer of AUTOSAR, but if they don't then the developer needs to write his own MCAL layer or outsource to companies providing such services.

Types Of AUTOSAR:

There are two types of AUTOSAR architectures named as Classic and Adaptive. The classic have all the modules which are generally needed for a application whereas the Adaptive can be configured and adapted according to application by removing unnecessary modules. Current Classic release version is 4.4.0 and current adaptive version is 19.03

1) Classic Platform:

The AUTOSAR classic platform is the standard for embedded real-time ECUs based on OSEK. Its main deliverable is specifications. The architecture distinguishes between three software layers that run on a microcontroller application, runtime environment (RTE) and basic software (BSW). The application software layer is mostly hardware independent. Communication between software components and access to BSW happens via RTE, which represents the full interface for applications.

The BSW is divided in three major layers and complex drivers:

- Services
- Electronic control unit (ECU) abstraction
- Microcontroller abstraction

Services are divided further, into functional groups representing the infrastructure for system, memory and communication services.

One essential concept of the Classic Platform is the Virtual Functional Bus (VFB). This virtual bus is an abstract set of RTEs that are not yet deployed to specific ECUs and decouples the applications from the infrastructure. It communicates via dedicated ports, which means that the communication interfaces of the application software must be mapped to these ports. The VFB handles communication within the individual ECU and between ECUs. From an application point of view, no detailed knowledge of lower-level technologies or dependencies is required. This supports hardware-independent development and usage of application software.

The Classic Platform also enables the integration of non-AUTOSAR systems such as GENIVI by using the Franca Interface Definition Language.

2) Adaptive platform:

New use-cases required the development of the adaptive platform. One example is automated driving, in the context of which the driver temporarily and/or partially transfers responsibility for driving to the vehicle. This can require communication with traffic infrastructure (e.g. traffic signs and -lights), cloud servers (e.g. to access the latest traffic information or map data), or the use of microprocessors and high-performance computing hardware for parallel processing, e.g., graphic processing units (GPU)

Further, Car-2-X applications require interaction to vehicles and off-board systems. That means that the system has to provide secure on-board communication, support of cross-domain computing platforms, smartphone integration, integration of non-AUTOSAR systems, and so on. Also, cloud-based services will require dedicated means for security, such as secure cloud interaction and emergency vehicle preemption. They will enable remote and distributed services, such as remote diagnostics, over the air (OTA) update, repair, and exchange handling.

To support dynamic deployment of customer applications and to provide an environment for applications that require high-end computing power AUTOSAR is currently standardizing the AUTOSAR Adaptive Platform. Its core is an operating system based on the POSIX standard. The operating system can be used from the

application via a subset of the POSIX according to IEEE 1003.13 (namely PSE51). One of the key features of the Adaptive Platform is service-oriented communication since the Platform is based on the Service - Oriented Architecture.

Adaptive AUTOSAR is developed and written using C++ which is an object-oriented programming language. The communication protocol used for the in-vehicle networking is SOME/IP, based on Ethernet. Two types of interfaces are available: services and APIs. The platform consists of functional clusters which are grouped in services and the AUTOSAR adaptive platform foundation.

Functional clusters:

- Assemble functions of the adaptive platform
- Define clustering of requirements specification
- Describe behavior of software platform from application and network perspective
- Do not constrain the final SW design of the architecture implementing the Adaptive Platform.

Functional clusters in AUTOSAR Adaptive Platform have to have at least one instance per (virtual) machine while services may be distributed in the in-car network.

Adaptive platform services include:

- Update and Configuration management
- State Management
- Network Management
- Diagnostics

The adaptive platform contains both specification and code. In comparison to the Classic Platform, AUTOSAR develops an implementation to shorten the validation cycle and illustrate the underlying concepts. This implementation is available to all AUTOSAR partners.

3.5 MATLAB:

MATLAB is a software package for high-performance mathematical computation, visualization, and programming environment. It provides an interactive environment with hundreds of built-in functions for technical computing, graphics, and animations.

MATLAB stands for Matrix Laboratory. MATLAB was written initially to implement a simple approach to matrix software developed by the LINPACK (Linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a modern programming language environment, and it has refined data structures, includes built-in editing and debugging tools, and supports object-oriented programming.

MATLAB is Multi-paradigm. So, it can work with multiple types of programming approaches, such as Functional, Object-Oriented, and Visual.

MATLAB is used in various disciplines of engineering, science, and economics.

MATLAB allows several types of tasks, such as manipulations with matrix, algorithm implementation, data, and functions plotting, and can interact with programs written in other programming languages.

MATLAB is a dynamic and weakly typed programming language.

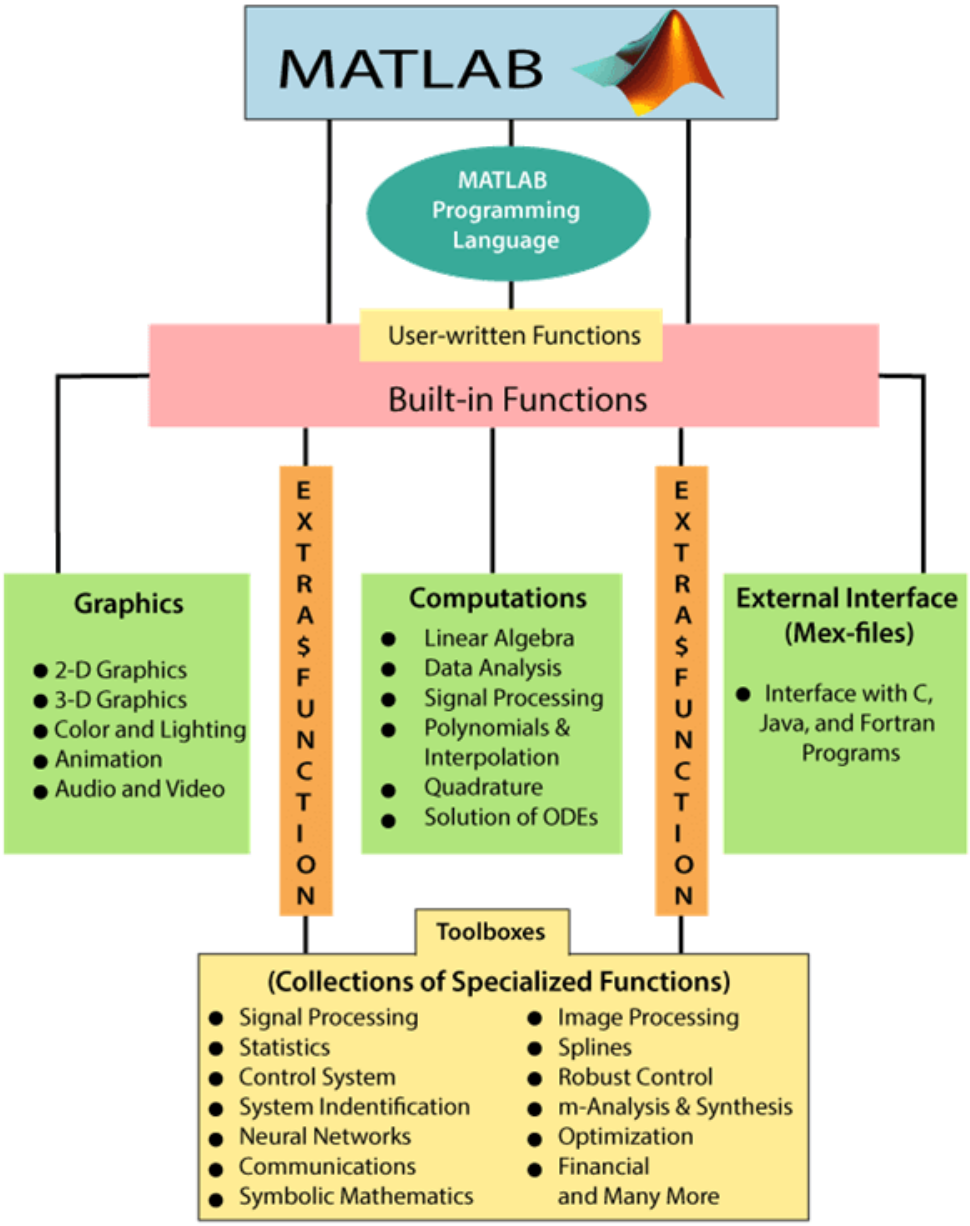
MATLAB environment handles tasks of the declaration of the data type of the variables and provision for an appropriate amount of storage for the variables.

History of MATLAB

The development of the MATLAB started in the late 1970s by Cleve Moler, the chairman of the Computer Science department at the University of New Mexico. Cleve wanted to make his students able to use LINPACK & EISPACK (software libraries for numerical computing, written in FORTRAN), and without learning FORTRAN. In 1984, Cleve Moler with Jack Little & Steve Bangert rewrote MATLAB in C and founded MathWorks. These libraries were known as JACKPAC at that time, later these were revised in 2000 for matrix manipulation and named as LAPACK.

Main Features and Capabilities of MATLAB

The diagram in the figure shows the main features and capabilities of MATLAB.



A schematic diagram of MATLAB's main features.

MATLAB's built-in functions provide excellent tools for linear algebra computations, data analysis, signal processing, optimization, numerical solution of ordinary differential equations (ODEs), quadrate, and many other types of scientific calculations.

Most of these functions use state-of-the-art algorithms. These are numerous functions for 2-D and 3-D graphics, as well as for animations.

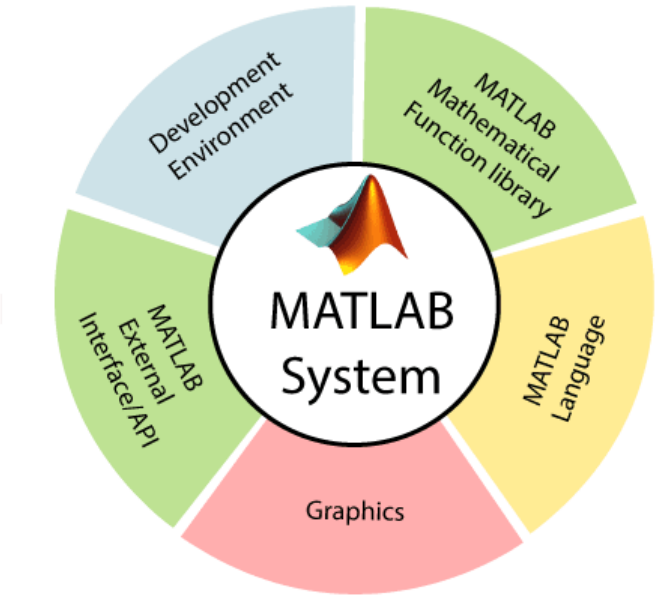
MATLAB supports an external interface to run those programs from within MATLAB. The user is not limited to the built-in functions; he can write his functions in the MATLAB language.

There are also various optional "toolboxes" available from the developers of MATLAB. These toolboxes are a collection of functions written for primary applications such as symbolic computations, image processing, statistics, control system design, and neural networks.

The necessary building components of MATLAB are the matrix. The fundamental data type is the array. Vectors, scalars, real matrices, and complex matrices are all automatically handled as special cases of the primary data type. MATLAB loves matrices and matrix functions. The built-in functions are optimized for vector functions. Therefore, Vectorized commands or codes run much faster in MATLAB.

MATLAB System

The MATLAB systems consist of five main elements:



Development Environment

This is the set of tools and facilities that help you use MATLAB operations and files. Many of these tools are the graphical user interface. It involves the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for considering help, the workspace, reports, and the search path.

MATLAB Mathematical Function Library

This is a vast compilation of computing design ranging from basic functions, like sum, sine, cosine, and complex mathematic, to more sophisticated features like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

MATLAB Language

This is a high level matrix/array language with control flow statement, function, data structure, input/output, and object-oriented programming characteristics. It allows both "programming in the small" to create quick and dirty throw-away programs rapidly and "programming in the large" to create large and complex application functions.

Graphics

MATLAB has extensive facilities for displaying vector and matrices as graphs, as well as annotating and printing these graphs. It contains high-level structures for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also involves low-level structures that allow us to customize the display of graphics fully as well as to build complete graphical user interfaces on our MATLAB applications.

MATLAB External Interfaces/API

This is a library that allows us to write C and FORTRAN programs that interact with MATLAB. It contains facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB Features:

As there are numerous features to describe, but here, we will focus on some of the key features:

- It is designed for numerical as well as symbolic computing.
- It's a high-level language used mainly for engineering and scientific computing.
- It works within a Desktop environment providing full features for iterative exploration, design, and problem-solving.
- Creation of custom plots for visualizing data and tools, with the help of built-in Graphics.
- Specific applications are designed to work with any particular type of problems, such as data classification, control system design and tuning, signal analysis.
- Provides several add-on toolboxes to build a wide range of engineering, scientific, and custom user interface applications.
- Provide interfaces to work with other programming languages such as C, C++, Java, .NET, Python, SQL, Hadoop.

MATLAB Footprints

MATLAB is currently being used by millions of engineers and scientists all over the world. It is nurturing organizations to design the systems and products of the future.

- Applied MATLAB
- MATLAB Capabilities
- Industrial MATLAB

So here for my project we do use the Industrial MATLAB.

Advantage of MATLAB

There are several advantages of MATLAB programming language:

- **Ease of Use**:

  The program can be used as a scratchpad to evaluate expressions typed at the command line, or it can be used to execute large prewritten programs. Applications may be written and changed with the built-in integrated development environment and debugged with the MATLAB debugger. Because the language is so simple to use, it is optimal for the fast prototyping of new applications. Many program development tools are supported to make the program easy to use. They contain an integrated editor/debugger, on-line documentation and manuals, a workspace browser, and extensive demos.

- **Platform Independence:**

  MATLAB is supported on different computer systems, providing a considerable measure of platform independence. The language is provided on Windows 2000/XP/Vista, Linux, various versions of UNIX, and the Macintosh. Applications written on any platform will run on the other entire platform, and information files written on any platform may be read apparently on any other platform. As a result, programs written in MATLAB can shift to new platforms when the needs of the user change.

- **Predefined Functions:**

  MATLAB comes complete with a huge library of predefined functions that provides tested and prepackaged solutions to many primary technical tasks. For example, suppose that we are writing a program that must evaluate the statistics associated with an input data set. In most languages, we would need to write our subroutines or functions to implement calculations such as the arithmetic mean, standard deviation, median, and so on. These and hundreds of other services are built right into the MATLAB language, making your job much more comfortable. In addition to the vast libraries of services built into the basic MATLAB language, there are many special-purpose toolboxes applicable to

help solve complex problems in particular areas. For example, a user can buy standard toolkits to solve problems in signal processing, control systems, communications, image processing, and neural networks, etc. There is also a broad compilation of free user-contributed MATLAB programs that are shared through the MATLAB Web site.

- **Device-Independent Plotting:**

  MATLAB has many basic plotting and imaging commands. The plots and pictures can be displayed on any graphical output device provided by the computer on which MATLAB is running. This facility makes MATLAB an outstanding tool for visualizing technical information.

- **Graphical User Interface:**

  MATLAB contains a tool that allows a programmer to interactively design a Graphical User Interface (**GUI**) for his program. With this capability, the programmer can design refined data-analysis programs that can be operated by relatively inexperienced users.

- **MATLAB Compiler:**

  MATLAB's adaptability and platform independence are produced by compiling MATLAB applications into a machine-independent p-code and then interpreting the p-code instruction at runtime. This method is equivalent to that used by Microsoft's Visual Basic language. Unfortunately, the resulting applications can sometimes execute slowly because the MATLAB code is interpreted rather than compiled.

  A separate MATLAB compiler is available. This compiler can compile MATLAB programs into a real executable that runs faster than the interpreted code. It is a great technique to convert a prototype MATLAB program into an executable suitable for sale and distribution to users.

Disadvantage of MATLAB

There is two major disadvantage of MATLAB programming language:

- **Interpreted language:**

  The first disadvantage is that it is an interpreted language and, therefore, may execute more slowly than compiled language. This problem can be check by properly structuring the MATLAB program.

- **Cost:**

  A full copy of MATLAB is five to ten times more costly than a conventional C or FORTRAN compiler. This comparatively high cost is more than offset by the decreased time necessary for an engineer or scientist to create a working program, so MATLAB is cost-effective for businesses. However, it is too expensive for most individuals to consider purchasing. Fortunately, there is also an inexpensive Student Edition of MATLAB, which is an excellent tool for students wishing to learn the language. The Student Edition of MATLAB is virtually identical to the full edition.

For my task I used a MATLAB tool called as PlotStr. PlotStr, formerly known as D'Accord, is a universal data visualization tool for ADAS systems and was developed in the context of SystemC (but is not limited to it).

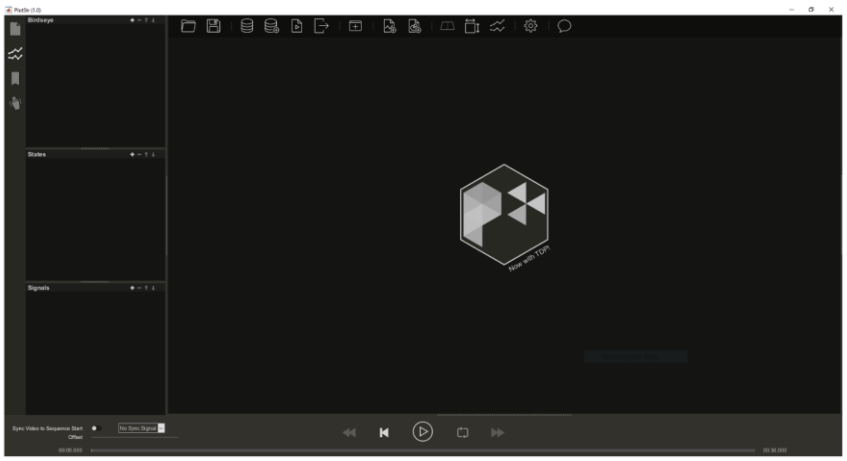


Fig: PlotStr window

Chapter-4

Screenshots of few Implementations

Note: According to security policy of Bosch Global software solutions this screenshots and code part of project not to be revealed

Fig-1: Animal detection frames

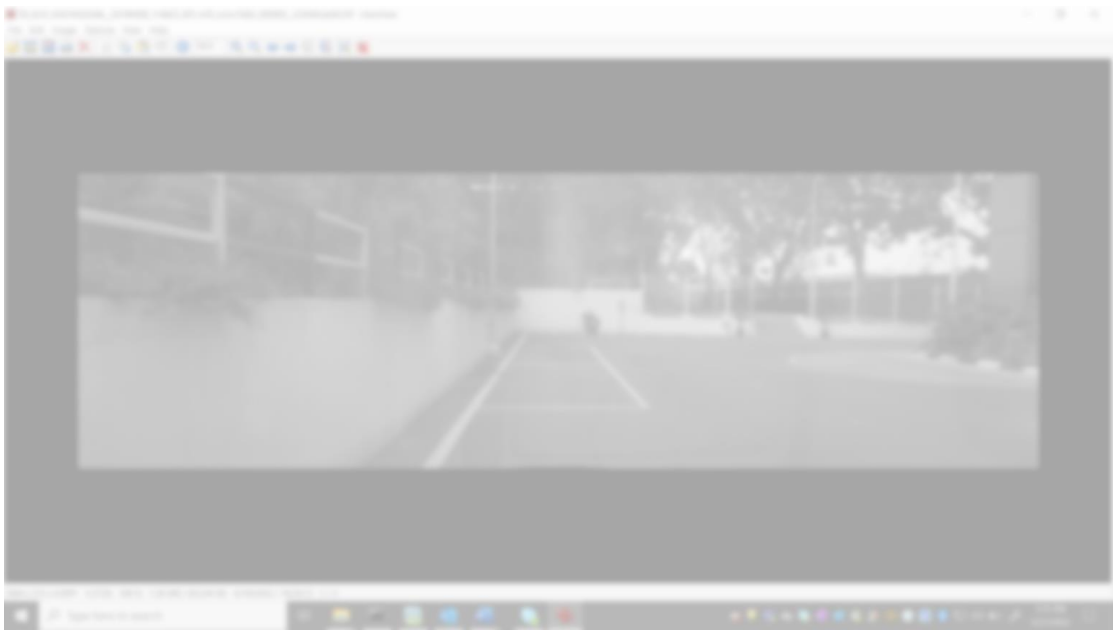


Fig-2: Image Classification

Codes used in image classification:

```
In [12]: import os
         import shutil

In [13]: path_folder = ""

In [14]: files = os.listdir(path_folder)

In [15]: path = "C:\\Users\\GJJ2KOR\\Desktop\\ValidationSeqequences"

In [16]: dstpath = "C:\\Users\\GJJ2KOR\\Desktop\\GOOD"

In [17]: with open("C:\\Users\\GJJ2KOR\\Desktop\\GOOD.txt") as f:
             line = f.readlines()
             for file in files:
                 if file == line:
                     src = path+file
                     dst = dstpath + file
                     shutil.copy(src,dst)

In [ ]:
```

```
In [1]: import os

In [2]: path = "C:\\15kTiff\\good"

In [3]: dir_list = os.listdir(path)

In [4]: print(dir_list)

['IN_XUV_KA01MQ3286__20190408_134404_008.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190408_134404_008.mf4_
color16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190408_134404_019.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_K
A01MQ3286__20190408_134404_019.mf4_color16Bit_001501__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_095545_001.mf4_color16Bit
_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_095545_001.mf4_color16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286_
_20190409_100130_003.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_100130_021.mf4_color16Bit_000002__G
3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_100130_021.mf4_color16Bit_001501__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_
103006_017.mf4_color16Bit_001501__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_111359_004.mf4_color16Bit_000002__G3NWide00.t
iff', 'IN_XUV_KA01MQ3286__20190409_111359_004.mf4_color16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_111359_14
1.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_111359_141.mf4_color16Bit_001501__G3NWide00.tiff', 'IN
_XUV_KA01MQ3286__20190409_111359_148.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_111359_148.mf4_colo
r16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_111359_158.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01M
Q3286__20190409_111359_158.mf4_color16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_074.mf4_color16Bit_000
002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_074.mf4_color16Bit_001500__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__201
90409_151618_075.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_075.mf4_color16Bit_001500__G3NWi
de00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_079.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_1516
18_079.mf4_color16Bit_001502__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_088.mf4_color16Bit_000002__G3NWide00.tif
f', 'IN_XUV_KA01MQ3286__20190409_151618_088.mf4_color16Bit_001501__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_096.m
f4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190409_151618_096.mf4_color16Bit_001500__G3NWide00.tiff', 'IN_XU
V_KA01MQ3286__20190409_151618_101.mf4_color16Bit_000002__G3NWide00.tiff', 'IN_XUV_KA01MQ3286__20190410_115558_080.mf4_color16

In [ ]:
```

```
In [15]: import os

In [17]: with open('[redacted]', 'r') as file1:
             with open('[redacted]', 'r') as file2:
                 same = set(file1).intersection(file2)

         same.discard('\n')

         with open('[redacted]', 'w') as file_out:
             for line in same:
                 file_out.write(line)

In [ ]:
```

In [1]:
```
import hashlib


output_file_path = "[redacted]
input_file_path = "[redacted]


completed_lines_hash = set()


output_file = open(output_file_path, "w")


for line in open(input_file_path, "r"):

  hashValue = hashlib.md5(line.rstrip().encode('utf-8')).hexdigest()

  if hashValue not in completed_lines_hash:
    output_file.write(line)
    completed_lines_hash.add(hashValue)

output_file.close()
```

In [ ]:

In [ ]:

Chapter 5
Conclusion

This is the project all about ADAS systems and how the autonomous vehicles are designed to react according to the situation and environment present around them with the use of the Video based sensors .

Bosch provided me an opportunity to work as a project trainee in Automatic Resource optimizer Project. This internship opportunity with the team has provided me with a golden opportunity to learn and explore my skills. I learnt extensively about Visual Studio, Python, MATLAB, Machine learning, usage of Bosch tools, etc. The programme has enhanced my critical thinking and analytical skills as well as improved my professional and ability to work in a multicultural environment. Working in Bosch as an intern was not only an honor and privilege but a lifelong experience that will forever shape my professional life. My Mentor, colleagues have inspired and supported me in my learning process and in achieving my tasks. I am very appreciative of this opportunity and forever grateful to Bosch for giving the opportunity to work as a project trainee.

REFERENCES

[1] AUTOSAR Development Partnership. Specification of RTE, www.autosar.org, 2010-09-22.

[2] MACHINE LEARNING. Introduction and its classification, www.tutorialspoint.com.

[3] ARTIFICIAL INTELIGENCE. Introduction, History and it's Development over decades, www.tutorialspoint.com.

[4] ADAS. Definition, Usages and its importance, www.synopsys.com/adas.

[5] MATLAB. Introduction and about PlotStr, www.bosch.com