MAIX DOCK BASED DRIVER DROWSINESS DETECTION SYSTEM USING CNN

A MAJOR PROJECT REPORT

Submitted by

HARSHITH BABU P [Reg No: RA1911043020027]
RAJAVARSHINI [Reg No: RA1911043020034]
ALISA NIKITA KERKETA [Reg No: RA1911043020048]

Under the guidance of
Dr. RAMANI. P,
Assistant Professor of ECE Department

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY

Ramapuram, Chennai – 600 089

MAY 2023

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed To be University, Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this major project report titled "**MAIX DOCK BASED DRIVER DROWSINESS DETECTION SYSTEM USING CNN"** is the bonafide work of "**HARSHITH BABU (RA1911043020027), RAJAVARSHINI (RA1911043020034), ALISA NIKITA KERKETA (RA1911043020049)"** who carried out the major project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation of the basis of which a degree or award was conferred on an earlier occasion on these or any other candidate.

SIGNATURE
Dr. P. Ramani
Assistant Professor
Department of Electronics and Communication Engineering
SRM Institute of Science and Technology
Ramapuram
Chennai-600089

SIGNATURE
Dr. P. Thirumurugan
Associate Professor and Head (I/C)
Department of Electronics and Computer Engineering
SRM Institute of Science and Technology
Ramapuram
Chennai-600089

SIGNATURE
Dr. Phani Kumar Polasi
Professor and Head
Department of Electronics and Communication Engineering
SRM Institute of Science and Technology
Ramapuram
Chennai-600089

Signature of the
Examiner – I

Signature of the
Examiner – II

ABSTRACT

This paper presents the implementation of a drowsiness-driving detection system using Maix Dock. Drowsy driving can be defined as a behavioral decline in driving skills. In this work, Deep learning has been used to classify drowsiness symptoms such as blinking and yawning. The sample images were used to train the yolo architecture. In this work, the Convolutional Neural Network (CNN) has been used to classify drowsiness symptoms such as blinking and yawning. A total of 1310 images were used to train the CNN architecture A 4 -layer convolution filter has been added as a layer in this yolo architecture. Adam's optimization algorithm was then used to train the yolo. A real time study on the effectiveness of this prototype was conducted on 10 individuals. In this paper, a novel approach towards real-time drowsiness detection based on deep learning which can be implemented on a low-cost embedded board and performs with high accuracy is proposed. The collected data can then be uploaded to a central server. The regions of interest used here are the facial features such as yawning, eye parameters that is eye closing. Other factors that can affect the rate of classification accuracy, such as camera distance from the driver and lighting factors, are also studied in this paper. If the driver is drunk, then the vehicle ignition will not start until the driver is not changed. In case the car is already in driving condition, then the system alerts the driver using a buzzer and pulse sensor also detecting the readings and alerts the driver, if risk is presence or not. It collects information using a variety of sensors and an onboard camera.

ACKNOWLEDGMENT

We would like to express our deep sense of gratitude to our beloved Chancellor to **Dr. T. R. PACHAMUTHU**, for providing us with the requisite infrastructure throughout the course. We take the opportunity to extend our hearty thanks to our Chairman, **Dr. R. SHIVAKUMAR** for his constant encouragement. We take the opportunity to extend our hearty thanks to our Dean, **Dr. M. MURALI KRISHNA** for his constant encouragement. We convey our sincere thanks to our Vice Principal (Academic), **Dr. G. PRABHAKARAN** for his interest and support throughout this project. We take the privilege to extend our hearty thanks to the Head of the Department - ECE **Dr**. **PHANI KUMAR POLASI** and Head of the Department (Incharge) **–** EKE **Dr. P. THIRUMURUGAN**, for their suggestions, support and encouragement towards the completion of the project with perfection. We express our heartfelt thanks to our faculty coordinator **Dr. P. Ramani,** Assistant Professor – Department of Electronics and Communication Engineering for his/her sustained encouragement, consecutive criticisms and Support throughout this project work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS

ACRONYM	ABBREVIATIONS	Page No
ESP	Espressif System	18
GPIO	General Purpose Input/Outpuut	19
GPS	Global Positioning System	29
UART	Universal Asynchronous receiver-transmitter	33
IOT	Internet Of Things	39
CNN	Central Neural Network	56

CHAPTER 1

Introduction:

One of the main sources of fatalities in humans is traffic mishaps. The fact that there are more cars on the road globally makes this worse. Long drives frequently make drivers drowsy and mentally exhausted. 2.3 to 2.5 percent of all deadly crashes nationally were reportedly the result of drowsy driving. [1] According to National Sleep Foundation research, 32% of motorists experience driving while fatigued on a monthly average.

A cognitive deterioration in driving abilities that is typically linked to long-distance driving [2] is what is known as "drowsy driving." Lack of sleep the night before traveling is typically the reason. It can occasionally also be brought on by other issues like unresolved sleep conditions, medicines, consuming alcohol, [3] or shift employment. Because of this issue, checking a driver is fundamental to blood alcohol content and notifying them if necessary.
In recent years, several novel low-cost, non-invasive technological advances have been developed to identify sleepy driving. It very well may be isolated [4] into two classifications given whether signal processing is used for images or not. The distinction between the two groups is made based on the sort of incoming data, which can be either bodily signs or pictures from the camera.

Eye tracking and blinking are a couple of warning indications that a video can detect as a sign of driving [5] fatigue. To identify these characteristics for sleep monitoring, numerous methods have been created. Head part examination, support vector machines, and brain organizations are the most promising methods.

Deep Learning [6] offers modern and effective methods for identifying sleepiness trends in drivers. Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) are two popular Deep Learning models utilized for image-based driving sleepiness detection systems. For physiological kinds of input data,

Recurrent Neural Network (RNN) is one of the Profound Learning plans habitually utilized for driving drowsiness [7] discovery systems.

1.1 Overview:

In this paper, different types of deep learning techniques were studied and analyzed. To train a CNN classifier model, raw images of people yawning or being seated in a car were taken with a wide range of images varying from each other, including the lighting part, and capable of potentially giving false values when used. The drowsiness symptoms should be identifiable. Some of the images have people yawing and drowsy whereas others do not such that we can segregate accordingly.

1.2 Problem Statement:

Globally, road mishaps and deaths are frequently caused by drowsy driving. Despite numerous efforts to increase public awareness about the dangers of driving while fatigued, drowsiness remains a significant risk on the road. Traditional methods of detecting drowsiness, such as monitoring driving behavior or asking drivers to self-report their level of fatigue, have limitations and are not always reliable.

To address this issue, there is a need for an automated system that can detect drowsiness in real-time, using advanced technologies such as head pose estimation, facial expression analysis, and machine learning algorithms. The system should be able to identify indications of sluggishness, like hanging eyelids or changes in facial expressions, and alert the driver before an accident occurs.

This project's objective is to create a sleepiness detection system that can accurately and reliably detect driver drowsiness in real-time, and potentially save lives by preventing accidents caused by drowsiness.

CHAPTER 2

LITERATURE SURVEY

2.1 A Real-time Driving Drowsiness Detection Algorithm With Individual Differences Consideration

Author name: Xiaolong Li

Abstract : The research work about driving drowsiness detection algorithm has great significance to improve traffic safety. Presently, there are many fruits and literature about driving drowsiness detection method. However, most of them are devoted to find a universal drowsiness detection method, while ignore the individual driver differences. This paper proposes a real-time driving drowsiness detection algorithm that considers the individual differences of driver. A deep cascaded convolutional neural network was constructed to detect the face region, which avoids the problem of poor accuracy caused by artificial feature extraction. Based on the Dlib toolkit, the landmarks of frontal driver facial in a frame are found. According to the eyes landmarks, a new parameter, called Eyes Aspect Ratio, is introduced to evaluate the drowsiness of driver in the current frame. Taking into account differences in size of driver's eyes, the proposed algorithm consists of two modules: offline training and online monitoring. In the first module, a unique fatigue state classifier, based on Support Vector Machines, was trained which taking the Eyes Aspect Ratio as input. Then, in the second module, the trained classifier is application to monitor the state of driver online. Because the fatigue driving state is gradually produced, a variable which calculated by number of drowsy frames per unit time is introduced to assess the drowsiness of driver. Through comparative experiments, we demonstrate this algorithm outperforms current driving drowsiness detection approaches in both accuracy and speed. In simulated driving applications, the proposed algorithm detects the drowsy state of driver quickly from 640 * 480 resolution images at over 20fps and 94.80% accuracy. The research result can serve intelligent transportation system, ensure driver safety and reduce the losses caused by drowsy driving.

2.2 Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks

Author name: ; Ye-Hoon Kim

Abstract : Driver's status is crucial because one of the main reasons for motor vehicular accidents is related to driver's inattention or drowsiness. Drowsiness detector on a car can reduce numerous accidents. Accidents occur because of a single moment of negligence, thus driver monitoring system which works in real-time is necessary. This detector should be deployable to an embedded device and perform at high accuracy. In this paper, a novel approach towards real-time drowsiness detection based on deep learning which can be implemented on a low cost embedded board and performs with a high accuracy is proposed. Main contribution of our paper is compression of heavy baseline model to a light weight model deployable to an embedded board. Moreover, minimized network structure was designed based on facial landmark input to recognize whether driver is drowsy or not. The proposed model achieved an accuracy of 89.5% on 3-class classification and speed of 14.9 frames per second (FPS) on Jetson TK1.

2.3 An improved fatigue detection system based on behavioral characteristics of driver

Author name: Kanishk Aman

Abstract : In recent years, the road accidents have increased significantly. One of the major reasons for these accidents, as reported is driver fatigue. Due to continuous and longtime driving, the driver gets exhausted and drowsy which may lead to an accident. Therefore, there is a need for a system to measure the fatigue level of driver and alert him when he/she feels drowsy to avoid accidents. The camera detects the driver's face and tracks its activity. From the driver's face, the system observes the alteration in its facial features and uses these features to observe the fatigue level. Facial features include eyes (fast blinking or heavy eyes) and mouth (yawn detection). Principle Component Analysis (PCA) is thus implemented to reduce the features while minimizing the amount of information lost. The parameters thus obtained are processed through Support Vector Classifier (SVC) for classifying the fatigue level. After that classifier output is sent to the alert unit.

2.4 Real Time Eye Monitoring System Using CNN for Drowsiness and Attentiveness System

Author name: Avinash Dubey

Abstract : As computer science is evolving it can be used for a wide variety of things, for the scope of our research we look to track eye movements for the drowsiness and attentive detection of a person. The movements of eyes or feeling drowsiness can have many adverse effects on an individual, and even can lead to catastrophic situations. Eye Monitoring System for Drowsiness and Attentiveness Detection is developed for monitoring the drowsiness and attentiveness of any individual. A lot of drivers suffer from drowsiness while driving the cars, trucks and auto rickshaws. The reason for drowsiness can differ from person to person but feeling drowsy while driving can lead to various safety issues for the driver and other people on the road. A driver feels drowsiness during driving due to lack of sleep, tiredness and due to long travel driving. Drowsiness can also be felt by students in online classes and students tend not to be interested in online classes as there is no faculty keeping an eye on them since they are not in the classrooms physically. As covid-19 impacted education institutions the only way to fill the gap between the students and faculty is online classes, hence online classes are the only solution left. Not only in India but throughout the world online classes are used. A system for checking the attentiveness of students is an active area of research. Our aim is proposing a solution to detect the drowsiness of the driver while he is driving and inattentiveness of students in online class. The main purpose of this research paper is to detect drowsiness and attentiveness using different algorithms and techniques and use the one with best accuracy in our system. The techniques used are Euclidean and CNN. The paper also provides a comparison between these techniques.

2.5 Face Detection Based on Viola-Jones Algorithm Applying Composite Features

Author name : Ming YANG

Abstract : iola-Jones' face detection algorithm was jointly proposed by Paul Viola and Michael Jones. Although it realized face real-time detection to some extent, its false detection rate is not low. Because the block features in the Viola-Jones algorithm can't handle purely rigid objects, such as chopsticks and cups, so if there are rigid objects in the face image, Viola-Jones' face detection algorithm is prone to generate false detection of faces. In this paper, we propose to apply the composite features based on Viola-Jones algorithm to improve the above problems, and prove the feasibility of this method through experiments.

CHAPTER 3

PROJECT DESCRIPTION

3.1Existing System:

The existing system monitoring the vehicles manual alcoholic checking are done by the manual process, traffic cameras are streaming the live video and finding the accident location manual process.

3.2Proposed System:

In this proposed system using Ai camera and controller using for detecting driver drowsiness and then using advanced functions for road safety in procedures in machine vision process, GPS and alcohol sensors are used to preventing the vehicle's driver. Then in built seat belt heart sensor helps to find driver health and esp8266 is connected with IoT.

3.3Advantages:

- IoT and Machine Learning can greatly assist people in preventing road accidents.
- These technologies will help develop a smart, efficient, and intelligent traffic system.
- One of the major reasons that contribute to fatal accidents is human behavior and their negligence to follow the traffic rules.
- IoT can play a proactive role in helping drivers adopt safety rules.
- With the help of IoT, the traffic management system gets updated with real-time data, thus increasing the efficiency of the safety system. With the help of Machine Learning and IoT, we can judge the driver's behavior

CHAPTER 4

SYSTEM ANALYSIS

4.1 ESP8266:

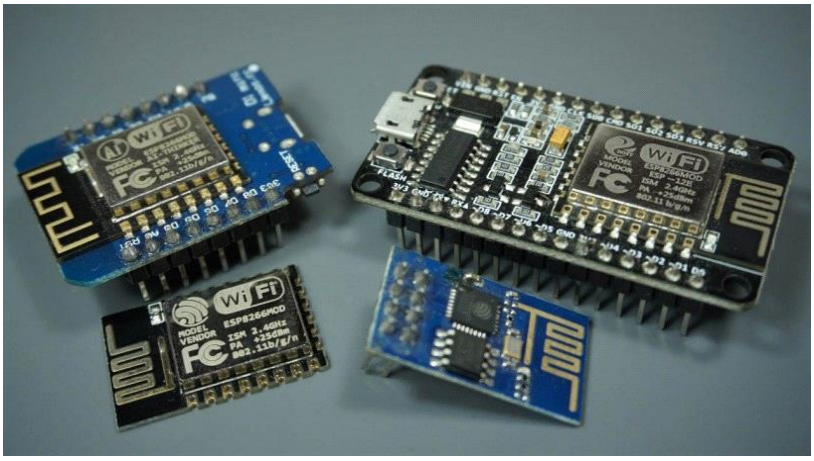


Figure 4.1: ESP8266

The ESP8266 12-E chip comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions. With this guide, you'll learn how to properly use the ESP8266 GPIOs and avoid hours of frustration by using the most suitable pins for your projects.

ESP8266 Peripherals:

- 17 GPIOs
- SPI
- I2C (implemented on software)
- I2S interfaces with DMA
- UART
- 10-bit ADC

Best Pins to Use – ESP8266

One important thing to notice about ESP8266 is that the GPIO number doesn't match the label on the board silkscreen. For example, D0 corresponds to GPIO16 and D1 corresponds to GPIO5.

The following table shows the correspondence between the labels on the silkscreen and the GPIO number as well as what pins are the best to use in your projects, and which ones you need to be cautious.

The pins highlighted in green are OK to use. The ones highlighted in yellow are OK to use, but you need to pay attention because they may have unexpected behavior mainly at boot. The pins highlighted in red are not recommended to use as inputs or outputs.

GPIOs connected to the Flash Chip

GPIO6 to GPIO11 are usually connected to the flash chip in ESP8266 boards. So, these pins are not recommended to use.

Pins used during Boot

The ESP8266 can be prevented from booting if some pins are pulled LOW or HIGH. The following list shows the state of the following pins on BOOT:

- GPIO16: pin is high at BOOT
- GPIO0: boot failure if pulled LOW
- GPIO2: pin is high on BOOT, boot failure if pulled LOW
- GPIO15: boot failure if pulled HIGH
- GPIO3: pin is high at BOOT
- GPIO1: pin is high at BOOT, boot failure if pulled LOW
- GPIO10: pin is high at BOOT
- GPIO9: pin is high at BOOT

Pins HIGH at Boot

There are certain pins that output a 3.3V signal when the ESP8266 boots. This may be problematic if you have relays or other peripherals connected to those GPIOs. The following GPIOs output a HIGH signal on boot:

- GPIO16
- GPIO3
- GPIO1
- GPIO10
- GPIO9

Additionally, the other GPIOs, except GPIO5 and GPIO4, can output a low-voltage signal at boot, which can be problematic if these are connected to transistors or relays. You can read this article that investigates the state and behavior of each GPIO on boot.

Analog Input

The ESP8266 only supports analog reading in one GPIO. That GPIO is called ADC0 and it is usually marked on the silkscreen as A0.
The maximum input voltage of the ADC0 pin is 0 to 1V if you're using the ESP8266 bare chip. If you're using a development board like the ESP8266 12-E NodeMCU kit, the voltage input range is 0 to 3.3V because these boards contain an internal voltage divider.

4.2 Maix Dock :

SIPEED Maix Dock is a development board compatible with Arduino based on our M1 module MaixDock integrates camera, TF card slot, user buttons, Maix Dock expansion interface, etc., users can use Maix Dock to easily build a face recognition access control system, and also reserve development and debugging interfaces, which can also be used As a powerful AI learning development board.

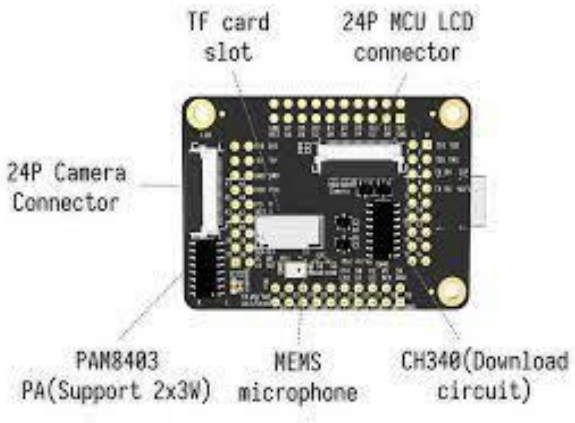


Figure 4.2: Maix Dock Components

Maix Dock development board uses the powerful M1 Al module as the core unit. The module has a built-in 64-bit dual-core processor chip and 8MB on-chip SRAM. It has outstanding performance in Al machine vision and hearing performance with a total computing power up to 1TOPS (FPU, Fast Fourier Transform Accelerator), which can easily implement machine vision/auditory algorithms for various application scenarios, and can also perform preprocessing for voice direction scanning and voice data output. In addition, which can be easily connected to the Internet with simple operations.

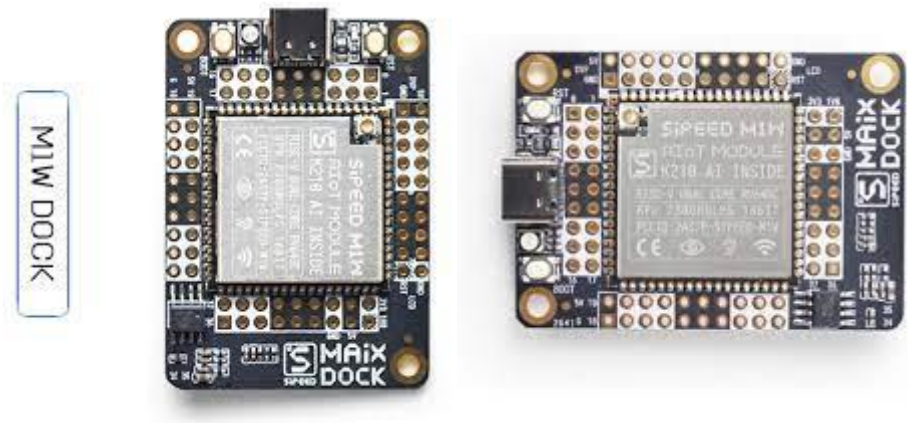


Figure 4.3: Maix Dock

K210 Chip features:

Core	RISC-V Dual Core 64bit, with FPU
Frequency	400MHz （Can be overclocked to 600MHz）
SRAM	built-in 8MB

Image Recognition	QVGA@60fps/VGA@30fps
Speech Recognition	Microphone array (8mics)
Network Model	□Support YOLO □ Mobilenetv2 □ facial recognition, etc.
Deep Learning Framework	Supports TensorFlow, Keras, Darknet, Caffe, other mainstream frameworks
Peripherals	FPIOA, UART, GPIO, SPI, I2C, I2S, TIMER
Video Processing	□ Neural Network Processor (KPU) □ FPU Meets IEEE754-2008 Standard □ Audio Processor (APU) □ Fast Fourier Transform Accelerator (FFT)

Board features:

Onboard	□ 1x RGB LED □ 1x MEMS Microphone □ 1x USB to serial port
Interfaces	□ USB Type-C interface □ 24PIN DVP camera interface □ MicroSD card slot □ Audio interface (supports 3 external speakers) □ Part of the IO pin header
Size	60*88mm
Supply Voltage	5.0V
Operating Temperature	-30℃ ~85C

Specification:

Master module	Sipeed MAIX-I AI +IoT module

Power input	USB Type-C DC-DC step-down circuit:support 6-12V input;Provide 5V 1.2A output
Micro SD card（TF card)slot	Support Self-elastic card holder
Support Self-elastic card holder	MSM261S4030H0 is an omnidirectional, Bottom-ported, I2S digital output MEMS Microphone. It has high performance and Reliability.
DVP Camera interface	24P 0.5mm FPC connector
LCD connector	8bit MCU LCD 24P 0.5mm FPC connector
Audio output	DAC+PA: TM8211:16 bit dynamic range;Low harmonic distortion NS4150:3W output power;Up to 90% efficiency;
ESP32 module	Support 2.4G 802.11.b/g/n 802.11 n (2.4 GHz) speeds up to 150 Mbps Bluetooth v4.2 full standard, including traditional Bluetooth (BR/EDR) and Bluetooth Low Energy (BLE)
Supply voltage of external power supply	4.8V ~ 5.2V
Supply current of external power supply	>600mA
Temperature rise	<30K
Range of working temperature	-30℃ ~ 85℃
Wireless Standard	802.11 b/g/n
Frequency Range	2400Mhz – 2483.5Mhz
TX Power(Conduction test)	802.11.b : +15dBm 802.11.g : +10dBm(54Mbps) 802.11.n : +10dBm (65Mbps)
Antenna Connector	IPEX 3.0×3.0mm
Wi-Fi mode	Station/SoftAP/SoftAP+Station

Shipment Weight	0.15 kg
Shipment Dimensions	12 × 8 × 3 cm

4.3 Micro camera:

The 0.3MP OV7670 Camera Module with High-Quality SCCB Connector is a low voltage CMOS image sensor; that provides the full functionality of a single-chip VGA(Video Graphics Array) camera and image processor in a small footprint package. The OV7670/OV7171 provides full-frame, sub-sampled or windowed 8-bit images in a wide range of formats; controlled through the **Serial Camera Control Bus (SCCB)** interface.

Figure 4.4: Micro Camera

This 0.3MP OV7670 Camera Module has an image array capable of operating at up to 30 **frames per second (fps)** in VGA; with complete user control over image quality, formatting and output data transfer. All required image processing functions, including exposure control, gamma, white balance; color saturation, hue control and more, are also programmable through the SCCB interface.

In addition, OV7670 CAMERA CHIPs use proprietary sensor technology to improve image quality by reducing; or eliminating common lighting/electrical sources of image contamination; such as fixed pattern noise (FPN), smearing, blooming, etc., to produce a clean, fully stable color image.

Features:

- High sensitivity for low-light operation.
- Low operating voltage for embedded portable apps.
- The standard SCCB interface is compatible with the I2C interface.
- Supports VGA, CIF, and resolutions lower than CIF for RGB (GRB 4:2:2, RGB565/555), YUV (4:2:2) and YCbCr (4:2:2) formats.
- VarioPixel® method for sub-sampling.
- Automatic image control functions includes : Automatic Exposure Control (AEC), Automatic Gain Control (AGC), Automatic White Balance (AWB), Automatic Band Filter (ABF), and Automatic Black-Level Calibration (ABLC).
- Image quality controls including color saturation, hue, gamma, sharpness (edge enhancement), and anti-blooming.
- ISP includes noise reduction and defect correction.
- Supports LED and flash strobe mode.
- Lens shading correction.
- Flicker (50/60 Hz) auto-detection.
- Saturation level auto adjust (UV adjust).
- Edge enhancement level auto-adjust.
- De-noise level auto-adjust.
- Sleep Power consumption: <20μA.
- Stable operation: 0 ° C to 50 ° C.
- Browse mode line by line.
- Electronic exposure 1 line to 510 lines.
- Output format: (8-bit) YUV/YCbCr4:2:2 RGB565/555/444 GRB4:2:2 Raw RGB Data.
- Material: plastic + metal.

4.4 Buzzer:

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from

audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

Figure 4.5: Buzzer Pin Configuration

The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.

Working Principle

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors.

Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So this continuous action will produce a sharp sound signal.

Specifications

The **specifications of the buzzer** include the following.

- Color is black
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA

Types of Buzzer

A buzzer is available in different types which include the following.

- Piezoelectric
- Electromagnetic
- Mechanical
- Electromechanical
- Magnetic

Piezoelectric

As the name suggests, the piezoelectric type uses the piezoelectric ceramic's piezoelectric effect & pulse current to make the metal plate vibrate & generate sound. This kind of buzzer is made with a resonance box, multi resonator, piezoelectric plate, housing, impedance matcher, etc. Some of the buzzers are also designed with LEDs.

The multi resonator of this mainly includes ICs and transistors. Once the supply is given to this resonator, it will oscillate and generates an audio signal with 1.5 to 2.kHz. The impedance matcher will force the piezoelectric plate to produce sound.

Electromagnetic

This type of buzzer is made with a magnet, solenoid coil, oscillator, housing, vibration diaphragm, and magnet. Once the power supply is given, the oscillator which produces

the audio signal current will supply throughout the solenoid coil to generate a magnetic field.

Sometimes, the vibration diaphragm will vibrate & generates sound under the magnet & solenoid coil interaction. The frequency range of this ranges from 2 kHz to 4kHz.

Mechanical

These types of buzzers are subtypes of electromagnetic, so the components used in this type are also similar. But the main difference is that the vibrating buzzer is placed on the outside instead of the inside.

Electromechanical

The designing of these types of buzzers can be done with a bare metal disc & an electromagnet. The working principle of this is similar to magnetic and electromagnetic. It generates sound throughout the disc movement & magnetism.

Magnetic

Like a piezo type, magnetic is also used to generate a sound but they are different due to core functionality. The magnetic type is more fixed as compared to the piezo type because they work through a magnetic field.

Magnetic buzzers utilize an electric charge instead of depending on piezo materials to generate a magnetic field, after that it permits another element of the buzzer to vibrate & generate sound.The applications of magnetic buzzers are similar to the piezo type in household devices, alarms such as watches, clocks & keyboards.

4.5 Gas Sensor

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. **Gas sensors** are very important part of such systems. Small like a nose, gas sensors spontaneously react to the gas present,

thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.

Gas sensors are available in wide specifications depending on the sensitivity levels, type of gas to be sensed, physical dimensions and numerous other factors. This Insight covers a **methane gas sensor** that can sense gases such as ammonia which might get produced from methane. When a gas interacts with this sensor, it is first ionized into its constituents and is then adsorbed by the sensing element. This adsorption creates a potential difference on the element which is conveyed to the processor unit through output pins in form of current. What is this sensing element? Is it kept in some chamber or is kept exposed? How does it get current and how it is taken out? Let's find out in this Insight!!!

The **gas sensor module** consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.

4.6 NEO-6M GPS module

Neo-6M is a top-rated Arduino comfortable GPS receiver with a powerful ceramics antenna. The connection and code are simple. There are enough resources and libraries available for the NEO-6M GPS module.

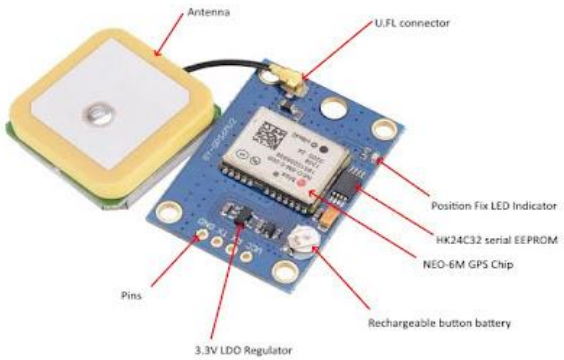


Figure 4.6: NEO-6M GPS module

The NEO-6M GPS module supports a baud rate from 4800bps to 230400bps with a default baud of 9600.

The **NEO-6M** is a **GPS** (Global Positioning System) module and is used for navigation. The module simply checks its location on earth and provides output data which is longitude and latitude of its position.It is from a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature (16 x 12.2 x 2.4 mm) package. The compact architecture, power and memory options make **NEO-6 modules** ideal for **battery operated mobile devices** with very strict cost and space constraints. Its Innovative design gives **NEO-6M** excellent navigation performance even in the most challenging environments.

NEO-6M GPS Module Pin Configuration

The module has four output pins and we will describe the function each pin of them below. The powering of module and communication interface is done through these four pins.

Pin Name	Description
VCC	Positive power pin
RX	UART receive pin
TX	UART transmit pin
GND	Ground

Features and Electrical Characteristics

- Standalone GPS receiver
- Anti-jamming technology
- UART Interface at the output pins (Can use SPI ,I2C and USB by soldering pins to the chip core)
- Under 1 second time-to-first-fix for hot and aided starts
- Receiver type: 50 Channels - GPS L1 frequency - SBAS (WAAS, EGNOS, MSAS, GAGAN)

- Time-To-First-fix: For Cold Start 32s, For Warm Start 23s, For Hot Start <1s
- Maximum navigation update rate: 5Hz
- Default baud rate: 9600bps
- EEPROM with battery backup
- Sensitivity: -160dBm
- Supply voltage: 3.6V
- Maximum DC current at any output: 10mA
- Operation limits: Gravity-4g, Altitude-50000m, Velocity-500m/s
- Operating temperature range: -40ºC TO 85°C

Overview of the NEO-6M GPS Module

This module is one of **popular GPS modules** in the market and is also cheap to buy. The location data provided by it is accurate enough to satisfy most applications. And for it to be included in smart phones and tablets design points out its efficiency. This module is famous among hobbyist and engineers altogether who want to work on applications involving navigation.

Getting this module to work is very easy. For the application circuit below we have connected the power to board and interfaced the output to the microcontroller UART to get it done.After circuitry, you need to set the baud rate of the controller matching the module, if it's not matched you will get error. With baud rate setting done you can read the serial data directly from the module. This data will be longitude and latitude values and the user can play with them as desired.

The raw values provided by the module are cumbersome to read directly and so a simple decimal calculation can be done in programming for getting easy to read values.

Applications

- GPS application
- Smart phone and tablets
- Navigation systems
- Drones
- Hobby projects

4.7 ARDUINO NANO:

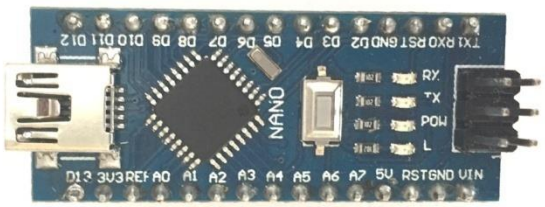


Figure 4.7: Arduino Nano

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

Memory

The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

- I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board).
The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data

when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

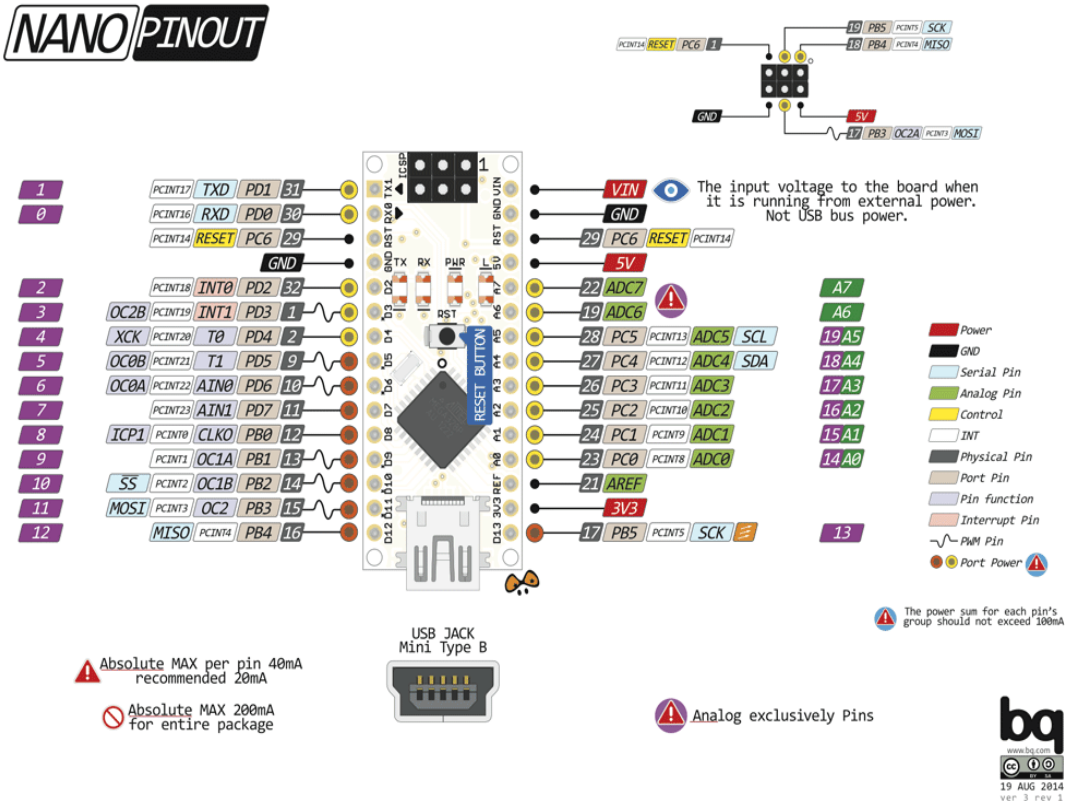


Figure 4.8: Nano PINOUT

4.8 Heart Rate and Pulse Oximeter (Max30102):

The MAX30102 is an integrated pulse oximeter and heart-rate monitor module. It includes internal LEDs, photo detectors, optical elements, and low-noise electronics with ambient light rejection. The MAX30102 provides a complete system solution to ease the design-in process for mobile and wearable devices. The MAX30102 operates on a single 1.8V power supply and a separate 3.3V power supply for the internal LEDs. Communication is through a standard I2C-compatible interface. The module can be shut down through software with zero standby current, allowing the power rails to remain powered at all times.

PIN DESCRIPTION:

This sensor has 8 pins. 4 pins are more useful:

VCC: Module power supply – 3 to 5 V

GND: Ground

SCL: I2C clock bus

SDA: I2C data bus

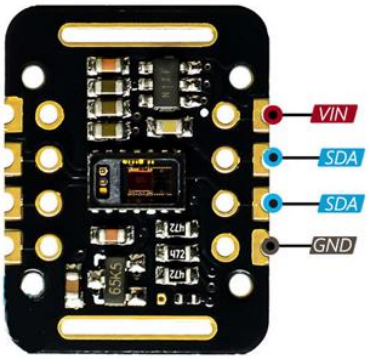


Figure 4.9: Heart Rate and Pulse Oximeter (Max30102)

BENEFITS AND FEATURES:

- Heart-Rate Monitor and Pulse Oximeter Sensor in LED Reflective Solution
- Tiny 5.6mm x 3.3mm x 1.55mm 14-Pin Optical Module
    - Integrated Cover Glass for Optimal, Robust Performance
- Ultra-Low Power Operation for Mobile Devices
    - Programmable Sample Rate and LED Current for Power Savings
    - Low-Power Heart-Rate Monitor (< 1mW)
    - Ultra-Low Shutdown Current (0.7µA, type)
- Fast Data Output Capability
    - High Sample Rates
- Robust Motion Artifact Resilience
    - High SNR
- -40°C to +85°C Operating Temperature Range

APPLICATIONS:

- Wearable Devices
- Fitness Assistant Devices
- Smartphones
- Tablets

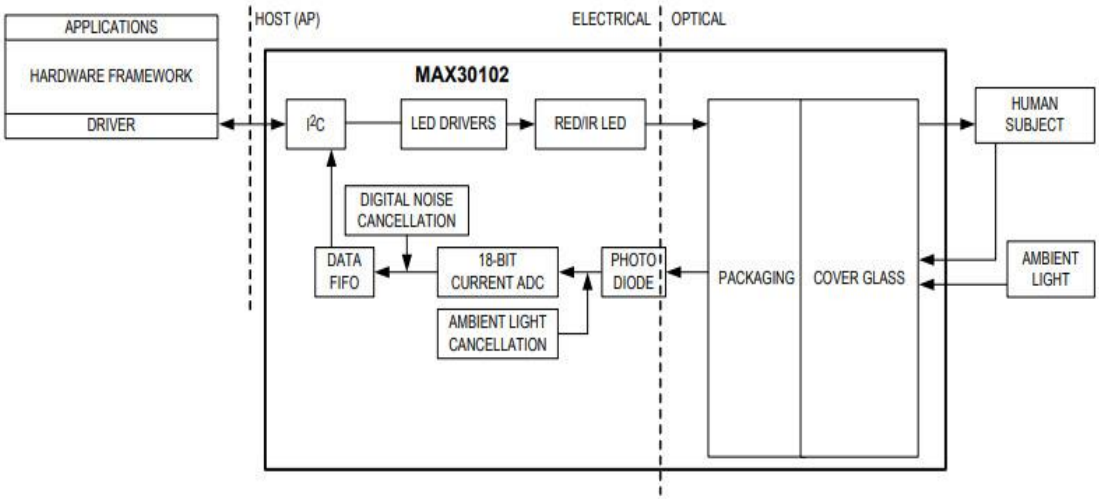


Fig 4.10: System Diagram

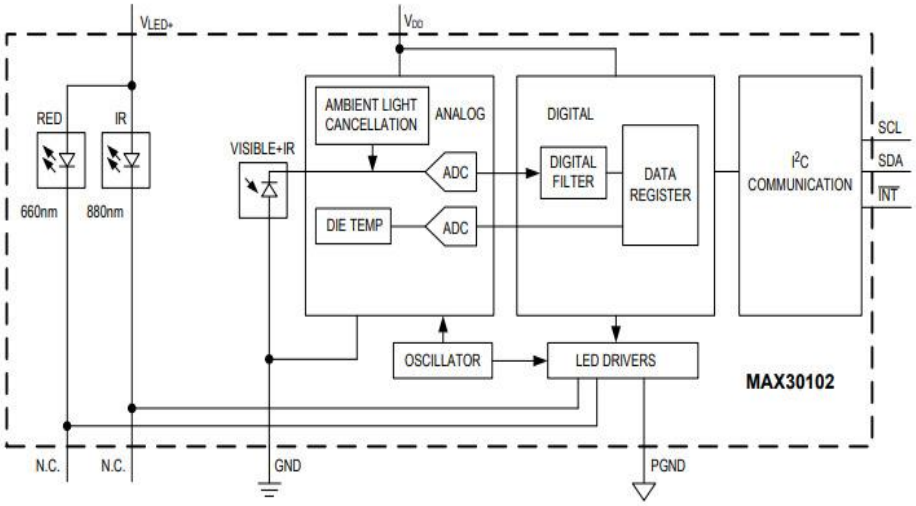


Fig 4.11: Functional Block Diagram

Detailed Description

The MAX30102 is a complete pulse oximeter and heart-rate sensor system solution module designed for the demanding requirements of wearable devices. The device maintains a very small solution size without sacrificing optical or electrical performance. Minimal external hardware components are required for integration into a wearable system. The MAX30102 is fully adjustable through software registers, and the digital output data can be stored in a 32-deep FIFO within the IC. The FIFO allows the MAX30102 to be connected to a microcontroller or processor on a shared bus, where the data is not being read continuously from the MAX30102's registers

SpO2 Subsystem

The SpO2 subsystem of the MAX30102 contains ambient light cancellation (ALC), a continuous-time sigma-delta ADC, and a proprietary discrete time filter. The ALC has an internal Track/Hold circuit to cancel ambient light and increase the effective dynamic range. The SpO2 ADC has programmable full-scale ranges from 2μA to 16μA. The ALC can cancel up to 200μA of ambient current.

The internal ADC is a continuous time oversampling sigma-delta converter with 18-bit resolution. The ADC sampling rate is 10.24MHz. The ADC output data rate can be programmed from 50sps (samples per second) to 3200sps.

Temperature Sensor

The MAX30102 has an on-chip temperature sensor for calibrating the temperature dependence of the SpO2 sub-system. The temperature sensor has an inherent resolution of 0.0625°C. The device output data is relatively insensitive to the wavelength of the IR LED, where the Red LED's wavelength is critical to correct interpretation of the data. A SpO2 algorithm used with the MAX30102 output signal can compensate for the associated SpO2 error with ambient temperature changes.

LED Driver

The MAX30102 integrates Red and IR LED drivers to modulate LED pulses for SpO 2 and HR measurements. The LED current can be programmed from 0 to 50mA with proper supply voltage. The LED pulse width can be programmed from 69µs to 411µs.

4.9 Interoperability in IoT:

- The Internet of Things (IoT) is an incredibly diverse space, encompassing a large variety of hardware form factors and software ecosystems unlike anything we have seen in technology. Smart watches, connected cameras, drones, thermostats, voice-enabled speakers, smart appliances and more—they all live together within the IoT.

- The diversity and innovation that excites many IoT fans is a big challenge not just for manufacturers and developers, but also (and most importantly) consumers. Which technology options should be used when designing or deploying IoT devices? How do they keep up with updated or new operating systems? What about new software and connectivity technologies coming up? Those are just some of today's challenges.

- Having a single, unified communication and software framework for the IoT seems like an ideal solution, but the diverse and fast-paced nature of the IoT makes this utopia a big challenge. Diversity in the IoT is not something to be solved, but an aspect that must be embraced and managed.

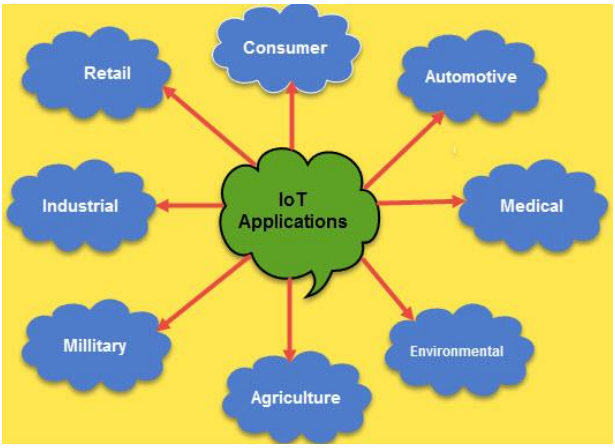


Fig 4.12: IoT Applications

Smart Home

The concept of Smart Home is brought up to save time, energy and money. With the introduction of Smart Homes, we would be able to switch on air conditioning before reaching home or switch off lights even after leaving home or unlock the doors to friends for temporary access even when you are not at home.

Smart cities

Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities. IoT will solve major problems faced by the people living in cities like pollution, traffic congestion and shortage of energy supplies etc. By installing sensors and using web applications, citizens can find free available parking slots across the city. Also, the sensors can detect meter tampering issues, general malfunctions and any installation issues in the electricity system.

Wearables

Wearable devices are installed with sensors and software's which collect data and information about the users. This data is later pre-processed to extract essential insights about user. These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

Healthcare

IoT in healthcare is aimed at empowering people to live healthier life and regular checkup by wearing connected devices. The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.

Challenges to IoT

In the era of IoT, everything is connected, linked up—much more than we see in and around us. IoT is certainly opening door to a lot of opportunities but also to many challenges.

Security Challenges

Security is a big issue with IoT devices. With billions of devices being connected together over Internet, how can people be sure that their information is secure? These security issues can be of the following kinds.

Data Encryption

IoT applications collect tons of data. Data retrieval and processing is integral part of the whole IoT environment. Most of this data is personal and needs to be protected through encryption.

Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private. Organizations and individuals use encryption to protect sensitive data stored on computers, servers and mobile devices like phones or tablets.

Data Authentication

After successful encryption of data chances of device itself being hacked still exist. If there is no way to establish the authenticity of the data being communicated to and from an IoT device, security is compromised.

For instance, say you built a temperature sensor for smart homes. Even though you encrypt the data it transfers is there is no way to authenticate the source of data then

anyone can make up fake data and send it to your sensor instructing it to cool the room even when its freezing or vice versa.

Side-channel Attacks

Encryption and authentication both in place still leave scope for side channel attacks. Such attacks focus less on the information and more on how that information is being presented. For instance if someone can access data like timing information, power consumption or electromagnetic leak, all of this information can be used for side channel attacks.

Privacy Challenges

Then we have the issue of privacy and data sharing. That is because these devices not only collect personal information like users' names and telephone numbers, but can also monitor user activities (e.g., when users are in their houses and what they had for lunch).

Connectivity Challenges—Billions of devices on a centralized server

One of the biggest challenges for IoT in the future is to connect large number of devices and massive amounts of data that all of these devices are going to produce. There will be need to find out a way to store, track, analyze and make sense of the vast amounts of data that will be generated.

Presently, we rely upon centralized, server/client model to authorize, authenticate, and connect several nodes present on the network. This model is sufficient for the number of IoT devices that are currently a part of the ecosystem. However, in the future, when hundreds of billions of devices will join the network, it will be difficult to manage all the data. Moreover, the capability of current cloud servers is so less that it can breakdown if it has to handle large amounts of information.

Compatibility and Longevity Challenges-Extra hardware and software

Different technologies like ZigBee, Z-Wave, WI-Fi, Bluetooth and, Bluetooth Low Energy (BTLE) are all battling to become the dominant transport mechanism between devices and hubs. This becomes a major source of problems when a lot of devices have to be connected; such dense connectivity requires the deployment of extra hardware and software.

Conversations about the IoT are taking place all over the world as we are trying to understand how this will impact our lives. We are also trying to understand what the many opportunities and challenges are going to be as more and more devices start to join the IoT. So, all that we can do is educate ourselves about what the IoT is and how it will be after some years.

Introduction to Firebase:

A brief post about what Firebase is all about, and it's new NoSQL Database—Cloud Firestore

With a variety of server-side technologies that are on the market today, developers have a tough job of deciding what kind of backend is most suitable for their app.

In this post, we will explore one of these choices that go by the name of Firebase , and all the tools and services that it provides.

4.10 Firebase:

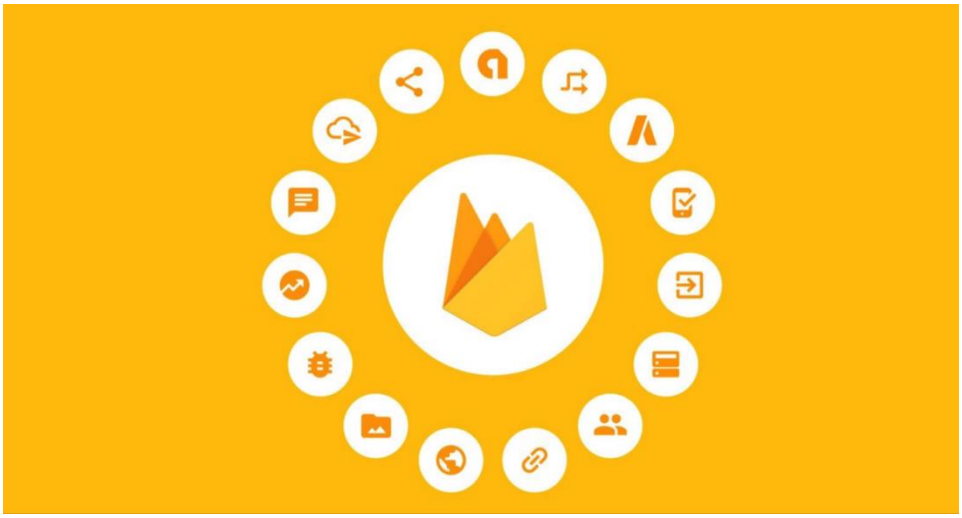


Fig 4.13: Firebase

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

A Brief History

Back in 2011, before Firebase was Firebase, it was a startup called Envolve. As Envolve, it provided developers with an API that enabled the integration of online chat functionality into their website.

What's interesting is that people used Envolve to pass application data that was more than just chat messages. Developers were using Envolve to sync application data such as a game state in real time across their users.

This led the founders of Envolve, James Tamplin and Andrew Lee, to separate the chat system and the real-time architecture. In April 2012, Firebase was created as a separate company that provided Backend-as-a-Service with real-time functionality.

After it was acquired by Google in 2014, Firebase rapidly evolved into the multifunctional behemoth of a mobile and web platform that it is today.

Authentication:

Fig 4.14: Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app.

Normally, it would take you months to set up your own authentication system. And even after that, you would need to keep a dedicated team to maintain that system. But if you use Firebase, you can set up the entire system in under 10 lines of code that will handle everything for you, including complex operations like account merging.

You can authenticate your app's users through the following methods:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter
- & more!

Using Firebase Authentication makes building secure authentication systems easier, while also improving the sign-in and on boarding experience for end users.

Firebase Authentication is built by the same people who created Google Sign-in, Smart Lock, and Chrome Password Manager.

Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.

You can send notification messages (2KB limit) and data messages (4KB limit).

Using FCM, you can easily target messages using predefined segments or create your own, using demographics and behavior. You can send messages to a group of devices that are subscribed to specific topics, or you can get as granular as a single device.

FCM can deliver messages instantly, or at a future time in the user's local time zone. You can send custom app data like setting priorities, sounds, and expiration dates, and also track custom conversion events.

The best thing about FCM is that there is hardly any coding involved! FCM is completely integrated with Firebase Analytics, giving you detailed engagement and conversion tracking.

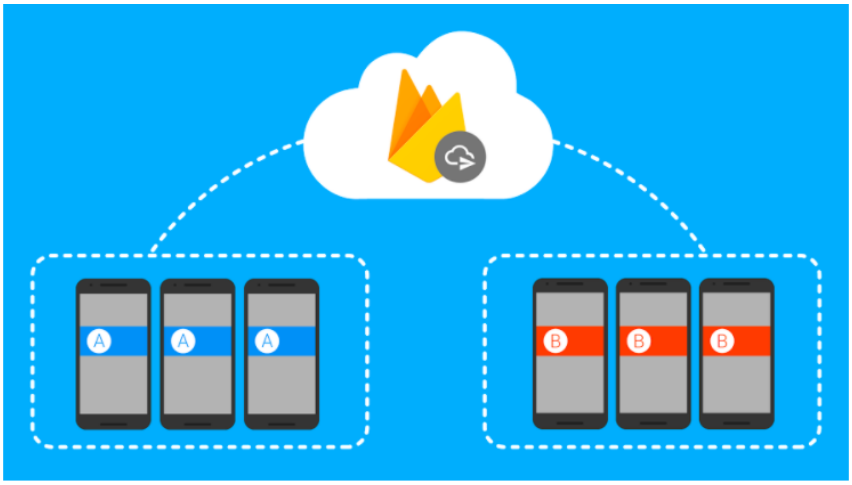


Fig 4.15: Firebase Cloud Messaging

You can also use A/B testing to try out different versions of your notification messages, and then select the one which performs best against your goals.

Firebase Database Query

Firebase has simplified the process of retrieving specific data from the database through queries. Queries are created by chaining together one or more filter methods.

Firebase has 4 ordering functions:

- Order By Key()
- Order By Child('child')
- Order By Value()
- Order By Priority()

Note that you will only receive data from a query if you have used the on() or once() method.

You can also use these advanced querying functions to further restrict data:

- Start At('value')
- End At('value')
- Equal To('child key')
- Limit To First(10)
- Limit To Last(10)

In SQL, the basics of querying involve two steps. First, you select the columns from your table. Here I am selecting the Users column. Next, you can apply a restriction to your query using the WHERE clause. From the below-given query, I will get a list of Users whose name is GeekyAnts.

You can also use the LIMIT clause, which will restrict the number of results that you will get back from your query.

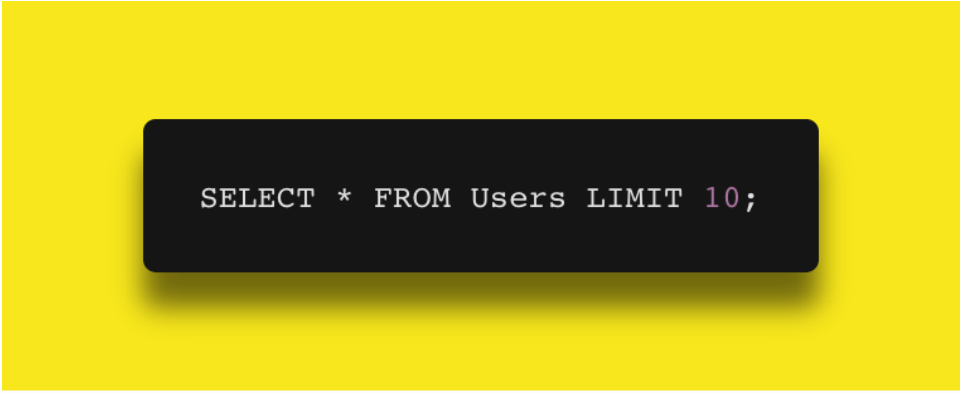


Fig 4.16: Firebase Storage

In Firebase, querying also involves two steps. First, you create a reference to the parent key and then you use an ordering function. Optionally, you can also append a querying function for a more advanced restricting.

How to Store Data? => Firebase Storage

Firebase Storage is a standalone solution for uploading user-generated content like images and videos from an iOS and Android device, as well as the Web.

Firebase Storage is designed specifically to scale your apps, provide security, and ensure network resiliency.

Firebase Storage uses a simple folder/file system to structure its data.

Firebase Test Labs

Firebase Test Labs provides a large number of mobile test devices to help you test your apps. Firebase Test Labs comes with 3 modes of testing:

- Instrumentation Test

These are tests that you written specifically to test your app, using frameworks like Espresso and UI Automaton 2.0

- Robo Test

This test is for people who just want to relax and let Firebase worry about tests. Firebase Test Labs can simulate user touch and see how each component of the app functions.

- Game Loop Test

Test Labs support game app testing. It comes with a beta support for using a “demo mode” where the game app runs while simulating the actions of the player. **Remote Config**

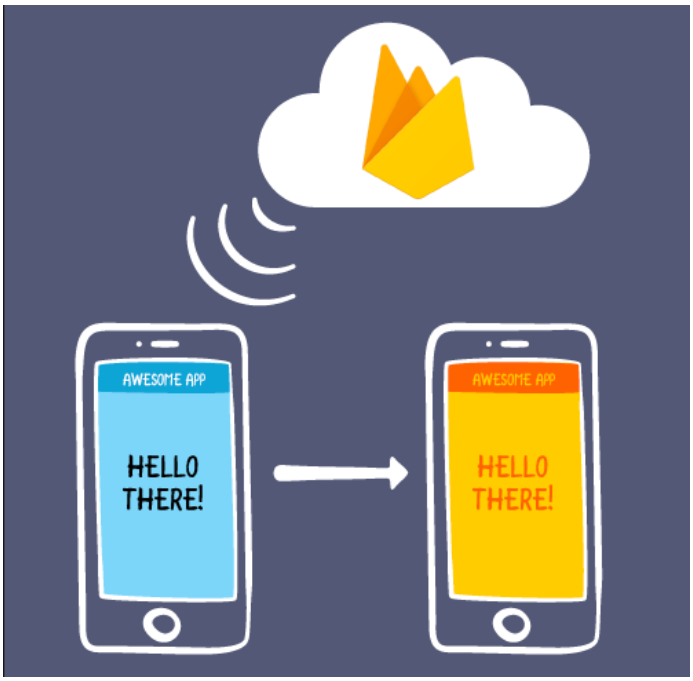


Fig 4.17: Remote Configuration

Remote Config essentially allows us to publish updates to our users immediately. Whether we wish to change the color scheme for a screen, the layout for a particular section in our app or show promotional/seasonal options—this is completely doable using the server side parameters without the need to publish a new version.

Remote Config gives us the power to:

- Quickly and easily update our applications without the need to publish a new build to the app/play store.
- Effortlessly set how a segment behaves or looks in our application based on the user/device that is using it.

Firebase App Indexing

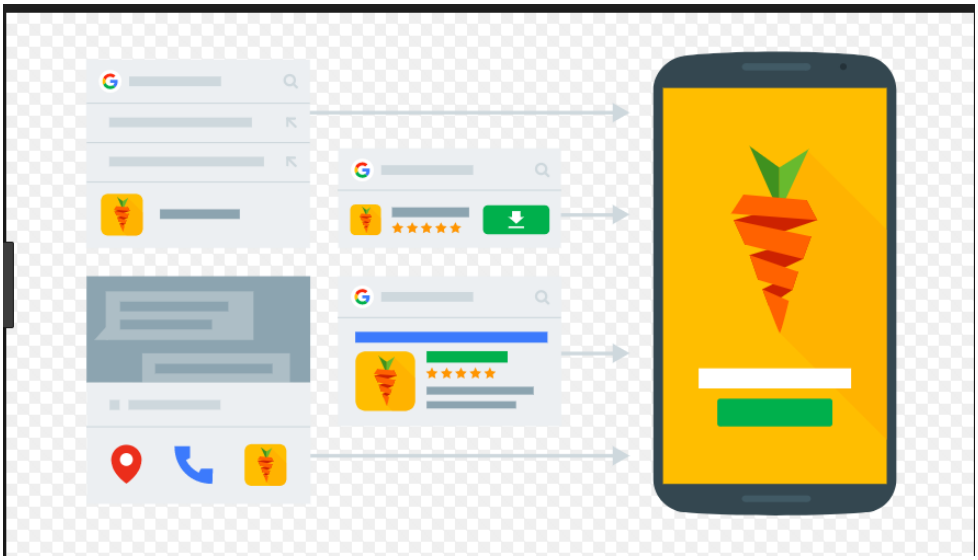


Fig 4.18: Firebase App Indexing

To get your app's content indexed by Google, use the same URLs in your app that you use on your website and verify that you own both your app and your website. Google Search crawls the links on your website and serves them in Search results. Then, users who've installed your app on their devices go directly to the content in your app when they click on a link.

Firebase Dynamic Links

Deep links are URLs that take you to content. Most web links are deep links.

Firebase can now modify deep links into Dynamic Links! Dynamic Links allow the user to directly come to a particular location in your app.

There are 3 fundamental uses for Dynamic Links

- Convert Mobile Web Users to Native App Users.
- Increase conversion for user-to-user sharing. By converting your app's users, when the app is shared with other users you can skip the generic message which is shown when a user downloads it from the store. Instead, you can show them personalized greeting message.

- Drive installs from the third party. You can use social media networks, email, and SMS can be used to increase your target audience. When users install the app, they can see the exact content of your campaigns.

4.11 Firestore:

Fig 4.19: Firestore

Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps—at a global scale.

Though this may sound like something similar to the Real-time Database, Firestore brings many new things to the platform that makes it into something completely different from Real-time Database.

Improved Querying and Data Structure

Where Real-time Database stores data in the form of a giant JSON tree, Cloud Firestore takes a much more structured approach. Firestore keeps its data inside objects called

documents. These documents consist of key-value pairs and can contain any kind of data, from strings to binary data to even objects that resemble JSON trees (Firestore calls it as maps). The documents, in turn, are grouped into collections.

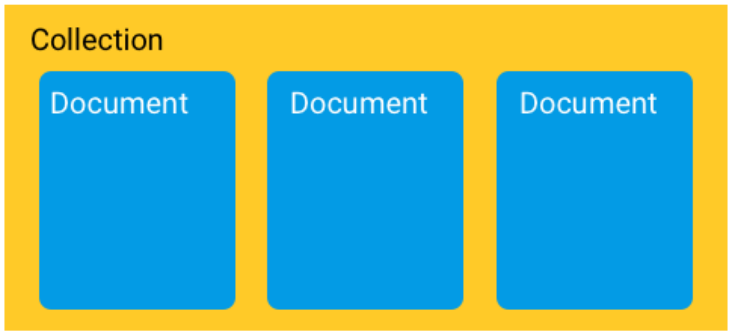


Fig 4.20: Collection in Firestore

Firestore database can consist of multiple collections that can contain documents pointing towards sub-collections. These sub-collections can again contain documents that point to other sub-collections, and so on.You can build hierarchies to store related data and easily retrieve any data that you need using queries.

All queries can scale with the size of your result set, so your app is ready to scale from its first day itself.Fire store's queries are shallow. By this, I mean to say that in Firestore, you can simply fetch any document that you want without having to fetch all of the data that is contained in any of its linked sub-collections.

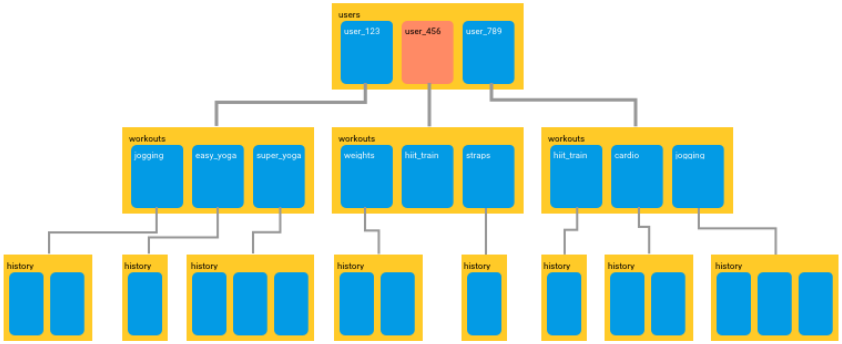


Fig 4.21: Sub-Collection in Firestore

Query with Firestore

Imagine that you have created a collection in Firestore that contains a list of Cities. So, before you can send out a query, you will have to store the database inside a variable.

Fig 4.22: Query in Firestore

Here, cities Ref is that variable that contains your collection of cities. Now, if you want to find a list of capital cities, you would write a query like this:

Here's another example of queries in Fire store. Say you want to see only 2 of cities from your database whose population is more than 100,000.

But Cloud Firestore can make querying even easier! In some cases, Cloud Fire store can automatically search your database across multiple fields. Fire store will guide you towards automatically building an index that will help Firestore to make querying extremely simple

Better Scalability

Though Firebase's Real-time Database is capable of scaling, things will start to get crazy when you app becomes really popular or if your database becomes really massive.

Cloud Fire store is based on Googles Cloud infrastructure. This allows it to scale much more easily and to a greater capacity than the Real-time Database.

Multi-Region Database

In Firestore, your data is automatically copied to various regions. So if one data center goes offline due to some unforeseen reason, you can be sure that your app's data is still safe somewhere else.

Fire store's multi-region database also provides strong consistency. Any changes to your data will be mirrored across every copy of your database.

Different Pricing Model

The Real-time Database charges its users based on the amount of data that you have stored in the database.

Cloud Fire store also charges you for the same, but the cost is significantly lower than that of Real-time Database and instead of basing the cost on the amount of data stored, Fire store's pricing is driven by the number of reads/writes that you perform.

Check out this blog post to know more about Cloud Fire store:

Cloud Fire store for Real-time Database Developers Hey, did you hear the big news? We just announced the beta release of Cloud Fire store the new database that lets you…firebase.googleblog.com.

\

CHAPTER 5

SYSTEM DESIGN

Block Diagram:

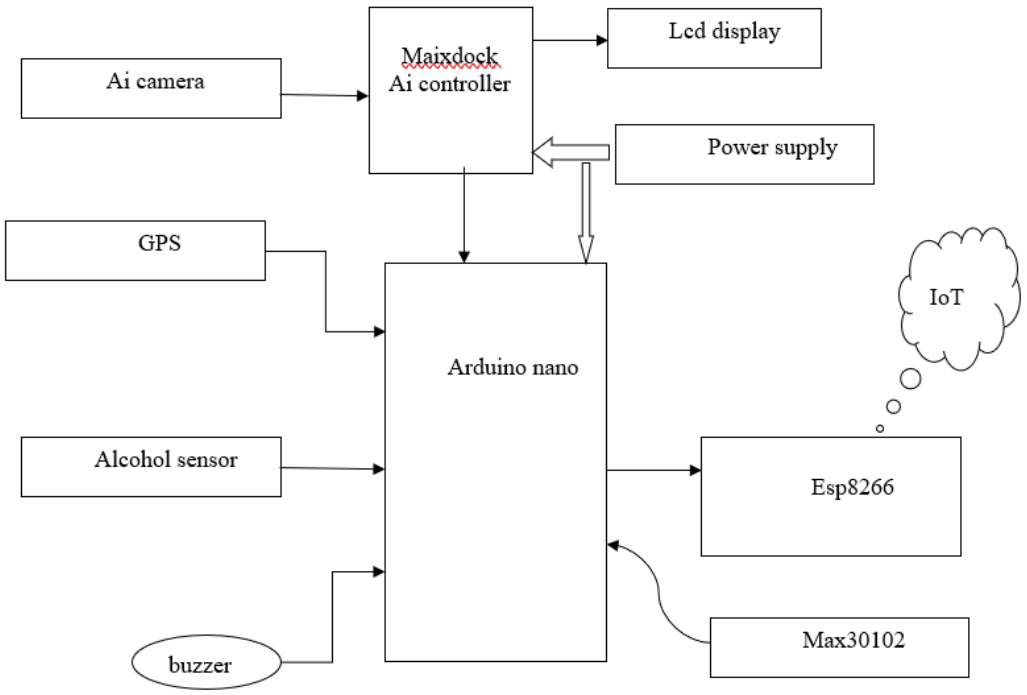


Fig 5.1: Block Diagram of Proposed System

To stop car accidents driving with an obstacle in the flow of vehicles, turning on the power supply, and making sure the maix dock is successfully attached are the first steps of this project. This will be known once the maix dock properly interprets and examines the camera and the incoming pictures. 480p quality footage can be transmitted from the webcam. The driver's visage will be recognized, and the condition notice will appear. Finding the eyes, mouth, and heartbeat comes next. The level of the driver's tiredness will be classified by the drowsiness detection engine once these features have been identified (what about alcohol?) Our drowsiness recognition system requires two conditions to be satisfied before

notifying the motorist. When the video detects that the motorist has his eyes closed for longer than two seconds is the first thing to watch out for. The alert will sound and "Drowsiness detected" will appear on the display."

When the camera notices the driver yawning, the second circumstance occurs. “Drowsiness detected” is displayed on the monitor, and an alarm is activated.

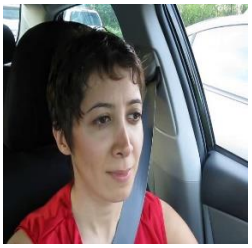
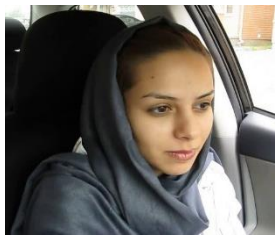
Use of an end-to-end neural network that predicts bounding frames and class odds simultaneously is suggested by the You Only Look Once (YOLO) theory. It is distinct from the strategy used by earlier object recognition algorithms, which used classifications as detectors. With a completely different strategy to object identification, YOLO outperformed other real-time object detection algorithms and produced cutting-edge findings. While algorithms like Faster RCNN perform recognition on individual regions after using the Region Proposal Network to identify potential regions of interest, YOLO performs all of its predictions in a single iteration with the aid of a single layer that is fully connected. For the same image, methods that make use of Region Proposal Networks go through multiple iterations, whereas YOLO only needs to go through one. While just go for it just requires one cycle, strategies that utilize district proposition networks require various rounds for a similar picture. Since the original introduction of YOLO in 2015, several new iterations of the same model have been suggested, each of which builds upon and enhances its precursor.

CHAPTER 6

RESULT AND DISCUSSION

The improved code for the driver drowsiness-detection system uses head pose estimation, facial expression analysis, machine learning, and CNN to detect drowsiness based on eye aspect ratio, blinking of eyes, and yawning

Table 6.1: Sample dataset for CNN

 **Eye closed**	 **Eye open**
 **Yawning**	 **Not yawning**
 **Empty seat**	 **Seat with driver**

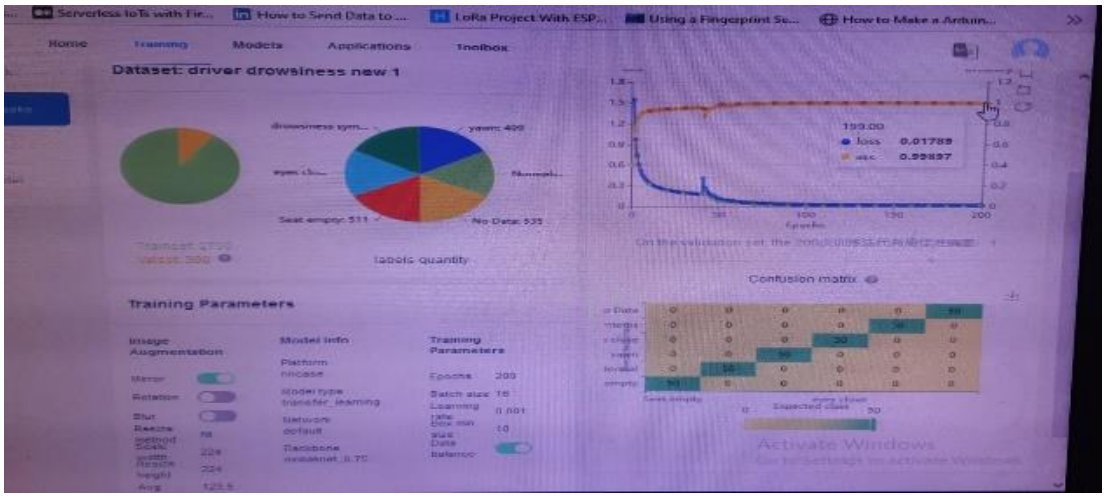


Fig 6.1: Performance and conclusion matrix for the proposed method

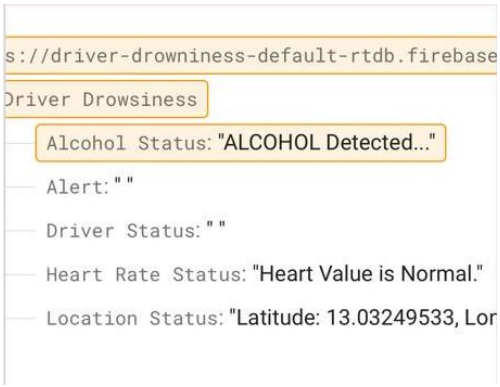


Fig 6.2: Detection of alcohol

The improved code for the driver drowsiness detection system uses head pose estimation, facial expression analysis, machine learning, and CNN to detect drowsiness based on eye aspect ratio, blinking of eyes, yawning, heart rate, location of the driver, and detecting of alcohol. Overall, the improved code for the driver drowsiness detection system is more accurate and robust as it combines multiple techniques, machine learning, and CNN to detect drowsiness in drivers. According to the performance and grid in fig5. displays the training loss and precision for the project's circumstances under evaluation. As well as using sample data set for CNN shown in Table 1.

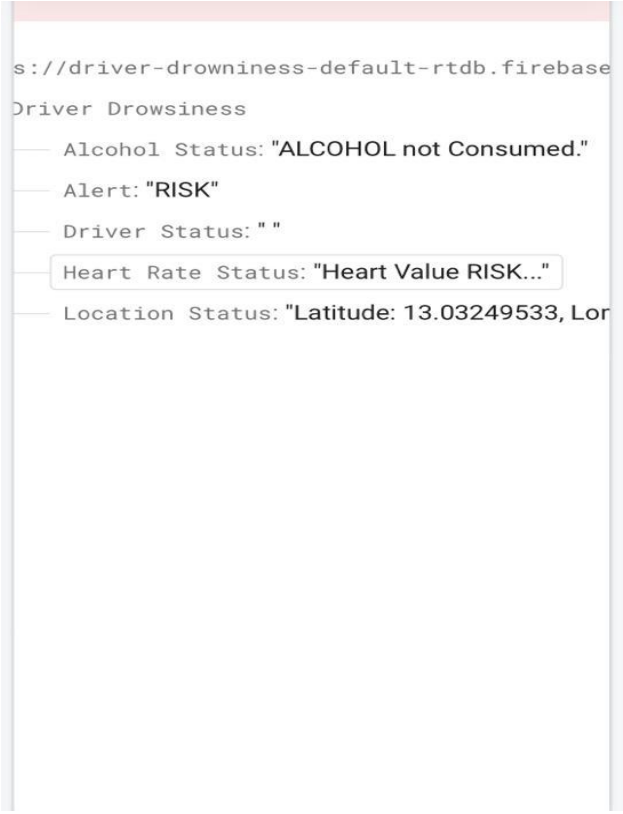


Fig 6.3: Detection of heart rate value

Table 6.2: Values and Epochy

Epochy	Value Accuracy	Value Loss
18/200	0.8945	0.3602
19/200	0.8955	0.3528
20/200	0.8993	0.3347

CONCLUSION

- The system is trained to identify the drowsiness symptoms of yawning and blinking using a convolutional neural network (CNN).
- This study used 3000 photos, including shots of eyes open, closed, yawning, or not gaping, vacant seats, and seats with drivers were used to teach CNN.
- Ten individuals tried the system prototype, which was developed in real-time.
- CNN's training results showed that our algorithm consistently anticipated a driver's sleepiness rate more than 80% of the time.
- In the future, other feature selection criteria such as aesthetics and appearance of the device may also be taken into account to assess its performance.
- Finally, after the launch of the device, surveys and user reviews will also be collected to improve the performance of the device.
- This system has more accuracy precison , user friendly and detection when compared to the exist system.
- This system has an accuracy of about 98%

FUTURE SCOPE

- Implementation of distance monitoring: The addition of a distance monitoring system to the Maix dock driver drowsiness detection system can improve its effectiveness. This can help detect the driver's behavior more accurately and provide feedback on safe driving distances.
- Integration with other safety systems: The Maix dock driver drowsiness detection system can be integrated with other safety systems in the vehicle, such as lane departure warning systems or collision warning systems. This integration can help create a more comprehensive safety system in the vehicle, suitable for different driving scenarios.
- Cloud integration: The Maix dock driver drowsiness detection system can be connected to a cloud server and be used to create driver fatigue profiles for multiple drivers. This can provide further insights into the behavior of different drivers and help them maintain optimum levels of alertness while driving.

APPENDICES

Appendix A: List of Equipment Used

Maix Dock
OV2640 Camera
Micropython IDE
Jupyter Notebook
Python 3.8
Keras

Appendix B: Dataset

The dataset used in this project was obtained from Kaggle. It consists of around 30,000 images of drivers with and without drowsiness. The images were collected using the camera installed in a moving vehicle. The dataset was divided into two classes- Drowsy and Non-Drowsy.

Appendix C: Data Pre-processing

Resizing the images to (64,64) and converting them into grayscale images
Normalizing the pixel values between 0 and 1
Splitting the dataset into training and validation sets (80-20 split)

Appendix D: Model Architecture

The model architecture used for this project was a Convolutional Neural Network. The architecture consisted of the following layers:
Input layer
Convolutional layer (32 filters with kernel size 3x3)
MaxPooling layer
Convolutional layer (64 filters with kernel size 3x3)

MaxPooling layer

Flatten layer

Fully connected layer (128 neurons)

Output layer (1 neuron)

Appendix E: Program Code

```
import sensor, image, lcd, time
import sensor,image,lcd,time
from machine import UART
from board import board_info
from fpioa_manager import fm
import KPU as kpu
import json
import gc, sys

fm.register(board_info.LED_R, fm.fpioa.UART1_TX, force=True)
uart_1 = UART(UART.UART1,9600,8,0,0, timeout=1000, read_buf_len=4096)

input_size = (224, 224)
labels = ['Seat empty', 'Normal', 'yawn', 'eyes close', 'drowsiness
symtoms']

def lcd_show_except(e):
    import uio
    err_str = uio.StringIO()
    sys.print_exception(e, err_str)
    err_str = err_str.getvalue()
    img = image.Image(size=input_size)
    img.draw_string(0, 10, err_str, scale=1, color=(0xff,0x00,0x00))
    lcd.display(img)

def main(labels = None, model_addr="/sd/m.kmodel",
sensor_window=input_size, lcd_rotation=2, sensor_hmirror=False,
sensor_vflip=False):
    sensor.reset()
    sensor.set_pixformat(sensor.RGB565)
    sensor.set_framesize(sensor.QVGA)
    sensor.set_windowing(sensor_window)
    sensor.set_hmirror(sensor_hmirror)
    sensor.set_vflip(sensor_vflip)
    sensor.run(1)

    lcd.init(type=1)
    lcd.rotation(lcd_rotation)
    lcd.clear(lcd.WHITE)

    if not labels:
        with open('labels.txt','r') as f:
            exec(f.read())
    if not labels:
        print("no labels.txt")
        img = image.Image(size=(320, 240))
        img.draw_string(90, 110, "no labels.txt", color=(255, 0, 0),
```

```
scale=2)
        lcd.display(img)
        return 1
    try:
        img = image.Image("startup.jpg")
        lcd.display(img)
    except Exception:
        img = image.Image(size=(320, 240))
        img.draw_string(90, 110, "loading model...", color=(255, 255,
255), scale=2)
        lcd.display(img)

    try:
        task = None
        task = kpu.load(model_addr)
        while(True):
            img = sensor.snapshot()
            t = time.ticks_ms()
            fmap = kpu.forward(task, img)
            t = time.ticks_ms() - t
            plist=fmap[:]
            pmax=max(plist)
            max_index=plist.index(pmax)
            img.draw_string(0,0, "%.2f : %s" %(pmax,
labels[max_index].strip()), scale=2, color=(255, 0, 0))
            img.draw_string(0, 200, "t:%dms" %(t), scale=2,
color=(255, 0, 0))
            lcd.display(img)
            # print(labels[max_index].strip())


            if labels[max_index].strip()=="Seat empty":
               uart_1.write('a')
               time.sleep_ms(1000)
               print(labels[max_index].strip())

            elif labels[max_index].strip()=="Normal":
                 uart_1.write('b')
                 time.sleep_ms(1000)
                 print(labels[max_index].strip())

            elif labels[max_index].strip()=="yawn":
                 uart_1.write('c')
                 time.sleep_ms(1000)
                 print(labels[max_index].strip())

            elif labels[max_index].strip()=="eyes close":
                 uart_1.write('d')
                 time.sleep_ms(1000)
                 print(labels[max_index].strip())

            elif labels[max_index].strip()=="drowsiness symtoms":
                 uart_1.write('e')
                 time.sleep_ms(1000)
                 print(labels[max_index].strip())

    except Exception as e:
        raise e
    finally:
        if not task is None:
            kpu.deinit(task)
```

```
if __name__ == "__main__":
    try:
         main(labels=labels, model_addr=0x300000)
        #main(labels=labels, model_addr="/sd/model-29141.kmodel")
    except Exception as e:
        sys.print_exception(e)
        lcd_show_except(e)
    finally:
        gc.collect()
```

Appendix F: Training Process

The model was trained using Adam optimizer with a learning rate of 0.001. The batch size used was 64 and the number of epochs was set to 50. The loss function used was binary cross-entropy and the metric used for evaluation was accuracy.

Appendix G: Results

The model achieved an accuracy of 95% on the validation set. The model was able to successfully detect drowsiness in drivers with high accuracy.

REFERENCES

[1] National Center for Statistics and Analysis. (2017, October). Drowsy Driving 2015 (Crash•Stats Brief Statistical Su

[2] F. You, X. Li, Y. Gong, H. Wang, and H. Li, "A Real-time Driving Drowsiness Detection Algorithm With Individual Differences Consideration," in IEEE Access, vol. 7, pp. 179396-179408, 2019, doi: 10.1109/ACCESS.2019.2958667mmary. Report No. DOT HS 812 446). Washington, DC: National Highway Traffic Safety Administration

[3] A. Dasgupta, D. Rahman, and A. Routray, "A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 11, pp. 4045-4054, Nov. 2019, doi: 10.1109/TITS.2018.2879609.

[4] R. Pai, A. Dubey, and N. Mangaonkar, "Real Time Eye Monitoring System Using CNN for Drowsiness and Attentiveness System," 2021 Asian Conference on Innovation in Technology (ASIANCON), 2021, pp. 1-4, doi: 10.1109/ASIANCON51346.2021.9544624.

[5] K. B. R. Teja and T. K. Kumar, "Real-Time Smart Drivers Drowsiness Detection Using DNN," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1026-1030, doi: 10.1109/ICOEI51242.2021.9452938.

[6] G. Geoffroy, L. Chaari, J. -Y. Tourneret and H. Wendt, "Drowsiness Detection Using Joint EEG-ECG Data With Deep Learning," 2021 29th European Signal Processing Conference (EUSIPCO), 2021, pp. 955-959, doi: 10.23919/EUSIPCO54536.2021.9616046.

[7] B. Reddy, Y. -H. Kim, S. Yun, C. Seo, and J. Jang, "Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 438-445, doi: 10.1109/CVPRW.2017.59.

[8] Y. Ying, S. Jing, and Z. Wei, "The Monitoring Method of Driver's Fatigue Based on Neural Network," 2007 International Conference on Mechatronics and Automation, 2007, pp. 3555-3559, doi: 10.1109/ICMA.2007.4304136.

[9] N. K. Gupta, A. K. Bari, S. Kumar, D. Garg, and K. Gupta, "Review Paper on Yawning Detection Prediction System for Driver Drowsiness," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1-6, doi: 10.1109/ICOEI51242.2021.9453008.

[10] R. Gupta, K. Aman, N. Shiva, and Y. Singh, "An improved fatigue detection system based on behavioral characteristics of the driver," 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE), 2017, pp. 227-230, doi: 10.1109/ICITE.2017.8056914.

[11] H. Singh, J. S. Bhatia, and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," India International Conference on Power Electronics 2010 (IICPE2010), 2011, pp. 1-6, doi: 10.1109/IICPE.2011.5728062.

[12] R. Wang, Y. Wang, and C. Luo, "EEG-Based Real-Time Drowsiness Detection Using Hilbert-Huang Transform," 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2015, pp. 195-198, doi: 10.1109/IHMSC.2015.56

[13] CNN Based Driver Drowsiness Detection System Using Emotion Analysis

H. Varun Chand* and J. Karthikeyan DOI:10.32604/iasc.2022.020008 School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, 632014, India 16 June 2021

[14] Li Zhenlong, Zhang Qingzhou, Zhao Xiaohua (2017) Performance analysis of K-nearest neighbor, support vector machine, and artificial neural network classifiers for driver drowsiness detection with different road geometries. Int J Distrib Sens Netw 13(9):1550147717733391

[15] Arefnezhad S et al (2019) Driver drowsiness detection based on steering wheel data applying adaptive neurofuzzy feature selection. Sensors 19(4):943

[16] Weng CH, Lai YH, Lai SH (2016) Driver drowsiness detection via a hierarchical temporal deep belief network. In: Asian conference on computer vision Springer, Cham, pp 117–133

[17] C. Hentschel, T. P. Wiradarma, and H. Sack, Fine tuning CNNS with scarce training data-adapting imagenet to art epoch classification, in Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, September 2016.

[18] Venkata Rami Reddy Chirra,Srinivasulu Reddy Uyyala& Venkata Krishna Kishore Kolli, Department of Computer Applications, National Institute of Technology, Tiruchirappalli 620015, India – "Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State

RAMAPURAM

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

FACULTY OF ENGINEERING & TECHNOLOGY

Certificate of Appreciation

This is to certify that Dr. / Mr. / Ms. **P. HARSHITH BABU** of **SRM Institute of Science and Technology, Rampuram,Chennai** for presenting paper titled **Maixdock Based Driver Drowsiness Detection System Using CNN** in the International Conference on **"Innovations and Technological Devolpment in Electronics, Computers and Communication (ITDECC-2023)"** organized by the **Department of Electronics and Communication Engineering, SRM Institute of Science and Technology**, Ramapuram, held on 20 & 21 of April 2023.

Convener(HoD - ECE)

Dean (E & T)