**CHAPTER 1**

**Introduction:**

COVID-19 (Corona Virus Disease 2019) is a pandemic, which has been spreading exponentially around the globe. Many countries adopted stay-at-home or lockdown policies to control its spreading. However, prolonged stay-at-home may cause worse effects like economical crises, unemployment, food scarcity, and mental health problems of individuals. This project presents smart consumer electronics solution to facilitate safe distancing between the peoples. This Wearable band can detect the person and intimate to the user and warn when the distance between individuals gets reduced than safe distance.

It consists of arm based micro-controller along with camera to detect the presence of human. It also used an ultrasonic sensor to detect the distance between the people from the user. On mounted buzzer will intimate to the user when distance reduced than safe range.

**Existing system:**

Existing system uses the contactless temperature sensing, mask detection and social distancing with indoor environment.

**Proposed system:**

This Proposed system detects the person using SSD model to alert the user. This model trained by CNN (convolutional neural network) algorithm.

**Modules:**

* **Image Capturing**
* **Feature Selection and Classification**
* **Sensor interface**

**Image Capturing:**

In this module we create a basic infrastructure to establish the proposed feature of image handling in python. Prepare the python environment to access video input device, in this case a camera.

**Feature Selection and Classification:**

Image annotation is a vital step of which the objective is to label the positions and classes of object spots in the images. For this stage, a convolution neural networks (CNN) algorithm that provides a frame selection function is developed in Python. The bounding box will be detected and labeled in frame where the people in the image frame.

**Sensor interface:**

Proximity sensor is interfaced with raspberry pi to measure the distance between person and the user. Sensor is triggered only after the person is detected by the system. User will notified by the buzzer sound.

**Block Diagram:**

Raspberry Pi Zero

Ultrasonic sensor

Raspberry pi camera

Buzzer

Push button

Power supply Unit

**Block Diagram Description:**

In this Block Diagram, Button, Ultrasonic sensor, Buzzer and camera connected with Raspberry Pi. When person identified in the input frame from camera, the controller triggers the ultrasonic sensor to measure the distance. When distance reaches its threshold buzzer will beep.

**Components requirement:**

**Hardware requirement:**

* Push button
* Camera
* Raspberry Pi
* Buzzer
* Ultrasonic sensor

**Software requirement:**

* Language : Embedded ‘C++’,Python
* Compiler : GCC Complier.
* OS : Linux

**CHAPTER 2**

**2.1 Power Supply:**

**Battery** is the primary power source for any electronics wireless gadget, be it a smartphone, laptop, watch or remote. Can you imagine the situation without these energy sources? We wouldn’t be able to build any wireless electronic device and have to rely on wired power source only, even electric cars and space missions would not be possible without Batteries. Today in this tutorial we discuss briefly about various types of batteries, their classification, terminology and specifications.

**What is Battery and why it is used?**

Let’s see the basic **difference between a battery and a cell**. Also let’s find out why we exactly need a battery and why can’t we use the Alternating power (i.e., AC power from the wall sockets) instead of DC power.

**Cell:**A cell is an energy source which can deliver only DC voltage and current which are in very small quantities. For example if we take cells that we use in watches or remote controls, it can give maximum of 1.5 – 3V.

**Battery:**The functionality of the battery is exactly same as that of a cell but a **battery is a pack of cells arranged is a series/parallel fashion** so that the voltage can be raised to desired levels. The best known example for a battery is a power bank which is used to charge up smart phones. If we ever see the inside of a power bank we can find set of batteries arranged serially/parallel based on the requirement. Batteries are arranged in series to increase the voltage and in parallel to increase the current.

Now **Why DC is preferred over AC**? In most of the portable electronics, AC can’t be stored where as DC can be stored without any difficulty. Even the losses due to AC power are more when compared to the DC power. That is the reason DC is preferred for portable electronic devices.

**Technical terms used while dealing with batteries**

We can’t just keep on using voltage and current alone to explain about a battery’s functionality, there are some unique terms that defines the characteristics of a battery like Watt-hour (mAh), C-rating, nominal voltage, charging voltage, charging current, discharging current, cut off voltage, shelf life, cycle life are the few terms used to define a batteries performance.

**Power capacity**:

It is the **energy stored in a battery** which is measured in **Watt-hour**

Watt-hour = V \* I \* hours {since voltage is kept constant, so it is measured in Ah/mAh}

**Power capability**:

It means the **amount of current that the battery can deliver**. It is also known as **C-rating.**

**Nominal voltage:**

While defining power capacity we have a unit called **Wh**which can be elaborated as V \* I \* hour but where did the voltage gone? As the voltage of the battery will be constant and will not be varied, it is considered as **nominal voltage** (fixed voltage). So since the voltage is fixed only Ampere and hour are considered as the unit (**Ah/mAh)**.

**Charging current:**

It is the **maximum current that can be applied to charge the battery** i.e., practically maximum of 1A/2A can be applied if a battery protecting circuit is in-built but still 500 mA is the best the range for charging the battery.

**Charging voltage:**

It is the maximum voltage that should be applied to the battery to efficiently charge a battery. Basically 4.2 V is the best/standard charging voltage. Though we apply 5 V to the battery it accepts only 4.2 V.

**Discharging current:**

It is the **current that can be drawn from the battery and is delivered to the load**.If the current drawn by the load is greater than the rated discharging current, the battery drains very fast which causes the battery heat up quickly which also causes the battery to explode. So it is cautious to determine the amount of current drawn by the load as well as the maximum discharging current a battery can withhold.

**Shelf life:**

There might be a situation where the batteries are kept idle/sealed especially in the stores/shops for a long period of time. *So***shelf life defines the time period a battery can be stay powered up and should be able to use it for a rated time period***.* Shelf life is mainly considered for non-rechargeable batteries because those are of use and throw. For rechargeable batteries even if the shelf time is less, we can still recharge it.

**Cut-off voltage:**

It is the **voltage at which the battery can be considered as fully discharged**, after which if we still try to discharge from it the battery gets damaged. So beyond the cut-off voltage the battery should be disconnected from the circuit and should be charged appropriately.

**Cycle life:**

Let’s consider a battery is fully charged and it is discharged to 80% of its actual capacity, then the battery is said to be completed one cycle. Likewise the **number of such cycles that a battery can charge and discharge defines the cycle life**. The more the cycle life the better will be the battery’s quality. But if a battery is discharged to say 40% of its actual capacity considering the battery is fully charged initially, it cannot be considered as a cycle life.

**Power density:**

It definespower capacity of battery for a given mass of volume.

**Types of Batteries**

Batteries are basically classified into 2 types:

* Non-rechargeable batteries (primary batteries)
* Rechargeable batteries (secondary batteries.

**Non-rechargeable Batteries**

These are basically considered as **primary batteries** because they can be used only once. These batteries cannot be recharged and used again. Let’s see about the regular, daily life primary batteries that we see.

* **Alkaline batteries:**It is basically constructed with the chemical composition of Zinc (Zn) and Manganese dioxide (MnO2), as the electrolyte used in it is potassium hydroxide which is purely an alkaline substance the battery is named as alkaline battery having he power density of 100 Wh/Kg.



**Advantages:**

1. Cycle life is more
2. More compatible and efficient for powering up portable devices.
3. Shelf life is more.
4. Small in size.
5. Highly efficient.
6. Low internal resistance so that discharge state in idle state is less.
7. Leakage is low.

**Disadvantages:**

1. Cost is a bit high. Except it everything is an advantage.

**Applications:**

It can used in torches, remotes, wall clocks, small portable gadgets etc.

* **Coin cell batteries:**The chemical composition of coil cell batteries is also alkaline in nature. Apart from alkaline composition, lithium and silver oxide chemicals will be used to manufacture these batteries which are more efficient in providing steady and stable voltage in such a small sizes.  It has Power density of 270 Wh/Kg.



**Advantages:**

1. Light in weight
2. Small in size
3. High density
4. Low cost
5. High nominal voltage (up to 3V)
6. Easy to get high voltages by arranging serially
7. Long shelf life

**Disadvantages:**

1. Needs a holder
2. Low current draw capability

**Applications:**

Used in watches, wall clocks, miniature electronic products etc.

**Rechargeable Batteries**

These are generally called as **secondary batteries** which can be recharged and can be reused. Though the cost is high, but they can be recharged and reused and can have a huge life span when properly used and safely charged.

**Lead-acid batteries**

It consists of lead-acid which is very cheap and seen mostly in cars and vehicles to power the lighting systems in it. These are more preferable in the products where the size/space and weight doesn’t matter. These comes with the nominal voltage starting 2V to24V and most commonly seen as 2V, 6V, 12V and 24V batteries. It has Power density of 7 Wh/Kg.



**Advantages:**

1. Cheap in cost
2. Easily rechargeable
3. High power output capability

**Disadvantages:**

1. Very heavy
2. Occupies much space
3. Power density is very low

**Applications:**

Used in cars, UPS (uninterrupted Power Supply), robotics, heavy machinery etc..

**Ni-Cd batteries**

These batteries are made of Nickel and Cadmium chemical composition. Though these are very rarely used, these are very cheap and their discharge rate is very low when compared to NiMH batteries. These are available in all standard sizes like AA, AAA, C and rectangular shapes. The nominal voltage is 1.2V, often connected together in a set of 3 which gives 3.6V. It has Power density of 60 Wh/Kg.



**Advantages:**

1. Cheap in cost
2. Easy to recharge
3. Can be used in all environments
4. Comes in all standard sizes

**Disadvantages:**

1. Lower power density
2. Contains toxic metal
3. Needs to be charged very frequently in order to avoid growth of crystals on the battery plate.

**Applications:**

Used in RC toys, cordless phones, solar lights and mostly in the applications where price is important.

**Ni-MH batteries**

The Nickel – Metal Hydride batteries are much preferable than Ni-Cad batteries because of their lower environmental impact. Its nominal voltage is 1.25 V which is greater than Ni-Cad batteries. It has less nominal voltage than alkaline batteries and they are good replacement due to its availability and less environmental impact. The power density of Ni-MH batteries is 100 Wh/Kg.



**Advantages:**

1. Available in all standard sizes.
2. High power density.
3. Easy to recharge.
4. A good alternative to alkaline which has almost all similarities and also it is rechargeable.

**Disadvantages:**

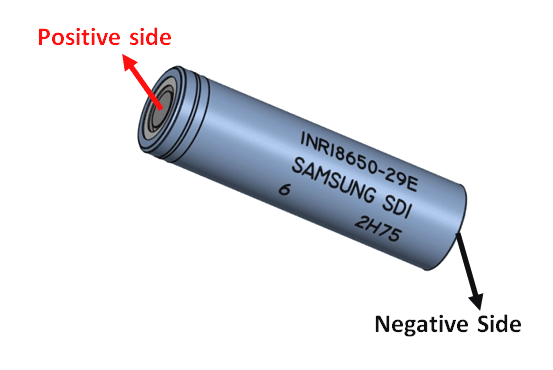
1. Self-discharge is very high.
2. Expensive than Ni-Cad batteries.

**Applications:**

Used in all applications similar to the alkaline and Ni-Cad batteries.

**Li-ion batteries**

These are made up of Lithium metal and are latest in rechargeable technology. As these are compact in size they can be used in most of the portable applications which need high power specifications. These are the best rechargeable batteries available. These have a nominal voltage of 3.7V (most commonly we have 3.6V and 7.2V) and have various ranges of power capacity (starting from 100s of mAh to 1000s of mAh). Even the C-rating ranges from 1C to 10C and Power density of Li-ion batteries is 126 Wh/Kg.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | **Lead Acid** | **Ni-Cd** | **Ni-MH** | **Alkaline** | ***Li-Ion*** | **Li-Polymer** |
| Cell Voltage | 2V | 1.2V | 1.2V | 1.5V | 3.6V | 3.7V |
| Cost | Low | Moderate | High | Very Low | Very High | Very High |
| Internal Resistance (IR) | Low | Very Low | Moderate | Varies | High | Low |
| Self Discharge (%/month) | 2% to 4% | 15% to 30% | 18% to 20% | 0.3% | 6% to 10% | 5% |
| Charge Cycle | 500 to 2000 | 500 to 1000 | 500 to 800 | Low | 1000 to 1200 | >1000 |
| Overcharge Tolerance | High | Medium | Low | Medium | Very Low | Very Low |
| Energy Density (Wh/kg) | 30 to 45 | 45 to 50 | 55 to 65 | 80 | 90 to 110 | 130 to 200 |
| Memory Effect | No | Yes | Yes | Yes | No | No |
| Maintenance | Very High | High | Low | Low | Low | Low |
| Safety | Highly Safe | Safe | Safe | Safe | Un-Safe | Un-Safe |

**18650 Cell Features and Technical Specifications**

* Nominal Voltage: 3.6V
* Nominal Capacity: 2,850 mAh
* Minimum Discharge Voltage: 3V
* Maximum Discharge current: 1C
* Charging Voltage: 4.2V (maximum)
* Charging current: 0.5C
* Charging Time: 3 hours (approx)
* Charging Method: CC and CV
* Cell Weight: 48g (approx)
* Cell Dimension: 18.4mm (dia) and 65mm (height)

### ****Where to use an 18650 Cell****

The 1**8650 Cell** is a **Li-ion type battery** which has found its application in many fields such as Portable electronics like torch lights, Electric Vehicles/Cars like Tesla and much more. The main reason for this battery being successful is its properties compared to its competitors. These properties include current carrying capability, voltage, cycle life, storage life, safety, and operating temperature and much more. Below table shows the comparison between popular batteries for key parameters.

**Lead Acid vs Ni-Cd vs Ni-MH vs Alkaline vs Li-ion vs Li-Polymer Batteries**

**Advantages:**

1. Very light in weight.
2. High C-rating.
3. Power density is very high.
4. Cell voltage is high.

**Disadvantages:**

1. These are a bit expensive.
2. If the terminals are short circuited the battery might explode.
3. Battery protection circuit is needed.

**Li-Po batteries**

These are also called as Lithium Ion polymer rechargeable batteries because it uses high conductivity polymer gel/polymers electrolyte instead of liquid electrolyte. These come under the Li-ion technology. These are a bit costly. But the battery is very highly protected when compared to the Li-ion batteries. It has Power density of 185 Wh/Kg.



**Advantages:**

1. These are highly protective compared to Li-ion batteries.
2. Very light in weight
3. Thin in structure when compared to Li-ion batteries.
4. Power density, nominal voltages are comparatively very high compared to Ni-Cad and Ni-MH batteries.

**Disadvantages:**

1. Expensive.
2. Might explode if wrongly connected.
3. Should not be bent or exposed to high temperature which may cause to explosion.

**Applications:**Can be used in all the portable devices which need rechargeable advantage like drones, robotics, RC toys etc.

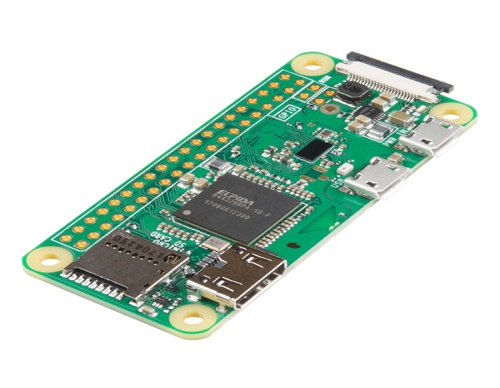
**2.2 RASPBERRY PI MICRO CONTROLLER**

Raspberry pi:

The Raspberry Pi is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation) to promote the teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries).[[5]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-6)[[7]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-7) The original model became far more popular than anticipated,[[8]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-1000x-8) selling outside its [target market](https://en.wikipedia.org/wiki/Target_market) for uses such as [robotics](https://en.wikipedia.org/wiki/Robotics). It does not include peripherals (such as [keyboards](https://en.wikipedia.org/wiki/Keyboard_(computing)) and [mice](https://en.wikipedia.org/wiki/Mouse_(computing))) and [cases](https://en.wikipedia.org/wiki/Computer_case). However, some accessories have been included in several official and unofficial bundles.[[8]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-1000x-8)

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with [Eben Upton](https://en.wikipedia.org/wiki/Eben_Upton" \t "Eben Upton) as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

According to the Raspberry Pi Foundation, more than 5 million Raspberry Pis were sold by February 2015, making it the best-selling [British computer](https://en.wikipedia.org/wiki/British_computer).[[9]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-9) By November 2016 they had sold 11 million units,[[10]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-11_million-10)[[11]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-11) and 12.5m by March 2017, making it the third best-selling "general purpose computer".[[12]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-12) In July 2017, sales reached nearly 15 million.[[13]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-13) In March 2018, sales reached 19 million



**Generations of released model:**

Several generations of Raspberry Pis have been released. All models feature a [Broadcom](https://en.wikipedia.org/wiki/Broadcom) [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC) with an integrated [ARM](https://en.wikipedia.org/wiki/ARM_architecture)-compatible [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU) and [on-chip graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit#_blank) (GPU).

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. [Secure Digital](https://en.wikipedia.org/wiki/Secure_Digital) (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. The boards have one to four USB ports. For video output, [HDMI](https://en.wikipedia.org/wiki/HDMI) and [composite video](https://en.wikipedia.org/wiki/Composite_video) are supported, with a standard 3.5 mm phono jack for audio output. Lower-level output is provided by a number of GPIO pins, which support common protocols like [I²C](https://en.wikipedia.org/wiki/I²C). The B-models have an [8P8C](https://en.wikipedia.org/wiki/8P8C) [Ethernet](https://en.wikipedia.org/wiki/Ethernet) port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth). Prices range from US$5 to $35.

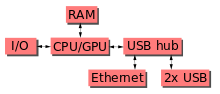
The first generation (Raspberry Pi 1 Model B) was released in February 2012, followed by the simpler and cheaper Model A. In 2014, the Foundation released a board with an improved design, Raspberry Pi 1 Model B+. These boards are approximately credit-card sized and represent the standard *mainline* form-factor. Improved A+ and B+ models were released a year later. A ["Compute Module"](https://en.wikipedia.org/wiki/Raspberry_Pi#Compute_module) was released in April 2014 for embedded applications. The Raspberry Pi 2, which added more RAM, was released in February 2015.

A Raspberry Pi Zero with smaller size and reduced [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) and [general-purpose input/output](https://en.wikipedia.org/wiki/General-purpose_input/output) (GPIO) capabilities was released in November 2015 for US$5. By 2017, it became the newest mainline Raspberry Pi. On 28 February 2017, the Raspberry Pi Zero W was launched, a version of the Zero with Wi-Fi and Bluetooth capabilities, for US$10.[[16]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-16)[[17]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-17) On 12 January 2018, the Raspberry Pi Zero WH was launched, the same version of the Zero W with pre-soldered GPIO headers.[[18]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-18)

Raspberry Pi 3 Model B was released in February 2016 with a 64 bit [quad core](https://en.wikipedia.org/wiki/Multi-core_processor) processor, on-board [WiFi](https://en.wikipedia.org/wiki/Wi-Fi" \t "Wi-Fi), [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) and USB boot capabilities.[[19]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-19) On [Pi Day](https://en.wikipedia.org/wiki/Pi_Day) 2018 model 3B+ appeared with a faster 1.4 GHz processor and a three times faster network based on [gigabit Ethernet](https://en.wikipedia.org/wiki/Gigabit_Ethernet) (300 Mbit / s) or 2.4 / 5 GHz dual-band [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) (100 Mbit / s).[[1]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-RapsberryPi3B+Release-1) Other options are: [Power over Ethernet](https://en.wikipedia.org/wiki/Power_over_Ethernet) (PoE), [USB boot](https://en.wikipedia.org/wiki/Universal_Serial_Bus) and [network boot](https://en.wikipedia.org/wiki/DHCP) (an [SD card](https://en.wikipedia.org/wiki/SD_card) is no longer required). This allows the use of the Pi in hard-to-reach places (possibly without electricity).

**Hardware:**

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.



This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the [Ethernet](https://en.wikipedia.org/wiki/Ethernet) and [USB](https://en.wikipedia.org/wiki/USB) hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

### Processor:



The Raspberry Pi 2B uses a 32-bit 900 MHz quad-core [ARM Cortex-A7](https://en.wikipedia.org/wiki/ARM_Cortex-A7)processor.

The [Broadcom](https://en.wikipedia.org/wiki/Broadcom) BCM2835 SoC used in the first generation Raspberry Pi[[20]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-Broadcom-BCM2835-Website-20) includes a 700 [MHz](https://en.wikipedia.org/wiki/Hertz) [ARM11](https://en.wikipedia.org/wiki/ARM11)76JZF-S processor, [VideoCore](https://en.wikipedia.org/wiki/VideoCore" \t "VideoCore) IV [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit)(GPU),[[21]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-grandmax_brose_2012-21) and RAM. It has a level 1 (L1) [cache](https://en.wikipedia.org/wiki/CPU_cache) of 16 [KB](https://en.wikipedia.org/wiki/Kibibyte) and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is [stacked](https://en.wikipedia.org/wiki/Package_on_package" \t "Package on package)underneath the RAM chip, so only its edge is visible. The 1176JZ(F)-S is the same CPU used in the [original iPhone](https://en.wikipedia.org/wiki/Original_iPhone),[[22]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-22) although at a higher clock rate, and mated with a much faster GPU.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core [ARM Cortex-A7](https://en.wikipedia.org/wiki/ARM_Cortex-A7) processor, with 256 KB shared L2 cache.[[23]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-2-B-Announcement-23) The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core [ARM Cortex-A53](https://en.wikipedia.org/wiki/ARM_Cortex-A53) processor,[[24]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-24) the same SoC which is used on the Raspberry Pi 3, but underclocked (by default) to the same 900 MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production (as of late 2016).

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.[[1]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-RapsberryPi3B+Release-1)

#### *Performance:*

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 [GFLOPS](https://en.wikipedia.org/wiki/FLOPS).[[25]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-eLinux-perf-25)[[26]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-hackaday-raspi-26) On the [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) level the performance is similar to a 300 MHz [Pentium II](https://en.wikipedia.org/wiki/Pentium_II) of 1997–99. The GPU provides 1 [Gpixel](https://en.wikipedia.org/wiki/Gpixel" \t "Gpixel)/s or 1.5 [Gtexel](https://en.wikipedia.org/wiki/Texel_(graphics)" \t "Texel (graphics))/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the [Xbox](https://en.wikipedia.org/wiki/Xbox_(console)) of 2001.

Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original.[[23]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-2-B-Announcement-23) In parallelised benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+.[[27]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-27)

The Raspberry Pi 3, with a quad-core [ARM Cortex-A53](https://en.wikipedia.org/wiki/ARM_Cortex-A53) processor, is described as having ten times the performance of a Raspberry Pi 1.[[28]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-:0-28) This was suggested to be highly dependent upon task [threading](https://en.wikipedia.org/wiki/Thread_(computing)) and [instruction set](https://en.wikipedia.org/wiki/Instruction_set) use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelised tasks.[[29]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-29)

#### *Overclocking:*

Most Raspberry Pi chips could be [overclocked](https://en.wikipedia.org/wiki/Overclocking) to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the [Raspbian](https://en.wikipedia.org/wiki/Raspberry_Pi#Software) [Linux distro](https://en.wikipedia.org/wiki/Linux_distro) the overclocking options on [boot](https://en.wikipedia.org/wiki/Booting) can be done by a software command running "sudo raspi-config" without voiding the warranty.[[30]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-Turbo_mode-30) In those cases the Pi automatically shuts the overclocking down if the chip temperature reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized [heat sink](https://en.wikipedia.org/wiki/Heat_sink) is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five overclock ("turbo") presets that when used, attempt to maximise the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip and the CPU load, and dynamically adjusting clock speeds and the core voltage. When the demand is low on the CPU or it is running too hot the performance is throttled, but if the CPU has much to do and the chip's temperature is acceptable, performance is temporarily increased with clock speeds of up to 1 GHz, depending on the individual board and on which of the turbo settings is used.

***The seven overclock presets are:***

* none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 [overvolting](https://en.wikipedia.org/wiki/Overvolting)
* modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolting,
* medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 overvolting,
* high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 overvolting,
* turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 overvolting,
* Pi 2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 overvolting,
* Pi 3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 overvolting. In system information the CPU speed will appear as 1200 MHz. When idling, speed lowers to 600 MHz.[[30]](https://en.wikipedia.org/wiki/Raspberry_Pi" \l "cite_note-Turbo_mode-30)[[31]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-turbosgithub-31)

In the highest (*turbo*) preset the SDRAM clock was originally 500 MHz, but this was later changed to 600 MHz because 500 MHz sometimes causes SD card corruption. Simultaneously in *high* mode the core clock speed was lowered from 450 to 250 MHz, and in *medium* mode from 333 to 250 MHz. The Raspberry Pi Zero runs at 1 GHz.

The CPU on the first and second generation Raspberry Pi board did not require cooling, such as a heat sink or [fan](https://en.wikipedia.org/wiki/Computer_fan), even when overclocked, but the Raspberry Pi 3 may generate more heat when overclocked.[[32]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-32)

### *RAM:*

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU.[[33]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-33) On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p [framebuffer](https://en.wikipedia.org/wiki/Framebuffer), and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC).[[34]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-34) Comparatively the [Nokia 701](https://en.wikipedia.org/wiki/Nokia_701) uses 128 MB for the Broadcom VideoCore IV.[[35]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-35)

For the later Model B with 512 MB RAM new standard memory split files (arm256\_start.elf, arm384\_start.elf, arm496\_start.elf) were initially released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM) respectively. But a week or so later the RPF released a new version of start.elf that could read a new entry in config.txt (gpu\_mem=*xx*) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 MB and 512 MB Raspberry Pis.[[36]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-36)

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM.[[37]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-37)[[38]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-38) The Raspberry Pi Zero and Zero W have 512 MB of RAM.

### Networking:

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip.[[39]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-SMSC-LAN9514-specs-39) The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi [802.11n](https://en.wikipedia.org/wiki/IEEE_802.11n-2009) (150 Mbit/s) and [Bluetooth 4.1](https://en.wikipedia.org/wiki/Bluetooth_4.1) (24 Mbit/s) based on the Broadcom BCM43438 [FullMAC](https://en.wikipedia.org/wiki/Wireless_network_interface_controller" \l "_blank) chip with no official support for [monitor mode](https://en.wikipedia.org/wiki/Monitor_mode) but implemented through unofficial firmware patching[[40]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-40) and the Pi 3 also has a 10/100 Mbit/s Ethernet port. The Raspberry Pi 3B+ features dual-band [IEEE 802.11b/g/n/ac WiFi](https://en.wikipedia.org/wiki/IEEE_802.11), [Bluetooth 4.2](https://en.wikipedia.org/wiki/Bluetooth_4.2), and [Gigabit Ethernet](https://en.wikipedia.org/wiki/1000BASE-T) (limited to approximately 300 Mbit/s by the [USB 2.0](https://en.wikipedia.org/wiki/USB_2.0) bus between it and the SoC).

### Peripherals:



The Model 2B boards incorporate four USB ports for connecting peripherals.

The Raspberry Pi may be operated with any generic [USB computer keyboard](https://en.wikipedia.org/wiki/USB_computer_keyboard) and [mouse](https://en.wikipedia.org/wiki/Mouse_(computing)).[[41]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-VerifiedPeripheralList-41) It may also be used with USB storage, USB to MIDI converters, and *virtually* any other device/component with USB capabilities.

Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.[[42]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-42)

### Video:



The early Raspberry Pi 1 Model A, with an HDMI port and a standard RCA composite video port for older displays

The video controller can generate standard modern TV resolutions, such as HD and [Full HD](https://en.wikipedia.org/wiki/Full_HD), and higher or lower monitor resolutions as well as older NTSC or PAL standard CRT TV resolutions. As shipped (i.e., without custom overclocking) it can support the following resolutions: 640×350 [EGA](https://en.wikipedia.org/wiki/Enhanced_Graphics_Adapter); 640×480 [VGA](https://en.wikipedia.org/wiki/Video_Graphics_Array); 800×600 [SVGA](https://en.wikipedia.org/wiki/Super_video_graphics_array); 1024×768 [XGA](https://en.wikipedia.org/wiki/XGA); 1280×720 [720p](https://en.wikipedia.org/wiki/720p) [HDTV](https://en.wikipedia.org/wiki/High-definition_television#_blank); 1280×768 [WXGA](https://en.wikipedia.org/wiki/Graphic_display_resolutions#_blank) variant; 1280×800 [WXGA](https://en.wikipedia.org/wiki/Graphic_display_resolutions" \l "_blank)variant; 1280×1024 [SXGA](https://en.wikipedia.org/wiki/SXGA); 1366×768 [WXGA](https://en.wikipedia.org/wiki/Graphic_display_resolutions#_blank) variant; 1400×1050 [SXGA+](https://en.wikipedia.org/wiki/SXGA%2B); 1600×1200 [UXGA](https://en.wikipedia.org/wiki/UXGA); 1680×1050 [WXGA+](https://en.wikipedia.org/wiki/WXGA%2B); 1920×1080 [1080p](https://en.wikipedia.org/wiki/1080p) [HDTV](https://en.wikipedia.org/wiki/High-definition_television#_blank); 1920×1200 [WUXGA](https://en.wikipedia.org/wiki/WUXGA).[[43]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-video-43)

Higher resolutions, up to 2048×1152, may work[[44]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-44)[[45]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-45) or even 3840×2160 at 15 Hz (too low a frame rate for convincing video).[[46]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-46) Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at these resolutions; in fact, the Pis are known to not work reliably for [H.265](https://en.wikipedia.org/wiki/H.265) (at those high resolutions), commonly used for very high resolutions (however, most common formats up to Full HD do work).

Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU is more powerful than its predecessors, potentially fast enough to allow the decoding of H.265-encoded videos in software.[[47]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-47) The GPU in the Raspberry Pi 3 runs at higher clock frequencies of 300 MHz or 400 MHz, compared to previous versions which ran at 250 MHz.[[48]](https://en.wikipedia.org/wiki/Raspberry_Pi" \l "cite_note-48)

The Raspberry Pis can also generate [576i](https://en.wikipedia.org/wiki/576i) and [480i](https://en.wikipedia.org/wiki/480i) [composite video](https://en.wikipedia.org/wiki/Composite_video) signals, as used on old-style ([CRT](https://en.wikipedia.org/wiki/Cathode_ray_tube)) TV screens and less-expensive monitors through standard connectors – either RCA or 3.5 mm phono connector depending on models. The television signal standards supported are [PAL-BGHID](https://en.wikipedia.org/wiki/PAL#_blank), [PAL-M](https://en.wikipedia.org/wiki/PAL-M), [PAL-N](https://en.wikipedia.org/wiki/PAL-N), [NTSC](https://en.wikipedia.org/wiki/NTSC) and [NTSC-J](https://en.wikipedia.org/wiki/NTSC-J).[[49]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-composite-49)

### Real-time clock:

None of the current Raspberry Pi models have a built-in [real-time clock](https://en.wikipedia.org/wiki/Real-time_clock), so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a [network time server](https://en.wikipedia.org/wiki/Network_Time_Protocol) or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the [file system](https://en.wikipedia.org/wiki/File_system), the Pi automatically saves the current system time on shutdown, and re-loads that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, may be added (often via the [I²C](https://en.wikipedia.org/wiki/I²C) interface).

### Operating systems

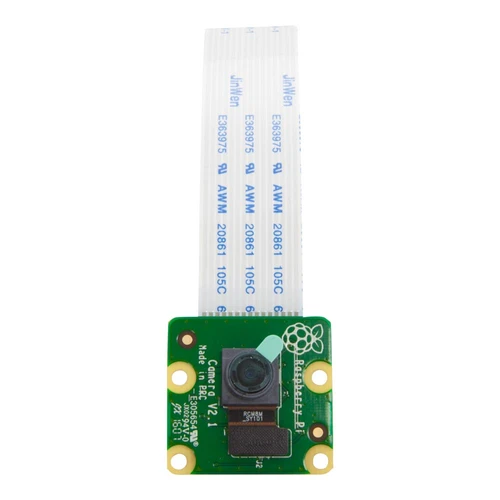


Various operating systems for the Raspberry Pi can be installed on a MicroSD, MiniSD or SD card, depending on the board and available adapters; seen here is the MicroSD slot located on the bottom of a Raspberry Pi 2 board.

The Raspberry Pi Foundation provides [Raspbian](https://en.wikipedia.org/wiki/Raspbian), a Debian-based [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution)for download, as well as third-party [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), [Windows 10 IoT Core](https://en.wikipedia.org/wiki/Windows_10_IoT_Core), [RISC OS](https://en.wikipedia.org/wiki/RISC_OS), and specialised [media centre](https://en.wikipedia.org/wiki/OpenELEC) distributions.[[106]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-raspberrypi_downloads-106) It promotes [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [Scratch](https://en.wikipedia.org/wiki/Scratch_(programming_language)) as the main programming languages, with support for many other languages.[[107]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-107) The default [firmware](https://en.wikipedia.org/wiki/Firmware) is [closed source](https://en.wikipedia.org/wiki/Closed_source), while an unofficial [open source](https://en.wikipedia.org/wiki/Open_source) is available.[[108]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-108)[[109]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-109)[[110]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-110) Many other operating systems can also run on the Raspberry Pi. Other third-party operating systems available via the official website include [Ubuntu MATE](https://en.wikipedia.org/wiki/Ubuntu_MATE), [Windows 10 IoT Core](https://en.wikipedia.org/wiki/Windows_10_IoT_Core), [RISC OS](https://en.wikipedia.org/wiki/RISC_OS) and specialised distributions for the [Kodi](https://en.wikipedia.org/wiki/Kodi_(software)) media centre and classroom management.

**2.3 Raspberry Pi Camera**

5 megapixel camera capable of taking photographs of 3280 x 2464 pixelsCapture video at 1080p30, 720p60 and 640x480p90 resolutionsAll software is supported within the latest version of Raspbian Operating System.



The Camera v2is the new official camera board released by the Raspberry Pi foundation.The Raspberry Pi Camera Module v2 is a high quality 8megapixel Sony IMX219 imagesensorcustom designed add-onboardfor Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras.

* 5 megapixel camera capable of taking photographs of 3280 x 2464 pixels
* Capture video at 1080p30, 720p60 and 640x480p90 resolutions
* All software is supported within the latest version of Raspbian Operating System

The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable.The high quality Sony IMX219 imagesensor itself has a native resolution of 8megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

**Applications**

-CCTV security camera

-motion detection

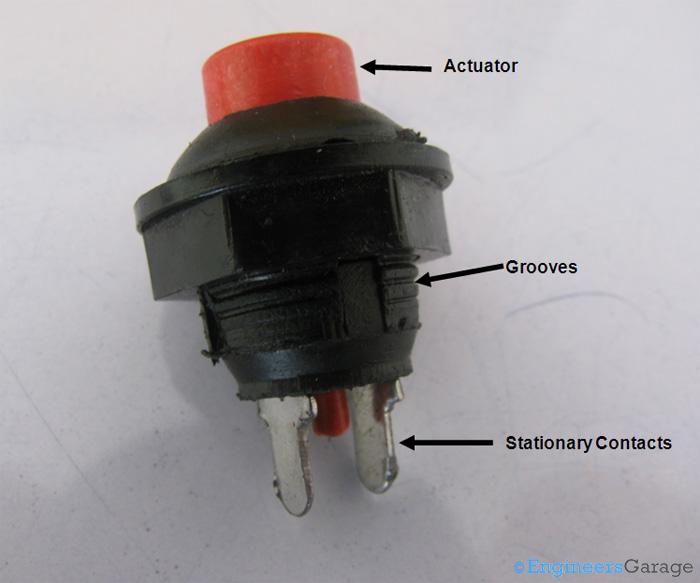
-time lapse photography

**2.4 Push Button Switch**

Ever imagined how quickly the horn of your vehicle responds? Not even a second or two, just push the button and there comes the sound. You put your hand off the button and it’s back to silent mode.   An instant response of these horns is due to what we call “push -button switches”.

 Push button switches are those which can be made to work with the force of a finger or two.  Not only vehicles but camera, lifts and several other common and uncommon interactions with machines/gadgets involve push button switches applications. But what makes this switch so user friendly? What makes it respond only for the time it is pressed no longer or shorter? How the contacts inside this switch work?

 Let us explore the internal architecture of the pushbutton and find out the answers to the questions through this insight.



The image above shows the external view of a conventional SPST push button switch. Almost all the parts of the switch can be figured out by observing its external structure. The red colored bulge is the actuator of the switch. The actuator extends towards the bottom of the switch and emerges out as a thin cylinder. Among other prominent extensions are the two stationary metal contacts legs at the bottom. A groovy pattern is also provided for the purpose of easy mounting.

 Usually, external body of push button switches is made from polymer plastics and can have multiple shapes, sizes and output terminals depending upon its use.

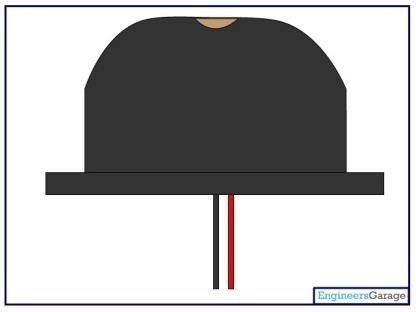
**2.5 Piezo Buzzer**



The **piezo buzzer** produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits.

 The buzzer produces a same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other. This, push and pull action, results in a sound wave. Most buzzers produce sound in the range of 2 to 4 kHz. The Red lead is connected to the Input and the Black lead is connected to Ground.

Click to read more about internal structure and working of a [piezo buzzer](http://www.engineersgarage.com/insight/how-piezo-buzzer-works).



**2.6 Ultrasonic Sensor**

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below



Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.



|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

### **Ultrasonic Sensor Pin Configuration**

**HC-SR04 Sensor Features**

* Operating voltage: +5V
* Theoretical  Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

**How to use the HC-SR04 Ultrasonic Sensor**

**HC-SR04 distance sensor** is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

  Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

**Applications**

* Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
* Used to measure the distance within a wide range of 2cm to 400cm
* Can be used to map the objects surrounding the sensor by rotating it
* Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water
* ots like biped robot, obstacle avoider robot, path finding robot etc.
* Used to measure the distance within a wide range of 2cm to 400cm
* Can be used to map the objects surrounding the sensor by rotating it
* Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

**CHAPTER 3**

Architecture Diagram:

CNN

Image Classification

Raspberry pi Camera

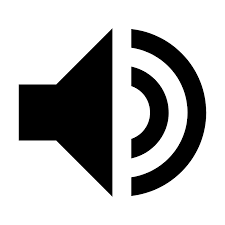
Ultrasonic Sensor

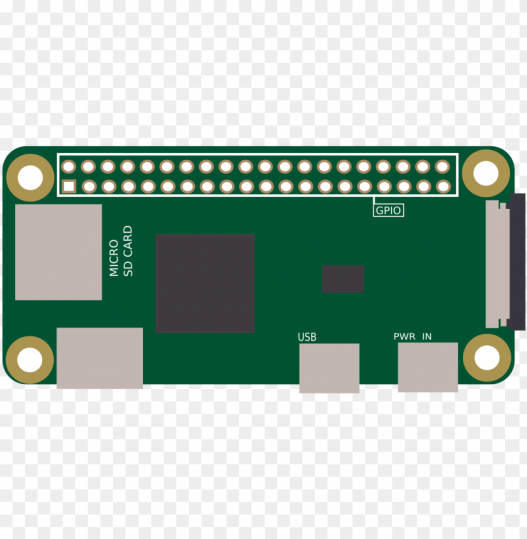
Raspberry Pi Zero

Buzzer

Capture images and send number of array







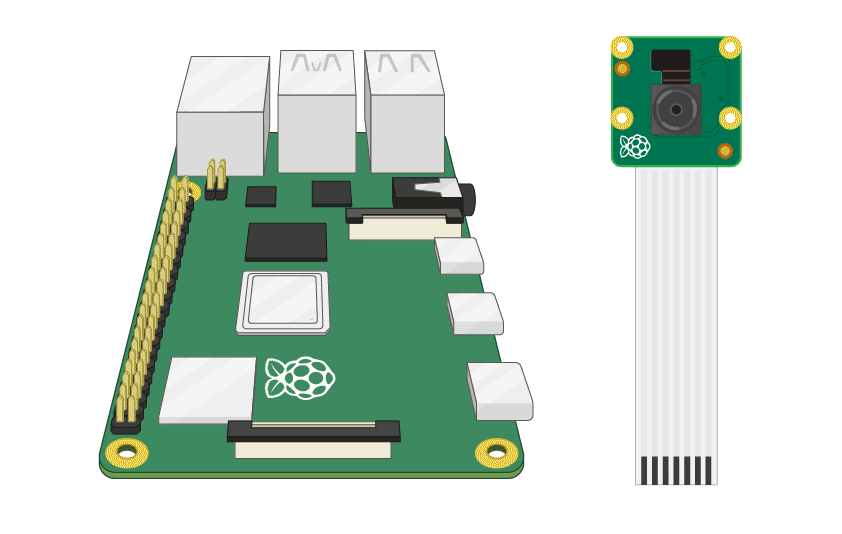


Image Preprocessing

Feature Selection

Send trigger signal and receive echo signal

Send signal to buzzer when person detected

**CHAPTER 4**

**4.1 Introduction to Python**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* System scripting.

**What can Python do?**

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

**Why Python?**

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.

**Good to know**

* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
* Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official [End Of Life date of January 1, 2020](https://pythonclock.org/) has been established for Python 2, after which time it will no longer be maintained.
* Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe [Monty Python’s Flying Circus](https://en.wikipedia.org/wiki/Monty_Python's_Flying_Circus), of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
* It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

**Python Syntax compared to other programming languages**

* Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

### Python is Interpreted

* Many languages are compiled, meaning the source code you create needs to be translated
* into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
* This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
* One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
* In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.
* For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and [object-oriented programming](https://realpython.com/python3-object-oriented-programming/).
* Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
* Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

**4.2 OpenCVPython**

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. ... OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

**4.3 Convolutional Neural Networks (CNN)**

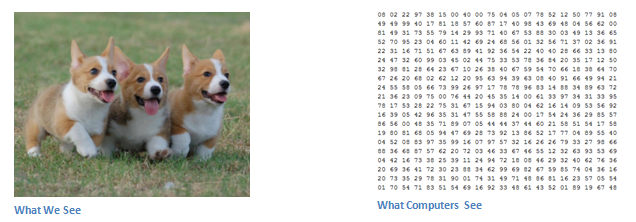
**Introduction**

Convolutional neural networks (CNN) sounds like a weird combination of biology and math with a little CS sprinkled in, but these networks have been some of the most influential innovations in the field of computer vision. 2012 was the first year that neural nets grew to prominence as Alex Krizhevsky used them to win that year’s ImageNet competition (basically, the annual Olympics of computer vision), dropping the classification error record from 26% to 15%, an astounding improvement at the time. Ever since then, a host of companies have been using deep learning at the core of their services. Facebook uses neural nets for their automatic tagging algorithms, Google for their photo search, Amazon for their product recommendations, Pinterest for their home feed personalization, and Instagram for their search infrastructure.

However, the classic, and arguably most popular, use case of these networks is for image processing. Within image processing, let’s take a look at how to use these CNNs for image classification.

**The Problem Space**

                Image classification is the task of taking an input image and outputting a class (a cat, dog, etc) or a probability of classes that best describes the image. For humans, this task of recognition is one of the first skills we learn from the moment we are born and is one that comes naturally and effortlessly as adults. Without even thinking twice, we’re able to quickly and seamlessly identify the environment we are in as well as the objects that surround us. When we see an image or just when we look at the world around us, most of the time we are able to immediately characterize the scene and give each object a label, all without even consciously noticing. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.



**Inputs and Outputs**

                When a computer sees an image (takes an image as input), it will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values). Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describe the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer.  The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for cat, .15 for dog, .05 for bird, etc).

What We Want the Computer to do

                Now that we know the problem as well as the inputs and outputs, let’s think about how to approach this. What we want the computer to do is to be able to differentiate between all the images it’s given and figure out the unique features that make a dog a dog or that make a cat a cat. This is the process that goes on in our minds subconsciously as well. When we look at a picture of a dog, we can classify it as such if the picture has identifiable features such as paws or 4 legs. In a similar way, the computer is able perform image classification by looking for low level features such as edges and curves, and then building up to more abstract concepts through a series of convolutional layers. This is a general overview of what a CNN does. Let’s get into the specifics.

**Biological Connection**

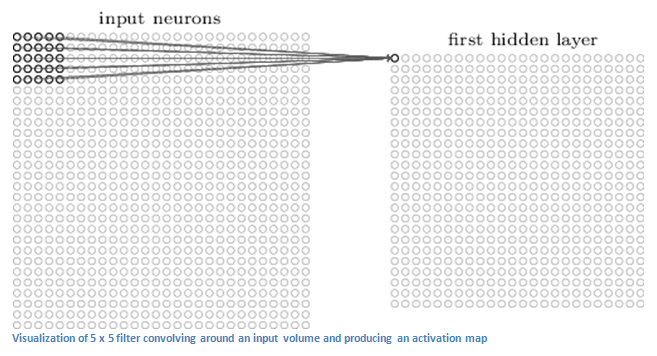
                But first, a little background. When you first heard of the term convolutional neural networks, you may have thought of something related to neuroscience or biology, and you would be right. Sort of. CNNs do take a biological inspiration from the visual cortex. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field. This idea was expanded upon by a fascinating experiment by Hubel and Wiesel in 1962 (Video) where they showed that some individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation. For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges. Hubel and Wiesel found out that all of these neurons were organized in a columnar architecture and that together, they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks (the neuronal cells in the visual cortex looking for specific characteristics) is one that machines use as well, and is the basis behind CNNs.

**Structure**

                Back to the specifics. A more detailed overview of what CNNs do would be that you take the image, pass it through a series of convolutional, nonlinear, pooling (down sampling), and fully connected layers, and get an output. As we said earlier, the output can be a single class or a probability of classes that best describes the image. Now, the hard part understands what each of these layers do. So let’s get into the most important one.

**First Layer – Math Part**

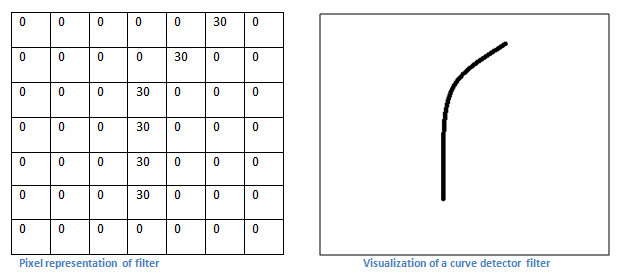
                The first layer in a CNN is always a Convolutional Layer. First thing to make sure you remember is what the input to this conv (I’ll be using that abbreviation a lot) layer is. Like we mentioned before, the input is a 32 x 32 x 3 array of pixel values. Now, the best way to explain a conv layer is to imagine a flashlight that is shining over the top left of the image. Let’s say that the light this flashlight shines covers a 5 x 5 area. And now, let’s imagine this flashlight sliding across all the areas of the input image. In machine learning terms, this flashlight is called a filter (or sometimes referred to as a neuron or a kernel) and the region that it is shining over is called the receptive field. Now this filter is also an array of numbers (the numbers are called weights or parameters). A very important note is that the depth of this filter has to be the same as the depth of the input (this makes sure that the math works out), so the dimension of this filter is 5 x 5 x 3. Now, let’s take the first position the filter is in for example.  It would be the top left corner. As the filter is sliding, or convolving, around the input image, it is multiplying the values in the filter with the original pixel values of the image (aka computing element wise multiplications). These multiplications are all summed up (mathematically speaking, this would be 75 multiplications in total). So now you have a single number. Remember, this number is just representative of when the filter is at the top left of the image. Now, we repeat this process for every location on the input volume. (Next step would be moving the filter to the right by 1 unit, then right again by 1, and so on). Every unique location on the input volume produces a number. After sliding the filter over all the locations, you will find out that what you’re left with is a 28 x 28 x 1 array of numbers, which we call an activation map or feature map. The reason you get a 28 x 28 array is that there are 784 different locations that a 5 x 5 filter can fit on a 32 x 32 input image. These 784 numbers are mapped to a 28 x 28 array.



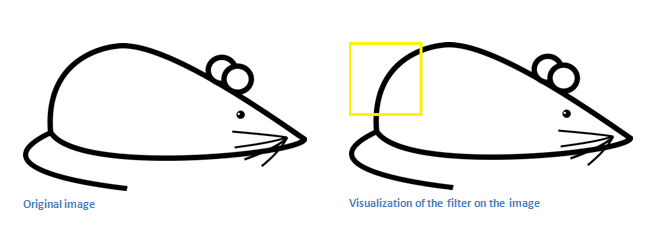
Let’s say now we use two 5 x 5 x 3 filters instead of one. Then our output volume would be 28 x 28 x 2. By using more filters, we are able to preserve the spatial dimensions better. Mathematically, this is what’s going on in a convolutional layer.

**First Layer – High Level Perspective**

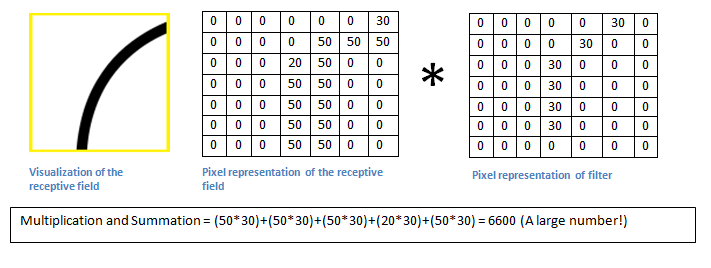
                However, let’s talk about what this convolution is actually doing from a high level. Each of these filters can be thought of as **feature identifiers**. When I say features, I’m talking about things like straight edges, simple colors, and curves. Think about the simplest characteristics that all images have in common with each other. Let’s say our first filter is 7 x 7 x 3 and is going to be a curve detector. (In this section, let’s ignore the fact that the filter is 3 units deep and only consider the top depth slice of the filter and the image, for simplicity.)As a curve detector, the filter will have a pixel structure in which there will be higher numerical values along the area that is a shape of a curve (Remember, these filters that we’re talking about as just numbers!).



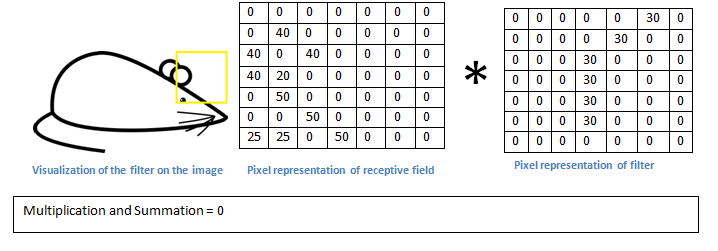
Now, let’s go back to visualizing this mathematically. When we have this filter at the top left corner of the input volume, it is computing multiplications between the filter and pixel values at that region. Now let’s take an example of an image that we want to classify, and let’s put our filter at the top left corner.



Remember, what we have to do is multiply the values in the filter with the original pixel values of the image.



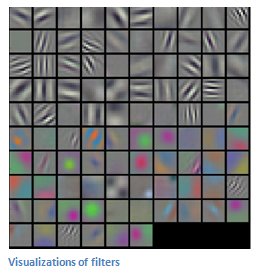
Basically, in the input image, if there is a shape that generally resembles the curve that this filter is representing, then all of the multiplications summed together will result in a large value! Now let’s see what happens when we move our filter.



The value is much lower! This is because there wasn’t anything in the image section that responded to the curve detector filter. Remember, the output of this conv layer is an activation map. So, in the simple case of a one filter convolution (and if that filter is a curve detector), the activation map will show the areas in which there at mostly likely to be curves in the picture. In this example, the top left value of our 26 x 26 x 1 activation map (26 because of the 7x7 filter instead of 5x5) will be 6600. This high value means that it is likely that there is some sort of curve in the input volume that caused the filter to activate. The top right value in our activation map will be 0 because there wasn’t anything in the input volume that caused the filter to activate (or more simply said, there wasn’t a curve in that region of the original image). Remember, this is just for one filter. This is just a filter that is going to detect lines that curve outward and to the right. We can have other filters for lines that curve to the left or for straight edges. The more filters, the greater the depth of the activation map, and the more information we have about the input volume.

**Disclaimer:**

  The filter I described in this section was simplistic for the main purpose of describing the math that goes on during a convolution. In the picture below, you’ll see some examples of actual visualizations of the filters of the first conv layer of a trained network. Nonetheless, the main argument remains the same. The filters on the first layer convolve around the input image and “activate” (or compute high values) when the specific feature it is looking for is in the input volume.



**Going Deeper Through the Network**

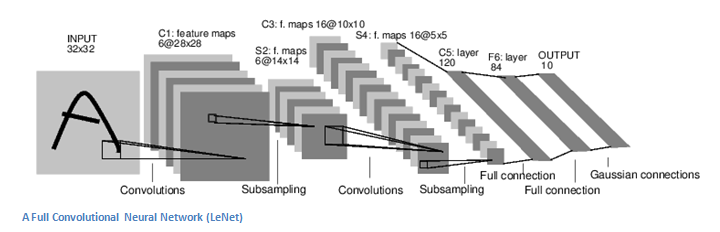
                Now in traditional convolutional neural network architecture, there are other layers that are interspersed between these conv layers. I’d strongly encourage those interested to read up on them and understand their function and effects, but in a general sense, they provide nonlinearities and preservation of dimension that help to improve the robustness of the network and control over fitting. A classic CNN architecture would look like this.

https://adeshpande3.github.io/assets/Table.png

The last layer, however, is an important one and one that we will go into later on. Let’s just take a step back and review what we’ve learned so far. We talked about what the filters in the first conv layer are designed to detect. They detect low level features such as edges and curves. As one would imagine, in order predicting whether an image is a type of object, we need the network to be able to recognize higher level features such as hands or paws or ears. So let’s think about what the output of the network is after the first conv layer. It would be a 28 x 28 x 3 volume (assuming we use three 5 x 5 x 3 filters).  When we go through another conv layer, the output of the first conv layer becomes the input of the 2nd conv layer.  Now, this is a little bit harder to visualize. When we were talking about the first layer, the input was just the original image. However, when we’re talking about the 2nd conv layer, the input is the activation map(s) that result from the first layer. So each layer of the input is basically describing the locations in the original image for where certain low level features appear. Now when you apply a set of filters on top of that (pass it through the 2nd conv layer), the output will be activations that represent higher level features. Types of these features could be semicircles (combination of a curve and straight edge) or squares (combination of several straight edges). As you go through the network and go through more conv layers, you get activation maps that represent more and more complex features. By the end of the network, you may have some filters that activate when there is handwriting in the image, filters that activate when they see pink objects, etc. If you want more information about visualizing filters in ConvNets, Matt Zeiler and Rob Fergus had an excellent research paper discussing the topic. Jason Yosinski also has a video on YouTube that provides a great visual representation. Another interesting thing to note is that as you go deeper into the network, the filters begin to have a larger and larger receptive field, which means that they are able to consider information from a larger area of the original input volume (another way of putting it is that they are more responsive to a larger region of pixel space).

**Fully Connected Layer**

                Now that we can detect these high level features, the icing on the cake is attaching a **fully connected layer** to the end of the network. This layer basically takes an input volume (whatever the output is of the conv or ReLU or pool layer preceding it) and outputs an N dimensional vector where N is the number of classes that the program has to choose from. For example, if you wanted a digit classification program, N would be 10 since there are 10 digits. Each number in this N dimensional vector represents the probability of a certain class. For example, if the resulting vector for a digit classification program is [0 .1 .1 .75 0 0 0 0 0 .05], then this represents a 10% probability that the image is a 1, a 10% probability that the image is a 2, a 75% probability that the image is a 3, and a 5% probability that the image is a 9 (Side note: There are other ways that you can represent the output, but I am just showing the softmax approach). The way this fully connected layer works is that it looks at the output of the previous layer (which as we remember should represent the activation maps of high level features) and determines which features most correlate to a particular class. For example, if the program is predicting that some image is a dog, it will have high values in the activation maps that represent high level features like a paw or 4 legs, etc. Similarly, if the program is predicting that some image is a bird, it will have high values in the activation maps that represent high level features like wings or a beak, etc. Basically, a FC layer looks at what high level features most strongly correlate to a particular class and has particular weights so that when you compute the products between the weights and the previous layer, you get the correct probabilities for the different classes.



**Training (AKA: What Makes this Stuff Work)**

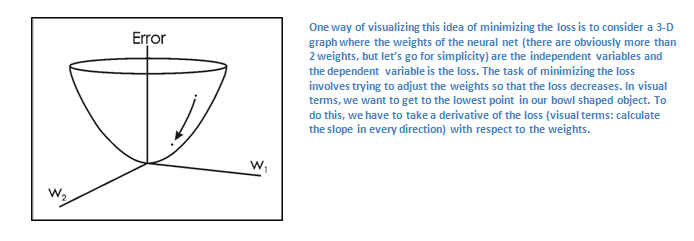
                Now, this is the one aspect of neural networks that I purposely haven’t mentioned yet and it is probably the most important part. There may be a lot of questions you had while reading. How do the filters in the first conv layer know to look for edges and curves? How does the fully connected layer know what activation maps to look at? How do the filters in each layer know what values to have? The way the computer is able to adjust its filter values (or weights) is through a training process called **backpropagation**.

Before we get into backpropagation, we must first take a step back and talk about what a neural network needs in order to work. At the moment we all were born, our minds were fresh. We didn’t know what a cat or dog or bird was. In a similar sort of way, before the CNN starts, the weights or filter values are randomized. The filters don’t know to look for edges and curves. The filters in the higher layers don’t know to look for paws and beaks. As we grew older however, our parents and teachers showed us different pictures and images and gave us a corresponding label. This idea of being given an image and a label is the training process that CNNs go through. Before getting too into it, let’s just say that we have a training set that has thousands of images of dogs, cats, and birds and each of the images has a label of what animal that picture is. Back to backprop.

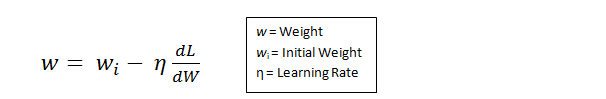
So backpropagation can be separated into 4 distinct sections, the forward pass, the loss function, the backward pass, and the weight update. During the **forward pass**, you take a training image which as we remember is a 32 x 32 x 3 array of numbers and pass it through the whole network. On our first training example, since all of the weights or filter values were randomly initialized, the output will probably be something like [.1 .1 .1 .1 .1 .1 .1 .1 .1 .1], basically an output that doesn’t give preference to any number in particular. The network, with its current weights, isn’t able to look for those low level features or thus isn’t able to make any reasonable conclusion about what the classification might be. This goes to the **loss function** part of backpropagation. Remember that what we are using right now is training data. This data has both an image and a label. Let’s say for example that the first training image inputted was a 3. The label for the image would be [0 0 0 1 0 0 0 0 0 0]. A loss function can be defined in many different ways but a common one is MSE (mean squared error), which is ½ times (actual - predicted) squared.

https://adeshpande3.github.io/assets/Equation.png

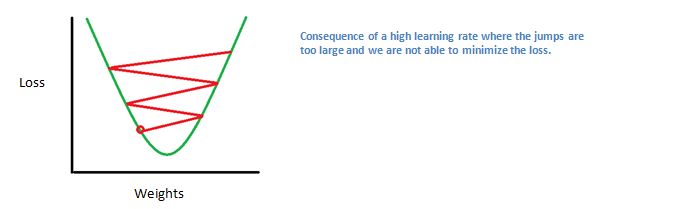
Let’s say the variable L is equal to that value. As you can imagine, the loss will be extremely high for the first couple of training images. Now, let’s just think about this intuitively. We want to get to a point where the predicted label (output of the ConvNet) is the same as the training label (This means that our network got its prediction right).In order to get there, we want to minimize the amount of loss we have. Visualizing this as just an optimization problem in calculus, we want to find out which inputs (weights in our case) most directly contributed to the loss (or error) of the network.



This is the mathematical equivalent of a **dL/dW** where W are the weights at a particular layer. Now, what we want to do is perform a **backward pass** through the network, which is determining which weights contributed most to the loss and finding ways to adjust them so that the loss decreases. Once we compute this derivative, we then go to the last step which is the **weight update**. This is where we take all the weights of the filters and update them so that they change in the opposite direction of the gradient.



The **learning rate** is a parameter that is chosen by the programmer. A high learning rate means that bigger steps are taken in the weight updates and thus, it may take less time for the model to converge on an optimal set of weights. However, a learning rate that is too high could result in jumps that are too large and not precise enough to reach the optimal point.



The process of forward pass, loss function, backward pass, and parameter update is one training iteration. The program will repeat this process for a fixed number of iterations for each set of training images (commonly called a batch). Once you finish the parameter update on the last training example, hopefully the network should be trained well enough so that the weights of the layers are tuned correctly.

**Testing**

Finally, to see whether or not our CNN works, we have a different set of images and labels (can’t double dip between training and test!) and pass the images through the CNN. We compare the outputs to the ground truth and see if our network works!

**How Companies Use CNNs**

                Data, data, data. The companies that have lots of this magic 4 letter word are the ones that have an inherent advantage over the rest of the competition. The more training data that you can give to a network, the more training iterations you can make, the more weight updates you can make, and the better tuned to the network is when it goes to production. Facebook (and Instagram) can use all the photos of the billion users it currently has, Pinterest can use information of the 50 billion pins that are on its site, Google can use search data, and Amazon can use data from the millions of products that are bought every day. And now you know the magic behind how they use it.