1 to 8 aathi introduction

9 to 19 chandru Subsystems, math modelling,

20 to 26 aathi constraints, objectives

27 to 31 chandru Conceptual design, block diag, Implementaion (Very very important)

31 to 37 aathi components

subsystems

mathematical modelling

block diagram

implementation

My friend aathith, presented a brief information to the problem statement and the proposed solution.

Next slide

The major sub systems involved in our solutions include

1. System to take photo
2. System that identifies the students in the photo

Next slide

The system that takes photo, it is evident from the name itself what action is it going to perform. It contains the components such as camera, a system to signal the camera and a system to transfer the taken photo. we will explain what is the need for explicitly signaling the camera there is a need for it.

This is simple.

Next slide

Coming on to the next major sub system, this is the part that is going to do the actual facial recognition. This contains a lot of sub system as it is going to do the important part of identifying somebody from the photo. The flow of operation is follows receive the photo, identify the faces in the photo, for each face identified give the unique descriptor of that face (think unique descriptor as a unique number generated for unique face that the machine can understand) and at last after knowing the descriptor I can easily mark the attendance.

Next slide

I am going to explain more about a few of the components of the systems that I mentioned before.

Next slide

“The system that outputs a descriptor for all the faces in the photo”. After we find “m” faces in photo we are going to iterate through each individual face. Now each face is fed into the facial recognition algorithm that gives us the unique descriptor for each face so at last we will get “m” descriptors for a photo and we will store that.

Next slide

“The system that iterates “m” descriptors and marks the attendance”. We are having “m” descriptors that the machine can understand, now how to mark the attendance. Well at first think we are having a data base that contains the student’s register number and the descriptor of the student’s face stored. Then our work becomes easy just, iterate through the “m” descriptors and check whether they are in the database of student, if it is available mark the attendance against that student.

Next slide

Next up we will be seeing the mathematical modelling of some of the systems explained.

Next slide

Seeing this slide you can understand why are we explicitly signaling the camera, actually we are triggering the camera every 5 min to take a photo.

Next slide

The facial recognition algorithm that takes the input as a image containing one face and then gives the descriptor of that face. I think you already got a good concept of what is the function of descriptor. To be more precise the algorithm actually outputs a vector of fixed size that contains floating numbers. The values of the number changes for different faces.

Next slide

When I was explaining you about the “system that takes m descriptor and marks the attendance” I told you about data base and how we are checking the descriptors are equal to mark the attendance.

But as I told the algorithm here does not return a number but a vector or simply an array of values. To

compare vectors for the similarity we need to use cosine similarity. Essentially what we are doing is telling how much similar these 2 faces are, one that we take in the class, one that was previously taken.

Now my friend will take on

Next

Coming on to the conceptual design,

Next

the answers to the questions were as follows

We needed a system that marks the attendance for the student present in the class, intelligently by assessing how much time the student was present.

From our proposed solution we emphasized the use of facial recognition and camera for our solution.

As the facial recognition overcomes the problems faced by the previously mentioned attendance system

Next slide

Coming to block diag

Next slide

Now to visually represent the systems that we talked about in a concise diagram.

Next slide

Coming on to our implementation we have finalized the use of these 2 algorithms, for facial detection we are going to use DLIB HOG (Histogram of oriented gradients) and for facial recognition a convolutional neural network that has been pre trained for giving out descriptors of size 128.

We are going to use raspberry pi for implementing all of the sub systems explained.

Content to be put in report

Sub systems

Our proposed solution should have a camera and it should a facial recognition algorithm. Keeping in mind these 2 points, we designed the major sub systems in our solution. To control and interface the camera we needed a system and to run the facial recognition we needed another system. So, 2 major sub systems were designed to addressing these 2 specifications.

The major sub systems involved in our solutions include

1. System to take photo
2. System that identifies the students in the photo

We wanted these 2 systems to be separate as when we design them keeping in mind, they are separate and interconnected by another system.

1. System that takes the photo:

The system that takes photo, it is evident from the name itself what action is it going to perform. It contains the components such as

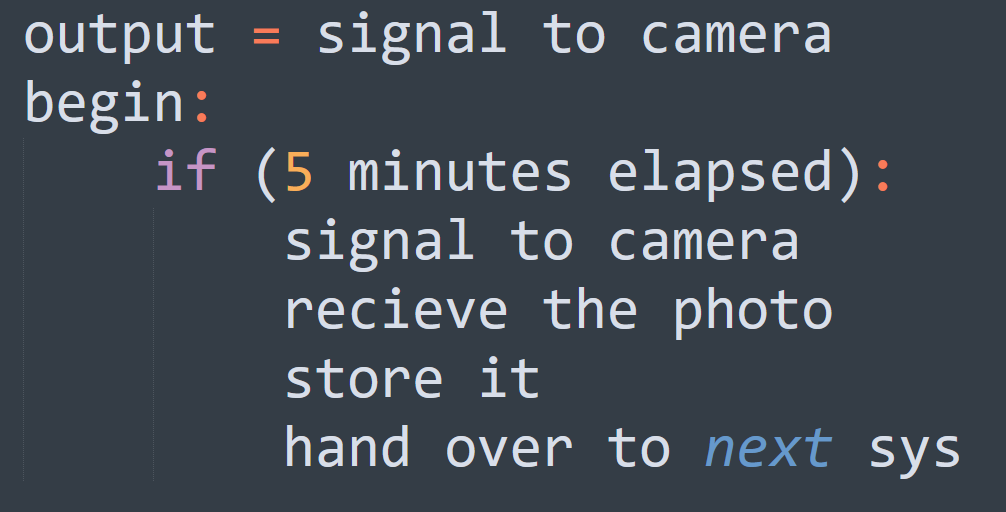
1. camera,
2. a system to signal the camera
3. a system to transfer the taken photo.

We kept another system to explicitly signal the camera and there is a need for it. As one of our motivation is that we wanted to give attendance to those students who are present in the class for its full duration. To tackle this problem, we have simply divided the whole class duration into separate chunks of 5-minute duration. So, every 5 minutes the camera is signaled to take photo and the photo taken is then sent to the next major sub system for facial recognition.

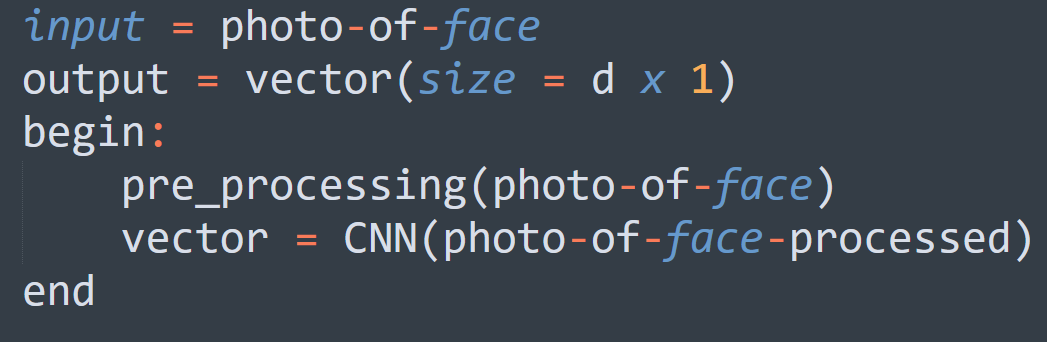
1. Camera
   1. A camera is an optical instrument used to record images. At their most basic, cameras are sealed boxes (the camera body) with a small hole (the aperture) that allow light in to capture an image on a light-sensitive surface (usually photographic film or a digital sensor)
   2. Now days everybody is familiar with the camera thanks to the smartphone revolution. The camera is going to be the important part and the decision of choosing which camera to use if very important. One of the obvious parameters of the camera which is wide known is the megapixels of the camera.
   3. A megapixel (MP) is a million pixels; the term is used not only for the number of pixels in an image but also to express the number of image sensor elements of digital cameras or the number of display elements of digital displays. For example, a camera that makes a 2048 × 1536-pixel image (3,145,728 finished image pixels) typically uses a few extra rows and columns of sensor elements and is commonly said to have "3.2 megapixels" or "3.4 megapixels", depending on whether the number reported is the "effective" or the "total" pixel count
   4. Sensor Size – If you had to guess camera quality based only on one parameter, size of the sensor, will be your best bet for most accurate response. The sensor size determines how much light it can use to create the image. This is also a reason why DSLR with same MP count often perform better than Smartphone Rear cameras.
2. A system to signal the camera
   1. This system actually takes care of signaling the camera. Now what is the need for explicitly signaling the camera
   2. Now lets us first consider another scenario where we get all the images from the camera and we will run the face recognition algorithm for all the images. For example, let’s say our camera outputs 60 frames per second, so that means just for one class we will get 60 \* 60 which is astonishing 3600 frames per second and each image has to transported and processed
   3. Again, the complexity increases when we consider multiple cameras for better detection inside the classroom
   4. This will again increase the burden on the facial recognition sub system this will force us to use a high-performance device for facial recognition. This invariably increase the cost and complexity of the system that we proposed.
   5. To implement this, we can use a microcontroller that is interfaced with the camera, the micro controller can be programmed to signal the camera every 5 minutes. Now the reason for 5 minutes is very simple, we already specified that every 5 minutes the camera takes photo so every 5 minutes the micro controller signals the camera and the camera takes the picture
   6. A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general-purpose applications consisting of various discrete chips.
   7. In modern terminology, a microcontroller is similar to, but less sophisticated than, a system on a chip (SoC). SoC may include a microcontroller as one of its components, but usually integrates it with advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or one or more coprocessors.
   8. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices.
   9. Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz for low power consumption (single-digit milliwatts or microwatts). They generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications.
3. A system to transfer the taken photo
   1. This part can also be programmed in the same micro controller that was discussed before for signaling the camera
   2. Now here the main aim is to transfer the photo that was taken by the camera after it was signaled by the specified micro controller
   3. The microcontroller should be interfaced with a transmitter so that it can send the photo inside a channel.
   4. Now the choice transmitter and the interfacing of Arduino with the transmitter all depends on which channel we select. These are the following channels that are popularly used
   5. Short-range wireless
      1. Bluetooth mesh networking – Specification providing a mesh networking variant to Bluetooth low energy (BLE) with increased number of nodes and standardized application layer (Models).
      2. Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.
      3. Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.
      4. Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.
      5. Wi-Fi – Technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point or directly between individual devices.
      6. ZigBee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.
      7. Z-Wave – Wireless communications protocol used primarily for home automation and security applications
   6. Medium-range wireless
      1. LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.
      2. 5G - 5G wireless networks can be used to achieve the high communication requirements of the IoT and connect a large number of IoT devices, even when they are on the move.
   7. Long-range wireless
      1. Low-power wide-area networking (LPWAN) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission. Available LPWAN technologies and protocols: LoRaWan, Sigfox, NB-IoT, Weightless, RPMA.
      2. Very small aperture terminal (VSAT) – Satellite communication technology using small dish antennas for narrowband and broadband data.
   8. Wired
      1. Ethernet – General purpose networking standard using twisted pair and fiber optic links in conjunction with hubs or switches.
      2. Power-line communication (PLC) – Communication technology using electrical wiring to carry power and data. Specifications such as Home Plug or G.hn utilize PLC for networking IoT devices.
   9. After choosing the channel, we will go on to select the transmitter that is appropriate for the channel so that we can send information properly on the selected channel.
   10. The choice of which channel it all depends on how are we going to implement the two sub systems and the distance between them
4. System that marks the attendance:
   1. Coming on to the next major sub system, this is the part that is going to do the facial recognition. This contains a lot of sub system as it is going to do the important part of identifying somebody from the photo. The flow of operation is as follows
      1. receive the photo
      2. identify the faces in the photo,
      3. For each face identified give the unique descriptor of that face
      4. using the descriptor marking the attendance
   2. A short explanation about unique descriptor is that it can be considered as a unique number generated for unique face that the machine can understand. Ideally the facial recognition algorithm should output the same number if you input the same face even if is the same person in a different outfit or in a different makeup, i.e. even when the physical characteristics changes. To be more technical we can divide this sub system into a number of components as follows, these components are divided in the same way as the flow is described above.
      1. System that receives the photo
      2. System that runs the face detection algorithm that outputs “m” detected faces
      3. System that outputs a descriptor for “m” faces
      4. System that iterates through “m” descriptor and marks the attendance
   3. System that receives the photo
      1. The action that it is going to perform is already prevalent from its name
      2. It actually is responsible for receiving the photo that is transmitted from the first major sub system that was present in the class
   4. System that runs the face detection algorithm that outputs “m” detected faces
      1. The face detection algorithm, finds the faces in the photos, essentially to visualize we can draw bounding boxes that contain the face.
      2. The face detection algorithm outputs a list of co ordinates with which we can draw the bounding box
   5. System that outputs a descriptor for “m” faces
      1. “The system that outputs a descriptor for all the faces in the photo”. After we find “m” faces in photo we are going to iterate through each individual face. Now each face is fed into the facial recognition algorithm that gives us the unique descriptor for each face so at last we will get “m” descriptors for a photo and we will store that.
      2. This can be realized for example using a Convolutional neural network which can be trained for outputting a different descriptor for different unique images fed as the input.
   6. System that iterates through “m” descriptor and marks the attendance
      1. “The system that iterates “m” descriptors and marks the attendance”. We are having “m” descriptors that the machine can understand, now how to mark the attendance. Well at first think we are having a data base that contains the student’s register number and the descriptor of the student’s face stored. Then our work becomes easy just, iterate through the “m” descriptors and check whether they are in the database of student, if it is available mark the attendance against that student.

Mathematical Modelling

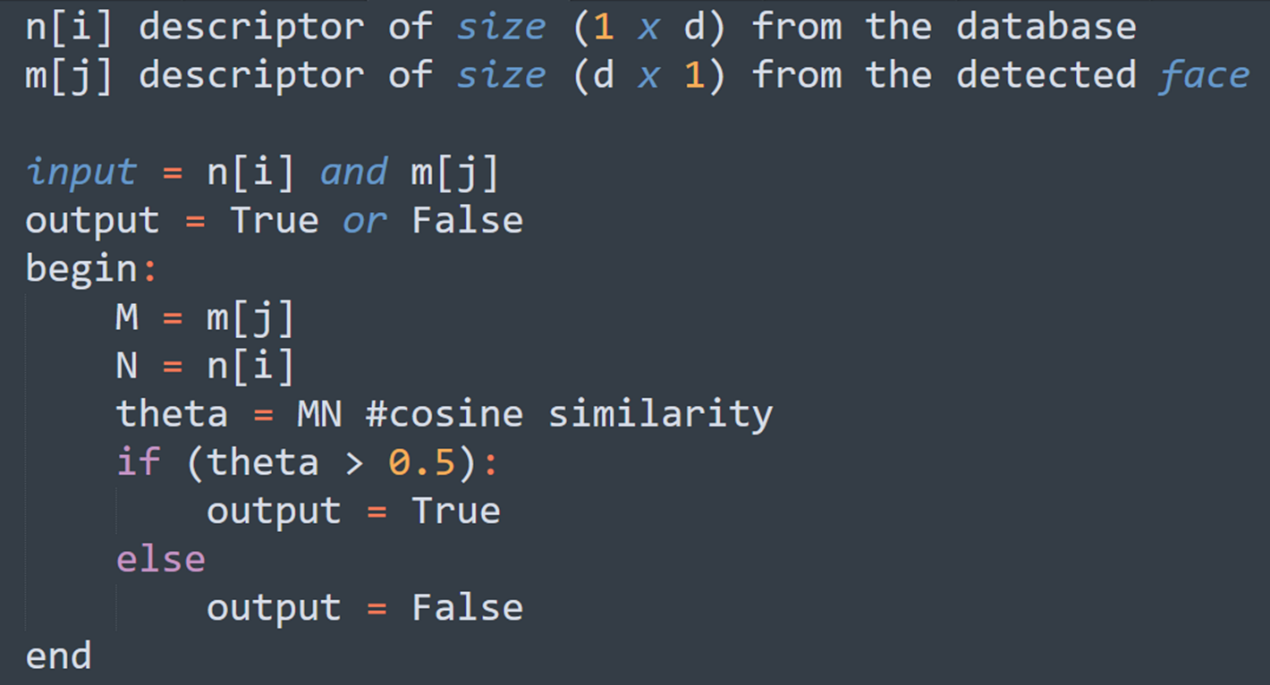
1. System that signals the camera
   1. This system is one of the components of the major sub system “System that takes photo”, this system should signal the camera every 5 minutes, a mathematical mode of this system is given below



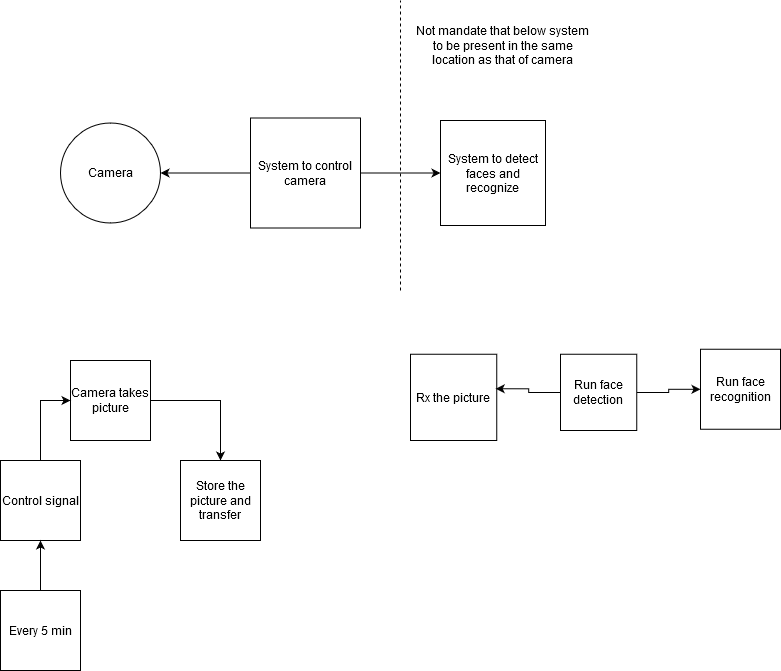
1. System that outputs a descriptor for one face
   1. This system is one of the components of the major sub system “System that marks the attendance”, this system is actually the main part, which actually performs the face recognition it is the algorithm.
   2. The facial recognition algorithm that takes the input as an image containing one face and then gives the descriptor of that face. We have already given a basic concept of what is descriptor and what is its function. To be more precise the algorithm actually outputs a vector of fixed size that contains floating numbers. The values of the number changes for different faces.
   3. a mathematical mode of this system is given below



1. System that checks whether two descriptors are same
   1. This system belongs to the major sub system “System that marks the attendance”, essentially this system compares whether two faces are same , it is kind of what when we do when we are identifying an unknown person in a crowd with the help of a photo graph of the unknown photo graph.
   2. In the explanation of “system that takes m descriptor and marks the attendance” there was mention about data base and how we are checking the descriptors whether they are equal to mark the attendance.
   3. But as algorithm here does not return a number but a vector or simply an array of values. To compare vectors for the similarity we need to use cosine similarity. Essentially what we are doing is telling how much similar these 2 faces are, one that we take in the class, one that was previously taken.

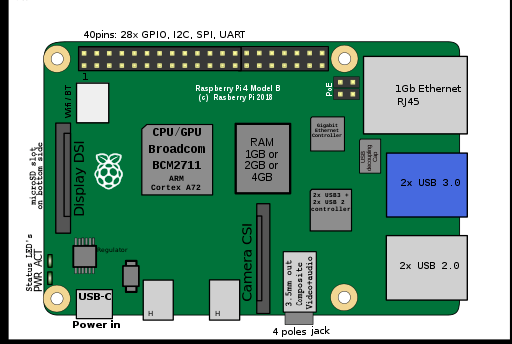


Block diagram



Implementation

1. Comparison of different face detection algorithm this is for deciding which algorithm to use
   1. Haar Cascade Face Detector in OpenCV
      1. Haar Cascade based Face Detector was the state-of-the-art in Face Detection for many years since 2001, when it was introduced by Viola and Jones. There have been many improvements in the recent years.
      2. Pros
         1. Works almost real-time on CPU
         2. Simple Architecture
         3. Detects faces at different scales
      3. Cons
         1. The major drawback of this method is that it gives a lot of False predictions.
            1. Doesn’t work on non-frontal images.
            2. Doesn’t work under occlusion
   2. HoG Face Detector in Dlib
      1. This is a widely used face detection model, based on HoG features and SVM. You can read more about HoG in our post. The model is built out of 5 HOG filters – front looking, left looking, right looking, front looking but rotated left, and a front looking but rotated right. The model comes embedded in the header file itself.
      2. The dataset used for training, consists of 2825 images which are obtained from LFW dataset and manually annotated by Davis King, the author of Dlib.
      3. Pros
         1. Fastest method on CPU
         2. Works very well for frontal and slightly non-frontal faces
         3. Light-weight model as compared to the other three.
         4. Works under small occlusion
      4. Cons
         1. The major drawback is that it does not detect small faces as it is trained for minimum face size of 80×80. Thus, you need to make sure that the face size should be more than that in your application. You can however, train your own face detector for smaller sized faces.
         2. The bounding box often excludes part of forehead and even part of chin sometimes.
         3. Does not work very well under substantial occlusion
         4. Does not work for side face and extreme non-frontal faces, like looking down or up.
   3. DNN Face Detector in OpenCV
      1. This model was included in OpenCV from version 3.3. It is based on Single-Shot-Multibox detector and uses ResNet-10 Architecture as backbone. The model was trained using images available from the web, but the source is not disclosed. OpenCV provides 2 models for this face detector.
      2. Floating point 16 version o
      3. f the original caffe implementation (5.4 MB )
      4. 8-bit quantized version using TensorFlow (2.7 MB)
      5. Pros
         1. The method has the following merits:
         2. Most accurate out of the four methods
         3. Runs at real-time on CPU
         4. Works for different face orientations – up, down, left, right, side-face etc.
         5. Works even under substantial occlusion
         6. Detects faces across various scales (detects big as well as tiny faces)
      6. This is the slowest among the face detectors
   4. CNN Face Detector in Dlib
      1. This method uses a Maximum-Margin Object Detector ( MMOD ) with CNN based features. The training process for this method is very simple and you don’t need a large amount of data to train a custom object detector.
      2. Pros
         1. Works for different face orientations
         2. Robust to occlusion
         3. Works very fast on GPU
         4. Very easy training process
      3. Cons
         1. Very slow on CPU
         2. Does not detect small faces as it is trained for minimum face size of 80×80. Thus, we need to make sure that the face size should be more than that in our application.
   5. For medium to large image sizes - Dlib HoG is the fastest method on CPU. But it does not detect small sized faces (< 70x70). So, if you know that your application will not be dealing with very small sized faces (for example a selfie app), then HoG based Face detector is a better option. Also, if you can use a GPU, then MMOD face detector is the best option as it is very fast on GPU and also provides detection at various angles.
   6. We have stated the comparison of face detection algorithms as we can see that the first 2 algorithms are the most popular in case of algorithms that does not use CNN.
   7. The Face detection algorithm using CNN actually gives the advantage over these algorithms but at the cost of performance
   8. Based on our environment considerations and our implementation we have decided to not to use CNN based face detection.
2. Raspberry Pi 4
   1. Raspberry Pi 4 Model B was released in June 2019 with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, full gigabit Ethernet (throughput not limited), two USB 2.0 ports, two USB 3.0 ports, and dual-monitor support via a pair of micro HDMI (HDMI Type D) ports for up to 4K resolution. The Pi 4 is also powered via a USB-C port, enabling additional power to be provided to downstream peripherals, when used with an appropriate PSU
   2. The Raspberry Pi 4 uses a Broadcom BCM2711 SoC with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor, with 1 MiB shared L2 cache.[44][45] Unlike previous models, which all used a custom interrupt controller poorly suited for virtualization, the interrupt controller on this SoC is compatible with the ARM Generic Interrupt Controller (GIC) architecture 2.0, providing hardware support for interrupt distribution when using ARM virtualization capabilities
   3. The Raspberry Pi 4, with a quad-core ARM Cortex-A72 processor, is described as having three times the performance of a Raspberry Pi 3.
   4. The Raspberry Pi 4 is available with 2, 4 or 8 GiB of RAM



1. Raspberry pi 8 MP camera
   1. The 8MP Raspberry Pi Camera Module v2 can be used to take high-definition video, as well as stills photographs. It uses high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It is 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. It is suitable for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. It’s easy to use for beginners.The camera works with all models of Raspberry Pi 1, 2, and 3. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library.
   2. Specifications of 8MP Raspberry Pi Camera V2 :-
      1. 8 megapixel camera capable of taking photographs of 3280x2464 pixels
      2. Capture video at 1080p30, 720p60 and 640x480p90 resolutions
      3. All software is supported within the latest version of Raspbian Operating System
      4. Supports Raspberry Pi 1,2 and 3
      5. Applications: CCTV security camera, motion detection, time lapse photography

Reference

<https://en.wikipedia.org/wiki/Internet_of_things>

<https://en.wikipedia.org/wiki/Microcontroller>

<https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/>