

EMOTION RECOGNITION FROM FACIAL EXPRESSIONS USING HYBRID FEATURE METHOD

Submitted in partial fulfilment of the requirements for the award of
Bachelor of engineering degree in
Electronics and Communication Engineering

By

J.BENITTA AROGANA (37130052)

C.SINDUJA (37130383)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2021



DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Benitta Arogana.J(37130052) and Sinduja.C(37130383) who carried out the project entitled "EMOTION RECOGNITION FROM FACIAL EXPRESSIONS USING HYBRID FEATURE METHOD" under my supervision from November 2020 to April 2021.

Internal Guide

Dr. G. MERLIN SHEEBA M.E., Ph.D.

Head of the Department

Dr.T.RAVI M.E.,Ph.D.

Submitted for Viva voce Examination held on **17th April 2021**

Internal Examiner

External Examiner

DECLARATION

We, Benitta Arogana. J (37130052) and Sinduja.c (37130383) hereby declare that the Project Report entitled "EMOTION RECOGNITION FROM FACIAL EXPRESSIONS USING HYBRID FEATURE METHOD" done by us under the guidance of Dr.G.MERLIN SHEEBA M.E.,Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

- 1. BENITTA AROGANA.J (37130052)**
- 2. SINDUJA (37130383)**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA M.E., Ph.D.** Dean, School of Electrical and Electronics and **Dr. T. RAVI M.E., Ph.D.** Head of the Department, Dept. of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. MERLIN SHEEBA M.E., Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

I wish to express my thanks to the Project Panel members for their valuable feedbacks during the Project Reviews which were useful in many ways for the completion of the project.

ABSTRACT

In face-to-face experiences, facial expressions transmit non-verbal human content. Since the early 90's, researchers have drawn growing attention to automatic face expression processing, which plays a key role in human computer interfaces. Classical methods of machine learning often involve a complicated process of extraction and yield bad results. This project apply the latest developments to profound learning to suggest efficient deep CNNs that can translate semantic zed knowledge available on the face correctly and automated without hand-designing function descriptors. In recent years, a number of papers have suggested a general paradigm for the recognition of face speech using models of profound learning. While the works have an improved result, a great improvement still seems to be required. We are offering a profound learning approach through a focused network that can concentrate on crucial parts of the face and significantly expand on various datasets compared to previous models. Thanks to experimental findings, we demonstrate the visualisation method that is capable of finding essential facial regions based on the classification performance to detect various emotions.

CONTENTS		
CHAPTER NO	CHAPTER NAME	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	
	1.1 BACKGROUND	1
	1.2 CONVOLUTIONAL	3
	1.3 RECTIFIED LINEAR UNIT	4
	1.4 POOLING	4
	1.5 PROBLEM DEFINITION	5
2	LITERATURE REVIEW	6
3	AIM AND SCOPE OF THE PROJECT	8
	3.1 EXISTING SYSTEM	8
	3.2 PROPOSED SYSTEM	9
	3.3 PROPOSED APPLICATION	9
4	MATERIALS AND ALGORITHM	9
	4.1 ALGORITHM	11
	4.2 HARDWARE REQUIREMENTS	12
	4.22 FEATURES	13
	4.23 VIDEO SECURITY	14
	4.24 INPUT CONTROL DEVICES	14
	4.25 TECHNOLOGY	15
	4.26 TECHNOLOGY CIRCUIT	15
	4.27 RASBERRY PI	16
	4.3 HARDWARE CONNECTIONS	17
	4.31 PROCESSOR	17
	4.32 PERFORMANCE	18
	4.33 OVER CLOCKING	19
	4.34 RAM	21
	4.35 NET WORKING	21
	4.36 PERIPHERALS	22
	4.37 VIDEO	22
	4.38 REAL-TIME CLOCK	24
	4.4 SOFTWARE REQUIREMENTS	25
	4.41 PYTHON FILES IN IDLE	31
	4.42 OPENCV	33
	4.43 TENSORFLOW	33

CONTENTS		
CHAPTER NO	CHAPTER NAME	PAGE NO.
5	RESULT AND ANALYSIS	35
6	SUMMARY AND CONCLUSION	36
7	REFERENCES	37

FIGURE NUMBER	FIGURE NAME	PAGE NUMBER
1.1	Extraction of images	3
1.2	Single depth slice	5
4.1	Camera	12
4.2	Hardware connections	17
4.3	Processor	18
4.4	Peripheral	23
4.6	Python logo	27
4.7	Opencv logo	34

CHAPTER 1

INTRODUCTION

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioural science and in clinical practice. An automatic Facial Expression Recognition system needs to perform detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification. Facial expression recognition system is implemented using Convolutional Neural Network (CNN). CNN model of the project. The facial expression dataset with seven facial expression labels as happy, sad, surprise, fear, anger, disgust, and neutral is used in this project. The system achieved 56.77 % accuracy and 0.57 precision on testing dataset. The Music can be play the emotions, and when users are in a negative mood, the SD card module will be automatically play the appropriate music list. This project explores a couple of machine learning algorithms as well as facial-emotion-recognition techniques which would help us in accurate identification of the human emotion.

1.1 BACKGROUND

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, disgust, and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind.

Automatic recognition of facial expressions can be an important component of natural human machine interfaces; it may also be used in behavioural science and in clinical practice. It had been studied for a long period of time and obtaining the progress recent decades. Though much progress has been made, recognizing facial expression with a high accuracy remains to be difficult due to the complexity and varieties of facial expressions. On day to day basics humans commonly recognize emotions by characteristic features, displayed as a part of a facial expression. For instance happiness is undeniably associated with a smile or an upward movement of the corners of the lips. Similarly other emotions are characterized by other deformations typical to a particular expression. Research into automatic recognition of facial expressions addresses the problems surrounding the representation and categorization of static or dynamic characteristics of these deformations of face pigmentation. In machine learning, a Convolutional neural network (CNN or Convents) is a type of feed forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field.

The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a Convolutional operation. Convolutional networks were inspired by biological processes ² and are variations of multilayer perception designed to use minimal amounts of pre-processing. They have wide applications in image and video recognition, recommender systems and natural language processing. The Convolutional neural network is also known as shift invariant or space invariant artificial neural network (SIANN), which is named based on its shared weights architecture and translation invariance characteristics. LeNet is one of the very first Convolutional neural networks which helped propel the field of Deep Learning. This pioneering work by YannLeCun was named LeNet5 was used mainly for character recognition tasks such as reading zip codes, digits, etc

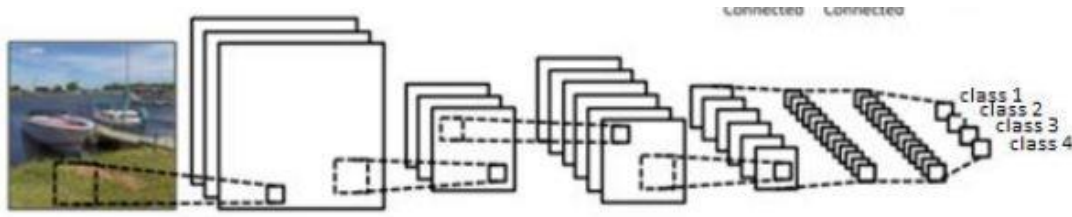


FIG 1.1: EXTRACTION OF IMAGES

1.2 Convolutional

The primary purpose of Convolutional in case of a CNN is to extract features from the input image. Convolutional preserves the spatial relationship between pixels by learning image features using small squares of input data. The Convolutional layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. For example, a typical filter on a first layer of a CNN might have size $3 \times 5 \times 5$ (i.e. images have depth 3 i.e. the colour channels, 5 pixels width and height). During the forward pass, each filter is convolved across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As the filter convolve over the width and height of the input volume it produces a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network.

Now, there will be an entire set of filters in each Convolutional layer (e.g. 20 filters), and each of them will produce a separate 2-dimensional activation map. The 2-dimensional Convolutional between image A and Filter B can be given as: $C(i,j) = \sum_{n=0}^{Na-1} \sum_{m=0}^{Ma-1} A(m,n) * B(i-m, j-n)$ (2.1) where size of A is $(Ma \times Na)$, size of B is $(Mb \times Nb)$, $0 \leq i < Ma + Mb - 1 \wedge 0 \leq j < Na + Nb - 1$. A filter convolves with the input image to produce a feature map. The Convolutional of another filter over the same image gives a different feature map. Convolutional operation captures the local dependencies in the original image. A CNN learns the values of these filters on its own during the training process (although parameters such as number of filters filter size, architecture of the network etc. still needed to specify before the training process). The

more number of filters, the more image features get extracted and the better network becomes at recognizing patterns in unseen images. The size of the Feature Map (Convolved Feature) is controlled by three parameters

- Depth: Depth corresponds to the number of filters we use for the Convolutional operation.
- Stride: Stride is the size of the filter; if the size of the filter is 5x5 then stride is 5.
- Zero-padding: Sometimes, it is convenient to pad the input matrix with zeros around the border, so that filter can be applied to bordering elements of input image matrix. Using zero padding size of the feature map can be controlled.

1.3 Rectified Linear Unit

An additional operation called ReLU has been used after every Convolutional operation. A Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation function to calculate its output given x : $R(x) = \text{Max}(0, x)$ (2.2) Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. When initializing the weights uniformly, half of the weights are negative. This helps creating a sparse feature representation. Another positive aspect is the relatively cheap computation. No exponential function has to be calculated. This function also prevents the vanishing gradient error, since the gradients are linear functions or zero but in no case nonlinear

1.4 POOLING (SUB-SAMPLING)

Spatial Pooling (also called sub-sampling or down-sampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc. In case of Max Pooling, a spatial neighbourhood (for example, a 2x2 window) is defined and the largest element is taken from the rectified feature map within that window. In case of average pooling the average or sum of all elements in that window is taken. In practice, Max Pooling has been shown to work better. Max Pooling reduces the input by applying the maximum function over the input x_i . Let m be the size of the filter, then the output calculates as follows: $M(x_i) = \max \{x_{i+k,l} \mid |k| \leq m/2, |l| \leq m/2, k, l \in \mathbb{N}\}$

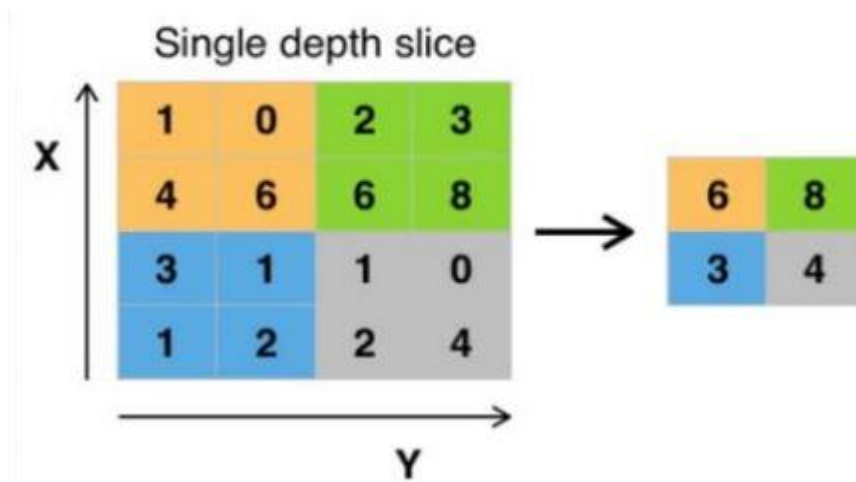


FIG 1.2: SINGLE DEPTH SLICE

The function of Pooling is to progressively reduce the spatial size of the input representation. In particular, pooling

- Makes the input representations (feature dimension) smaller and more manageable
- Reduces the number of parameters and computations in the network, therefore, controlling over-fitting
- Makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in input will not change the output of Pooling).
- Helps us arrive at an almost scale invariant representation. This is very powerful since objects can be detected in an image no matter where they are located.

1.5 PROBLEM DEFINITION

Human emotions and intentions are expressed through facial expressions and deriving an efficient and effective feature is the fundamental component of facial expression system. Facial expressions convey non-verbal cues, which play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be

used in behavioural science and in clinical practice. An automatic Facial Expression Recognition system needs to solve the following problems: detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification.

CHAPTER 2

LITERATURE REVIEW

W. Swinkels, L. Claesen, F. Xiao and H. Shen, et al "SVM point-based real-time emotion detection," 2017 IEEE Conference on Dependable and Secure Computing, Taipei, 2017

Facial expressions plays important role in communication without speaking in social interaction. It is the one of the core application highly used in research area. Emotion recognition through facial expressions is used in self driving cars, entertainment, security systems etc. The ERS is considered as HMI model. Emotion recognition usually uses of speech processing, gesture signal processing, science image processing, and physiological signal processing. Recognizing emotion is a challenging task for machines, therefore we have used specific feature extraction techniques. Different machine learning algorithms are used for this purpose with the help of training algorithms and testing it on appropriate dataset. This system works on four processes face Detection, pre-processing, feature extraction, classification. In this paper we have used some machine learning algorithms and feature extraction techniques that will provide accurate facial emotion

Neerja and E. Walia, et al "Face Recognition Using Improved Fast PCA Algorithm," 2008 Congress on Image and Signal Processing, Sanya, Hainan, 2008

Face detection has been around for ages. Taking a step forward, human emotion displayed by face and felt by brain, captured in either video, electric signal (EEG) or image form can be approximated. Human emotion detection is the need of the hour so that modern artificial intelligent systems can emulate and gauge reactions from face. This can be helpful to make informed decisions be it regarding identification of intent, promotion of offers or security related threats. Recognizing emotions from images or video is a trivial task for human eye, but proves to be very challenging for machines and requires many image processing techniques for feature extraction. Several machine learning algorithms are suitable for this job. Any detection or recognition by machine learning requires training algorithm and then testing them on a suitable dataset. This paper explores a couple of machine learning algorithms as well as feature extraction techniques which would help us in accurate identification of the human emotion.

A. C. Le Ngo, Y. H. Oh, R. C. W. Phan and J. See, et al "Eulerian emotion magnification for subtle expression recognition," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016

Over the last few years, automatic facial micro-expression analysis has garnered increasing attention from experts across different disciplines because of its potential applications in various fields such as clinical diagnosis, forensic investigation and security systems. Advances in computer algorithms and video acquisition technology have rendered machine analysis of facial micro-expressions possible today, in contrast to decades ago when it was primarily the domain of psychiatrists where analysis was largely manual. Indeed, although the study of facial micro-expressions is a well-established field in psychology, it is still relatively new from the computational perspective with many interesting problems.

In this survey, we present a comprehensive review of state-of-the-art databases and methods for micro-expressions spotting and recognition. Individual stages involved in the automation of these tasks are also described and reviewed at length. In addition, we also deliberate on the challenges and future directions in this growing field of automatic facial micro-expression analysis.

CHAPTER 3

AIM AND SCOPE

SCOPE OF THE PROJECT

In this project facial expression recognition system is implemented using Convolutional neural network. Facial images are classified into seven facial expression categories namely Anger, Disgust, Fear, Happy, Sad, Surprise and 'Neutral' Kaggle dataset is used to train and test the classifier.

3.1 EXISTING SYSTEM

This problem of Face Alignment for a single image, We show how an ensemble of regression trees can be used to estimate the face's landmark positions directly from a sparse subset of pixel intensities, achieving super-real-time performance with high quality predictions. We present a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labelled data.

We show how using appropriate priors exploiting the structure of image data helps with efficient feature selection. Different regularization strategies and its importance to combat over-fitting are also investigated. In addition, we analyse the effect of the quantity of training data on the accuracy of the predictions and explore the effect of data augmentation using synthesized data.

3.2 PROPOSED SYSTEM

This paper presents an algorithm to precisely estimate the position of facial landmarks in a computationally efficient way. Similar to previous works [8, 2] our proposed method utilizes a haar cascade. In the rest of this section we describe the details of the form of the individual components of the cascade and how we perform training.

3.3 PROPOSED APPLICATIONS

a) This system can be used in medical field for help monitoring in rehabilitation, in counselling of patient's emotional state and in monitoring while driving a car.

b) This system can be accessed anytime and from anywhere.

- c) This system can be used by individual, state or country.
- d) This system can be used in E- learning by detecting the state of learner and adjusting the presentation style according online tutor.
- e) Using this system security threats become limited in shops, ATM machines, banks etc.

CHAPTER 4

MATERIALS AND METHODS

4.1: ALGORITHM:

Step 1: Collection of a data set of images.

Step 2: Pre-processing of images.

Step 3: Detection of face form each image.

Step 4: The cropped face is converted into gray scale images.

Step 5: Every image is fed into the input layer as array. Array gets passed into Convolution2D layer.

Step 6: Convolution generates feature maps.

Step 7: Pooling method called MaxPooling2D that uses (2,2) window across the feature map only keeping the maximum pixel value.

Step 8: During training, Neural network Forward propagation and Backward Propagation performed on the pixel values.

Step 9: The Softmax function presents itself as probability for each emotion class.

The model is able to show the detail probability composition of the emotion in the face.

FACE DETECTION:

For Face detection itself, several approaches have been used in the related literature. Knowledge based method to encode human knowledge is on such.

FACIAL MAPPING:

The algorithm is implemented using Opencv python library that contains a landmarks facial detector with pre trained models. The 68 point shape predictor data task was used to train the facial land mark predictor and is the source of these markings.



Facial land mark in opencv

4.2: HARDWARE REQUIREMENTS:

4.21: CAMERA



FIG: 4.1: CAMERA

Typical low-cost webcam used with many personal computers

A webcam is a video capture device that is connected to a computer or computer network, often using a USB port or, if connected to a network, Ethernet or Wi-Fi, or build in for certain types of laptops.

Their most popular uses are for video links, permitting computers to act as videophones or videoconferencing stations. Webcams can also be used with various computer video telecommunication programs such as Windows Live Messenger, Skype and Yahoo! Messenger. Other popular uses, which include the security surveillance and the recording of video files or even still-images, are accessible via numerous other software programs, applications and devices.

Webcams are known for their low manufacturing costs and flexibility,^[1] making them the lowest cost form of video telephony. They have also become the source of security and privacy issues, as some built-in webcams can be remotely activated via spyware

4.22: Features

As webcam capabilities have been added to instant messaging, text chat services such as AOL Instant Messenger, one-to-one live video communication over the Internet has now reached millions of mainstream PC users worldwide. Improved video quality has helped webcams encroach on traditional video conferencing systems. New features such as automatic lighting controls, real-time enhancements (retouching, wrinkle smoothing and vertical stretch), automatic face tracking and autofocus assist users by providing substantial ease-of-use, further increasing the popularity of webcams.

Webcam features and performance can vary by program, computer operating system, and also by the computer's processor capabilities.

For example, *'high-quality video'* is principally available to users of certain Logitech webcams if their computers have dual-core processors meeting certain

specifications. Video calling support has also been added to several popular instant messaging programs.

Some online video broadcasting sites have taken advantage of this technology to create Internet television programs centred on two (or more) people "dialoguing" with each other from different locations. Among others, BloggingHeads.tv uses this technology to enable conversations between prominent journalists, scientists, bloggers, and philosophers.

4.23: Video security

Webcams are also used as security cameras. Software is available to allow PC-connected cameras to watch for movement and sound, recording both when they are detected; these recordings can then be saved to the computer, e-mailed or uploaded to the Internet. In one well-publicised case,^[6] a computer e-mailed out images as the burglar stole it, allowing the owner to give police a clear picture of the burglar's face even after the computer had been stolen.

4.24: Input control device

Special software can use the video stream from a webcam to assist or enhance a user's control of applications and games. Video features, including faces, shapes, models and colours can be observed and tracked to produce a corresponding form of control. For example, the position of a single light source can be tracked and used to emulate a mouse pointer; a head mounted light would allow hands-free computing and would greatly improve computer accessibility. This can also be applied to games, providing additional control, improved interactivity and immersiveness.

Special software can use the video stream from a webcam to assist or enhance a user's control of applications and games. Video features, including faces, shapes, models and colours can be observed and tracked to produce a corresponding form of control. For example, the position of a single light source can be tracked and used to emulate a mouse pointer; a head mounted light would allow hands-free computing and

would greatly improve computer accessibility. This can also be applied to games, providing additional control, improved interactivity and immersiveness.

4.25: Technology

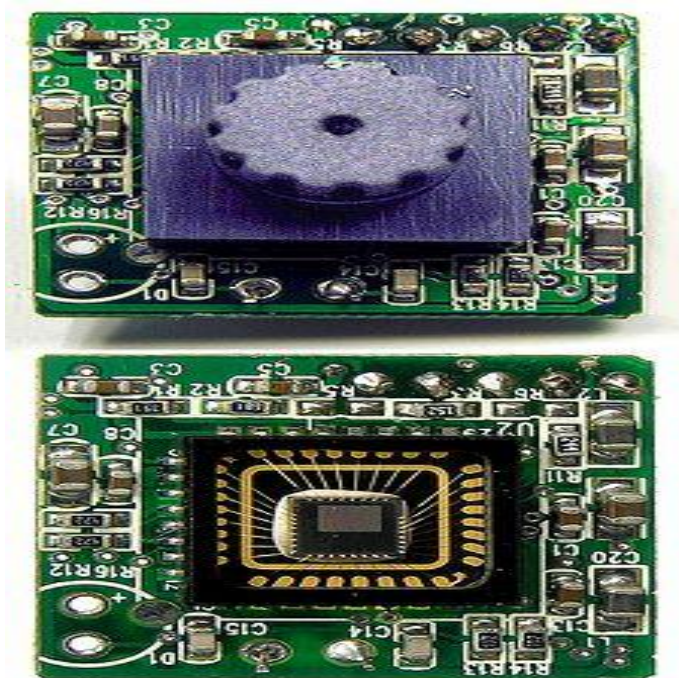


FIG: 4.2: TECHNOLOGY CIRCUIT

Webcams typically include a lens (shown at top), an image sensor (shown bottom), and supporting circuitry.

Webcams typically include a lens, an image sensor, and some support electronics. Various lenses are available, the most common in consumer-grade webcams being a plastic lens that can be screwed in and out to set the camera's focus. Fixed focus lenses, which have no provision for adjustment, are also available. As a camera system's depth of field is greater for small imager formats and is greater for lenses with a large f-number (small aperture), the systems used in webcams have

sufficiently large depth of field that the use of a fixed focus lens does not impact image sharpness much. Image sensors can be CMOS or CCD, the former being dominant for low-cost cameras, but CCD cameras do not necessarily outperform CMOS-based cameras in the low cost price range. Most consumer webcams are capable of providing VGA-resolution video at a frame rate of 30 frames per second. Many newer devices can produce video in multi-megapixel resolutions, and a few can run at high frame rates such as the PlayStation Eye, which can produce 320×240 video at 120 frames per second.

Support electronics are present to read the image from the sensor and transmit it to the host computer. The camera pictured to the right, for example, uses a Sonic SN9C101 to transmit its image over USB. Some cameras, such as mobile phone cameras, use a CMOS sensor with supporting electronics "on die", i.e. the sensor and the support electronics are built on a single silicon chip to save space and manufacturing costs. Most webcams feature built-in microphones to make video calling and videoconferencing more convenient.

The USB video device class (UVC) specification allows for interconnectivity of webcams to computers even without proprietary drivers installed. Microsoft Windows XP SP2, Linux ^[7] and Mac OS X (since October 2005) have UVC drivers built in and do not require extra drivers, although they are often installed in order to add additional features.

4.27: RASPBERRY Pi

The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.

4.3: Hardware Connections:

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

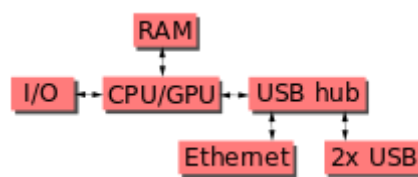


FIG: 4.2 HARDWARE CONNECTIONS

This block diagram depicts Models A, B, A+, and B+. Model A, A+ and the Pi Zero lacks the Ethernet and 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 (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.

4.31: Processor



FIG: 4.3 PROCESSOR

The Raspberry Pi 2 uses a 32-bit 900 MHz quad-core ARM Cortex-A7 processor.

The Broadcom BCM2835 SOC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first generation smart phones (its CPU is an older ARMv6 architecture) which includes a 700 MHz ARM1176JZF-S processor, Video Core IV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) caches of 128 KB. The level 2 cache is used primarily by the GPU. The SOC is stacked underneath the RAM chip, so only its edge is visible.

The Raspberry Pi 2 uses a Broadcom BCM2836 SOC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor (as do many current smart phones), with 256 KB shared L2 cache.

The Raspberry Pi 3 uses a Broadcom BCM2837 SOC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

4.32: Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

Raspberry Pi 2 includes a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It is described as 4–6 times more powerful than its predecessor. The GPU is

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Pixel/s or 1.5 Texel/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 of 2001.

The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board. A cluster of 64 Raspberry Pi Model B computers, labelled "Iridise-pi", achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts for c. US\$4000.

4.33: Over clocking

The CPU chips of the first and second generation Raspberry Pi board did not require cooling, such as a heat sink, unless the chip was over clocked, but the Raspberry Pi 2 SOC may heat more than usual under over clocking.

Most Raspberry Pi chips could be over clocked to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly over clocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the Raspberry Linux distro the over clocking options on boot can be done by a software command running "sudo raspi-config" without voiding the warranty. In those cases the Pi automatically shuts the over clocking down if the chip reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and over clocking settings (voiding the warranty); an appropriately sized heat sink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five over clock ("turbo") presets that when used, attempt to maximize the performance of the SOC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip, 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 over clock presets are:

- none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 over volt,
- modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 over volt,
- medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 over volt,
- high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 over volt,
- turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 over volt,
- Pi2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 over volt,
- Pi3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 over volt. In system information CPU speed will appear as 1200 MHz When in idle speed lowers to 600 MHz

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.

4.34: RAM

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. 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 frame buffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom Video Core IV.

For the new Model B with 512 MB RAM initially there were new standard memory split files released(arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). 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 and 512 MB Raspberry Pies.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

4.35: 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 adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip.^[32] The Raspberry Pi 3 and Pi Zero W (Wireless) are equipped with 2.4 GHz Wi-Fi 802.11n (150 Bit/s) and Bluetooth 4.1 (24 Bit/s) based on Broadcom BCM43438 chip and the Pi 3 also has a 10/100 Ethernet port.

4.36: Peripherals



FIG: 4.4 PERIPHERALS

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

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse. 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.

4.37: Video [edit]



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 emit standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. As shipped (i.e., without custom over clocking) it can emit these: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768 WXGA variant; 1280×800 WXGA variant; 1280×1024 SXGA; 1366×768 WXGA variant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680×1050 WXGA+; 1920×1080 1080p HDTV; 1920×1200 WUXGA.

Higher resolutions, such as, up to 2048×1152, may work or even 3840×2160 at 15 Hz (too low a frame rate for convincing video). Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pies are known to not work reliably for H.264 (at those high resolutions), commonly used for very high resolutions (most formats, commonly used, 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 decode H.265-encoded videos in software. The GPU in the RPI 3 runs at higher clock frequencies of 300 MHz or 400 MHz than previous versions which run at 250 MHz

The Raspberry Pies can also generate 576i and 480i composite video signals, as used on old-style (CRT) TV screens through standard connectors – either RCA or 3.5 mm phone connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

4.38: Real-time clock [edit]

None of the current Raspberry Pi models have a built-in real-time clock, and so they don't "know" the time of day. As a workaround, a program running on the Pi can get the time from a network time server or user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the PI does automatically save the time it has on shutdown, and re-installs that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, which is fully binary coded, may be added (often via the I²C interface).

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Compared to the Raspberry Pi 2 it has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

Like the Pi 2, it also has:

- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot (now push-pull rather than push-push)
- Video Core IV 3D graphics core

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2.

Monitor or TV

A monitor or TV with HDMI in can be used as a display with a Raspberry Pi.

HDMI to HDMI Cable

Connect Raspberry Pi to a Monitor or TV with a HDMI to HDMI cable.

USB Keyboard

A USB keyboard is used to input text into a computer, laptop or a Raspberry Pi.

USB Mouse

A USB mouse is used to control a cursor or mouse pointer when using a computer, laptop or a Raspberry Pi.

SD Card

SD Card and micro SD card are used to store files and software to use on your Raspberry Pi.

.

4.4: SOFTWARE REQUIREMENTS

PYTHON

Python is a wonderful and powerful programming language that's easy to use (easy to read **and** write) and with Raspberry Pi lets you connect your project to the real world.



FIG 4.6: PYTHON LOGO

Python syntax is very clean, with an emphasis on readability and uses Standard English keywords. Start by opening IDLE from the desktop.

IDLE

The easiest introduction to Python is through IDLE, a Python development environment. Open IDLE from the Desktop or applications menu:



FIG 4.22 PROGRAMING PROCESS

IDLE gives you a REPL (Read-Evaluate-Print-Loop) which is a prompt you can enter Python commands in to. As it's a REPL you even get the output of commands printed to the screen without using `print`.

```
>>>1+2
3
>>>name="Sarah"
>>>"Hello "+ name
'Hello Sarah'
```

IDLE also has syntax highlighting built in and some support for auto completion. You can look back on the history of the commands you've entered in the REPL with **Alt + P** (previous) and **Alt + N** (next).

BASIC PYTHON USAGE

Hello world in Python:

```
print("Hello world")
```

Simple as that!

INDENTATION

Some languages use curly braces **{** and **}** to wrap around lines of code which belong together, and leave it to the writer to indent these lines to appear visually nested. However, Python does not use curly braces but instead requires indentation for nesting. For example a loop in Python:

```
for i in range(10):
    print("Hello")
```

The indentation is necessary here. A second line indented would be a part of the loop, and a second line not indented would be outside of the loop. For example:

```
for i in range(2):
    print("A")
    print("B")
```

Would print:

```
A
B
A
B
```

Whereas the following:

```
for i in range(2):
    print("A")
    print("B")
```

Would print:

```
A
A
B
```

VARIABLES

To save a value to a variable, assign it like so:

```
Name="Bob"
Age=15
```

Note here I did not assign types to these variables, as types are inferred, and can be changed (it's dynamic).

```
Age=15
Age+=1# increment age by 1
Print (age)
```

This time I used comments beside the increment command.

COMMENTS

Comments are ignored in the program but there for you to leave notes, and are denoted by the hash `#` symbol. Multi-line comments use triple quotes like so:

```
.....
```

This is a very simple Python program that prints "Hello".
That's all it does.

```
.....
```

```
Print ("Hello")
```

LISTS

Python also has lists (called arrays in some languages) which are collections of data of any type:

```
Numbers= [1, 2, 3]
```

Lists are denoted by the use of square brackets `[]` and each item is separated by a comma.

ITERATION

Some data types are iterable, which means you can loop over the values they contain. For example a list:

```
Numbers= [1, 2, 3]
```

```
For number in numbers:
```

```
Print (number)
```

This takes each item in the list `numbers` and prints out the item:

```
1  
2  
3
```

Note I used the word `number` to denote each item. This is merely the word I chose for this - it's recommended you choose descriptive words for variables - using plurals for lists, and singular for each item makes sense. It makes it easier to understand when reading.

Other data types are Iterable, for example the string:

```
Dog name="BINGO"
```

```
For char in dog_name:
```

```
Print (char)
```

This loops over each character and prints them out:

```
B  
I  
N  
G  
O
```

RANGE

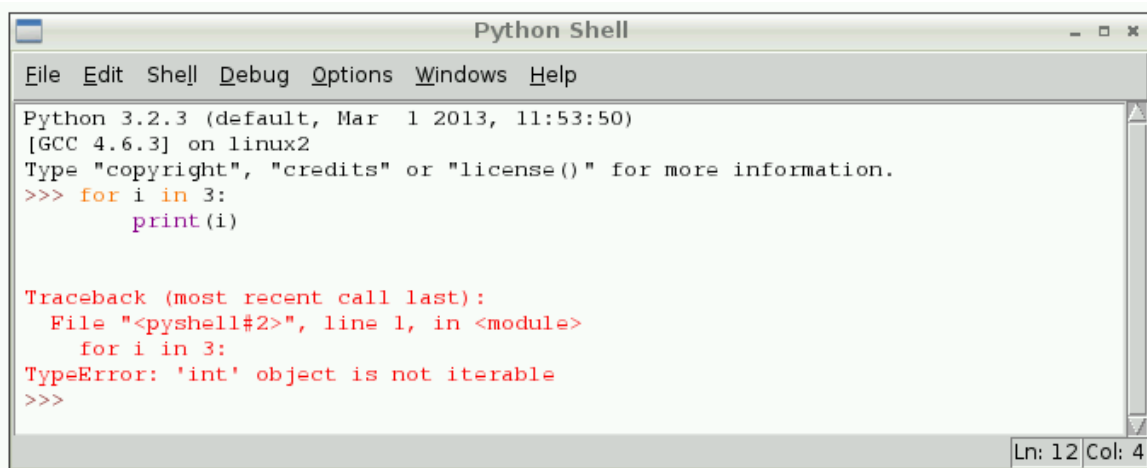
The integer data type is not Iterable and trying to iterate over it will produce an error.
For example:

```
For i in 3:
```

```
Print (i)
```

Will produce:

TypeError:'int' object is not iterable



The screenshot shows a 'Python Shell' window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The main text area displays the following content:

```
Python 3.2.3 (default, Mar 1 2013, 11:53:50)  
[GCC 4.6.3] on linux2  
Type "copyright", "credits" or "license()" for more information.  
>>> for i in 3:  
    print(i)  
  
Traceback (most recent call last):  
  File "<pyshell#2>", line 1, in <module>  
    for i in 3:  
TypeError: 'int' object is not iterable  
>>>
```

The status bar at the bottom right indicates 'Ln: 12 Col: 4'.

However you can make an iterable object using the `range` function:

```
For I in range (3):  
Print (I)
```

`Range (5)` contains the numbers `0`, `1`, `2`, `3` and `4` (five numbers in total). To get the numbers `1` to `5` use `range (1, 6)`.

LENGTH

You can use functions like `Len` to find the length of a string or a list:

```
Name="Jamie"  
Print (len (name)) # 5  
  
Names= ["Bob","Jane","James","Alice"]  
Print (len (names)) # 4
```

IF STATEMENTS

You can use `if` statements for control flow:

```
Name="Joe"  
  
File (name)>3:  
Print ("Nice name,")  
Print (name)  
Else:  
Print ("That's a short name,")  
Print (name)
```

4.41: PYTHON FILES IN IDLE

To create a Python file in IDLE, click `File > New File` and you'll be given a blank window. This is an empty file, not a Python prompt. You write a Python file in this window, save it, then run it and you'll see the output in the other window.

For example, in the new window, type:

```
n = 0

for i in range(1,101):
    n += i

Print ("The sum of the numbers 1 to 100 is :")
Print (n)
```

Then save this file (`File > Save` or `Ctrl + S`) and run (`Run > Run Module` or hit `F5`) and you'll see the output in your original Python window.

EXECUTING PYTHON FILES FROM THE COMMAND LINE

You can write a Python file in a standard editor like Vim, Niño or Leaf Pad, and run it as a Python script from the command line. Just navigate to the directory the file is saved (use `CD` and `IS` for guidance) and run with `python` , e.g. `python hello.py` .

4.42: OPENCV

Open CV-Python is a library of Python bindings designed to solve computer vision problems. ... Open CV-Python makes use of Numpy, which is a highly optimized library for numerical operations with MATLAB-style syntax. All the Open CV array structures are converted to and from Numpy arrays.

Open CV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. In simple language it is library used for Image Processing. It is mainly used to do all the operation related to Images.



FIG 4.7: OPENCV LOGO

Human eyes provide lots of information based on what they see. Machines are facilitated with seeing everything, convert the vision into numbers and store in the memory. Here the question arises how computer convert images into numbers. So the answer is that the pixel value is used to convert images into numbers. A pixel is the smallest unit of a digital image or graphics that can be displayed and represented on a digital display device.

Open CV is available for free of cost.

Since the Open CV library is written in C/C++, so it is quit fast. Now it can be used with Python.

It require less RAM to usage, it maybe of 60-70 MB.

Computer Vision is portable as Open CV and can run on any device that can run on C.

4.43: TENSORFLOW:

Tensor Flow is an open source library for fast numerical computing.

It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API.

Unlike other numerical libraries intended for use in Deep Learning like Thane, Tensor Flow was designed for use both in research and development and in production systems, not least Rank Brain in Google search and the fun Deep Dream project.

It can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines.

How to Install Tensor Flow

Installation of Tensor Flow is straightforward if you already have a Python SCI PY environment.

Tensor Flow works with Python 2.7 and Python 3.3+. You can follow the Download and Setup instructions on the Tensor Flow website. Installation is probably simplest via PYPI and specific instructions of the pip command to use for your Linux or Mac OS X platform are on the Download and Setup webpage.

There are also ENC and docked that you can use if you prefer

Computation is described in terms of data flow and operations in the structure of a directed graph.

Notes: Nodes perform computation and have zero or more inputs and outputs. Data that moves between nodes are known as tensors, which are multi-dimensional arrays of real values.

Edges: The graph defines the flow of data, branching, looping and updates to state. Special edges can be used to synchronize behaviour within the graph, for example waiting for computation on a number of inputs to complete.

Operation: An operation is a named abstract computation which can take input attributes and produce output attributes. For example, you could define an add or multiply operation

Tensor Flow is a Python-friendly open source library for numerical computation that makes machine learning faster and easier. ... It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. Tensor Flow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation

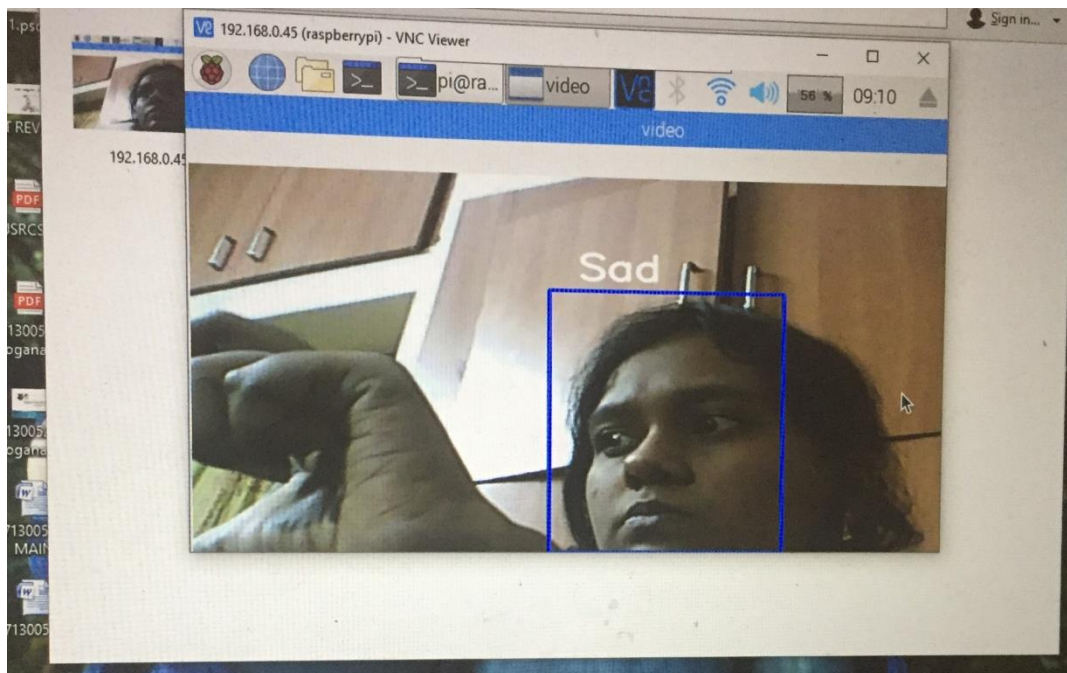
Tensor Flow isn't the easiest of languages, and people are often discouraged with the steep learning curve. There are other languages that are easier and worth learning as well like PyTorch and Keras. ... It's helpful to learn the different architectures and types of neural networks so you know how they can be used

CHAPTER 5

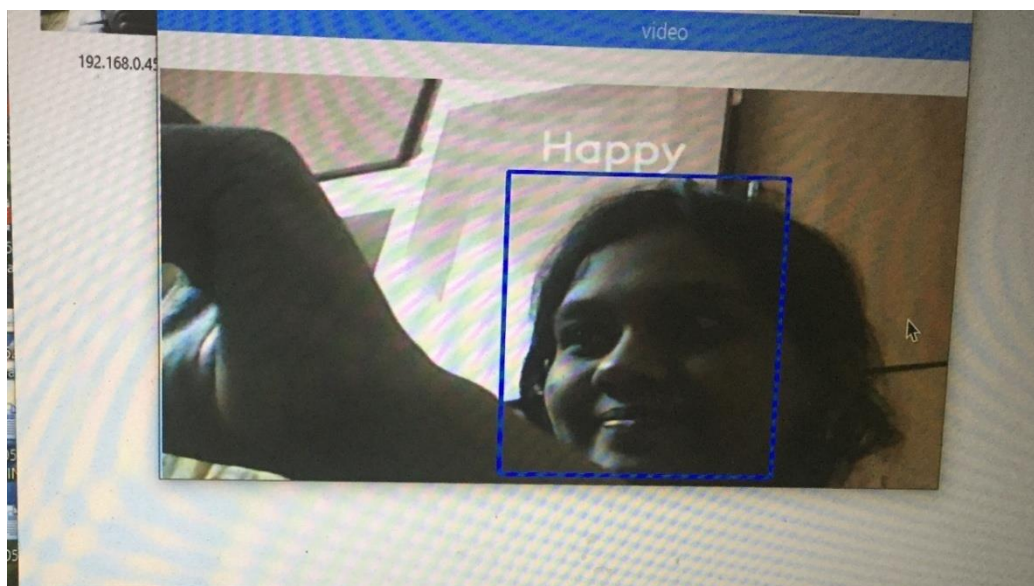
RESULT AND ANALYSIS

The following emotions as sadness, happiness, surprise, anger and fear in the database, are all judged precisely by the model are shown in the pictures.

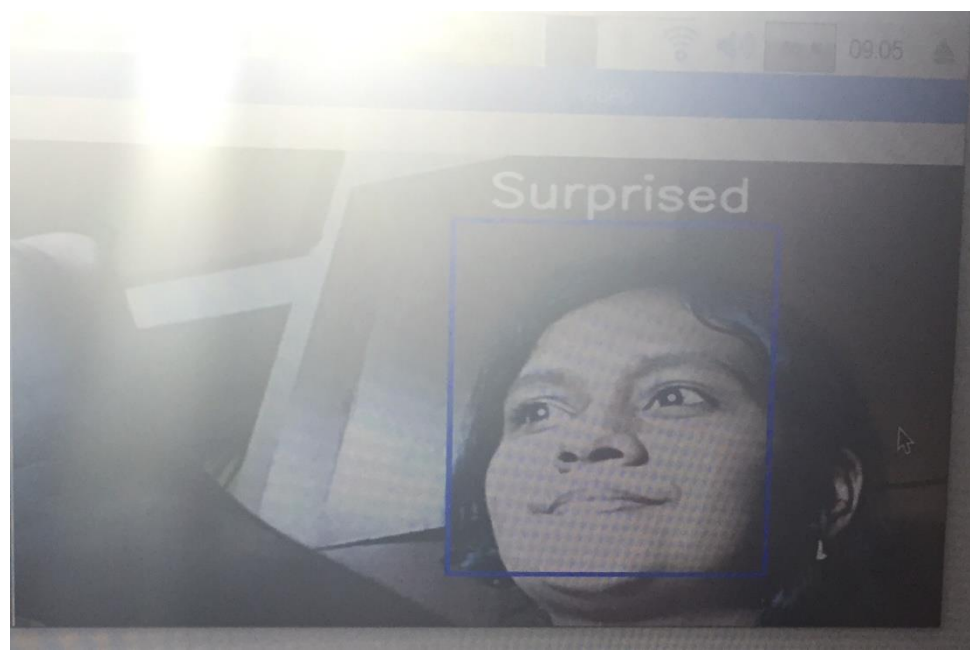
SADDNESS



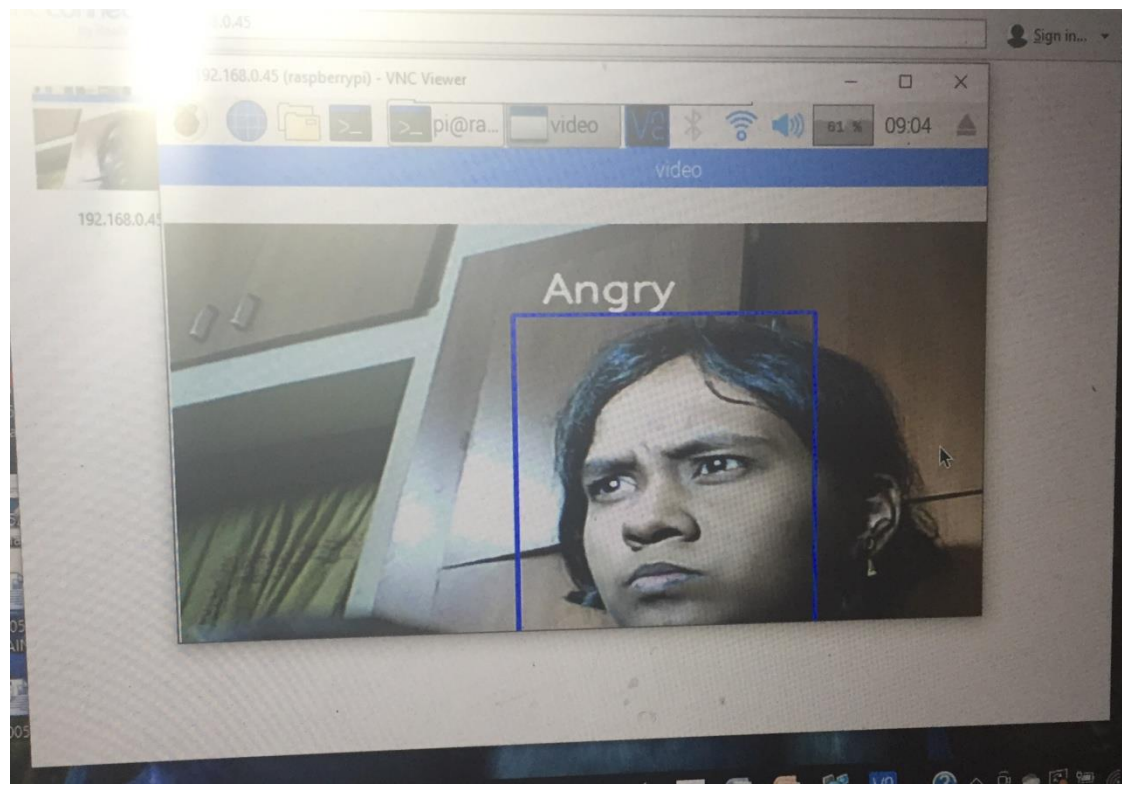
HAPPINESS



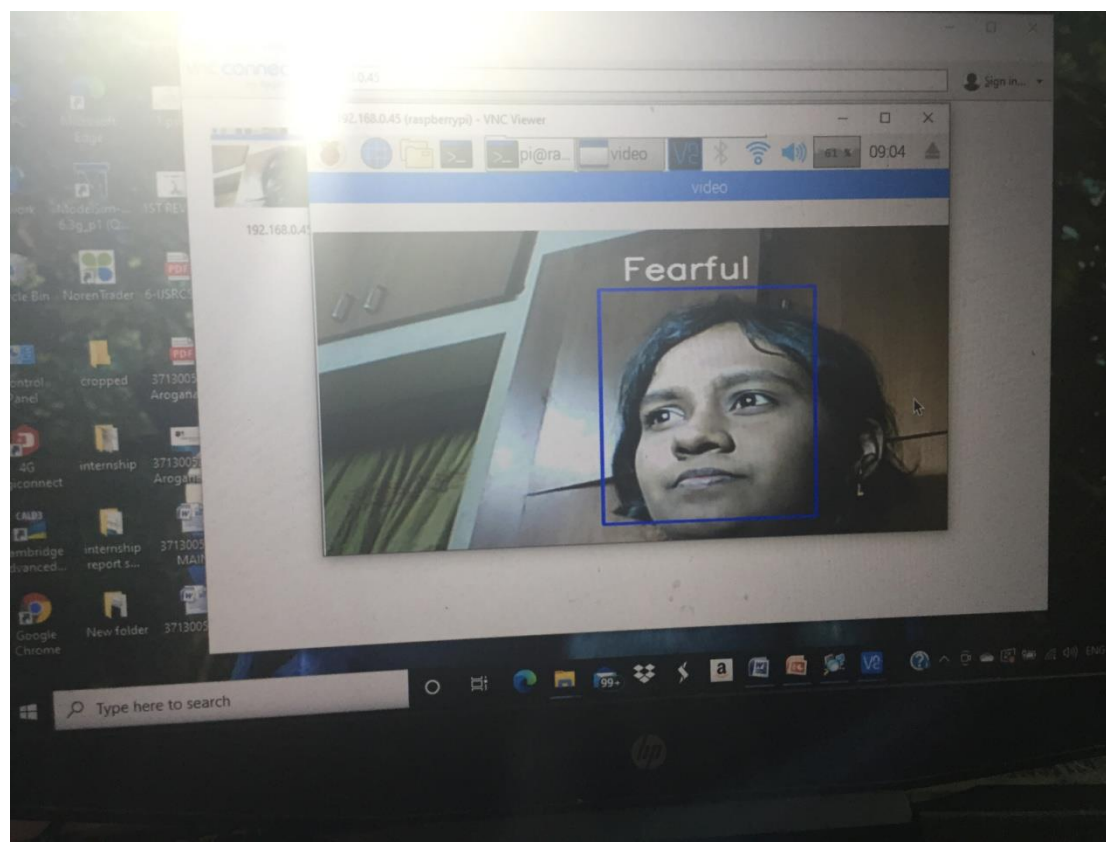
SURPRISE



ANGER



FEARFULL



LIMITATIONS

1. Dependence on ambient light

With poor lighting conditions even though face is easily detected, sometimes the system is unable to detect the eyes. So it gives an erroneous result which must be taken care of. In real time scenario infrared backlights should be used to avoid poor lighting conditions.

4. Problem with multiple faces

If more than one face is detected by the webcam, then our system gives an erroneous result. This problem is not important as we want to detect the drowsiness of a single driver.

CHAPTER 6

SUMMARY AND CONCLUSION

Although people are mostly in a progressive and convenient medical environment, their psychological problems should be relieved by time and proper media. In particular, people can be caught with so-called modern diseases such as Internet addiction and interpersonal alienation. Therefore, people are more prone to emotional obstacles or improper expression in modern daily life. This paper introduces ICT into the life of the elderly to ease their emotions, which not only addresses the problem of Internet addiction, but uses multimedia to improve life quality or life satisfaction. The communication technology is applied in series to the device of the IOS platform, and the camera detected if anyone is there automatically. The image would be transmitted to the desktop immediately. And then, the music playlist is played based on the facial expression. The App has quite simple user interface and ease to use. And the playlist could be alternative by one's preference. We hope that every adult, especially the old, could release depression in a short time and live well on one's own. This application may help one's not only mental health but also physical health. To a person, it is important and needed to get along with self. Music response may be a great treatment.

REFERENCES

- [1] [T. F. Cootes, C. J. Taylor, et al. Statistical models of appearance for computer vision, 2004.
- [2] P. Ekman and E. L. Rosenberg. What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA, 1997.
- [3] Gudi. Recognizing semantic features in faces using deep learning. ArXiv preprint arXiv: 1512.00743, 2015.
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770– 778, 2016.
- [5] S. Ioffe and C. Szeged. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ArXiv preprint arXiv: 1502.03167, 2015.
- [6] R. T. Ionesco, M. Popescu, and C. Grozea. Local learning to improve bag of visual words model for facial expression recognition. In Workshop on challenges in representation learning, ICML, 2013.
- [7] Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep Convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
- [8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Back propagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.
- [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278– 2324, 1998.
- [10] K. Simonyan and A. Zisserman. Very deep Convolutional networks for large-scale image recognition. ArXiv preprint arXiv: 1409.1556, 2014. [12] L. Sirovich and M. Kirby. Low-dimensional proceed.
- [11] N. Srivastava. Improving neural networks with dropout. PhD thesis, University of Toronto, 2013.
- [12] Y. Tang. Deep learning using support vector machines. CoRR, abs/1306.0239, 2, 2013. [15] H. Van Kuilenburg, M. Wiering, and M. Den Uyl. A model based method for automatic facial expression recognition. In European Conference on Machine Learning, pages 194–205. Springer, 2005.
- [12] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. ArXiv preprint arXiv: 1611.05431, 2016.
- [14] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016

R1.docx

ORIGINALITY REPORT

09%

SIMILARITY INDEX

PRIMARY SOURCES

1	en.wikipedia.org Internet	607 words-1%
2	www.raspberrypi.org Internet	823 words-1%
3	www.coursehero.com Internet	441 words-1%
4	DaiLun Chiang, JihHsiang Yang, ZiYuan Huang, FeiPei Lai. "Music Response Based on Real-time Facial Expression Recognition", 2019 29th International Facial Expression Recognition", 2019 29th International Crossref	231 words-1%
5	scholarworks.sjsu.edu Internet	224 words -1%
6	Machinelearningmastery.com Internet	175 words -1%
7	rcciit.org Internet	172 words -1%
8	www.froniersin.org Internet	159 words -1%

9 Sumit Das, Manas Kumar Synyal, Sourav Kumar 137 words -1%

Upadhyay, Supriyo Chatterjee. "An Intelligent

Approach for Predicting Emotion Using Convolution Neural

Approach for Predicting Emotion Using Convolution Neural

Crossref
