FACE RECOGNITION AND LIP MOVEMENTS BASED SPEAKER VERIFICATION

SK. Nayab Rasool[1], P.Kalpana[2]
[1]Assistant Professor, Dept of ECE, Anurag Group Of Institutions, Hyderabad
[2]Assistant Professor, Dept of ECE, Jagruthi College of Engineering, Hyd

Abstract
The existing password verification scheme is based on keyboard using which we are entering our password. For giving some level of security the keys which we are pressing on then keyboard will not shown on the screen and instead of it some '*' marks will show. But if anybody watch keyboard while typing the keys then he can easily find out our password.In the proposed system we overcome the drawback present in existing system by using single camera. Our system is designed by using ARM 32-bit micro controller which supports different features and algorithms for the development of automotive vision systems. Here the camera is connected to ARM controller. The camera will capture the lip movement when we trying to pronounce it without creating any voice. If the lip movement is corresponding with the previously recorded movement then only verification will conform and secure password of locker will matches then it opens.In this method user has to stand in front of camera and make the lip movement as per password without speaking the password. The Open CV algorithms in our ARM development board match the received lip movement from camera with the previously recorded database.

Keywords: Lip Password, Face Reorganization, Raspberry Pi, Linux OS.

1. Introduction:

An embedded system is a special purpose computer system that is designed to perform very small sets of designated activities. Embedded systems date back as early as the late 1960s where they used to control electromechanical telephone switches. The first recognizable embedded system was the Apollo Guidance Computer developed by Charles Draper and his team. Later they found their way into the military, medical sciences and the aerospace and automobile industries. Today they are widely used to serve various purposes like Network equipment such as firewall, router, switch, and so on. Embedded Systems have and require very few resources in terms of ROM or other I/O devices as compared to a desktop computer.

In an embedded system, when there is only a single task that is to be performed, then only a binary is to loaded into the target controller and is to be executed. However, when there are multiple tasks to be executed or multiple events to be handled, then there has to be a program that handles and prioritizes these events. This program is the Operating System (OS), which one is very familiar with, in desktop PCs.Various Operating Systems:

Embedded Operating Systems are classified into two categories:

1. Real-time Operating Systems (RTOS):

Real Time Operating Systems are those which guarantee responses to each event within a defined amount of time. This type of operating system is mainly used by time-critical applications such as measurement and control systems. Some commonly used RTOS for embedded systems are: V x Works, OS-9, System bi an, RT Linux.

2. Non-Real-time Operating Systems:

Non-Real Time Operating Systems do not guarantee defined response times. These systems are mostly used if multiple applications are needed. Windows CE and Palm OS are examples for such embedded operating systems.

Why Linux?

There are a wide range of motivations for choosing Linux over a traditional embedded OS.

The following are the criteria due to which Linux is preferred:

Quality and Reliability of Code:

Quality and reliability are subjective measures of the level of confidence in the code that comprises software such as the kernel and the applications that are provided by distributions. Some properties that professional programmers expect from a "quality" code are modularity and structure, readability, extensibility and configurability. "Reliable" code should have features like predictability, error recovery and longevity. Most programmers agree that the Linux kernel and other projects used in a Linux system fit this description of quality and reliability. The reason is the open source development model, which invites many parties to contribute to projects, identify existing problems, debate possible solutions, and fix problems effectively.

2.Existing System:

Traditionally, the acoustic speech signals may probably be the most natural modality to achieve speaker verification. Although a purely acoustic-based speaker verification system has shown the effectiveness in its application domain, its performance would be degraded dramatically in the environment corrupted by the background noise or multiple talkers. Nevertheless, the access-controlled systems utilizing the still face images are very susceptible to the poor quality of pictures, variations in pose or facial expressions. Further, such a system may be easily deceived by a face photograph placed in front of the camera as well.

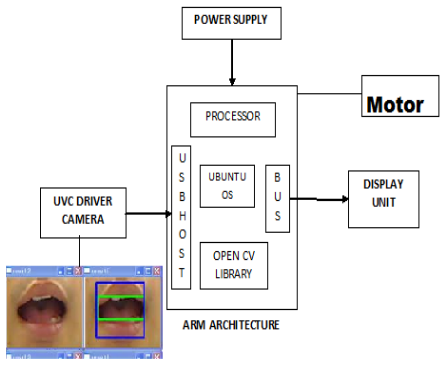
The existing password verification method is based on keyboard using which we are entering our password. For giving some level of security the keys which we are pressing on then keyboard will not shown on the screen and instead of it some '*' marks will show. But if anybody watch keyboard while typing the keys then he can easily find out our password. The performance of the existing lip motion based speaker verification systems is far behind our expectation

3. Proposed Method:

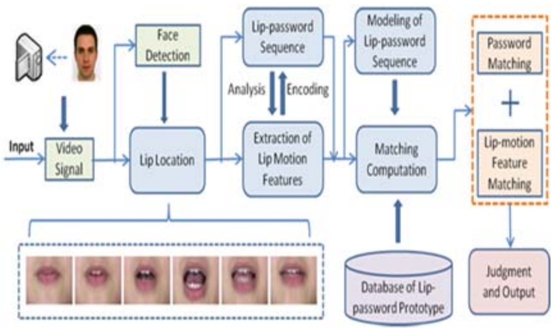
In the proposed method we overcome the drawback present in existing system by using single camera. Our system is designed by using ARM 32-bit micro controller which supports different features and algorithms for the development of automotive vision systems. Here the camera is connected to ARM controller. The camera will capture the lip movement when we trying to pronounce it without creating any voice. If the lip movement is matched with the previously recorded movement then only verification will conform and secure password of locker will matches then it opens.

In this method we have to stand in front of camera and make the lip movement as per password without speaking the password. The Open CV algorithms in our ARM development board match the received lip movement from camera with the previously recorded database. We are developing the application in ARM11 Platform. Through mjpg-streamer video streaming will be done. That video get displayed on the webpage.The images captured by the camera should be processed very fast to provide real time visualization of environment to the user.

For this purpose along with low cost we think to use ARM. The images captured by the camera should be processed very fast to provide real time visualization of environment to the user.



BLOCK DIAGRAM



In this section, we are giving the complete description on the proposed system architecture. Here we are using Raspberry Pi board as our platform. It has an ARM-11 SOC with integrated peripherals like USB, Ethernet and serial etc. On this board we are installing Linux operating system with necessary drivers for all peripheral devices and user level software stack which includes a light weight GUI based on X Server, V4L2 API for interacting with video devices like cameras, TCP/IP stack to communicate with network devices and some standard system libraries for system level general IO operations. The Raspberry Pi board equipped with the above software stack is connected to the outside network and a camera is connected to the Raspberry Pi through USB bus. On the other side we have to host a web server with cloud facility.

The system uses webcam which is places at fore head of visually impaired person and is connected to Raspberry PI board through USB device. Install CMOS camera device driver in Raspberry pi board. Now the camera will capture the images. The board will take the images from camera and display it on monitor. Also it saves the images in the form of frames in pen drive.

4.Hardware Requirements:

Raspberry pi:

The Raspberry Pi is a small, powerful and lightweight ARM based computer which can do many of the things a desktop PC can do. The powerful graphics capabilities and HDMI video output make it ideal for multimedia applications such as media centers and narrowcasting solutions. The Raspberry Pi is based on a Broadcom BCM2835 chip. It does not feature a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage. Raspberry Pi has a strong processing capacity because of using the ARM11 architecture and Linux-based system. In terms of control and interface, it has 8 GPIO, 1 UART, 1 I2C and 1 SPI, which are basically meet the control requirement are simple and easy-used open source peripheral driver libraries.

Fig.Raspberry pi board

While Raspberry pi can be used without any additional hardwar (except perhaps a power supply of some kind), it won't be much use as a general computer. As with any normal PC, it is likely you will need some additional hardware.
The following are more or less essential:
• Raspberry Pi board
• Prepared Operating System SD Card
• USB keyboard
• Display (with HDMI, DVI, Composite or SCART input)
• Power Supply
• Cables

Prepared operating system SD card

As the Raspberry Pi has no internal storage or built-in operating system it requires an SD-Card that is set up to boot the R Pi. You can create your own preloaded card using any suitable SD card you have. Be sure to backup any existing data on the card. Preloaded SD cards will be available from the R Pi Shop. This guide will assume you have a preloaded SD card.

Keyboard & mouse

Most standard USB keyboards will work with the R Pi. Wireless keyboard should also function, and only require a single USB port for an RF dongle. In order to use a Bluetooth keyboard or mouse you would need to use a Bluetooth dongle, which again uses a single port. Remember that the Model A has a single USB port and the Model B only has two (typically a keyboard and mouse will use a USB port each).

Display: There are two main connection options for the Raspberry Pi display, HDMI (high definition) and Composite (low definition).

Power supply

The unit uses a Micro USB connection to power itself (only the power pins are connected - so it will not transfer data over this connection). A standard modern phone charger with a micro-USB connector will do, but needs to produce at least 700mA at 5 volts• Computer

USB Port or powered USB hub (will depend on power output)

Cables

You will probably need a number of cables in order to connect your R Pi up.

1. Micro-B USB Power Cable
2. HDMI-A or Composite cable, plus DVI adaptor or SCART adaptor if required, to connect your R Pi to the Display/Monitor/TV of your choice.
3. Audio cable, this is not needed if you use a HDMI TV/monitor.
4. Ethernet/LAN Cable

5.Operating System :

Linux:

Linux or **GNU/Linux** is a free and open source software operating system for computers. The operating system is a collection of the basic instructions that tell the electronic parts of the computer what to do and how to work. Free and open source software (FOSS) means that everyone has the freedom to use it, see how it works, and changes it. There is a lot of software for Linux, and since Linux is free software it means that none of the software will put any license restrictions on users. This is one of the reasons why many people like to use Linux.

A Linux-based system is a modular UNIX like operating system. It derives much of its basic design from principles established in UNIX during the 1970s and 1980s. Such a system uses a monolithic kernel, the Linux kernel, which handles process control, networking, and peripheral and file system access. Device drivers are either integrated directly with the kernel or added as modules loaded while the system is running.

Separate projects that interface with the kernel provide much of the system's higher-level functionality. The GNU user land is an important part of most Linux-based systems, providing the most common implementation of the C library, a popular shell, and many of the common UNIX tools which carry out many basic operating system tasks. The graphical user interface (or GUI) used by most Linux systems is built on top of an implementation of the X Window System. Some components of an installed Linux system are:

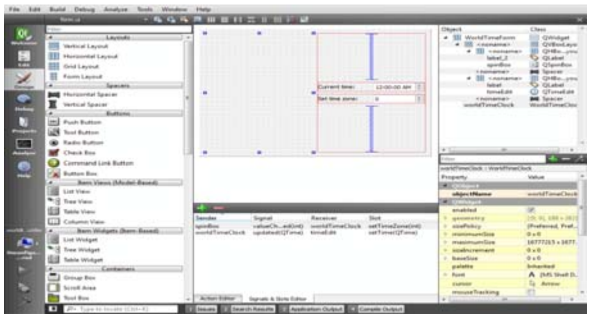
A boot loader, for example GNU GRUB or LILO. This is a program which is executed by the computer when it is first turned on, and loads the Linux kernel into memory.

- An init program. This is the first process launched by the Linux kernel, and is at the root of the process tree: in other terms, all processes are launched through in it. It starts processes such as system services and login prompts (whether graphical or in terminal mode).
- Software libraries which contain code which can be used by running processes. On Linux systems using ELF-format executable files, the dynamic linker which manages use of dynamic libraries is "ld-linux.so". The most commonly used software library on Linux systems is the GNU C Library. If the system is set up for the user to compile software themselves, header files will also be included to describe the interface of installed libraries.
- User interface programs such as command shells or windowing environments.

Designing User Interface

Qt Creator provides a fully integrated visual editor, Qt Designer. Qt Designer is a tool for designing and building graphical user interfaces from Qt widgets. Users can compose and customize widgets or dialogs and test those using different styles and resolutions. Widgets and forms created with Qt Designer are integrated seamlessly with programmed code, using the Qt signals and slots mechanism, which lets users easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code. Furthermore, features such as widget promotion and custom plug in allow users to use their own widgets with Qt Designer. Qt Designer is used for editing user interface files. It presents users with an intuitive drag-and-drop interface for composing new user interfaces. The user interfaces that are designed with Qt Designer are fully functional and can be previewed immediately to ensure that the design is as intended. There is no need to recompile the entire project to test out a new design. The following figure shows the integrated Qt Designer being used to edit a simple form.

Fig. Integrated Qt Designer



The figure shows how Qt Designer is integrated into Qt Creator. The center of the view is used for the construction of the user interface, with the available user interface components (widgets) kept in the container on the left side of the window.

Other tools are positioned around the edge of the working area. These include the **Object Inspector** (top-right), which shows the hierarchy of the objects in the current user interface, and the **Property Editor** (right), used to configure the currently selected widget. In addition, developers can use the integrated Qt Designer for many other tasks, such as connecting signals and slots, creating actions for toolbars and menus, and setting the tab order. The Qt Designer integration also includes project management and code completion.

Fig. The code editor in Edit mode

Design Approach

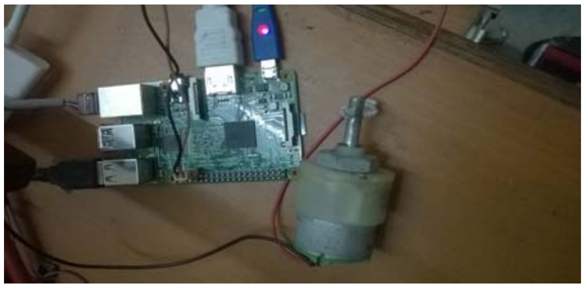


Fig. Connections on the Raspberry PI Board

Fig. Output window

Results and discussions

In this system we have to stand in front of camera and make the images of face.

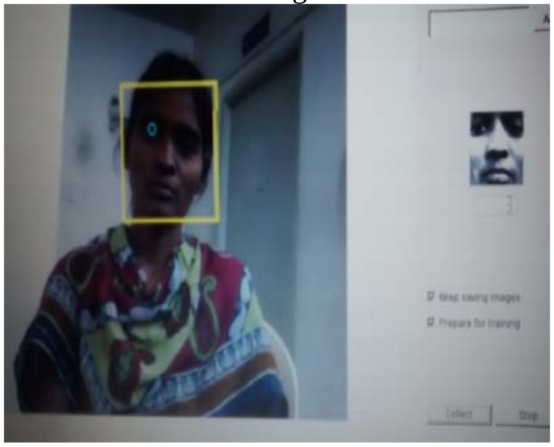


Fig. Face recognition

First face is recognized then only it will go to lip password lip movement as per password without speaking the password. Otherwise it will be stop .

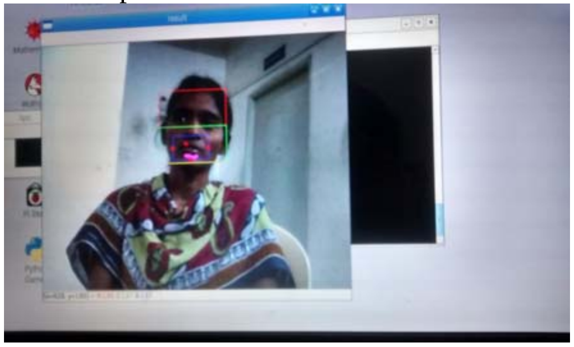


Fig. Lip movements

Finally face recognition &lip password these two are matched then only the person is authenticated & automatically motor will be on condition.

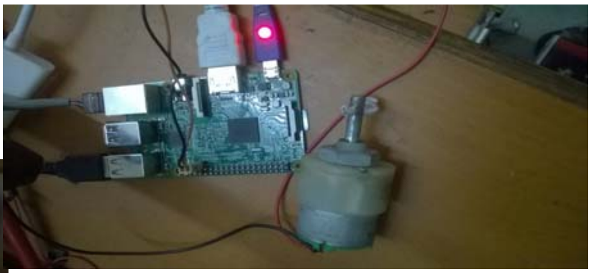


Fig. System is in on condition

Conclusion:

This paper proposes a concept of face recognition and lip motion movements (simply called lip-password hereinafter), which is composed of a password embedded in the lip movement. It provides a double security. In the proposed system we overcome the drawback present in existing system by using single camera. Our system is designed by using ARM 32-bit micro controller which supports different features and algorithms for the development of automotive vision systems. Here the camera is connected to ARM controller. The camera will capture the lip movement when we trying to pronounce it without creating any voice. If the lip movement is matched with the previously recorded movement then only verification will conform and secure password of locker will matches then it opens.

In this system user has to stand in front of camera and make the lip movement as per password without speaking the password. The OPENCV algorithms in our ARM development board first the face is recognize after that it will be go to lip password. If suppose the face is not recognize then process will be stop. Finally the face is recognized and lip password is also match the previous password.

REFERENCES:

[1] A. B. Hassanat and S. Jassim, "A special purpose knowledge-based face localization method," in SPIE, Florida, 2008, pp. 69820-69829.

[2] A. B. Hassanat and S. Jassim, "Visual words for lip-reading," in SPIE, Florida, 2010, p. 77080B.

[3] J. P. Barker and F. Berthommier, "Estimation of speech acoustics from visual speech features: A comparison of linear and non-linear models," in Auditory-Visual Speech Processing, Santa Cruz, 1999, p. 112–117

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 980–988, Jul. 2008.

[5] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "A comparison of session variability compensation approaches for speaker verification," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 802–809, Dec. 2010.

[6] A. Roy, M. Magimai-Doss, and S. Marcel, "A fast parts-based approach to speaker verification using boosted slice classifiers," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 241–254, Feb. 2012.

[7] E. Engin, Y. Yemez, and A. M. Tekalp, "Multimodal speaker identification using an adaptive classifier cascade based on modality reliability," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 840–852, Oct. 2005.

[8] G. Chetty and M. Wagner, "Robust face-voice based speaker identity verification using multilevel fusion," *Image Vis. Comput.*, vol. 26, no. 9, pp. 1249–1260, Sep. 2008.

[9] A. K. Sao and B. Yegnanarayana, "Face verification using template matching," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 636–641, Sep. 2007.

[10] C. Chi Ho, B. Goswami, J. Kittler, and W. Christmas, "Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 602–612,Apr. 2012.

[11] M. I. Faraj and J. Bigun, "Synergy of lip-motion and acoustic features in biometric speech and speaker recognition," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1169–1175, Sep. 2007 recognition," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 4, pp. 564–570, Jul. 2004